



ELSEVIER

COLEÇÃO

CAMPUS - ABEPRO

ENGENHARIA DE PRODUÇÃO

MARCOS  
ARENALES

VINÍCIUS  
ARMENTANO

REINALDO  
MORABITO

HORACIO  
YANASSE

# Pesquisa Operacional

*PARA CURSOS DE ENGENHARIA*

Cadastre-se em **[www.elsevier.com.br](http://www.elsevier.com.br)**  
para conhecer nosso catálogo completo,  
ter acesso a serviços exclusivos no site  
e receber informações sobre nossos  
lançamentos e promoções.

MARCOS  
ARENALES

VINÍCIUS  
ARMENTANO

REINALDO  
MORABITO

HORACIO  
YANASSE

# Pesquisa Operacional



© 2007, Elsevier Editora Ltda.  
Todos os direitos reservados e protegidos pela Lei 9.610 de 19/02/1998.  
Nenhuma parte deste livro, sem autorização prévia por escrito da editora,  
poderá ser reproduzida ou transmitida sejam quais forem os meios empregados:  
eletrônicos, mecânicos, fotográficos, gravação ou quaisquer outros.

*Copidesque:* Cláudia Mello Belhassof  
*Editoração Eletrônica:* Rosane Guedes  
*Revisão Gráfica:* Marília Pinto de Oliveira e Marco Antonio Correa

*Projeto Gráfico*  
Elsevier Editora Ltda.  
A Qualidade da Informação.  
Rua Sete de Setembro, 111 – 16º andar  
20050-006 Rio de Janeiro RJ Brasil  
Telefone: (21) 3970-9300 FAX: (21) 2507-1991  
E-mail: [info@elsevier.com.br](mailto:info@elsevier.com.br)  
*Escritório São Paulo:*  
Rua Quintana, 753/8º andar  
04569-011 Brooklin São Paulo SP  
Tel.: (11) 5105-8555

ISBN 13: 978-85-352-Í FJH

**Nota:** Muito zelo e técnica foram empregados na edição desta obra. No entanto, podem ocorrer erros de digitação, impressão ou dúvida conceitual. Em qualquer das hipóteses, solicitamos a comunicação à nossa Central de Atendimento, para que possamos esclarecer ou encaminhar a questão.

Nem a editora nem o autor assumem qualquer responsabilidade por eventuais danos ou perdas a pessoas ou bens, originados do uso desta publicação.



Central de atendimento  
Tel.: 0800-265340  
Rua Sete de Setembro, 111, 16º andar – Centro – Rio de Janeiro  
e-mail: [info@elsevier.com.br](mailto:info@elsevier.com.br)  
site: [www.campus.com.br](http://www.campus.com.br)

CIP-Brasil. Catalogação-na-fonte.  
Sindicato Nacional dos Editores de Livros, RJ

---

P564

Pesquisa operacional [recurso eletrônico] / Marcos Arenales...  
[et al.]. - Rio de Janeiro : Elsevier : ABEP, 2011.  
recurso digital (ABEP-Campus)

Formato: FLASH  
Requisitos do sistema: Adobe Flash Player  
Modo de acesso: World Wide Web  
Inclui índice  
Anexos  
ISBN 978-85-352-5193-7 (recurso eletrônico)

1. Pesquisa operacional. 2. Pesquisa operacional -  
Modelos matemáticos. 3. Livros eletrônicos. I. Arenales, Marcos  
II. Associação Brasileira de Engenharia da Produção. III. Série

---

11-5920. CDD: 658.4034  
CDU: 005.31

# Os Autores

**Marcos Arenales** é professor do Departamento de Matemática Aplicada e Estatística (SME) da Universidade de São Paulo (USP) – campus de São Carlos. É graduado em Matemática pela Universidade Estadual Paulista (UNESP) e obteve os títulos de mestre em Otimização e Pesquisa Operacional pela Universidade Estadual de Campinas (IMECC-UNICAMP), de doutor em Engenharia de Sistemas e Automação (FEEC-UNICAMP) e de livre-docente em Matemática Aplicada pela USP. Seus principais interesses de pesquisa situam-se no desenvolvimento de métodos e aplicações na área de otimização e pesquisa operacional. Tem publicado artigos em diversos periódicos científicos, incluindo *Canadian Journal of Operational Research and Information Processing, Computers and Operations Research, European Journal of Operational Research, Gestão & Produção, International Journal of Production Research, International Transactions in Operations Research e Pesquisa Operacional*.

**Vinícius Amaral Armentano** é professor titular da Faculdade de Engenharia Elétrica e de Computação (FEEC) da Universidade Estadual de Campinas (UNICAMP). É graduado em Engenharia Elétrica pela Universidade Mackenzie, obteve os títulos de mestre em Engenharia Elétrica pela FEEC-UNICAMP, e de Ph.D em Engenharia Elétrica pelo Imperial College, University of London. Foi professor visitante da Leeds School of Business da University of Colorado, nos Estados Unidos. Seus principais interesses de pesquisa situam-se na área de otimização combinatória, com a modelagem de problemas práticos em produção e logística, e desenvolvimento de métodos exatos e heurísticos em programação inteira. Tem publicado artigos em diversos periódicos científicos, incluindo *Computers and Operations Research, European Journal of Operational Research, Journal of Heuristics, Gestão e Produção, Journal of the Operational Research Society, Pesquisa Operacional, Production and Operations Management e SIAM Journal on Control and Optimization*.

**Reinaldo Morabito** é professor do Departamento de Engenharia de Produção (DEP) da Universidade Federal de São Carlos (UFSCar). É graduado em Engenharia Civil pela Universidade Estadual de Campinas (UNICAMP), obteve os títulos de mestre em Ciências da Computação e Matemática Computacional, de doutor em Engenharia de Transportes e de livre-docente em Engenharia Mecânica, todos pela Universidade de São Paulo (USP). Fez pós-doutorado no Massachusetts Institute of Technology (MIT) em Cambridge, nos Estados Unidos. Seus principais interesses de pesquisa situam-se na área de pesquisa operacional aplicada a sistemas de produção e logística, com a modelagem matemática de problemas práticos e desenvolvimento de métodos de solução exatos e heurísticos. Tem publicado artigos em diversos periódicos científicos, incluindo *Canadian Journal of Operational Research and Information Processing, Computers and Operations Research, Gestão & Produção, European Journal of Operational Research, International Journal of Production Research, International Transactions in Operations Research, Journal of the Operational Research Society, Pesquisa Operacional, Production and Operations Management e Transportation Research*.

**Horacio Hideki Yanasse** é pesquisador do Laboratório Associado de Computação e Matemática Aplicada (LAC) do Instituto Nacional de Pesquisas Espaciais (INPE). É graduado em Engenharia Eletrônica pelo Instituto Tecnológico de Aeronáutica (ITA), obteve o título de mestre em Análise de Sistemas e Aplicações pelo Inpe, e de PhD em Pesquisa Operacional pelo Massachusetts Institute of Technology (MIT) em Cambridge, nos Estados Unidos. Foi pesquisador visitante da Universidade de Sheffield, na Inglaterra. Seus principais interesses de pesquisa situam-se na área de otimização combinatória, com a modelagem de problemas práticos oriundos de ambientes produtivos, e desenvolvimento de métodos exatos e heurísticos em programação inteira. Tem publicado artigos em diversos periódicos científicos, incluindo *Computers and Operations Research, Discrete Applied Mathematics, European Journal of Operational Research, IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems, International Transactions in Operations Research, Journal of the Operational Research Society, Management Science, Operations Research, Parallel Computing e Pesquisa Operacional*.

Página deixada intencionalmente em branco

# Dedicatórias

*À minha esposa Selma, aos meus pais Dom Mariano e Dona Isabel  
e à minha enorme família de muitos amigos.*

*À minha esposa Ana Luiza, aos meus amigos  
Artur, Bottura, Giampietro e Hermano,  
e aos meus pais Emília e Augusto.*

*Às minhas filhas Julia e Paula,  
e aos meus pais Pedro e Myriam.*

*À minha esposa Rita, aos meus filhos Ricardo, Denise e Eduardo,  
e aos meus pais Tomossaburo e Hatsumi.*

Página deixada intencionalmente em branco

# Prefácio

Pesquisa operacional é a aplicação de métodos científicos a problemas complexos para auxiliar no processo de tomada de decisões, tais como projetar, planejar e operar sistemas em situações que requerem alocações eficientes de recursos escassos. Neste livro, introduzimos modelos matemáticos determinísticos e probabilísticos e vários dos principais métodos de solução e algoritmos utilizados na pesquisa operacional para melhor entendimento, análise e solução de problemas de decisão.

Esta obra é direcionada, principalmente, a cursos de graduação em engenharia de produção, mas também pode ser utilizada em outros cursos de engenharia (civil, elétrica, mecânica, sistemas), ciências de administração, ciências de computação, economia, estatística e matemática. Também pode ser usada em cursos introdutórios de graduação e pós-graduação (*stricto sensu* e *lato sensu*) em tópicos como: otimização linear (programação linear), otimização discreta (programação linear inteira), otimização (fluxos) em redes, programação dinâmica (determinística e estocástica) e teoria de filas. Os requisitos para a leitura deste livro são cursos introdutórios de cálculo integral e diferencial, álgebra linear e teoria de probabilidade.

A habilidade de elaborar modelos matemáticos é fundamental e, por esse motivo, o livro apresenta uma grande variedade de modelos de problemas nas áreas de engenharias e administração, entre outras, com o intuito de mostrar aos estudantes a versatilidade e a aplicabilidade de modelos de pesquisa operacional. Procuramos destacar a importância da pesquisa operacional na engenharia de produção, em particular no gerenciamento de sistemas de produção e logística, envolvendo problemas de planejamento, programação e controle da produção, problemas de distribuição de produtos, problemas de localização de facilidades e instalações, entre outros.

O livro está organizado em duas partes: na Parte I abordam-se modelos determinísticos de programação matemática, tais como: otimização linear, otimização discreta, otimização em redes e programação dinâmica determinística, juntamente com métodos de solução e algoritmos. Na Parte II trata-se de modelos probabilísticos que consideram incertezas, como modelos de programação dinâmica estocástica e teoria de filas. Devido à limitação de espaço, alguns tópicos importantes em pesquisa operacional não estão incluídos neste volume, tais como: otimização não-linear (programação não-linear), otimização multiobjetivo (exceto programação de metas em otimização linear), programação estocástica e otimização robusta, métodos de controle de estoques e de previsão de demanda, teoria de jogos e simulação discreta, que poderão ser objeto de um futuro volume.

Procuramos escrever o livro de forma a minimizar a dependência entre as seções de cada capítulo, para que o professor possa explorar o material de interesse para a disciplina que ministra. O asterisco em algumas seções é indicativo de material mais avançado, ou material apresentado de forma mais detalhada, e que pode ser omitido em cursos introdutórios. Acreditamos que o conteúdo deste livro seja suficiente para dois cursos semestrais ou um curso anual de introdução às técnicas de pesquisa operacional e suas aplicações em diversas áreas. Também pode ser utilizado em um curso semestral, selecionando alguns capítulos ou partes de cada capítulo, e como bibliografia complementar em outros cursos de engenharia de produção, como planejamento e controle da produção e logística.

Este material é resultado da nossa experiência em ministrar cursos de graduação e pós-graduação em pesquisa operacional na Universidade de São Paulo em São Carlos – USP (engenharia civil, engenharia de computação, ciências de computação e informática, matemática), Universidade Estadual de Campinas – UNICAMP (engenharia elétrica, engenharia de compu-

tação, ciência da computação), Universidade Federal de São Carlos – UFSCar (engenharias de produção materiais, produção química e produção agroindustrial, estatística) e no Instituto Nacional de Pesquisas Espaciais – INPE (computação aplicada).

Gostaríamos de agradecer diversos colegas e alunos de graduação e pós-graduação que contribuíram com comentários críticos, sugestões e revisão do material de alguns dos capítulos. Somos gratos aos professores: Maristela Santos e Franklina Toledo, da USP; Vitória Pureza, da UFSCar; Rita de Cássia Meneses Rodrigues, do INPE; Maria do Socorro Rangel, José Gilberto S. Rinaldi e Silvio A. Araujo, da Universidade Estadual Paulista – UNESP; Lana Santos, da Universidade Federal de Viçosa; e Gisele F. Pileggi, da Universidade Mackenzie. Agradecemos também aos alunos de pós-graduação: Adriana C. Cherri, Kelly C. Poldi e Carla Lucke Silva, da USP; Sara L. A. Fonseca, Luís H. Sacchi, André L. Shiguemoto e, em especial, Olinto C.B. Araújo, da UNICAMP; e Deisemara Ferreira, Ana Paula Iannoni e Cristiane A. Parreira, da UFSCar. Também agradecemos ao editor convidado, Mário O. Batalha (UFSCar), da Coleção Abepro de Engenharia de Produção, pelo apoio e incentivo durante a elaboração deste trabalho. Este livro contou com apoio financeiro parcial do CNPq.

***Marcos Nereu Arenales***, ICMC-USP ([arenales@icmc.usp.br](mailto:arenales@icmc.usp.br))

***Vinícius Amaral Armentano***, FEEC-UNICAMP ([vinicius@densis.fee.unicamp.br](mailto:vinicius@densis.fee.unicamp.br))

***Reinaldo Morabito***, DEP-UFSCar ([morabito@power.ufscar.br](mailto:morabito@power.ufscar.br))

***Horacio Hideki Yanasse***, LAC-INPE ([horacio@lac.inpe.br](mailto:horacio@lac.inpe.br))

# Apresentação

Pesquisa operacional envolve o desenvolvimento de métodos científicos para analisar sistemas complexos e tomar decisões. O papel desta disciplina tem se tornado cada vez mais importante com a crescente complexidade e incerteza no século XXI: a economia eletrônica requer decisões operacionais e táticas mais rápidas e melhores; globalização, telecomunicações e Internet definem novas relações entre clientes, fornecedores, parceiros e competidores. Tradicionalmente, pesquisa operacional tem desempenhado papel importante nas áreas de projeto, planejamento, e operação de cadeias ou redes de suprimento. No entanto, sua aplicação tem se estendido rapidamente para uma grande variedade de áreas, tais como agricultura, finanças, medicina, *marketing*, recursos naturais, energéticos e ambientais, e políticas no setor público, envolvendo serviços de saúde, educação, saúde pública, justiça criminal, serviços urbanos e segurança pública.

Três requisitos são necessários para a utilização da pesquisa operacional. O primeiro envolve a compreensão de características e atributos de um sistema complexo e a habilidade de abstrair e traduzir os pontos mais importantes em um modelo matemático ou de simulação. O segundo consiste da habilidade para desenvolver métodos de resolução para os modelos e utilizar pacotes comerciais com conhecimento sobre os métodos utilizados nestes. O terceiro envolve a comunicação com clientes para compreender o problema e explicar resultados não intuitivos, mas importantes, gerados pela aplicação de pesquisa operacional.

Este livro é direcionado, principalmente, para cursos de graduação, com destaque para a aplicação da pesquisa operacional na engenharia de produção, em particular no gerenciamento de sistemas de produção e logística, envolvendo problemas de planejamento, programação e controle da produção, problemas de distribuição de produtos, problemas de localização de facilidades e instalações, entre outros. O livro apresenta uma grande variedade de modelos com o intuito de mostrar aos estudantes a versatilidade e a aplicabilidade de modelos de pesquisa operacional, além de tratar da abordagem de métodos de resolução de classes importantes de modelos determinísticos de otimização e de modelos probabilísticos.

**Gabriel R. Bitran**

*Professor da Sloan School of Management,  
Massachusetts Institute of Technology,  
e ex-presidente da Production and Operations  
Management Society dos Estados Unidos.*

Página deixada intencionalmente em branco

# Apresentando a Coleção ABEPRO de Engenharia de Produção

O cenário competitivo no qual o Brasil está inserido demanda às instituições nacionais aprimoramento na utilização de instrumentos e conceitos modernos de gerenciamento. A importação pura e simples de modelos de gestão que obtiveram sucesso no exterior não garante igual sucesso no Brasil. Esta constatação é o resultado direto das peculiaridades dos sistemas produtivos do País.

Existe consenso no mundo acadêmico, empresarial e governamental de que tecnologias de produto e processo, quando dissociadas de sistemas de gestão adequados, não são suficientes para o aumento da competitividade. Este entendimento explica, em grande parte, o sucesso que os cursos de Engenharia de Produção vêm encontrando na comunidade acadêmica, seus empregadores e em relação à sociedade em geral. Não é raro encontrar outros cursos de Engenharia redirecionando seus currículos de maneira a enfatizar disciplinas ligadas à gestão empresarial.

Uma boa maneira de avaliar este sucesso é o aumento de novos cursos de Engenharia de Produção no Brasil. Este fato, que tem como ponto positivo o reconhecimento da importância da Engenharia de Produção para o desenvolvimento nacional, revela, no entanto, alguns perigos. Dentre eles, pode-se destacar a possibilidade do surgimento de cursos de graduação em Engenharia de Produção incapazes de transmitir o conhecimento mínimo exigido para a formação de um bom profissional. A criação de novos cursos de Engenharia de Produção fora dos centros mais tradicionais de ensino e pesquisa acentua este perigo. Muitas vezes o corpo docente responsável pelas disciplinas desses cursos tem dificuldades em acessar referências bibliográficas adequadas que assegurem a qualidade para os cursos. Desta forma, a possibilidade de encontrar em uma única obra de referência o conteúdo mínimo necessário para uma disciplina ou conjunto de disciplinas do curso de Engenharia de Produção é relevante.

Consciente deste novo contexto e também da responsabilidade para com a qualidade do ensino e da pesquisa em Engenharia de Produção, a ABEPRO (Associação Brasileira de Engenharia de Produção) desenvolve, em parceria com a Editora Campus, a Coleção ABEPRO-CAMPUS de Livros Didáticos em Engenharia de Produção. Esta Coleção reúne uma série de livros-texto que pretende cobrir as áreas essenciais da Engenharia de Produção. O conteúdo presente no conjunto de livros da Coleção ABEPRO-CAMPUS contribuirá para garantir os conhecimentos mínimos necessários à formação de um Engenheiro de Produção competente e sintonizado com os anseios nacionais de desenvolvimento.

Os conhecimentos que estes livros reúnem, em suas respectivas áreas de conhecimento, representam o conteúdo mínimo recomendável que a ABEPRO julga necessário para este campo de conhecimento. Embora os livros tenham sido organizados e escritos com a atenção voltada prioritariamente para o ensino de graduação, eles também podem ser utilizados na pós-graduação, sobretudo nos cursos de especialização.

Os livros da Coleção ABEPRO-CAMPUS de Engenharia de Produção - como é o caso desta segunda obra - são escritos por professores e pesquisadores com experiência nas suas áreas e atuando em algumas das mais renomadas instituições de ensino e pesquisa do Brasil. Desta forma, eles refletem anos de experiência destes profissionais no ensino, na pesquisa e na extensão em suas áreas de atuação. O resultado mais imediato desta situação é que os livros são ilustrados por exemplos nacionais, seja no caso dos exercícios e casos resolvidos, seja no caso dos exercícios propos-

tos. A capacidade de elaboração destas obras, com a qualidade que a ABEPRO julga desejável, é um dos resultados mais imediatos da maturidade da Engenharia de Produção brasileira.

Além de ser examinado e aprovado pelos Editores da Coleção ABEPRO-CAMPUS de Engenharia de Produção, o conteúdo do livro que está agora em suas mãos foi também analisado por dois importantes professores e pesquisadores da área. A aprovação do projeto por estes dois assessores *ad hoc*, além da aprovação também pelo corpo técnico da Editora Campus, indica a qualidade do trabalho concretizado neste livro de Pesquisa Operacional.

Este é o segundo de uma série de livros que devem cobrir os aspectos principais do ensino e da pesquisa em Engenharia de Produção. Três outros livros, igualmente em áreas centrais da Engenharia de Produção, já foram contratados e estão sendo elaborados, devendo ser lançados em breve. Disponibilizando este livro à comunidade da Engenharia de Produção do país, a ABEPRO acredita que está contribuindo para o aumento da competitividade da economia brasileira, para a ampliação da base de conhecimentos relativos às especificidades dos problemas da Engenharia de Produção nacional e, sobretudo, para a ampliação e capacitação dos recursos humanos atuantes na Engenharia de Produção do país.

Finalmente, a Coleção de Livros Didáticos em Engenharia de Produção ABEPRO-CAMPUS faz agradecimento especial aos autores. Eles souberam reconhecer a importância deste projeto para a Engenharia de Produção nacional. Assobrados por agendas carregadas, souberam encontrar o tempo necessário para escrever este livro. Em um tempo onde prioridades profissionais e pessoais se acumulam, escrever um livro é sempre um exercício de dedicação ao qual nem todos estão dispostos.

Obrigado e parabéns aos autores. A ABEPRO espera que este pioneirismo seja seguido por outros autores e que, em breve, nós vejamos capazes de disponibilizar a Coleção completa de livros à comunidade de Engenharia de Produção. Aos leitores agradecemos a confiança depositada na ABEPRO, na Editora Campus e nos autores, ao mesmo tempo em que desejamos que esta obra seja um instrumento do sucesso profissional de cada um.

*Prof. Dr. Mário Otávio Batalha*

*Prof. Dr. Osvaldo L. G. Quelhas*

Editores da Coleção de Livros Didáticos ABEPRO-CAMPUS de EP

# Sumário

## Prefácio IX

### Capítulo 1. Introdução à pesquisa operacional 1

- 1.1 Origens e definição de pesquisa operacional 1
- 1.2 Resolução de problemas com modelos matemáticos 3
- 1.3 Exemplo ilustrativo de modelagem matemática 5
- 1.4 Organização do livro 12

### Capítulo 2 – Otimização linear 15

- 2.1 Introdução 15
- 2.2 Algumas aplicações e modelagens matemáticas 15
  - 2.2.1 Problemas de mistura 15
  - 2.2.2 Problemas de transporte, transbordo e designação 21
  - 2.2.3 Problemas de planejamento da produção 26
  - 2.2.4 Problemas de programação de projetos 34
  - 2.2.5 Problemas de gestão financeira (fluxo de caixa) 35
  - 2.2.6 Problemas de meio ambiente 37
  - 2.2.7 Problemas de corte e empacotamento 39
  - 2.2.8 Ajuste de curvas 44
  - 2.2.9 Controle ótimo de sistemas lineares 46
  - 2.2.10 Problemas lineares por partes 47
- 2.3 Hipóteses de linearidade 49
- 2.4 Conceitos básicos 50
  - 2.4.1 Um problema de otimização linear 50
  - 2.4.2 Transformação de problemas na forma padrão 52
- 2.5 Resolução gráfica 54
- 2.6 Teoria básica e o método simplex 66
  - 2.6.1 Soluções básicas 67
  - 2.6.2\* O método simplex 74
  - 2.6.3\* O algoritmo simplex 84
  - 2.6.4\* Exemplos numéricos e interpretações geométricas 85
  - 2.6.5\* Considerações sobre implementações do método simplex 92
- 2.7 Método simplex em tabelas 93
- 2.8 Determinação de uma solução básica factível inicial 99
- 2.9 Programação de metas 103
- 2.10 Dualidade 109
  - 2.10.1 Relaxação lagrangiana e o problema dual 110
  - 2.10.2 Relações primais-duais 120
  - 2.10.3 Análise de sensibilidade (pós-otimização) 124
  - 2.10.4\* O método dual simplex 129
  - 2.10.5\* O algoritmo dual simplex 137
  - 2.10.6\* Reotimização após a inclusão de novas restrições 142
- 2.11 Otimização linear em teoria de jogos 146
- 2.12 Exercícios 150

### Capítulo 3. Otimização discreta 163

- 3.1 Introdução 163
- 3.2 Relaxação linear 165
- 3.3 Modelagem com variáveis binárias 167
  - 3.3.1 Implicações “se – então” 167
  - 3.3.2 Restrição ativada ou desativada 169

- 3.3.3 Restrições disjuntivas 169
- 3.3.4 Representação de função linear por partes 170
- 3.3.5 Relações lógicas 170
- 3.3.6 Representação de valores discretos 171
- 3.4 Formulações de problemas clássicos 172
  - 3.4.1 Problemas da mochila 172
  - 3.4.2 Problemas de corte 174
  - 3.4.3 Problemas de designação 178
  - 3.4.4 Problemas de cobertura, partição e empacotamento de conjuntos 181
  - 3.4.5 Problemas de caixeiro-viajante 186
  - 3.4.6 Problemas de carteiro chinês 192
- 3.5 Problemas de logística 195
  - 3.5.1 Roteamento de veículos 195
  - 3.5.2 Localização de facilidades 200
- 3.6 Problemas de produção 205
  - 3.6.1 Planejamento da produção 206
  - 3.6.2 Dimensionamento e programação de lotes 213
  - 3.6.3 Programação da produção 215
  - 3.6.4 Outros problemas de programação 227
- 3.7\* Complexidade computacional 232
- 3.8\* Formulações alternativas 234
- 3.9 Métodos de enumeração implícita e de planos de corte 239
  - 3.9.1 Método *branch-and-bound* 239
  - 3.9.2 Algoritmo de planos de cortes de Gomory 251
  - 3.9.3 Método *branch-and-cut* 256
- 3.10\* Pré-processamento 260
- 3.11\* Outros métodos em otimização discreta 263
- 3.12 Notas 273
- 3.13 Exercícios 275

#### **Capítulo 4. Otimização em redes 289**

- 4.1 Noções básicas de redes e grafos 289
- 4.2 Algumas aplicações e métodos de resolução 297
  - 4.2.1 O problema de caminho mínimo 297
  - 4.2.2 O problema de caminho máximo 327
  - 4.2.3 O problema da árvore geradora mínima 331
  - 4.2.4 Problemas de fluxo 335
- 4.3\* O método simplex para redes 350
- 4.4 Exercícios 368

#### **Capítulo 5. Programação dinâmica determinística 375**

- 5.1 Um exemplo simples 375
  - 5.1.1 Cálculos do exemplo 377
  - 5.1.2 Determinação do plano ótimo 380
- 5.2 Indução progressiva 381
  - 5.2.1 Cálculos da indução progressiva 382
  - 5.2.2 Determinação do plano ótimo com a indução progressiva 385
- 5.3 Formalização da programação dinâmica 386
  - 5.3.1 Conceitos básicos 386
  - 5.3.2 Princípio da otimalidade 387
  - 5.3.3 Otimização recursiva 388
  - 5.3.4 Exemplo de aplicação de programação dinâmica com custos não-lineares 389
- 5.4 Indução da decomposição na programação dinâmica 391
- 5.5 O problema da dimensionalidade em programação dinâmica 402
- 5.6 Exercícios 403

**Capítulo 6. Programação dinâmica estocástica 407**

- 6.1 A equação recursiva regressiva 407
- 6.2 Exemplos numéricos 408
  - 6.2.1 Controle de estoque 408
  - 6.2.2 Jogo de soma zero 414
- 6.3\* Processos markovianos de decisão 423
  - 6.3.1 Probabilidade de transição 425
  - 6.3.2 Exemplo de manutenção de equipamentos 426
- 6.4 Exercícios 428

**Capítulo 7. Sistemas de filas e otimização 433**

- 7.1 Definição e classificação de um sistema de filas 435
- 7.2 Medidas de desempenho de um sistema de filas 439
  - 7.2.1 Análise em equilíbrio 441
  - 7.2.2 Fórmula de Little 442
- 7.3 Modelo de filas fundamental 445
  - 7.3.1 A relação das distribuições exponencial e Poisson 445
  - 7.3.2 Modelo de nascimento e morte 449
- 7.4 Modelos de filas com um servidor e distribuições exponenciais 454
  - 7.4.1 Modelo de fila  $M/M/1/GD/\infty/\infty$  – um servidor 454
  - 7.4.2 Modelo de fila  $M/M/1/GD/K/\infty$  – um servidor e capacidade limitada 459
  - 7.4.3 Modelo de fila  $M/M/1/GD/\infty/N$  – um servidor e população finita 461
  - 7.4.4 Modelos de decisão de filas – um servidor 462
- 7.5 Modelos de filas com múltiplos servidores e distribuições exponenciais 463
  - 7.5.1 Modelo de fila  $M/M/m/GD/\infty/\infty$  – múltiplos servidores 463
  - 7.5.2 Modelo de fila  $M/M/\infty/GD/\infty/\infty$  – infinitos servidores (auto-serviço) 466
  - 7.5.3 Modelo de fila  $M/M/m/GD/K/\infty$  (com  $m \leq K$ ) – múltiplos servidores com capacidade limitada 467
  - 7.5.4 Modelo de fila  $M/M/m/GD/m/\infty$  – múltiplos servidores sem fila de espera 470
  - 7.5.5 Modelo de fila  $M/M/m/GD/K/K$  (com  $m \leq K$ ) – múltiplos servidores com capacidade limitada e população finita (manutenção de máquinas) 470
  - 7.5.6 Modelos de decisão de filas – múltiplos servidores 473
- 7.6 Outros modelos de filas com distribuições exponenciais 474
  - 7.6.1 Modelo com desistência na chegada 474
  - 7.6.2 Modelo com abandono na espera ou com aceleração do servidor 474
- 7.7 Modelos de filas com distribuições genéricas 476
  - 7.7.1 Modelo de fila  $M/G/1/GD/\infty/\infty$  – um servidor 476
  - 7.7.2 Modelo de fila  $M/G/1/GD/\infty/\infty$  – um servidor e chegadas em lote 478
  - 7.7.3 Modelo de fila  $M/G/m/GD/m/\infty$  – múltiplos servidores sem fila de espera 478
  - 7.7.4 Modelo de fila  $M/G/\infty/GD/\infty/\infty$  – infinitos servidores (auto-serviço) 479
  - 7.7.5 Modelos de fila  $G/G/1/GD/\infty/\infty$  e  $G/G/m/GD/\infty/\infty$  480
- 7.8\* Modelos de filas com múltiplas classes de usuários e prioridade 483
  - 7.8.1 Modelo de fila  $M/G/1/NPRP/\infty/\infty$  – um servidor e prioridade sem interrupção 483
  - 7.8.2 Modelo de fila  $M/M/1/PRP/\infty/\infty$  – um servidor e prioridade com interrupção 485
  - 7.8.3 Modelo de fila  $M/M/m/NPRP/\infty/\infty$  – múltiplos servidores e prioridade sem interrupção 487
- 7.9\* Modelos de redes de filas 488
  - 7.9.1 Modelos de redes de filas com distribuições exponenciais 489
  - 7.9.2 Modelos de decisão em rede de filas 493
- 7.10 Exercícios 494
- 7.11 Resumo dos resultados do capítulo 497

Apêndice – Elementos de sistemas de equações lineares 501

Referências Bibliográficas 509

Índice Remissivo 519

Página deixada intencionalmente em branco

# Introdução à pesquisa operacional

## 1.1 ORIGENS E DEFINIÇÃO DE PESQUISA OPERACIONAL

O termo *pesquisa operacional* é uma tradução (brasileira) direta do termo em inglês *operational research*, que em Portugal foi traduzido por *investigação operacional* e nos países de língua hispânica, por *investigación operativa*. O surgimento deste termo está ligado à invenção do radar na Inglaterra em 1934. Dois anos depois, o ministério britânico da Aviação criou a Estação de Pesquisa Manor Bawdsey, em Suffolk, para estudar como a tecnologia do radar poderia ser utilizada para interceptar aviões inimigos. O termo pesquisa operacional é atribuído ao superintendente da estação A. P. Rowe,<sup>1</sup> que, em 1938, coordenava equipes para examinar a eficiência de técnicas de operações advindas de experimentos com interceptação de radar. Em 1941, foi inaugurada a Seção de Pesquisa Operacional do Comando da Força Aérea de Combate, com equipes envolvidas em problemas de operações de guerra, como manutenção e inspeção de aviões, escolha do tipo de avião para uma missão e melhoria na probabilidade de destruição de submarinos. Outros problemas incluíam controle de artilharia antiaérea e dimensionamento de comboios de frota. A análise científica do uso operacional de recursos militares de maneira sistemática foi iniciada na Segunda Guerra Mundial.

Após o final da guerra, a pesquisa operacional evoluiu rapidamente na Inglaterra e nos Estados Unidos. Em 1947, foi implantado o projeto SCOOP (*Scientific Computation of Optimal Programs*) no Pentágono, com o objetivo de apoiar decisões de operações na força aérea americana. O projeto continha um grupo de pesquisa coordenado pelo economista Marshall Wood e pelo matemático George Dantzig. Durante esse projeto, Dantzig desenvolveu, formalizou e testou o método simplex para resolver problemas de programação linear (otimização linear). Esses desenvolvimentos se basearam em trabalhos precursores, notadamente do matemático russo Leonid Kantorovich<sup>2</sup>, e o termo *programação linear* foi sugerido a Dantzig pelo economista T. C. Koopmans (tanto Koopmans quanto Kantorovich ganharam posteriormente o prêmio Nobel de economia de 1975, pelas contribuições à teoria de alocação ótima de recursos).

Em 1952, foi fundada a sociedade científica americana de pesquisa operacional (ORSA – *Operations Research Society of America*) e, em 1953, a sociedade inglesa de pesquisa operacional (ORS – *Operational Research Society*) e a americana de ciências de administração (TIMS – *The Institute of Management Sciences*). Em 1957, foi realizada a primeira conferência internacional de pesqui-

1 Ver, por exemplo, Kirby (2003).

2 O trabalho de 1939 de Kantorovich (intitulado “Métodos matemáticos na organização e no planejamento de produção”) é considerado um dos precursores da pesquisa operacional, ao apontar que uma das maneiras de se aumentar a eficiência em uma empresa seria por meio de melhorias na organização e no planejamento da produção.

sa operacional em Oxford, na Inglaterra. Nessa conferência, foi possível constatar claramente os focos diferentes nos trabalhos apresentados pelos cientistas ingleses e pelos cientistas americanos. Os trabalhos dos ingleses enfatizaram estudos de caso ou problemas específicos, enquanto os trabalhos dos americanos abordavam modelos e métodos matemáticos (metodologias) em diversos temas, tais como teoria de estoques, substituição de equipamentos, teoria de filas, programação (*scheduling*) de tarefas em máquinas, teoria de jogos, fluxos em redes e otimização linear.

A partir do início da década de 1950 até o final da década de 1960, a pesquisa operacional foi aplicada em uma variedade de problemas oriundos dos setores público e privado. Em um levantamento feito em 1953 com 160 organizações da Grã-Bretanha, 45 responderam que possuíam um departamento de pesquisa operacional, ou pelo menos uma pessoa nessa atividade. Exemplos de aplicações envolviam diversos setores industriais e financeiros, como: mineração, metalúrgico, construção civil e militar, têxtil, farmacêutico, bancário e transportes. Em 1967, foram identificados 766 grupos de pesquisa operacional, dos quais 553 estavam envolvidos no setor industrial ou em serviços financeiros. Exemplos de aplicações no setor público envolviam coleta de lixo, transporte e polícia, entre outros. Desde então, a pesquisa operacional tem sido aplicada às mais diversas áreas de produção e logística, incluindo indústrias de alimentação, automóveis, aviação, computadores, eletrônica, metalurgia, mineração, mísseis, móveis, papel, petróleo, telecomunicações, transportes; além de organizações de serviço (públicas e privadas), como: bancos e seguradoras, hospitais, bibliotecas, sistemas judiciais, agências de viagens e turismo, energia, esportes, trânsito, agências de governo (federais, estaduais e municipais) etc.

Do ponto de vista educacional, na década de 1960, a pesquisa operacional era estudada apenas em cursos de pós-graduação, mas, a partir da década de 1970, passou a ser objeto também de cursos de graduação. Os primeiros livros influentes na área de otimização foram: *Linear Programming: Methods and Applications* (Gass, 1958), *Applied Dynamic Programming* (Bellman e Dreyfus, 1962), *Flows in Networks* (Ford e Fulkerson, 1963) e *Linear Programming and Extensions* (Dantzig, 1963). Neste último, no Capítulo 1 é apresentada uma classificação dos problemas de otimização, além da seguinte definição do problema de programação: “construir um enunciado de ações a serem executadas, os instantes da execução, suas quantidades (chamado programa ou *schedule*), que permite que um sistema evolua de um dado estado a um objetivo definido”. Na língua inglesa, *schedule* e *program* são sinônimos e, no contexto da época, a meta era programar atividades de sistemas complexos. Isso gerou a área de *programação matemática*, em que a programação linear é um caso particular de otimização de funções lineares sujeito a restrições lineares.

Atualmente, existem várias sociedades científicas em diversos países que agregam pessoas e entidades interessadas na teoria e prática da pesquisa operacional. Por exemplo, na Inglaterra existe a anteriormente citada ORS; nos Estados Unidos, o INFORMS (*Institute for Operations Research and the Management Sciences*), criado em 1995 agregando as sociedades anteriores ORSA e TIMS; na Alemanha, a GOR (*German Operations Research Society*); no Canadá, a CORS (*Canadian Operations Research Society*); e, em Portugal, a APDIO (Associação Portuguesa de Investigação Operacional). Existem também grupos regionais de associações de sociedades científicas de pesquisa operacional, como a ALIO (*Asociación Latino-Ibero-Americana de Investigación Operativa*), que reúne doze sociedades científicas, a EURO (Associação das Sociedades de Pesquisa Operacional da Europa), que reúne 29 sociedades científicas, a APORS (Associação das Sociedades de Pesquisa Operacional da Ásia e do Pacífico), que reúne dez sociedades científicas, e a NORAM (Associação das Sociedades de Pesquisa Operacional da América do Norte). Há também a Federação Internacional das Sociedades de Pesquisa Operacional (IFORS – *International Federation of Operational Research Societies*), com 52 membros de sociedades nacionais e mais os grupos regionais.

A pesquisa operacional iniciou-se no Brasil basicamente na década de 1960. O primeiro simpósio brasileiro de pesquisa operacional foi realizado em 1968 no ITA, em São José dos Campos,

SP. Em seguida, foi fundada a SOBRAPO (Sociedade Brasileira de Pesquisa Operacional), que publica o periódico científico *Pesquisa Operacional* há mais de 25 anos. O livro *25 Anos de Pesquisa Operacional no Brasil*, edição comemorativa do Jubileu de Prata da SOBRAPO, lançado em 1993, contém relatos da história da pesquisa operacional no Brasil e de pesquisadores pioneiros no país.<sup>3</sup>

Uma definição de pesquisa operacional foi proposta na primeira página do periódico inglês *Operational Research Quarterly* em 1967, que, de forma resumida, consiste no desenvolvimento de métodos científicos de sistemas complexos, com a finalidade de prever e comparar estratégias ou decisões alternativas. O objetivo é dar suporte à definição de políticas e determinação de ações de forma científica. Para alguns autores, pesquisa operacional significa abordagem científica para tomada de decisões, que procura determinar como melhor projetar e operar um sistema, usualmente sob condições que requerem a alocação de recursos escassos.<sup>4</sup> A sociedade europeia EURO define pesquisa operacional como uma abordagem científica para a solução de problemas no gerenciamento de sistemas complexos. De forma sucinta, podemos dizer que pesquisa operacional é um enfoque científico sobre a tomada de decisões. A denominação pesquisa operacional é comumente motivo de críticas e reflexões, pois não revela a abrangência da área e pode dar a falsa impressão de estar limitada à análise de operações. Alguns autores sugerem outras denominações preferíveis, por exemplo, análise de decisões, mas o termo pesquisa operacional é bastante difundido no âmbito das engenharias (em particular, da engenharia de produção), ciências de administração, ciências de computação, economia, estatística e matemática, entre outros, de modo que, apesar das limitações, preferimos usar este termo no presente livro.

Mais recentemente, a pesquisa operacional também tem sido chamada de ciência e tecnologia de decisão.<sup>5</sup> O componente científico está relacionado a idéias e processos para articular e modelar problemas de decisão, determinando os objetivos do tomador de decisão e as restrições sob as quais se deve operar. Também está relacionado a métodos matemáticos para otimizar sistemas numéricos que resultam quando se usam dados nos modelos. O componente tecnológico está relacionado a ferramentas de *software* e *hardware* para coletar e comunicar dados, e organizar esses dados, usando-os para gerar e otimizar modelos e reportar resultados, ou seja, a pesquisa operacional está se tornando um importante elemento nas metodologias de tecnologia da informação. A sociedade americana INFORMS atualmente define pesquisa operacional e ciências de administração como disciplinas profissionais que tratam da aplicação da tecnologia da informação para a tomada de decisões informada.

## 1.2 RESOLUÇÃO DE PROBLEMAS COM MODELOS MATEMÁTICOS

Se fazer ciência é a capacidade de observar e descrever fenômenos naturais, sociais, econômicos, entre outros, a matemática tem uma importância fundamental na descrição desses fenômenos. A partir da observação de fenômenos, processos ou sistemas, que podem ser físicos, químicos, biológicos, econômicos, buscam-se *leis* que os regem. Essas leis, se passíveis de serem descritas por relações matemáticas, dão origem aos *modelos matemáticos*. O termo *modelo* neste texto é usado como objeto abstrato, que procura imitar as principais características de um objeto real para fins de representar o objeto real.

Em geral, para formular um modelo matemático, simplificações razoáveis do sistema ou problema real precisam ser consideradas (em diferentes níveis) e a validação do modelo depende de a solução do modelo matemático ser coerente com o contexto original. Com isso, o modelo matemático é uma representação simplificada (abstração) do problema real. Ele deve ser suficiente-

<sup>3</sup> Em Campello *et al.* (2003) e Bornstein *et al.* (2004), estão descritos os feitos de outros dois pesquisadores que deram uma contribuição significativa para o crescimento e a consolidação da pesquisa operacional no país.

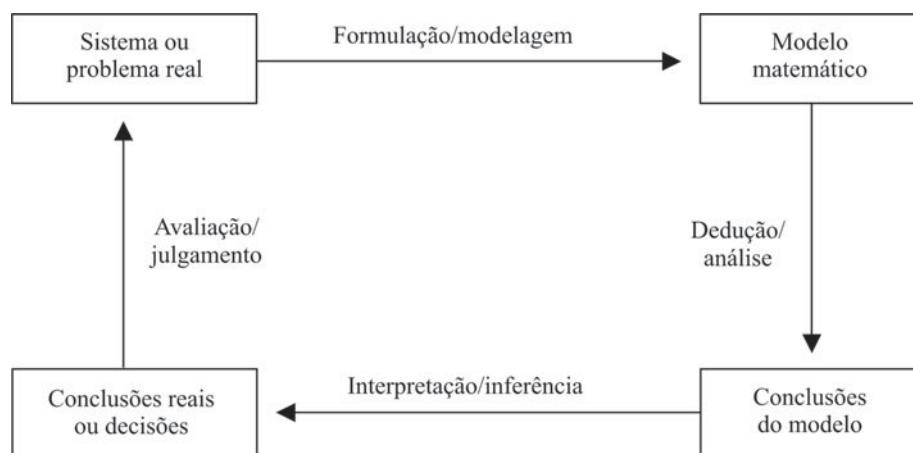
<sup>4</sup> Veja, por exemplo, Winston (2004).

<sup>5</sup> Veja, por exemplo, Shapiro (2001).

mente detalhado para captar os elementos essenciais do problema, mas suficientemente tratável por métodos de resolução. O diagrama da Figura 1.1 ilustra um processo simplificado da abordagem de solução de um problema usando a modelagem matemática.<sup>6</sup>

A formulação (modelagem) define as variáveis e as relações matemáticas para descrever o comportamento relevante do sistema ou problema real. A dedução (análise) aplica técnicas matemáticas e tecnologia para resolver o modelo matemático e visualizar quais conclusões ele sugere. A interpretação (inferência) argumenta que as conclusões retiradas do modelo têm significado suficiente para inferir conclusões ou decisões para o problema real. Freqüentemente, uma avaliação (julgamento) dessas conclusões ou decisões inferidas mostra que elas não são adequadas e que a definição do problema e sua modelagem matemática precisam de revisão e, então, o ciclo é repetido. Exemplos de modelos matemáticos são os modelos de programação matemática (otimização matemática), como: programação linear (otimização linear), programação linear inteira (otimização discreta), programação em redes (otimização em redes) e programação não-linear (otimização não-linear). Outros exemplos são modelos de teoria de filas para estudar a congestão em sistemas e determinar medidas de avaliação de desempenho e políticas ótimas de operação.

A pesquisa operacional e, em particular, a programação matemática tratam de problemas de decisão e faz uso de modelos matemáticos que procuram representar (em certo sentido, imitar) o problema real. Variáveis (incógnitas) são definidas e relações matemáticas entre essas variáveis são estabelecidas de forma a descrever o comportamento do sistema. O modelo matemático é resolvido (isto é, são determinados valores para as incógnitas, produzindo soluções, que dependem de dados do problema) e o passo seguinte consiste na validação do modelo, isto é, verificar se as soluções obtidas pela resolução do modelo matemático, para diversas situações alternativas (por exemplo, alterando-se dados do problema como demandas, custos etc.) são compatíveis com a realidade, conforme evidenciado na Figura 1.1. A solução do modelo apóia o processo de tomada de decisões, mas em geral diversos outros fatores pouco tangíveis, não quantificáveis, também devem ser levados em consideração para a decisão final (por exemplo, soluções que não levem em conta o comportamento humano podem falhar). Convém salientar que modelos não substituem tomadores de decisão.



**Figura 1.1** Processo de modelagem.

<sup>6</sup> Embora se concentre em modelos matemáticos, o diagrama da Figura 1.1 também pode ser interpretado para outros modelos de pesquisa operacional, por exemplo, modelos de simulação discreta.

Alguns autores sugerem que pesquisa operacional é tanto “ciência” quanto “arte”: ciência por causa das técnicas matemáticas envolvidas (objetivo), arte porque o sucesso de todas as fases que precedem e sucedem a solução do modelo matemático depende muito da criatividade e experiência do pessoal de pesquisa operacional (subjetivo). A abordagem de resolução de um problema por meio de pesquisa operacional envolve várias fases baseadas no diagrama da Figura 1.1: (i) definição do problema; (ii) construção do modelo; (iii) solução do modelo; (iv) validação do modelo; e (v) implementação da solução.

A fase (i) define o escopo do problema em estudo. A fase (ii) traduz a fase (i) em relações matemáticas ou lógicas de simulação, ou uma combinação delas. A fase (iii) utiliza métodos de solução e algoritmos<sup>7</sup> conhecidos para resolver o modelo da fase (ii). A fase (iv) verifica se o modelo proposto representa apropriadamente o problema, ou seja, se o modelo prediz adequadamente o comportamento do sistema. A qualidade da solução de um modelo depende da precisão com que este modelo representa o problema. Um modelo mais preciso, mesmo que resolvido de forma aproximada, pode ser bem mais útil do que um modelo menos preciso resolvido de forma exata. A qualidade da solução do modelo também depende da qualidade dos dados de entrada do modelo; a conhecida expressão inglesa *garbage in, garbage out* (entra lixo, sai lixo) sintetiza isso. Finalmente, a fase (v) preocupa-se com a implementação da solução na prática, traduzindo os resultados do modelo em decisões.

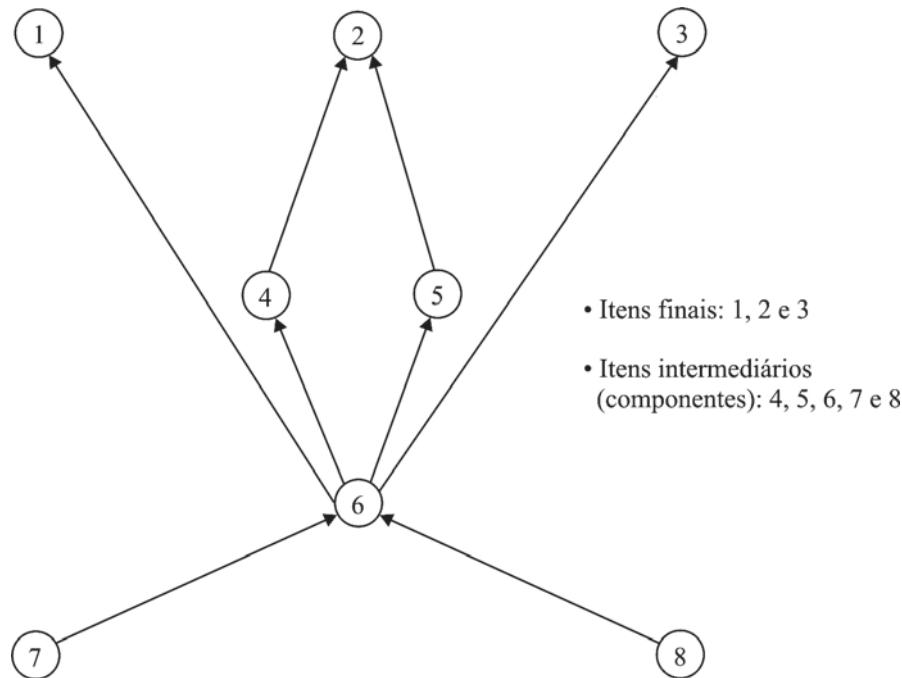
Obviamente, um modelo matemático nem sempre é formulado de uma só vez e podemos ter ciclos entre as fases (i)–(v) para revisão do modelo. Equívocos não são incomuns, mesmo para os mais experientes modeladores. Algumas vezes a análise da solução do modelo é suficiente para evidenciar suas inconsistências e, então, reformulamos o modelo; outras vezes, somente após a operação do modelo na prática é que somos capazes de detectar falhas e, então, repetimos todo o ciclo.

Com o aprimoramento dos métodos de solução dos modelos e o desenvolvimento tecnológico dos computadores e da informática (principalmente a partir dos anos 80, com a popularização dos microcomputadores), tem sido possível resolver modelos de pesquisa operacional cada vez mais complexos, outrora intratáveis. O leitor poderá se questionar sobre a necessidade do estudo desses métodos diante das facilidades do uso direto dos diversos *softwares* comerciais já existentes, baseados nas técnicas de pesquisa operacional. Entretanto, nem sempre nos deparamos com problemas de aplicação direta nesses aplicativos computacionais, sendo necessário um certo domínio da teoria em que se baseia o método. Mesmo com a facilidade do uso desses aplicativos, o desconhecimento da abrangência e das limitações do aplicativo inibe seu uso ou pode conduzir a um uso equivocado. Embora não seja necessário entender a eletrônica de uma máquina de calcular para se fazer bom uso dela, nem ser conhecedor da teoria dos números, qual seria a importância de uma máquina de calcular para uma pessoa sem noção das operações aritméticas e de como operá-la?

### 1.3 EXEMPLO ILUSTRATIVO DE MODELAGEM MATEMÁTICA

**Sistema MRP:** O sistema computacional MRP II (*Manufacturing Resource Planning*) é amplamente usado para o planejamento e controle de sistemas de manufatura. Este sistema é uma versão expandida do sistema MRP (*Material Requirements Planning*) e inclui finanças, contabilidade e outros aspectos comerciais. O MRP é usado para determinar quanto e quando produzir ou comprar componentes e matérias-primas para produzir itens finais. Na Figura 1.2, mostra-se uma estrutura em forma de árvore dos itens componentes 4, 5, 6, 7 e 8, usados para produzir os itens finais 1, 2 e 3. Esta árvore possui quatro níveis: o nível 1 corresponde aos itens finais 1, 2 e 3; o nível 2 está associado aos componentes 4 e 5; o nível 3 ao componente 6; e o nível 4 aos componentes 7 e 8.

<sup>7</sup> Um algoritmo é uma seqüência finita e não ambígua de instruções computáveis para solucionar um problema.



**Figura 1.2** Estrutura de itens finais e componentes.

De modo geral, a lógica do sistema MRP é composta de duas fases interativas, descritas a seguir a partir da Figura 1.2:

*Fase I:* Suponha que a demanda externa dos itens finais seja conhecida ao longo de um horizonte de planejamento<sup>8</sup> de  $T$  períodos, e que um lote de um item componente esteja disponível no mesmo período em que foi encomendado, para facilitar a explicação. Calcula-se, então, o tamanho dos lotes dos itens finais em cada período do horizonte de planejamento que geram demandas internas dos componentes 4 e 5 do nível 2. Prosseguindo desta forma, calcula-se o tamanho dos lotes dos componentes 6 no nível 3 e o tamanho dos lotes dos componentes 7 e 8 no nível 4 em cada período ao longo do horizonte de planejamento. O cálculo dos tamanhos dos lotes de todos os itens ignora restrições de capacidade, tais como horas de trabalho e horas de máquinas disponíveis.

*Fase II:* Calcula-se a capacidade utilizada para a produção dos lotes de cada item em cada período. Se essa capacidade for maior que a capacidade disponível em um ou mais períodos, retorna-se à fase I e alteram-se os planos de produção, isto é, os tamanhos dos lotes dos itens finais e itens componentes ao longo do horizonte de planejamento. Esse procedimento interativo entre as duas fases prossegue até que se ache uma solução que respeite a capacidade disponível. Esse procedimento pode consumir muito tempo computacional ou, ainda, pode não encontrar uma solução que atenda as restrições de capacidade.

Para ilustrar essas duas fases, considere um exemplo simples de produção de dois itens:<sup>9</sup> um item final, denotado por item 1, que possui um único item componente, denotado por item 2. Uma unidade do item 1 requer uma unidade do item 2. Considere que o estoque inicial dos dois itens é zero, o horizonte de planejamento consiste em 4 semanas, e que 80 horas de trabalho estão dis-

<sup>8</sup> Horizonte de planejamento é o tempo em que a empresa planeja sua produção.

<sup>9</sup> Este exemplo foi inspirado em Drexl e Kimms (1997).

poníveis a cada semana. A demanda do item 1 em cada semana é conhecida e deve ser atendida pela produção ou estoque do item 1. O custo total de um plano de produção, ao longo do horizonte de planejamento, compreende:

- Custo fixo e custo variável de produção de cada item em cada semana: o custo fixo independe da quantidade produzida e está associado à preparação de máquinas, enquanto o custo variável depende do tamanho do lote.
- Custo de estoque: quando um lote de um item é produzido em uma semana para atender a demanda de uma semana futura.

O problema de decisão pode ser enunciado da seguinte forma: quanto deve ser produzido de cada item em cada semana de forma a minimizar o custo total, respeitando a restrição de capacidade? A solução deste problema está longe de ser trivial, e envolve um balanceamento entre os custos de produção e estoque. Ao se ignorar a restrição de capacidade, tem-se dois planos de produção extremos com relação a estes custos. O plano de produção que atende exatamente a demanda de cada semana leva a um custo zero de estoque, porém, em geral, a um alto custo de produção. O outro plano corresponde a produzir na primeira semana a demanda total, isto é, a demanda das 4 semanas. Esse plano, em geral, corresponde a um baixo custo de produção, mas resulta em um alto custo de estoque. Provavelmente, o melhor plano de produção consiste em decisões intermediárias, com produção em alguns períodos do horizonte de planejamento.

A seguinte notação é utilizada para a descrição de um modelo matemático. Itens são representados por  $i$ ,  $i = 1, 2$ , e períodos de tempo são denotados por  $t$ ,  $t = 1, 2, 3, 4$ . Considere os seguintes parâmetros do problema:

- $d_{1t}$  = demanda do item 1 no período  $t$ ;
- $h_i$  = custo unitário de estoque por período do item  $i$ ;
- $S_i$  = custo fixo de produção do item  $i$ ;
- $c_i$  = custo unitário de produção do item  $i$ ;
- $a_i$  = tempo de trabalho para produzir uma unidade do item  $i$ .

Os valores dos parâmetros do exemplo são mostrados na Tabela 1.1.

A modelagem de um problema de pesquisa operacional e, em particular, de programação matemática envolve três aspectos fundamentais.

- Definição das *decisões* a serem tomadas.
- *Restrições* que limitam as escolhas das decisões.
- *Objetivos* que determinam preferências na escolha de decisões.

Voltando ao nosso exemplo, quais as decisões a serem tomadas? Como já ressaltado, as decisões compreendem o tamanho dos lotes produzidos dos itens 1 e 2 em cada semana. A quantidade em estoque dos itens 1 e 2 também envolve decisões que dependem das decisões acerca dos

**Tabela 1.1**  
**Dados do exemplo.**

	$d_{1t}$				$h_i$	$S_i$	$a_i$	$c_i$
	$t = 1$	$t = 2$	$t = 3$	$t = 4$				
$i = 1$	30	25	25	20	25	900	1	1
$i = 2$	-	-	-	-	10	850	1	1

tamanhos dos lotes produzidos em cada semana. Quais são as restrições que limitam o tamanho dos lotes dos itens 1 e 2? A capacidade disponível em horas de trabalho em cada período restringe as decisões. Além disso, a demanda precisa ser atendida sem atraso. O que faz uma decisão ser melhor que outra? No caso deste exemplo, o objetivo é minimizar a soma dos custos de produção e estoque. Portanto, a decisão que leve ao valor mínimo desse custo total é a preferida.

Considere as variáveis de decisão:

$x_{it}$  = quantidade do item  $i$  produzida no período  $t$ ;

$I_{it}$  = estoque do item  $i$  no final do período  $t$  (igual ao estoque no início do período  $t+1$ ).

Defina o custo de produção do item  $i$  no período  $t$  como:

$$K_{it}(x_{it}) = \begin{cases} S_i + c_i x_{it} & \text{se } x_{it} > 0 \\ 0 & \text{se } x_{it} = 0. \end{cases}$$

O modelo matemático de otimização para este exemplo pode ser formulado da seguinte maneira: minimizar a soma dos custos  $K_{11}(x_{11}) + h_1 I_{11} + K_{12}(x_{12}) + h_1 I_{12} + \dots + K_{24}(x_{24}) + h_2 I_{24}$ , ou seja:

$$\text{Minimizar} \quad \sum_{i=1}^2 \sum_{t=1}^4 K_{it}(x_{it}) + h_i I_{it} \quad (1.1)$$

sujeito às restrições:

$$I_{1t} = I_{1,t-1} + x_{1t} - d_{1t} \quad t = 1, 2, 3, 4 \quad (1.2)$$

$$I_{2t} = I_{2,t-1} + x_{2t} - x_{1t} \quad t = 1, 2, 3, 4 \quad (1.3)$$

$$a_1 x_{1t} + a_2 x_{2t} \leq 80 \quad t = 1, 2, 3, 4 \quad (1.4)$$

$$x_{it} \geq 0, I_{it} \geq 0, i = 1, 2 \quad t = 1, 2, 3, 4 \quad (1.5)$$

com  $I_{10} = I_{20} = 0$ , isto é, o estoque inicial dos itens no início do período  $t = 1$  é nulo.

A expressão (1.1) é chamada de *função objetivo* e, neste modelo, expressa a minimização dos custos de produção e de estoque ao longo do horizonte de planejamento. As *restrições* (1.2) e (1.3) correspondem ao balanceamento de estoque dos itens 1 e 2 em cada período  $t$ , ou seja, o estoque de cada item no final do período  $t$  deve ser igual ao estoque desse item no final do período anterior  $t-1$ , mais a quantidade produzida desse item no período  $t$ , menos a quantidade demandada desse item no período  $t$ . As restrições (1.4) garantem que o consumo de capacidade em cada período  $t$  é limitado a 80 horas por semana, e as restrições (1.5) indicam que as variáveis são não-negativas. Note que as variáveis  $x_{1t}$  e  $x_{2t}$  são dependentes, pois competem pela capacidade disponível em cada período.

Uma *solução factível* (ou viável) para o modelo (1.1)–(1.5) é uma escolha para os valores das variáveis que satisfaz todas as restrições. Uma *solução ótima* é uma solução factível que minimiza a função objetivo (1.1). Pode-se ter uma ou múltiplas soluções ótimas para um modelo, todas com o mesmo valor da função objetivo. Um *método ótimo* (ou exato) para um modelo de otimização é aquele que gera uma solução ótima. Em casos em que o modelo é complexo, determinar uma solução ótima em tempo computacional razoável (considerando as decisões envolvidas) pode ser muito difícil ou até impossível com as ferramentas e técnicas conhecidas na atualidade, então, podemos utilizar um *método heurístico* (ou não-exato) para gerar uma boa solução (*solução subótima*) para o modelo.

As fases I e II descritas anteriormente constituem um procedimento para tentar obter uma solução factível, não necessariamente ótima. Para o exemplo anterior, a fase I pode ser descrita mais detalhadamente pelos seguintes passos:

*Passo 1:* Aplique um método para achar a solução ótima para o item 1 isoladamente, isto é, resolva o subproblema:

$$\begin{aligned} \text{Minimizar} \quad & \sum_{t=1}^4 K_{1t}(x_{1t}) + h_1 I_{1t} \\ I_{1t} = & I_{1,t-1} + x_{1t} - d_{1t} \quad t = 1, 2, 3, 4 \\ x_{1t} \geq & 0, I_{1t} \geq 0 \quad t = 1, 2, 3, 4 \end{aligned}$$

e denote a solução ótima por  $x_{1t}^*, I_{1t}^*, t = 1, 2, 3, 4$ .

*Passo 2:* Aplique um método para achar a solução ótima para o item 2 isoladamente, isto é, resolva o subproblema:

$$\begin{aligned} \text{Minimizar} \quad & \sum_{t=1}^4 K_{2t}(x_{2t}) + h_2 I_{2t} \\ I_{2t} = & I_{2,t-1} + x_{2t} - x_{1t}^* \quad t = 1, 2, 3, 4 \\ x_{2t} \geq & 0, I_{2t} \geq 0 \quad t = 1, 2, 3, 4 \end{aligned}$$

e denote a solução ótima por  $x_{2t}^*, I_{2t}^*, t = 1, 2, 3, 4$ . Note que os valores  $x_{1t}^*, t = 1, 2, 3, 4$ , tornam-se demandas para a equação de balanceamento de estoque do item 2 em cada período  $t$ . Essas demandas são chamadas de demandas dependentes na literatura sobre MRP.

*Passo 3:* Verifique se a restrição de capacidade (1.4) é satisfeita, isto é, se:

$$a_1 x_{1t}^* + a_2 x_{2t}^* \leq 80 \quad t = 1, 2, 3, 4.$$

Se estas restrições de capacidade são satisfeitas, então a solução  $x_{1t}^*, I_{1t}^*, x_{2t}^*, I_{2t}^*, t = 1, 2, 3, 4$ , é uma solução factível, não necessariamente ótima. Caso contrário, o procedimento falha, pois não gera uma solução factível para o problema.

Os passos 1–3 constituem um método heurístico para resolução do modelo (1.1)–(1.5) que, ao contrário de um método ótimo, não garante que a solução encontrada seja factível (mesmo que o problema tenha pelo menos uma solução factível) ou ótima. Existem heurísticas propostas na literatura que formalizam procedimentos adicionais de transferência de quantidades de produção dos itens entre os períodos, em busca de uma solução factível e de baixo custo (fase II). A heurística descrita anteriormente é razoável no sentido de que decompõe um problema de múltiplos itens interdependentes em subproblemas independentes, um para cada item. No entanto, sua desvantagem é não levar em consideração a restrição de capacidade (1.4). Mesmo que essa restrição não exista no problema, este procedimento ainda é heurístico, pois a solução ótima dos subproblemas independentes não corresponde, em geral, à solução ótima do problema original.

A seguir, aplicamos ao exemplo os passos de 1 a 3 da heurística. O método que gera a solução de custo mínimo para cada item nos passos 1 e 2 é apresentado em detalhes na seção 5.4 do Capítulo 5. A solução obtida no passo 1 é mostrada na Tabela 1.2. O plano de produção do item 1 é a demanda dependente do item 2, e na Tabela 1.3 mostramos o plano de produção do item 2. Ao executarmos o passo 3, observamos que a capacidade consumida na produção dos itens 1 e 2 é de 110 horas no período 1 e de 90 horas no período 3, excedendo a capacidade disponível de 80 horas em cada período. Este poderia ser o plano de produção gerado por um *software* comercial MRP, alertando que o plano não é factível do ponto de vista da capacidade disponível.

**Tabela 1.2**  
**Plano de produção do item 1 obtido no passo 1.**

Item 1	Período	1	2	3	4
	Demanda	30	25	25	20
	Plano de produção	55	0	45	0
	Estoque	25	0	20	0

**Tabela 1.3**  
**Plano de produção do item 2 obtido no passo 2.**

Item 2	Período	1	2	3	4
	Demanda	55	0	45	0
	Plano de produção	55	0	45	0
	Estoque	25	0	20	0

Para gerar um plano factível a partir do plano anterior, analisemos inicialmente a antecipação da produção de 10 unidades do item 1 do período 3 para o período 2 (a antecipação para o período 1 violaria ainda mais a capacidade disponível). Essa transferência faz com que a restrição de capacidade no período 3 seja atendida. No entanto, não existem 10 unidades do item 2 para atender essa demanda de 10 unidades do item 1, a menos que a produção do item 2 no período 1 seja elevada de 10 unidades, o que aumentaria a violação de capacidade nesse período. Portanto, essa antecipação gera uma solução infactível.

Analisemos agora o adiamento da produção do item 1. A quantidade máxima de produção do item 1 que pode ser transferida do período 1 para o período 2 é de 25 unidades, devido à demanda do item 1 no período 1. Transferindo-se essa quantidade para o período 2, o novo plano de produção do item 1 é mostrado na Tabela 1.4. A capacidade necessária no período 1 para a produção dos itens 1 e 2 é de 85 unidades, 5 unidades a mais do que a disponível. Para eliminar esse excesso de capacidade de 5 unidades, transferem-se 5 unidades de produção do item 2 do período 1 para o período 2, como mostrado na Tabela 1.5.

Para eliminar o excesso de capacidade de 10 unidades do período 3, podemos transferir 45 unidades do item 2 para o período 2, eliminando assim o gasto de custo fixo desse item no período 3. A solução mostrada nas Tabelas 1.4 e 1.6<sup>10</sup> é factível e tem custo 5.750. Note que 5 unidades do item 2 estão desnecessariamente em estoque no fim do período 1, e a transferência dessas 5 unidades para o período 2 resulta em outra solução factível com custo 5.700, conforme mostrado na Tabela 1.7. Pode-se mostrar que a solução das Tabelas 1.4 e 1.7 é ótima aplicando-se um dos métodos ótimos descritos no Capítulo 3 para resolver o modelo (1.1)–(1.5).

Este exemplo ilustra a conveniência e a importância de utilizar modelos de pesquisa operacional. Uma análise adicional importante é verificar como essa solução se altera em função de possíveis perturbações nos valores dos parâmetros do problema (*análise de sensibilidade*), por exemplo, devido a incertezas nas demandas e capacidades dos períodos. Com o uso de modelos, essa análise pode ser feita, por exemplo, resolvendo-se o modelo para diferentes cenários (valores de parâmetros) e comparando-se as soluções obtidas.

10 Solução apresentada em Drexel e Kimms (1997).

**Tabela 1.4**  
**Plano de produção do item 1 – factível.**

Item 1	Período	1	2	3	4
	Demanda	30	25	25	20
	Plano de produção	30	25	45	0
	Estoque	25	0	20	0

**Tabela 1.5**  
**Plano de produção do item 2 – revisado.**

Item 2	Período	1	2	3	4
	Demanda	30	25	45	0
	Plano de produção	50	5	45	0
	Estoque	20	0	20	0

**Tabela 1.6**  
**Plano de produção do item 2 – factível.**

Item 2	Período	1	2	3	4
	Demanda	30	25	45	0
	Plano de produção	50	50	0	0
	Estoque	20	45	0	0

**Tabela 1.7**  
**Plano de produção do item 2 – ótimo.**

Item 2	Período	1	2	3	4
	Demanda	30	25	45	0
	Plano de produção	45	55	0	0
	Estoque	15	45	0	0

## 1.4 ORGANIZAÇÃO DO LIVRO

Este livro está organizado em duas partes: a Parte I estuda modelos determinísticos de programação matemática, como: otimização linear (programação linear), otimização discreta (programação linear inteira), otimização em redes (fluxos em redes) e programação dinâmica determinística, juntamente com métodos de solução e algoritmos. No Capítulo 2, estudamos a otimização linear, que envolve modelos com função objetivo e restrições lineares. Apresentamos exemplos de aplicações de modelos lineares em problemas de mistura, problemas de transporte, transbordo e designação, problemas de planejamento da produção (incluindo mix de produtos, seleção de processos, dimensionamento de lotes), problemas de programação de projetos, problemas de gestão financeira, problemas de gestão do meio ambiente, problemas de corte e empacotamento de materiais, problemas de ajuste de curvas, problemas de controle ótimo, entre outros. Para resolver os modelos lineares, apresentamos em detalhes o método simplex e o algoritmo primal simplex. Apresentamos também uma versão mais resumida do método simplex em tabelas, como uma alternativa para disciplinas de menor ênfase em algoritmos. Discutimos também em detalhes a teoria de dualidade, a análise de sensibilidade (pós-otimização), o algoritmo dual simplex, a programação de metas e uma introdução da otimização linear na teoria dos jogos.

No Capítulo 3, estudamos a otimização discreta, que envolve modelos com função objetivo e restrições lineares; porém, com variáveis que só podem assumir valores inteiros (0, 1, 2, ...). Apresentamos exemplos de aplicações de modelos lineares inteiros em problemas da mochila, problemas de corte e empacotamento de materiais, problemas de designação, problemas de cobertura, partição e empacotamento de conjuntos, problemas logísticos de caixeiro-viajante e carteiro chines, problemas de roteamento<sup>11</sup> e programação de veículos, problemas de localização de facilidades<sup>12</sup> e instalações (incluindo  $p$ -medianas,  $p$ -centros), problemas de planejamento da produção (incluindo dimensionamento de lotes e dimensionamento de lotes combinado com programação da produção) e problemas de programação (*scheduling*) da produção (incluindo uma máquina, máquinas paralelas, *flow-shop* e *job-shop*), entre outros. Para resolver os modelos lineares inteiros, apresentamos em detalhes o algoritmo *branch-and-bound*, o algoritmo de planos de corte e o algoritmo *branch-and-cut*. Apresentamos também uma introdução aos métodos de Benders, Dantzig-Wolfe, relaxação lagrangiana e métodos heurísticos.

No Capítulo 4, estudamos a otimização (fluxo) em redes, que trata de problemas que podem ser modelados como uma rede (grafo). Introduzimos noções básicas da teoria de grafos e fluxos em redes e apresentamos alguns exemplos de aplicações de modelos de redes em problemas de caminho mínimo, problemas de caminho máximo, problemas de árvore geradora mínima, problemas de fluxo de custo mínimo, problemas de transporte, entre outros. Para resolver os modelos em redes, apresentamos em detalhes os algoritmos de Dijkstra, Floyd, Ford e Fulkerson e o método simplex adaptado para redes.

No Capítulo 5, estudamos a programação dinâmica determinística, em que o problema original é decomposto em subproblemas menores. Apresentamos exemplos de aplicação da técnica de programação dinâmica em problemas de planejamento da produção, dimensionamento de lotes de produção e problemas da mochila, ilustrando a utilização de fórmulas recursivas progressivas e regressivas. Na seção de exercícios desse capítulo, também propomos exemplos de aplicações em problemas de gestão financeira, gestão de mão-de-obra, programação da produção e problemas de otimização linear e não-linear.

<sup>11</sup> O termo *roteirização* de veículos, embora não encontrado nos dicionários de língua portuguesa, tem sido usado como equivalente à palavra inglesa *routing* (ou *routeing*). O termo *roteamento* de veículos também é utilizado alternativamente por diversos autores brasileiros.

<sup>12</sup> A palavra *facilities* é mais abrangente em inglês, com significado mais amplo que *instalações*, que são prédios onde são gerados produtos e/ou serviços. Embora a palavra *facilidade* não tenha o mesmo significado de *facilities*, ela é utilizada por diversos autores brasileiros.

Na Parte II do livro tratamos de modelos probabilísticos que consideram incertezas, como modelos de programação dinâmica estocástica e teoria de filas. No Capítulo 6, estudamos a programação dinâmica estocástica (ou probabilística). Assim como o caso determinístico, a programação dinâmica estocástica decompõe o problema original em subproblemas menores. Apresentamos exemplos de aplicação dessa técnica em problemas de controle de estoques e no jogo da velha. Discutimos também o processo markoviano de decisão com uma aplicação ao problema de manutenção de equipamentos. Na seção de exercícios desse capítulo, outros exemplos de aplicações são propostos em problemas de distribuição, confiabilidade de sistemas, planejamento da produção e gestão de recursos.

Finalmente, no Capítulo 7, estudamos a teoria de filas, também chamada de teoria de congestionamento. Essa teoria estuda sistemas de filas de espera, determinando medidas de desempenho para esses sistemas, tais como: número médio de usuários na fila, tempo médio de espera em fila, utilização dos dispositivos de serviço etc. Também discutimos alguns modelos de otimização usando teoria de filas. Apresentamos modelos de filas com um e múltiplos servidores (em paralelo), uma e múltiplas filas (em paralelo), um e múltiplos estágios em série e combinação em série e paralelo (redes de filas). Discutimos modelos com processos de chegada e serviço dos usuários com distribuições de probabilidade exponenciais e modelos com distribuições de probabilidade genéricas. Também apresentamos modelos com múltiplas classes de usuários e prioridades de atendimento diferentes para cada classe. Os modelos são ilustrados com exemplos de aplicações em sistemas de manufatura e sistemas de serviço.

Devido a limitações de espaço, alguns tópicos importantes em pesquisa operacional não estão incluídos neste volume, tais como: a otimização não-linear (programação não-linear), em que as funções do modelo são não-lineares; a otimização multiobjetivo (além de programação de metas em otimização linear), em que o modelo envolve vários objetivos em conflito, a programação estocástica e a otimização robusta, os métodos de controle de estoques e de previsão de demanda, a teoria de jogos e a simulação discreta, que poderão ser objeto de um futuro volume.

Página deixada intencionalmente em branco

# 2

## Otimização linear

### 2.1 INTRODUÇÃO

Este capítulo é dedicado ao estudo da otimização linear e abrange aplicações, teoria e desenvolvimento dos principais métodos de solução. Os modelos de otimização linear têm sido amplamente utilizados na prática. Muitas situações práticas podem ser representadas por modelos desse tipo; também é comum encontrarmos esses modelos representando subproblemas de casos mais complexos. Embora tais modelos já tivessem sido propostos há mais tempo, o ano de 1947 foi um marco na área da otimização, quando o *método simplex* foi publicado, seguindo-se intensas pesquisas de novos métodos e implementações eficientes e aplicações em diversas áreas, tais como agricultura, planejamento da produção industrial, logística, telecomunicações, finanças e muitas outras. Outro marco importante em otimização linear ocorreu em 1984, com a publicação de um *método de pontos interiores*, ao qual também seguiram-se intensas pesquisas. Os métodos do tipo *simplex* e do tipo *pontos interiores* são, atualmente, as principais ferramentas computacionais para a resolução de problemas de otimização linear.

### 2.2 ALGUMAS APLICAÇÕES E MODELAGENS MATEMÁTICAS

Exemplos de problemas que podem ser formulados como um problema de otimização linear aparecem nas mais variadas áreas. Nesta seção, apresentamos alguns exemplos que, embora simplificados, fornecem uma idéia concreta dos problemas tratados pela otimização linear.

#### 2.2.1 Problemas de mistura

Problemas deste tipo consistem em combinar materiais obtidos na natureza (ou restos de outros já combinados anteriormente) para gerar novos materiais ou produtos com características convenientes. Estão entre os primeiros problemas de otimização linear implementados com sucesso na prática. A seguir apresentamos alguns exemplos particulares desta classe de problemas.

#### Rações

Fábricas de rações produzem vários tipos de rações para determinados animais, como bovinos, equinos, caninos de pequeno e grande portes, felinos, galináceos etc. Essas rações são produzidas pela mistura de alimentos ou farinhas de restos de alimentos: milho, farelo de arroz, farinha de osso, soja, farinha de peixe, entre outros (na prática, são dezenas de ingredientes utilizados),

cujos preços de mercado são conhecidos. A composição nutricional desses ingredientes é conhecida, isto é, a quantidade de proteína, cálcio, ferro, manganês, calorias etc.; na prática, o número de nutrientes é de cerca de duas dezenas. A nutrição veterinária especifica as necessidades mínimas e máximas desses nutrientes por quilo de ração para cada tipo de animal. Um problema de otimização surge para determinar quais devem ser as quantidades ideais de cada ingrediente por quilo de cada ração de modo que as necessidades nutricionais sejam atendidas e o custo total dos ingredientes seja o menor possível. Note que as necessidades nutricionais impõem restrições, de modo que nem toda mistura de ingredientes é aceitável, e o custo é o critério para se caracterizar a melhor solução.

Problemas de mistura surgem em vários outros processos industriais, como na produção de adubos, sucos concentrados de laranja, ligas metálicas etc. e geralmente são parte de um problema mais geral de planejamento e controle da produção.<sup>1</sup> Apresentamos a seguir mais alguns exemplos do problema da mistura.

### Ligas metálicas

Fundições produzem diversos tipos de aço a partir de vários insumos, como: lingotes de ferro, grafite, sucatas industriais de diversos tipos de aço, entre outros. Esses insumos são colocados em um forno de alta temperatura, no qual em estado líquido, se misturam para formar uma liga metálica, isto é, uma mistura. A composição da liga a ser produzida, em termos de carbono, silício, manganês etc., é determinada por normas técnicas da metalurgia (depende da aplicação da liga), bem como a composição dos produtos a serem misturados. Os preços dos insumos podem variar substancialmente e são conhecidos. Deve-se determinar as quantidades de cada insumo a serem fundidas, de modo que a composição da liga obtida satisfaça as normas técnicas da metalurgia e que o preço final da liga seja o menor possível.

### Composição de areias para filtro

Devido à grande permeabilidade, areias são usadas na constituição de filtros de Estações de Tratamento de Águas de abastecimento (ETA) como meio filtrante por interceptar impurezas existentes na água afluente. Para a construção das unidades de filtração de uma ETA, dispõe-se de areias provenientes de alguns portos passíveis de exploração com composições granulométricas distintas. Os custos de dragagem, transporte, seleção e preparo para a utilização da areia variam para cada porto. As areias devem ser dispostas em camadas que devem obedecer às composições granulométricas estabelecidas por norma, em quantidades predeterminadas. O problema consiste em combinar os volumes de areia provenientes de cada porto, de modo a atender às especificações da norma, com o menor custo possível.

### Formulação matemática do problema da mistura

O problema da mistura, em geral, consiste na obtenção ou fabricação de um *produto*, aqui chamado de *mistura*, combinando-se alguns materiais disponíveis na natureza ou disponíveis no mercado (no exemplo da ração, pode ser milho, farinha de osso etc.). A mistura é produzida a partir de ingredientes que possuem os *componentes* desejados no novo produto (na ração, os componentes são proteínas, vitaminas etc.) e que devem satisfazer determinadas especificações (na ração, as especificações são resultados de pesquisa em nutrição animal). A composição de cada ingrediente é conhecida, isto é, as proporções dos componentes de cada ingrediente são dadas, como também o seu custo unitário. Deseja-se determinar as quantidades de cada ingrediente que devemos utilizar para obter uma mistura com a composição especificada e com o menor custo possível.

<sup>1</sup> Para uma aplicação em indústrias cítricas veja, por exemplo, Munhoz e Morabito (2001).

Seja  $n$  o número de ingredientes que podem ser utilizados na produção da mistura e  $m$  o número de componentes relevantes para a mistura. Um dos passos fundamentais para se escrever um modelo matemático é identificar as incógnitas (variáveis do problema), ou seja, o que se deseja determinar. No problema da mistura, as variáveis são as quantidades dos ingredientes. Assim, definimos a variável:

$x_j$ : a quantidade do ingrediente  $j$  que deve ser utilizada em uma unidade de mistura,  $j=1, 2, \dots, n$ .

Uma unidade de mistura pode ser, por exemplo, 1kg. Essas variáveis devem ser não-negativas, pois um valor negativo para  $x_j$  não tem significado, isto é:  $x_j \geq 0, j = 1, 2, \dots, n$  são restrições do modelo. Para escrever as demais restrições, relativas à composição da mistura e seu custo, usamos a seguinte notação:

$a_{ij}$  a fração do componente  $i$  no ingrediente  $j$ ,  $i = 1, \dots, m$  e  $j = 1, \dots, n$ ,

$b_i$  a fração do componente  $i$  na mistura,  $i = 1, \dots, m$ ,

$c_j$  o custo de uma unidade do ingrediente  $j$ ,  $j = 1, \dots, n$ .

Como  $a_{ij}$  é a fração do componente  $i$  no ingrediente  $j$ , isto é, a quantidade do componente  $i$  em uma unidade da mistura, então  $a_{ij}x_j$  é a quantidade do componente  $i$  em  $x_j$  unidades do ingrediente  $j$ . Portanto,  $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n$  é a quantidade total do componente  $i$  em uma unidade da mistura (ver as hipóteses de linearidade na Seção 2.3). Como a quantidade do componente  $i$  em uma unidade da mistura deve ser  $b_i$ , escrevemos

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i \quad i = 1, 2, \dots, m.$$

Nas  $m$  equações anteriores, supõe-se que não há alterações na composição dos ingredientes quando estes se misturam. Por exemplo, no caso da ração, a quantidade total de cálcio na mistura é a soma das quantidades de cálcio presentes nos ingredientes. Essas hipóteses são fundamentais para se escrever um modelo linear e são discutidas na Seção 2.3.

Como  $x_j, j = 1, \dots, n$ , são as quantidades dos ingredientes a serem utilizadas em uma unidade da mistura, então a soma dessas quantidades deve resultar em uma unidade da mistura, ou seja,  $x_1 + \dots + x_n = 1$ .

O custo de uma unidade da mistura é a soma dos custos de todos os ingredientes utilizados para sua obtenção, ou seja,  $c_1x_1 + c_2x_2 + \dots + c_nx_n$ .

Desejamos minimizar esse custo, portanto, o problema da mistura é escrito como:

$$\begin{aligned} \text{Minimizar } & f(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n \\ & a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ & a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ & \dots \\ & a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \\ & x_1 + x_2 + \dots + x_n = 1 \\ & x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0. \end{aligned}$$

**Exemplo 2.1** Uma agroindústria deve produzir um tipo de ração para determinado animal. Essa ração é produzida pela mistura de farinhas de três *ingredientes* básicos: osso, soja e resto de peixe. Cada um desses três ingredientes contém diferentes quantidades de dois *nutrientes* necessários a uma dieta nutricional balanceada: proteína e cálcio. O nutricionista especifica as necessidades mínimas desses nutrientes em 1kg de ração. Cada ingrediente é adquirido no mercado com um certo custo unitário (\$/kg). Na Tabela 2.1, os dados do problema são apresentados. Por exemplo, a farinha de osso é constituída de 20% de proteína e 60% de cálcio; a ração deve ser composta de *pelo menos* 30% de proteína e 50% de cálcio; 1kg da farinha de osso custa \$0,56

**Tabela 2.1**  
**Dados para o problema da ração.**

Nutrientes	Ingredientes			Ração
	Ossو	Soja	Peixe	
Proteína	0,2	0,5	0,4	0,3
Cálcio	0,6	0,4	0,4	0,5
Custos (\$/kg)	0,56	0,81	0,46	

(os ingredientes podem ser constituídos por outros elementos, mas que não são importantes para o problema em questão).

Deve-se determinar em que quantidades os ingredientes devem ser misturados de modo a produzir uma ração que satisfaça às restrições nutricionais com o mínimo custo. Defina a variável de decisão  $x_j$  como a quantidade (em kg) do ingrediente  $j$  que deve ser utilizada em uma unidade (1kg) da ração,  $j = \text{osso, soja, peixe}$ . Com isso, o custo da mistura é dado por:

$$f(x_{\text{osso}}, x_{\text{soja}}, x_{\text{peixe}}) = 0,56 x_{\text{osso}} + 0,81 x_{\text{soja}} + 0,46 x_{\text{peixe}}$$

e as restrições de composição são dadas por:

$$\begin{aligned} 0,2 x_{\text{osso}} + 0,5 x_{\text{soja}} + 0,4 x_{\text{peixe}} &\geq 0,3 \\ 0,6 x_{\text{osso}} + 0,4 x_{\text{soja}} + 0,4 x_{\text{peixe}} &\geq 0,5. \end{aligned}$$

Observe a pequena diferença no enunciado, que resulta em restrições de desigualdade (em vez de igualdade), pois foram estabelecidos percentuais mínimos dos componentes na mistura.

Temos também que a soma dos ingredientes resulta em uma unidade da mistura, ou seja,

$$x_{\text{osso}} + x_{\text{soja}} + x_{\text{peixe}} = 1$$

e que esses ingredientes podem ser utilizados ou não, isto é,

$$x_{\text{osso}} \geq 0, \quad x_{\text{soja}} \geq 0, \quad x_{\text{peixe}} \geq 0.$$

O modelo matemático completo fica, então:

$$\begin{aligned} \text{Minimizar } f(x_{\text{osso}}, x_{\text{soja}}, x_{\text{peixe}}) &= 0,56 x_{\text{osso}} + 0,81 x_{\text{soja}} + 0,46 x_{\text{peixe}} \\ 0,2 x_{\text{osso}} + 0,5 x_{\text{soja}} + 0,4 x_{\text{peixe}} &\geq 0,3 \\ 0,6 x_{\text{osso}} + 0,4 x_{\text{soja}} + 0,4 x_{\text{peixe}} &\geq 0,5 \\ x_{\text{osso}} + x_{\text{soja}} + x_{\text{peixe}} &= 1 \\ x_{\text{osso}} \geq 0, \quad x_{\text{soja}} \geq 0, \quad x_{\text{peixe}} \geq 0. \end{aligned}$$

A melhor solução, chamada solução ótima deste modelo, é dada por:  $x^*_{\text{osso}} = 0,5$ ,  $x^*_{\text{soja}} = 0$  e  $x^*_{\text{peixe}} = 0,5$ , o que significa que a ração deve ser constituída de 50% de farinha de osso e 50% de farinha de peixe. O leitor pode determinar o custo dessa solução e procurar por soluções alternativas. Um dos principais objetivos deste capítulo é o de fornecer ferramentas para determinar soluções ótimas de modelos de otimização linear.

O exemplo apresentado, bem como os demais que se seguem, tem como objetivo apenas ilustrar uma situação. Os dados utilizados não correspondem a valores reais, que podem ser obtidos em tabelas nutricionais ou por análise laboratorial. ■

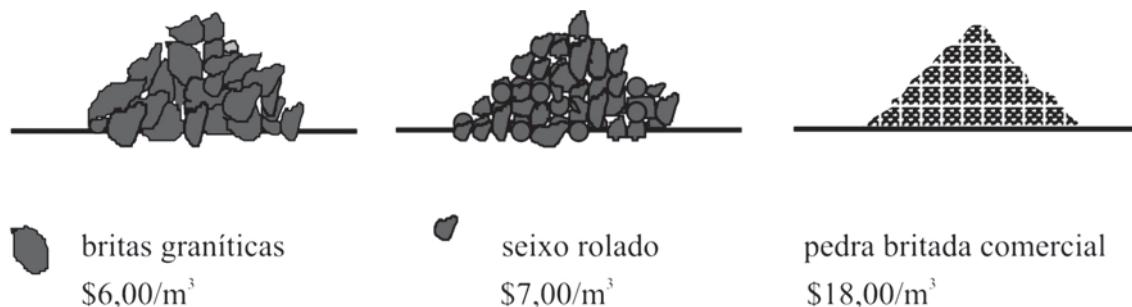
Para uma refeição humana,<sup>2</sup> na qual a palatabilidade é importante, além das necessidades nutricionais, deve-se considerar novas restrições, como a proporcionalidade entre os alimentos. Por

<sup>2</sup> Bornstein e Namen (2004) apresentam uma revisão dos modelos de dietas e desenvolvem uma ferramenta computacional para avaliação dos modelos.

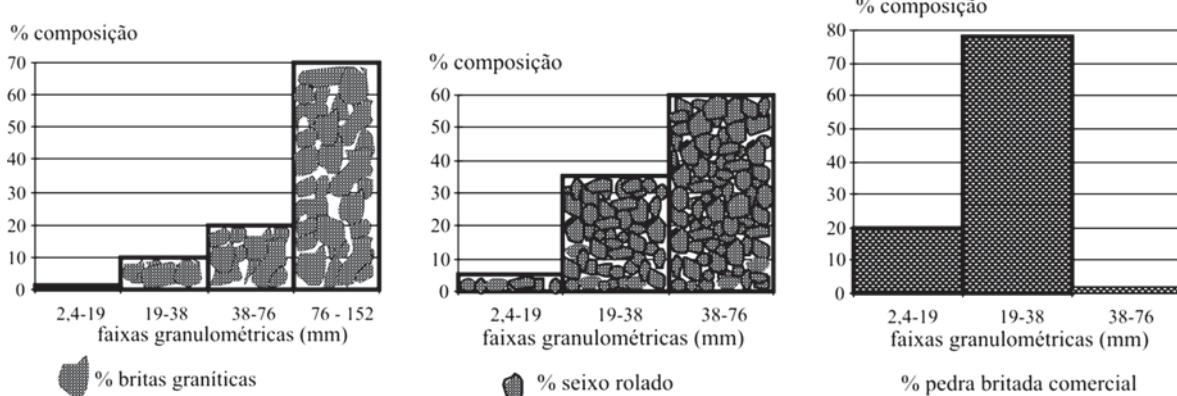
exemplo, uma refeição pode incluir uma salada composta dos ingredientes: maionese, salsa, maçã e batata. Sejam  $x_1, x_2, x_3$  e  $x_4$  as quantidades em gramas desses ingredientes na salada (consequentemente, na refeição). Tais ingredientes devem aparecer nas restrições nutricionais, contribuindo para o valor nutricional da refeição. Entretanto, a quantidade total da salada deve estar entre 100g e 200g e deve ser composta de 4% a 6% de maionese, 4% a 6% de salsa, 30% a 40% de maçã e 50% a 60% de batata. Note que a quantidade de salada é dada por:  $x_1 + x_2 + x_3 + x_4$ . Assim, deve-se adicionar ao modelo da mistura as seguintes restrições:

$$\begin{aligned} 0,04(x_1 + x_2 + x_3 + x_4) &\leq x_1 \leq 0,06(x_1 + x_2 + x_3 + x_4) \\ 0,04(x_1 + x_2 + x_3 + x_4) &\leq x_2 \leq 0,06(x_1 + x_2 + x_3 + x_4) \\ 0,30(x_1 + x_2 + x_3 + x_4) &\leq x_3 \leq 0,40(x_1 + x_2 + x_3 + x_4) \\ 0,50(x_1 + x_2 + x_3 + x_4) &\leq x_4 \leq 0,60(x_1 + x_2 + x_3 + x_4) \\ 100 &\leq x_1 + x_2 + x_3 + x_4 \leq 200. \end{aligned}$$

**Exemplo 2.2** Na implantação de uma barragem de grande consumo de concreto, decidiu-se utilizar como fontes de agregados graúdos: (i) britas graníticas obtidas pelo desmonte da rocha local; (ii) seixos rolados disponíveis nos vales próximos à barragem; e (iii) pedra britada comercial, ilustrados na Figura 2.1. Os custos baseados na exploração, britagem, transporte e preparo para a utilização desses agregados e composições granulométricas estão nas Figuras 2.1 e 2.2.



**Figura 2.1** Agregados graúdos.



**Figura 2.2** Composições granulométricas dos agregados graúdos.

Com o objetivo de otimizar a qualidade do concreto, chegou-se, por meio de pesquisas baseadas na compacidade (número de vazios intergranulares) e ensaios laboratoriais, a uma faixa de *composição granulométrica ideal* para a dosagem desses agregados, ilustrada na Figura 2.3.

O problema consiste em determinar as frações de cada fonte para a constituição da *composição granulométrica ideal*, de forma a minimizar o custo da produção de concreto.

As variáveis de decisão são:

- $x_1$  quantidade ( $m^3$ ) de britas graníticas obtidas por desmonte de rochas
- $x_2$  quantidade ( $m^3$ ) de seixo rolado
- $x_3$  quantidade ( $m^3$ ) de brita comercial.

Com isso, o custo da mistura é dado por  $f(x_1, x_2, x_3) = 6x_1 + 7x_2 + 18x_3$ .

A restrição  $0,05x_2 + 0,20x_3 \geq 0,10$  impõe que a mistura tenha uma proporção *mínima* de 10% de agregado de faixa granulométrica entre 2,4 e 19mm na mistura (veja Figuras 2.2 e 2.3).

Da mesma forma, considerando as demais granulometrias, obtemos:

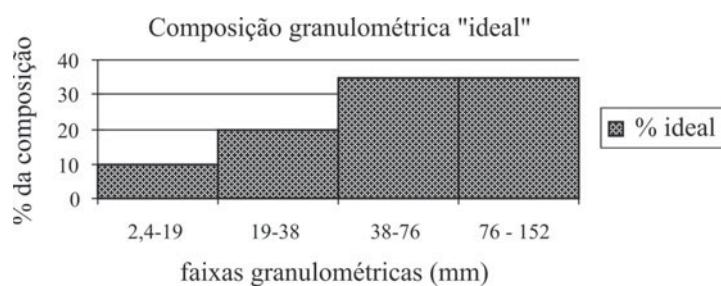
$$\begin{aligned} 0,10x_1 + 0,35x_2 + 0,78x_3 &\geq 0,20 & (19-38\text{mm}) \\ 0,20x_1 + 0,60x_2 + 0,02x_3 &\geq 0,35 & (38-76\text{mm}) \\ 0,70x_1 &\geq 0,35 & (76-152\text{mm}). \end{aligned}$$

Os ingredientes utilizados produzem uma unidade de mistura, ou seja,  $x_1 + x_2 + x_3 = 1$ , e, completando as restrições, temos as condições de não-negatividade das variáveis:  $x_1 \geq 0$ ,  $x_2 \geq 0$ ,  $x_3 \geq 0$ . O modelo matemático completo é dado por:

$$\begin{aligned} \text{Minimizar } f(x_1, x_2, x_3) &= 6x_1 + 7x_2 + 18x_3 \\ 0,05x_2 + 0,20x_3 &\geq 0,10 \\ 0,10x_1 + 0,35x_2 + 0,78x_3 &\geq 0,20 \\ 0,20x_1 + 0,60x_2 + 0,02x_3 &\geq 0,35 \\ 0,70x_1 &\geq 0,35 \\ x_1 + x_2 + x_3 &= 1 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{aligned}$$
■

Observação: Nem sempre um problema de otimização tem uma solução possível. No exemplo anterior novas fontes de agregados graúdos, com características diversas, podem ser consideradas, o que significa novas variáveis que podem proporcionar soluções alternativas.

Nos exercícios propostos no final deste capítulo são apresentadas algumas extensões para o problema da mistura, como a produção de várias misturas a partir do mesmo conjunto de ingredientes, os quais estão disponíveis em estoque em quantidades limitadas ou, ainda, quando a demanda pela mistura ocorre ao longo de um horizonte de planejamento de  $T$  períodos.



**Figura 2.3** Faixas granulométricas ideais.

## 2.2.2 Problemas de transporte, transbordo e designação

Esses problemas referem-se, por exemplo, ao transporte ou distribuição de produtos dos centros de produção aos mercados consumidores. Os produtos podem ser os mais variados possíveis: petróleo, equipamentos, máquinas, produção agrícola, energia elétrica etc. O problema consiste em transportar o produto dos centros de produção aos mercados consumidores de modo que o custo total de transporte seja o menor possível. Admite-se, geralmente, que as quantidades produzidas ou ofertadas em cada centro e as quantidades demandadas em cada mercado consumidor são conhecidas. O transporte deve ser efetuado respeitando-se as limitações de oferta e atendendo à demanda.

### Formulação matemática do problema de transporte

Denominemos os centros de produção de *origens* e os mercados consumidores, de *destinos*. Suponha que existam  $m$  origens e  $n$  destinos para um produto e que o custo de transportar uma unidade desse produto da origem  $i$  para o destino  $j$  é  $c_{ij}$ . Além disso, a oferta do produto na origem  $i$  é  $a_i$  e a demanda do produto no destino  $j$  é  $b_j$ . Um esquema gráfico para representar as origens, os destinos, as disponibilidades do produto e as demandas é apresentado na Figura 2.4.

As variáveis do problema são as quantidades transportadas das origens aos destinos:

$x_{ij}$  quantidade transportada do produto da origem  $i$  para o destino  $j$ .

Essas quantidades não podem ser negativas, portanto, as restrições  $x_{ij} \geq 0$ ,  $i = 1, \dots, m$  e  $j = 1, \dots, n$ , fazem parte do modelo matemático. Se  $x_{ij}$  é a quantidade transportada do produto da origem  $i$  para o destino  $j$ , então  $c_{ij}x_{ij}$  é o custo incorrido para se realizar esse transporte. O custo total de transporte é a soma dos custos de transporte de todas as quantidades transportadas de

todas as origens  $i$  a todos os destinos  $j$ , ou seja,  $\sum_{i=1}^m \sum_{j=1}^n c_{ij}x_{ij}$ . Esse custo deve ser minimizado. Sabemos que o que é transportado de cada origem  $i$  a todos os destinos  $j$ ,  $j = 1, 2, \dots, n$ , não pode ultrapassar a quantidade disponível de produto na origem  $i$ , ou seja,

$$\sum_{j=1}^n x_{ij} \leq a_i.$$

Desejamos também que as quantidades transportadas das diversas origens ao destino  $j$  satisfaçam a demanda requerida neste destino, ou seja,

$$\sum_{i=1}^m x_{ij} = b_j.$$

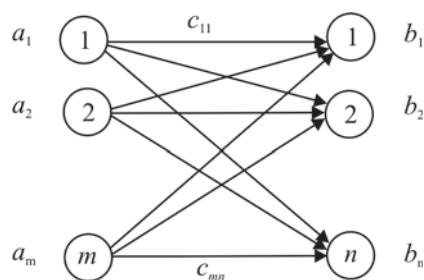


Figura 2.4 Rede de transporte.

O modelo completo do problema de transporte é definido por:

$$\begin{aligned}
 \text{Minimizar} \quad f(x_{11}, x_{12}, \dots, x_{mn}) &= \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\
 \sum_{j=1}^n x_{ij} &\leq a_i \quad i = 1, \dots, m \\
 \sum_{i=1}^m x_{ij} &= b_j \quad j = 1, \dots, n \\
 x_{ij} &\geq 0 \quad i = 1, \dots, m \text{ e } j = 1, \dots, n.
 \end{aligned}$$

**Exemplo 2.3** Considere uma companhia distribuidora de bebidas que tem 2 centros de produção ( $m = 2$ ) – Araraquara e São José dos Campos – e 3 mercados consumidores principais ( $n = 3$ ) – São Paulo, Belo Horizonte e Rio de Janeiro. O custo unitário ( $c_{ij}$ ) de se transportar uma unidade do produto de cada centro de produção a cada mercado consumidor é dada na Tabela 2.2. Nessa tabela, também são apresentadas as demandas ( $b_j$ ) de cada mercado e a quantidade máxima disponível do produto em cada centro de produção ( $a_i$ ) no próximo período.

Definindo a variável  $x_{ij}$  como a quantidade do produto a ser enviada do centro de produção  $i$ ,  $i = 1$  (Araraquara), 2 (São José dos Campos), ao mercado  $j$ ,  $j = 1$  (São Paulo), 2 (Belo Horizonte), 3 (Rio de Janeiro), o modelo que representa o problema é dado por:

$$\text{Minimizar} \quad f(x_{11}, \dots, x_{23}) = 4x_{11} + 2x_{12} + 5x_{13} + 11x_{21} + 7x_{22} + 4x_{23} \quad (2.1)$$

$$x_{11} + x_{12} + x_{13} \leq 800 \quad (2.2)$$

$$x_{21} + x_{22} + x_{23} \leq 1000 \quad (2.3)$$

$$x_{11} + x_{21} = 500 \quad (2.4)$$

$$x_{12} + x_{22} = 400 \quad (2.5)$$

$$x_{13} + x_{23} = 900 \quad (2.6)$$

$$x_{11} \geq 0, x_{12} \geq 0, x_{13} \geq 0, x_{21} \geq 0, x_{22} \geq 0, x_{23} \geq 0. \quad (2.7)$$

A função em (2.1) representa o custo total de se transportar o produto dos centros de produção aos mercados. As restrições (2.2) e (2.3) indicam que a quantidade que pode ser transportada não pode ultrapassar a disponibilidade do produto no centro de produção. As restrições (2.4), (2.5) e (2.6) impõem que as demandas dos mercados sejam atendidas, ou seja, os produtos devem ser transportados dos diversos centros produtores em quantidade suficiente para suprir as

**Tabela 2.2**  
**Dados para o problema de transporte da companhia distribuidora de bebidas.**

Centro de suprimento	Mercado			Suprimento disponível( $a_i$ )
	São Paulo (1)	Belo Horizonte (2)	Rio de Janeiro (3)	
Araraquara (1)	4	2	5	800
S. J. Campos (2)	11	7	4	1000
Demandas dos mercados ( $b_j$ )	500	400	900	

demandas dos mercados. As restrições (2.7) garantem que as quantidades a serem transportadas não podem ser negativas. ■

### **Exemplo 2.4** Transporte de agregados para a construção de uma rodovia.

Suponha que, para a construção de uma rodovia, não estejam disponíveis na região jazidas de rochas adequadas à obtenção de pedra britada. Faz-se necessário, portanto, o transporte desse material de jazidas mais próximas para alguns pontos convenientes preestabelecidos ao longo do caminho onde será implantada a estrada (ver Figura 2.5), ao menor custo. Na Figura 2.5, são apresentados todos os caminhos possíveis que ligam cada pedreira aos pontos de depósito.

Neste problema, temos  $m = 4$  jazidas correspondentes às origens e  $n = 3$  depósitos correspondentes aos destinos, cujos dados estão na Tabela 2.3. As quantidades ofertadas ( $a_i$  – última coluna) e demandadas ( $b_j$  – última linha), em toneladas, bem como os custos de transportar 1 tonelada de pedra da pedreira  $i$  para o depósito  $j$  ( $c_{ij}$  – que é função de vários fatores, como tempo de viagem, condições das estradas de acesso, condições dos veículos que servem a trajetória em questão etc.), são dados na Tabela 2.3.

A unidade de referência pode ser alterada para outra mais apropriada, como, em vez de tonelada, poderia ser a carga de um caminhão, de modo que o custo de transportar uma unidade seria o custo de uma viagem de caminhão.

Se  $x_{ij}$  é a variável de decisão que representa a quantidade transportada de rochas da jazida  $i$  para o ponto de depósito  $j$ , podemos formular este problema da seguinte maneira:

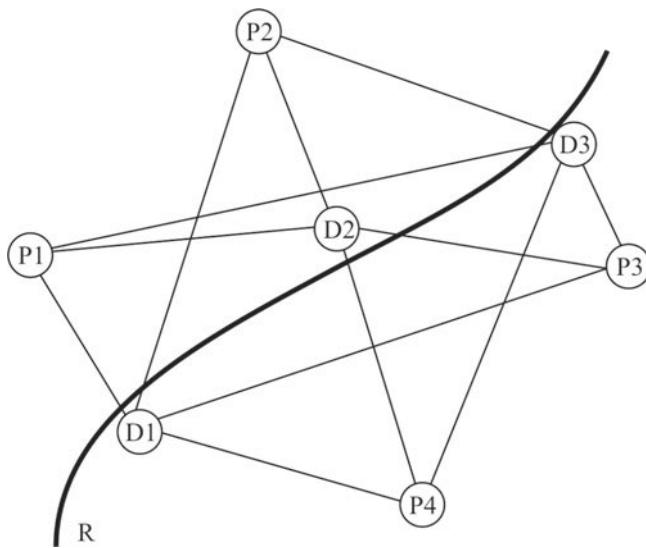
$$\text{Minimizar} \quad f(x_{11}, x_{12}, \dots, x_{43}) = 30x_{11} + 13x_{12} + 21x_{13} + 12x_{21} + 40x_{22} + 26x_{23} + 27x_{31} + \\ 15x_{32} + 35x_{33} + 37x_{41} + 25x_{42} + 19x_{43}$$

$$\begin{array}{ll} x_{11} + x_{12} + x_{13} & \leq 433 \\ x_{21} + x_{22} + x_{23} & \leq 215 \\ x_{31} + x_{32} + x_{33} & \leq 782 \\ x_{41} + x_{42} + x_{43} & \leq 300 \\ x_{11} + x_{21} + x_{31} + x_{41} & = 697 \\ x_{12} + x_{22} + x_{32} + x_{42} & = 421 \\ x_{13} + x_{23} + x_{33} + x_{43} & = 612 \\ x_{11} \geq 0, x_{12} \geq 0, x_{13} \geq 0, x_{21} \geq 0, x_{22} \geq 0, x_{23} \geq 0, x_{31} \geq 0, x_{32} \geq 0, x_{33} \geq 0, x_{41} \geq 0, x_{42} \geq 0, x_{43} \geq 0. \end{array}$$

Em alguns problemas de transporte, para se fazer o deslocamento, podem-se usar localidades intermediárias (ou de transbordo), ou seja, localidades que não são nem centros produtores nem mercados consumidores dos produtos.<sup>3</sup> Essas localidades intermediárias podem representar, por exemplo, depósitos ou centros de distribuição. Tais problemas são denominados de *problemas de transbordo*. O que é transportado de cada localidade intermediária aos mercados consumidores não pode ultrapassar a quantidade de produto que chega dos centros produtores a esses locais. Além disso, transportar além do necessário é mais caro (ou no mínimo igual) do que transportar apenas o necessário. Portanto, devemos observar que a quantidade transportada de cada localidade intermediária aos mercados consumidores é igual à quantidade de produto que chega dos centros produtores a esses locais de transbordo. Na Figura 2.6 está ilustrada uma situação em que um centro de distribuição  $j$  recebe produtos de dois centros produtores e abastece três mercados consumidores.

Assim, devemos adicionar ao modelo de transporte as restrições  $\sum_i x_{ij} = \sum_k x_{jk}$  para toda localidade intermediária  $j$  que represente um centro de distribuição.

<sup>3</sup> Veja, por exemplo, Ahuja *et al.* (1993) e Ghiani *et al.* (2004). Um exemplo de aplicação do problema de transbordo é encontrado em Yoshizaki *et al.* (1996).



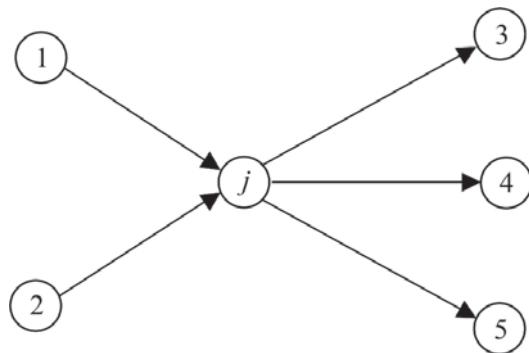
**Figura 2.5** Pedreiras fornecedoras de pedra britada: P1, P2, P3, P4; pontos convenientes para depósito de material: D1, D2, D3; trajeto da rodovia: R.

**Tabela 2.3**  
**Dados do problema de transporte de agregados.**

Pedreiras	Depósitos			Oferta $a_i$
	1	2	3	
1	30	13	21	433
2	12	40	26	215
3	27	15	35	782
4	37	25	19	300
demandas $b_j$	697	421	612	

**Exemplo 2.5** Considere o problema da companhia distribuidora de bebidas do Exemplo 2.3 e suponha que ela dispõe de dois depósitos para abastecer os mercados consumidores. Para evitar ambigüidade, enumeramos cada localidade distintamente – os centros de produção: Araraquara (1), S. J. dos Campos (2); os depósitos: Campinas (3), Barra Mansa (4); e os mercados consumidores: São Paulo (5), Belo Horizonte (6) e Rio de Janeiro (7). Suponha que os mercados sejam abastecidos somente a partir dos depósitos. Os custos unitários de se transportar uma unidade do produto de cada centro de produção para cada depósito e de cada depósito para cada mercado consumidor são dados nas Tabelas 2.4 e 2.5, respectivamente.

Se definirmos a variável  $x_{ij}$  como a quantidade do produto enviada da localidade  $i$  à localidade  $j$ , o modelo que representa o problema é dado a seguir. Note que algumas variáveis  $x_{ij}$  não estão definidas, por exemplo, a variável  $x_{15}$  não é definida, pois São Paulo (5) não é abastecido por Araraquara (1).



**Figura 2.6** Centro de distribuição  $j$ :  $x_{1j} + x_{2j} = x_{j3} + x_{j4} + x_{j5}$ .

**Tabela 2.4**  
Custos unitários de transporte de centros de suprimento aos depósitos.

Centros de suprimento	Depósitos	
	Campinas (3)	Barra Mansa (4)
Araraquara (1)	1	3
S. J. Campos (2)	1	2

**Tabela 2.5**  
Custos unitários dos depósitos aos mercados consumidores.

Depósitos	Mercados consumidores		
	São Paulo (5)	Belo Horizonte (6)	Rio de Janeiro (7)
Campinas (3)	1	3	3
Barra Mansa (4)	3	4	1

Minimizar  $f(x_{13}, \dots, x_{47}) = 1x_{13} + 3x_{14} + 1x_{23} + 2x_{24} + 1x_{35} + 3x_{36} + 3x_{37} + 3x_{45} + 4x_{46} + 1x_{47}$

$$x_{13} + x_{14} \leq 800$$

$$x_{23} + x_{24} \leq 1000$$

$$x_{35} + x_{45} = 500$$

$$x_{36} + x_{46} = 400$$

$$x_{37} + x_{47} = 900$$

$$x_{13} + x_{23} = x_{35} + x_{36} + x_{37}$$

$$x_{14} + x_{24} = x_{45} + x_{46} + x_{47}$$

$$x_{13} \geq 0, x_{14} \geq 0, \dots, x_{46} \geq 0, x_{47} \geq 0.$$
■

Modelos de transporte também podem representar outras situações. Por exemplo, suponha o problema em que temos  $n$  tarefas que precisam ser atribuídas a  $n$  pessoas e que  $p_{ij}$  mede o interesse do indivíduo  $i$  na realização da tarefa  $j$ . Seja a variável de decisão  $x_{ij}$  igual a 1 se o indivíduo  $i$  for designado para a realização da tarefa  $j$  e 0, caso contrário. A formulação deste problema de *designação* (ou *atribuição*) é

$$\text{Maximizar} \quad f(x_{11}, \dots, x_{nn}) = \sum_{i=1}^n \sum_{j=1}^n p_{ij} x_{ij} \quad (2.8)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, 2, \dots, n. \quad (2.9)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, 2, \dots, n \quad (2.10)$$

$$x_{ij} = 0 \text{ ou } 1, \quad i = 1, 2, \dots, n, j = 1, 2, \dots, n. \quad (2.11)$$

A função objetivo em (2.8) representa o interesse total a ser maximizado. O conjunto de restrições (2.9) impõe que cada indivíduo é designado a uma e somente uma tarefa. O conjunto de restrições (2.10) impõe que cada tarefa é realizada por apenas um e somente um indivíduo. As restrições (2.11) impõem que as variáveis de decisão tomem os valores inteiros 0 ou 1. Devido a essas restrições, este é um modelo de otimização linear inteira, a ser estudado no Capítulo 3. No entanto, devido a características particulares da estrutura deste modelo de designação, as restrições (2.11) podem ser substituídas por  $x_{ij} \geq 0$  e, mesmo assim, pode-se mostrar que o modelo tem solução ótima inteira (ver Propriedade 4.4, de integralidade, no Capítulo 4).

Problemas de transporte e transbordo podem ser estendidos para tratar de situações mais gerais, por exemplo, envolvendo múltiplos períodos. Esses problemas também são discutidos no Capítulo 4, uma vez que podem ser vistos como problemas de otimização em redes.

### 2.2.3 Problemas de planejamento da produção

A classe de problemas de planejamento e programação da produção é bastante ampla, e vários desses problemas podem ser modelados por meio de otimização linear. Neste capítulo, introduzimos alguns desses problemas; outros são estudados nos próximos capítulos.

#### *Mix de produção (planejamento estático)*

Problemas de *mix de produção*, isto é, fabricação de diversos produtos, aparecem em diversas situações reais e envolvem decidir quais produtos e quanto fabricar de cada produto em um período (mais adiante discutimos o planejamento da produção dinâmico, que envolve vários períodos com decisões interdependentes). Tendo em vista a capacidade limitada de produção (máquinas, recursos humanos, capital, armazenagem etc.) e os diversos produtos que a empresa pode fabricar e vender, deseja-se determinar quais fabricar e quanto fabricar de cada produto, de modo a maximizar a margem de contribuição ao lucro da empresa.

#### Formulação matemática do problema de *mix de produção*

Seja  $x_j$  a quantidade do produto  $j$ ,  $j = 1, 2, \dots, n$ , a ser produzida em um período do planejamento (por exemplo, um mês). Seja  $C_i$  a capacidade do recurso  $i$ ,  $i = 1, 2, \dots, m$ , disponível no período. Considere que, para a produção de uma unidade do produto  $j$ , são consumidas  $a_{ij}$  unidades do recurso  $i$ . Uma produção mínima do produto  $j$ , digamos  $d_j$ , precisa ser realizada no período, devi-

do aos pedidos em carteira e a uma política de estoque mínimo para preservação do produto no mercado. O departamento de vendas da empresa acredita que as vendas desse produto não excedam  $v_j$  unidades no período em estudo. Cada unidade do produto  $j$  resulta em uma contribuição ao lucro de  $l_j$  para a empresa. O problema de *mix* de produção pode ser formulado como:

$$\text{Maximizar } f(x_1, \dots, x_n) = \sum_{j=1}^n l_j x_j \quad (2.12)$$

$$\sum_{j=1}^n a_{ij} x_j \leq C_i \quad i = 1, 2, \dots, m \quad (2.13)$$

$$d_j \leq x_j \leq v_j \quad j = 1, 2, \dots, n. \quad (2.14)$$

A função em (2.12) representa a contribuição ao lucro da empresa, a ser maximizada. As restrições (2.13) limitam a fabricação dos produtos devido à disponibilidade dos diversos recursos e as restrições (2.14) impõem que a quantidade fabricada de cada produto não pode ser inferior à mínima preestabelecida, nem exceder o que o mercado pode absorver.

**Exemplo 2.6** Um fabricante de geladeiras precisa decidir quais modelos deve produzir em uma nova fábrica recentemente instalada. O departamento de marketing e vendas realizou uma pesquisa de mercado que indicou que, no máximo, 1.500 unidades do modelo de luxo e 6.000 unidades do modelo básico podem ser vendidas no próximo mês. A empresa já contratou um certo número de empregados e, com isso, dispõe de uma força de trabalho de 25.000 homens-hora por mês. Cada modelo de luxo requer dez homens-hora e cada modelo básico requer oito homens-hora para ser montado. Além disso, uma mesma linha de montagem é compartilhada pelos dois modelos e considere que a capacidade de produção desta linha seja de 4.500 geladeiras por mês. O lucro unitário do modelo de luxo é de \$100,00, e do modelo básico é de \$50,00. Deseja-se determinar quanto produzir de cada modelo de modo a maximizar o lucro da empresa.

Definimos a variável  $x_j$  como a quantidade de geladeiras do tipo  $j$ ,  $j = \text{luxo}, \text{básico}$ , a ser produzida no mês, de modo que o lucro da empresa é representado por:  $f(x_{\text{luxo}}, x_{\text{básico}}) = 100x_{\text{luxo}} + 50x_{\text{básico}}$ . As restrições de produção devido à limitação de capacidade ficam:  $10x_{\text{luxo}} + 8x_{\text{básico}} \leq 25.000$ , devido à limitação da força de trabalho por mês e,  $x_{\text{luxo}} + x_{\text{básico}} \leq 4.500$ , devido à limitação da linha de montagem. As restrições devido ao mercado e à não-negatividade são:  $0 \leq x_{\text{luxo}} \leq 1.500$  e  $0 \leq x_{\text{básico}} \leq 6.000$ . O modelo completo para este problema é dado por:

$$\begin{aligned} \text{Maximizar } & f(x_{\text{luxo}}, x_{\text{básico}}) = 100x_{\text{luxo}} + 50x_{\text{básico}} \\ & 10x_{\text{luxo}} + 8x_{\text{básico}} \leq 25.000 \\ & x_{\text{luxo}} + x_{\text{básico}} \leq 4.500 \\ & 0 \leq x_{\text{luxo}} \leq 1.500 \text{ e } 0 \leq x_{\text{básico}} \leq 6.000. \end{aligned}$$

#### Restrições de chance

No Exemplo 2.6, o potencial máximo de vendas de geladeiras foi estimado pelo departamento de marketing e vendas em 1.500 unidades do modelo de luxo e 6.000 unidades do modelo básico. Em geral, há incertezas com respeito a esses valores estimados da demanda, de modo que há interesse em um balanceamento (em inglês, *trade-off*) entre o potencial de lucro (produzir menos do que a demanda e deixar de lucrar mais) e o risco de superprodução (produzir mais do que a demanda e incorrer em mais custos). Para controlar melhor o risco de superproduzir, podemos utilizar *restrições de chances* baseadas na distribuição de probabilidade da demanda do produto.

Tomemos, por exemplo, o modelo de geladeira de luxo e a restrição do modelo anterior:  $x_{luxo} \leq 1.500$ . Em vez de usar essa restrição no modelo, podemos usar uma abordagem alternativa, baseada na variável aleatória de demanda  $D_{luxo}$  e na sua distribuição de probabilidade com média  $E(D_{luxo}) = 1.500$ . Seja  $\alpha_{luxo}$  um limite superior para a probabilidade de superprodução de geladeiras de luxo (fixado pelo gerente). Ou seja, a probabilidade de a demanda  $D_{luxo}$  ser menor do que a quantidade produzida  $x_{luxo}$  deve ser menor do que este limite  $\alpha_{luxo}$ :  $P(D_{luxo} \leq x_{luxo}) \leq \alpha_{luxo}$ . Por exemplo, se esse limite for fixado pelo gerente em  $\alpha_{luxo} = 0,10$ , então, a quantidade a ser produzida  $x_{luxo}$  deve ser tal que a probabilidade de superprodução  $P(D_{luxo} \leq x_{luxo})$  seja menor ou igual a 10%.

Admitindo-se que a demanda  $D_{luxo}$  tem distribuição normal com média  $E(D_{luxo})$  e variância  $V(D_{luxo})$  (ou desvio-padrão  $\sigma(D_{luxo}) = \sqrt{V(D_{luxo})}$ ) conhecidas, segue-se que:<sup>4</sup>

$$P(D_{luxo} \leq x_{luxo}) = P\left(\frac{D_{luxo} - E(D_{luxo})}{\sigma(D_{luxo})} \leq \frac{x_{luxo} - E(D_{luxo})}{\sigma(D_{luxo})}\right) \leq \alpha_{luxo}$$

Seja  $z_{\alpha_{luxo}}$  o valor que satisfaz a igualdade:  $P\left(\frac{D_{luxo} - E(D_{luxo})}{\sigma(D_{luxo})} \leq z_{\alpha_{luxo}}\right) = \alpha_{luxo}$ , isto é, a função distribuição normal acumulada no ponto  $z_{\alpha_{luxo}}$  é igual a  $\alpha_{luxo}$ . Dado  $\alpha_{luxo}$ , o valor  $z_{\alpha_{luxo}}$  é facilmente obtido, por exemplo, em tabelas de distribuição normal acumulada, e é negativo para valores de  $\alpha_{luxo}$  menores que 0,50. Considerando a desigualdade anterior e o fato de que a distribuição normal acumulada é uma função crescente, segue-se que  $\frac{x_{luxo} - E(D_{luxo})}{\sigma(D_{luxo})} \leq z_{\alpha_{luxo}}$ , ou seja,

$$x_{luxo} \leq E(D_{luxo}) + z_{\alpha_{luxo}} \sigma(D_{luxo})$$

que é uma restrição (linear) de chance para substituir a restrição anterior  $x_{luxo} \leq E(D_{luxo}) = 1.500$  no modelo de mix de produção. Por exemplo, se  $E(D_{luxo}) = 1.500$ ,  $\sigma(D_{luxo}) = 100$  e  $\alpha_{luxo} = 0,10$ , obtemos  $z_{0,10} = -1,282$  (tabela de distribuição normal), o que resulta em:  $x_{luxo} \leq E(D_{luxo}) + 1372$ .<sup>5</sup> Restrições de chance também podem ser usadas para abordar incertezas de outros parâmetros além de demanda; por exemplo, capacidade de produção.

### *Seleção de processos (planejamento estático)*

Uma empresa fabrica vários produtos em um período e, para isso, dispõe de processos alternativos. Esses processos podem representar tecnologias diferentes, opções de hora extra, turno adicional, subcontratação etc. O problema consiste em determinar quanto produzir de cada produto em cada processo no período, com o objetivo de minimizar os custos de produção, sujeito a limitações de recursos e requisitos de demanda. Considere os seguintes dados:

- $c_{ij}$  custo de fabricar uma unidade do produto  $i$  no processo  $j$ ,  $j = 1, 2, \dots, n$ ,
- $d_i$  demanda do produto  $i$ ,  $i = 1, 2, \dots, m$ , no período,
- $b_k$  quantidade disponível do recurso  $k$ ,  $k = 1, 2, \dots, K$ , no período,
- $a_{ijk}$  quantidade do recurso  $k$  necessária para fabricar uma unidade do produto  $i$  no processo  $j$ ,
- $J_i$  conjunto de processos alternativos que fabricam o produto  $i$ .

Definimos as variáveis de decisão por:

$x_{ij}$  número de produtos do tipo  $i$  fabricados no processo  $j$ .

O modelo completo para este problema é dado por:<sup>6</sup>

<sup>4</sup> Para detalhes de cálculo de probabilidades com distribuições normais, veja, por exemplo, Meyer (1983) e Ross (1993).

<sup>5</sup> Bitran e Yanasse (1984) analisam os erros incorridos devido a aproximações feitas com modelos determinísticos de problemas estocásticos de produção.

<sup>6</sup> Para mais detalhes deste problema, veja Jonhson e Montgomery (1974); para um exemplo de aplicação deste problema, combinado a um problema de dimensionamento de lotes, na produção de grãos eletrofundidos, veja também Luche e Morabito (2005).

$$\text{Minimizar } f(x_{11}, x_{12}, \dots) = \sum_{i=1}^n \sum_{j \in J_i} c_{ij} x_{ij}$$

$$\begin{aligned} \sum_{j \in J_i} x_{ij} &= d_i & i = 1, \dots, n \\ \sum_{i=1}^n \sum_{j \in J_i} a_{ijk} x_{ij} &\leq b_k & k = 1, \dots, K \\ x_{ij} &\geq 0 & i = 1, \dots, n, j \in J_i \end{aligned}$$

### Dimensionamento de lotes (planejamento dinâmico)

Empresas de manufatura, em geral, fabricam diversos tipos de produtos solicitados por diferentes clientes, muitas vezes em grandes quantidades, os quais devem estar prontos para entrega em diferentes datas previamente agendadas. Como as fábricas têm capacidades de produção limitadas (máquinas, mão-de-obra etc.), é necessário planejar a produção, isto é, decidir o *quê* e *quanto* produzir (em outras palavras, dimensionar os lotes de produção) em cada período de um horizonte de planejamento. A necessidade de antecipação da fabricação de produtos (estocados de um período para o outro) acarreta custos de estocagem e algumas dificuldades operacionais. No planejamento da produção, deseja-se determinar o tamanho dos lotes de produção, para atender a demanda na data solicitada e de modo que a soma dos custos de produção e estocagem seja mínima.

#### *Formulação matemática de um problema de dimensionamento de lotes – monoestágio*

Considere uma empresa que fabrica  $n$  produtos e deseja programar sua produção nos próximos  $T$  períodos de tempo. Este conjunto de períodos de tempo para a qual a empresa planeja sua produção é denominado *horizonte de planejamento* (conforme Capítulo 1). Supomos que a demanda de cada produto em cada período do horizonte de planejamento é conhecida. Em cada período, os recursos necessários para a produção são limitados e renováveis, isto é, em cada período, uma quantidade de recursos está disponível e não depende de como foram utilizados nos períodos anteriores. Exemplos de recursos renováveis são mão-de-obra, energia elétrica e horas de máquinas, enquanto recursos não renováveis são, por exemplo, matérias-primas que sobram em um período e podem ser utilizadas nos períodos seguintes. Há a possibilidade de estocagem de produtos de um período para o outro. Considere os seguintes dados do problema:

- $d_{it}$  demanda do item  $i$  no período  $t$ ,  $i = 1, \dots, n$ ,  $t = 1, \dots, T$ ,
- $R_t$  disponibilidade de recursos (renováveis) no período  $t$ ,
- $r_i$  quantidade de recursos necessários para a produção de uma unidade do item  $i$ ,
- $c_{it}$  custo de produzir uma unidade do item  $i$  no período  $t$ ,
- $h_{it}$  custo de estocar uma unidade do item  $i$  no período  $t$ .

As variáveis de decisão são:

- $x_{it}$  o número de itens do tipo  $i$  produzidos no período  $t$ ,
- $I_{it}$  o número de itens do tipo  $i$  em estoque no final do período  $t$ .

Os estoques iniciais do horizonte de planejamento  $I_{i0}$  são dados. A seguir, são detalhadas as restrições do problema.

- *Equações de conservação de estoque*

Para cada item  $i$ , o nível de estoque no final do período  $t$  é igual ao que se tinha em estoque no final do período anterior ( $t-1$ ), adicionado ao montante que foi produzido no período  $t$ , menos o que foi demandado no período  $t$ , ou seja,

$$I_{it} = I_{i,t-1} + x_{it} - d_{it} \quad i = 1, \dots, n, t = 1, \dots, T.$$

Tais restrições são típicas em modelos dinâmicos. Por exemplo, podem modelar a conservação de água em um reservatório: o volume de água em um reservatório  $i$  no instante  $t$  é igual ao volume de água que havia no instante  $t-\Delta t$ , mais o volume de água que entrou no reservatório no intervalo de tempo  $\Delta t$ , menos o volume de água utilizada para abastecer uma população no intervalo de tempo  $\Delta t$ .

- *Restrições de capacidade de produção*

A capacidade requerida para a produção dos itens em cada período  $t$  não pode superar a capacidade disponível da fábrica, ou seja,

$$r_1 x_{1t} + r_2 x_{2t} + \dots + r_n x_{nt} \leq R_t \quad t = 1, \dots, T.$$

- *Garantia de atendimento às demandas*

A demanda de cada produto  $i$  em cada período  $t$  precisa ser atendida. Isso significa que a quantidade total disponível no período deve ser maior ou igual à demanda, isto é, a quantidade produzida no período ( $x_{it}$ ), mais a quantidade em estoque no final do período anterior ( $I_{i,t-1}$ ), deve ser maior ou igual a  $d_{it}$ , ou seja,  $x_{it} + I_{i,t-1} \geq d_{it}$  ou, ainda,  $x_{it} + I_{i,t-1} - d_{it} \geq 0$ . Essa quantidade foi definida anteriormente como  $I_{it}$ , de modo que podemos garantir o atendimento às demandas impondo:

$$I_{it} \geq 0 \quad i = 1, \dots, n, t = 1, \dots, T.$$

Além disso, por definição das variáveis de produção  $x_{ij}$ , temos que

$$x_{it} \geq 0 \quad i = 1, \dots, n, t = 1, \dots, T.$$

Os custos incorridos são os de produção e de estocagem dos produtos. Desta forma, a função custo a ser minimizada é

$$f(x_{11}, I_{11}, x_{12}, I_{12}, \dots) = \sum_{i=1}^n \sum_{t=1}^T c_{it} x_{it} + \sum_{i=1}^n \sum_{t=1}^T h_{it} I_{it}.$$

O modelo completo de otimização linear é dado por:

$$\begin{aligned} \text{Minimizar} \quad & f(x_{11}, I_{11}, x_{12}, I_{12}, \dots) = \sum_{i=1}^n \sum_{t=1}^T c_{it} x_{it} + \sum_{i=1}^n \sum_{t=1}^T h_{it} I_{it} \\ & x_{it} + I_{i,t-1} - I_{it} = d_{it} \quad i = 1, \dots, n, t = 1, \dots, T \\ & r_1 x_{1t} + r_2 x_{2t} + \dots + r_n x_{nt} \leq R_t \quad t = 1, \dots, T \\ & x_{it}, I_{it} \geq 0 \quad i = 1, \dots, n, t = 1, \dots, T. \end{aligned}$$

Modelos desse tipo também são utilizados para apoiar decisões no planejamento agregado da produção, em que os produtos são agregados em famílias de produtos, as demandas dos produtos são agregadas por região, e as máquinas são agregadas em centros de trabalho.<sup>7</sup> Os problemas de dimensionamento de lotes podem ser facilmente estendidos para incorporar casos de pedi-

<sup>7</sup> Veja, por exemplo, Hax e Candea (1984) e Nahmias (1995); veja também Carvalho *et al.* (1998). Gonçalves Filho e Marçola (2001) estudam o problema com banco de horas e Caixeta-Filho *et al.* (2002) abordam o problema de plantio e vendas de flores.

dos pendentes (em inglês, *backlogging*) e vendas perdidas, por meio de uma simples substituição de variáveis:  $I_{it} = I_{it}^+ - I_{it}^-$ , em que  $I_{it}^+ \geq 0$  é o estoque em mãos e  $I_{it}^- \geq 0$  é o estoque devido (nível de pedidos pendentes) do produto  $i$  no final do período  $t$  (veja Seção 2.2.10 e discussão na Seção 3.6.1 do Capítulo 3). Nesta classe de problemas, também é comum a necessidade de preparar máquinas para a produção de um item e, nessa preparação, consomem-se recursos e incorre-se em custos.<sup>8</sup> Esses problemas são da classe de otimização discreta discutida no Capítulo 3, pois surgem decisões do tipo produzir ou não um item do tipo  $i$  no período  $t$ , as quais são modeladas por variáveis que assumem apenas dois valores: 0 ou 1.

**Exemplo 2.7** Considere uma fábrica de pré-moldados que produz dois tipos de vigas, cujas demandas para as próximas três semanas são conhecidas, conforme a Tabela 2.6.

Os produtos utilizam os mesmos tipos de recursos, porém em quantidades diferentes. Suponha, por simplicidade, que apenas um centro de trabalho esteja disponível para a produção dos dois itens, cuja disponibilidade é de 40 horas por período e que a produção de uma unidade do item 1 consuma 15 minutos e uma unidade do item 2 consuma 20 minutos. Os custos de produção por período são conhecidos e dados pela Tabela 2.7.

Admite-se que a produção possa ser antecipada e estocada para ser utilizada nos períodos seguintes. Os custos de estocagem são dados na Tabela 2.8 (por exemplo, uma unidade do item 1 pode ser produzida no período 2 e guardada em estoque para atender a demanda no período 3, por \$ 3,00/unidade).

Deseja-se definir um plano da produção de modo que os pedidos sejam atendidos ao menor custo de produção e estocagem. Os estoques iniciais dos dois produtos são nulos e deseja-se que seus estoques ao final do horizonte de planejamento também sejam nulos. Seja a variável de decisão  $x_{it}$  a quantidade da viga tipo  $i$  produzida no período  $t$ , e  $I_{it}$  a quantidade da viga tipo  $i$  estocada no final do período  $t$ . Então podemos escrever as restrições seguindo o modelo genérico apresentado anteriormente, obtendo:

**Tabela 2.6**  
**Demandas de vigas.**

Demandas de vigas	Período 1	Período 2	Período 3
Item 1	100	90	120
Item 2	40	50	80

**Tabela 2.7**  
**Custos de produção.**

Custos de produção	Período 1	Período 2	Período 3
Item 1	20	20	30
Item 2	20	20	30

<sup>8</sup> Bitran e Yanasse (1982) abordam algumas questões teóricas de complexidade; Araújo *et al.* (2004), Santos-Meza *et al.* (2002) e Rangel e Ferreira (2003) apresentam modelos e métodos para planejamento da produção em fundições de pequeno e médio porte e fábricas de refrigerantes; Taube Netto (1996) apresenta um estudo de caso em uma agroindústria brasileira.

**Tabela 2.8**  
**Custos de estocagem.**

<i>Custos de estocagem</i>	<i>Período 1</i>	<i>Período 2</i>
Item 1	2	3
Item 2	2,5	3,5

- *Restrições de conservação de estoque*

De  $I_{it} = I_{i,t-1} + x_{it} - d_{it}$ ,  $i = 1, 2$  e  $t = 1, 2, 3$ , temos:

$$I_{11} = x_{11} - 100$$

$$I_{12} = I_{11} + x_{12} - 90$$

$$I_{13} = I_{12} + x_{13} - 120 = 0 \text{ (estoque final da viga tipo 1 foi fixado em zero)}$$

$$I_{21} = x_{21} - 40$$

$$I_{22} = I_{21} + x_{22} - 70$$

$$I_{23} = I_{22} + x_{23} - 80 = 0 \text{ (estoque final da viga tipo 2 foi fixado em zero).}$$

- *Restrições de capacidade de produção*

De  $r_1 x_{1t} + r_2 x_{2t} + \dots + r_n x_{nt} \leq R_p$ ,  $t = 1, 2, 3$ , temos:

$$1/4x_{11} + 1/3x_{21} \leq 40$$

$$1/4x_{12} + 1/3x_{22} \leq 40$$

$$1/4x_{13} + 1/3x_{23} \leq 40$$

- *Garantia de atendimento às demandas*

De  $I_{it} \geq 0$ ,  $i = 1, 2$  e  $t = 1, 2$  (note que para  $t = 3$  o estoque é dado), temos:

$$I_{11}, I_{12}, I_{21}, I_{22} \geq 0$$

Além disso, por definição das variáveis  $x_{it}$ , devemos ter

$$x_{it} \geq 0, i = 1, 2, t = 1, 2, 3, \text{ ou seja,}$$

$$x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23} \geq 0.$$

- *Custo*

De  $f(x_{11}, I_{11}, x_{12}, I_{12}, \dots) = \sum_{i=1}^n \sum_{t=1}^T c_{it} x_{it} + \sum_{i=1}^n \sum_{t=1}^{T-1} h_{it} I_{it}$  (estoque final é fixado), temos

$$f(x_{11}, x_{12}, \dots, I_{22}) = 20x_{11} + 20x_{12} + 30x_{13} + 20x_{21} + 20x_{22} + 30x_{23} + 2I_{11} + 3I_{12} + 2,5I_{21} + 3,5I_{22}$$

$$\begin{aligned} x_{11} & -I_{11} & = 100 \\ x_{12} & + I_{11} - I_{12} & = 90 \\ x_{13} & + I_{12} & = 120 \\ x_{21} & -I_{21} & = 40 \\ x_{22} & + I_{21} - I_{22} & = 70 \\ x_{23} & + I_{22} & = 80 \end{aligned}$$

$$\begin{aligned}
 1/4x_{11} &+ 1/3x_{21} && \leq 40 \\
 1/4x_{12} &+ 1/3x_{22} && \leq 40 \\
 1/4x_{13} &+ 1/3x_{23} && \leq 40 \\
 x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}, I_{11}, I_{12}, I_{21}, I_{22} &\geq 0.
 \end{aligned}$$

■

### Formulação matemática de um problema de dimensionamento de lotes – multiestágios

De maneira análoga ao caso anterior, monoestágio, considere uma empresa que fabrica  $n$  produtos e deseja planejar sua produção em um horizonte de  $T$  períodos de tempo. Os pedidos são feitos antecipadamente, de modo que a demanda de cada produto é conhecida em cada período do horizonte de planejamento. Porém, a produção de uma unidade de um produto demandado requer a produção prévia ou concomitante de seus componentes (por exemplo, para produzir um computador é necessário produzir um gabinete, um teclado, uma placa-mãe etc., e para produzir um teclado é necessário produzir uma estrutura de teclado, 52 teclas, entre outros). Um componente também pode ter demanda externa (por exemplo, teclado). A fabricação dos itens (que podem ser produtos ou componentes) usa diversos recursos renováveis limitados por período (tempo de máquinas, mão-de-obra etc.). Considere os seguintes dados do problema:

- $d_{it}$  demanda externa do item  $i$  no período  $t$ ,  $i = 1, \dots, n$ ,  $t = 1, \dots, T$
- $b_{ji}$  quantidade do item  $j$  para a produção de uma unidade do item  $i$
- $R_{kt}$  disponibilidade do recurso (renovável) do tipo  $k$  no período  $t$
- $r_{ki}$  quantidade do recurso  $k$  necessária para a produção de uma unidade do item  $i$
- $c_{it}$  custo de produzir uma unidade do item  $i$  no período  $t$
- $h_{it}$  custo de estocar uma unidade do item  $i$  no final do período  $t$

As variáveis de decisão são:

$x_{it}$  número de itens do tipo  $i$  produzidos durante o período  $t$

$I_{it}$  número de itens do tipo  $i$  em estoque no final do período  $t$

Para a produção de  $x_{it}$  unidades do item  $i$  no período  $t$ , é necessário que a quantidade  $b_{ji}x_{it}$  do item  $j$  esteja disponível no período  $t$ . Assim, a demanda do item  $j$  no período  $t$  deve ser acrescida de:  $b_{j1}x_{1t} + b_{j2}x_{2t} + \dots + b_{jn}x_{nt}$ , a qual é chamada de *demandas interna* (ou *demandas derivadas*) e depende do plano de produção. Se o item  $j$  for um produto final, então não há demanda interna, isto é,  $b_{ji} = 0$  para  $i = 1, \dots, n$ . A Figura 1.2 (Capítulo 1) ilustra um exemplo de uma estrutura de produto em que cada item é ligado ao seu item componente por um arco. O modelo (1.1) – (1.5) do Capítulo 1 é um exemplo simples de apenas um produto final que depende de apenas um item componente. Note que a função custo a ser minimizada no modelo (1.1) – (1.5) não é linear, devido a um custo fixo da produção de um item.

O modelo matemático para o dimensionamento de lotes em multiestágios é:

$$\begin{aligned}
 \text{Minimizar } f(x_{11}, I_{11}, \dots) &= \sum_{i=1}^n \sum_{t=1}^T c_{it}x_{it} + \sum_{i=1}^n \sum_{t=1}^T h_{it}I_{it} \\
 x_{jt} + I_{j,t-1} - I_{jt} &= d_{jt} + \sum_{i=1}^n b_{ji}x_{it} \quad j = 1, \dots, n, \quad t = 1, \dots, T \\
 r_{k1}x_{1t} + r_{k2}x_{2t} + \dots + r_{kn}x_{nt} &\leq R_{kt} \quad k = 1, \dots, K, \quad t = 1, \dots, T \\
 x_{it}, I_{it} &\geq 0 \quad i = 1, \dots, n, \quad t = 1, \dots, T
 \end{aligned}$$

## 2.2.4 Problemas de programação de projetos

Nesta classe de problemas, determinamos a ordem em que um conjunto de atividades é realizado e, por simplicidade de exposição, ilustramos com um exemplo. Considere o problema de construir pilares de uma edificação, a qual é constituída basicamente por oito atividades, que estão relacionadas na Tabela 2.9, juntamente com a descrição dessas atividades, a duração e os predecessores imediatos de cada uma. Por exemplo, a atividade D só pode ser iniciada depois da conclusão das atividades B e C, enquanto as atividades A, B e E não têm predecessores, de modo que podem ser iniciadas a qualquer instante a partir do início do projeto. Uma questão importante para a execução de um projeto (isto é, a realização de todas as atividades) é a determinação do menor tempo necessário para sua conclusão. A simplicidade dos dados na Tabela 2.9 permite que o problema de determinar o menor tempo para a conclusão do projeto seja resolvido por inspeção, diferentemente se houvesse milhares de atividades. Recomendamos ao leitor que determine o menor tempo necessário para que todas as oito atividades sejam concluídas e note que certas atividades têm seus instantes de início bem definidos na solução do menor tempo, enquanto outras têm mais flexibilidade para serem iniciadas.

Uma formulação matemática para este problema consiste em definir a variável de decisão  $t_i$ ,  $i = A, B, C, D, E, F, G, H$ , o instante mais cedo em que a atividade  $i$  pode ser iniciada. Definimos o início do projeto no instante  $t = 0$ . Note na Tabela 2.9 que o projeto não pode ser concluído antes do instante  $t_H + 3$ . Assim, o objetivo é minimizar  $t_H + 3$ . Os instantes mais cedo  $t_i$  em que as diversas atividades podem ser iniciadas, devem satisfazer as relações de precedência indicadas na Tabela 2.9. Por exemplo, a atividade C não pode ser iniciada antes que a atividade A tenha sido completada, ou seja,  $t_C \geq t_A + 6$ ; a atividade A pode ser iniciada a qualquer instante depois do início (por definição, o início é o instante  $t = 0$ ); isto é,  $t_A \geq 0$ ; a atividade D não pode ser iniciada antes que as atividades B e C tenham sido completadas, ou seja,  $t_D \geq t_B + 5$ ,  $t_D \geq t_C + 4$ . De maneira similar, determinamos as demais restrições de precedência do problema. O modelo matemático completo para este exemplo é:

$$\begin{aligned}
 & \text{Minimizar} \quad f(t_A, t_B, \dots, t_H) = t_H + 3 \\
 & t_A \geq 0 \\
 & t_B \geq 0 \\
 & t_C \geq t_A + 6 \\
 & t_D \geq t_B + 5 \\
 & t_D \geq t_C + 4 \\
 & t_E \geq 0 \\
 & t_F \geq t_E + 2 \\
 & t_F \geq t_D + 2 \\
 & t_G \geq t_F + 3 \\
 & t_H \geq t_G + 72.
 \end{aligned}$$

Problemas de programação de projetos podem envolver outras restrições, além das relações de precedência, conforme o exemplo anterior, tais como limitação de recursos. Por exemplo, cada tarefa tem um orçamento para ser realizada e os recursos financeiros são limitados ao longo da execução do projeto. Estes problemas também são discutidos nos Capítulos 3 (Seção 3.6.4) e 4 (Seção 4.2.2).

**Tabela 2.9**  
**Dados das atividades do projeto.**

Atividades	Descrição	Predecessor imediato	Duração (horas)
<b>A</b>	<i>Preparo da armadura</i>	—	6
<b>B</b>	<i>Preparo da forma</i>	—	5
<b>C</b>	<i>Lançamento da armadura</i>	<b>A</b>	4
<b>D</b>	<i>Lançamento da forma</i>	<b>B, C</b>	2
<b>E</b>	<i>Providências para concretagem</i>	—	2
<b>F</b>	<i>Aplicação do concreto</i>	<b>E, D</b>	3
<b>G</b>	<i>Cura do concreto</i>	<b>F</b>	72
<b>H</b>	<i>Desforma do pilar</i>	<b>G</b>	3

## 2.2.5 Problemas de gestão financeira (fluxo de caixa)<sup>9</sup>

Modelos lineares também podem ser utilizados para apoiar decisões em problemas de gestão financeira, por exemplo, no gerenciamento do fluxo de caixa. Considere uma empresa que gostaria de maximizar o retorno do seu fluxo de caixa ao final de um horizonte de planejamento de  $n$  períodos (por exemplo, 6 meses). A empresa tem boas previsões dos fluxos de entrada de caixa (isto é, contas a receber) e dos fluxos de saída de caixa (isto é, contas a pagar) no início de cada período. Os seguintes dados são conhecidos:

$e_t$  entrada de caixa no início do período  $t$ ,  $t = 1, \dots, n$ ,

$s_t$  saída de caixa no início do período  $t$ ,  $t = 1, \dots, n$ .

Apenas duas opções de investimento estão disponíveis para o dinheiro do caixa, no início de cada período:

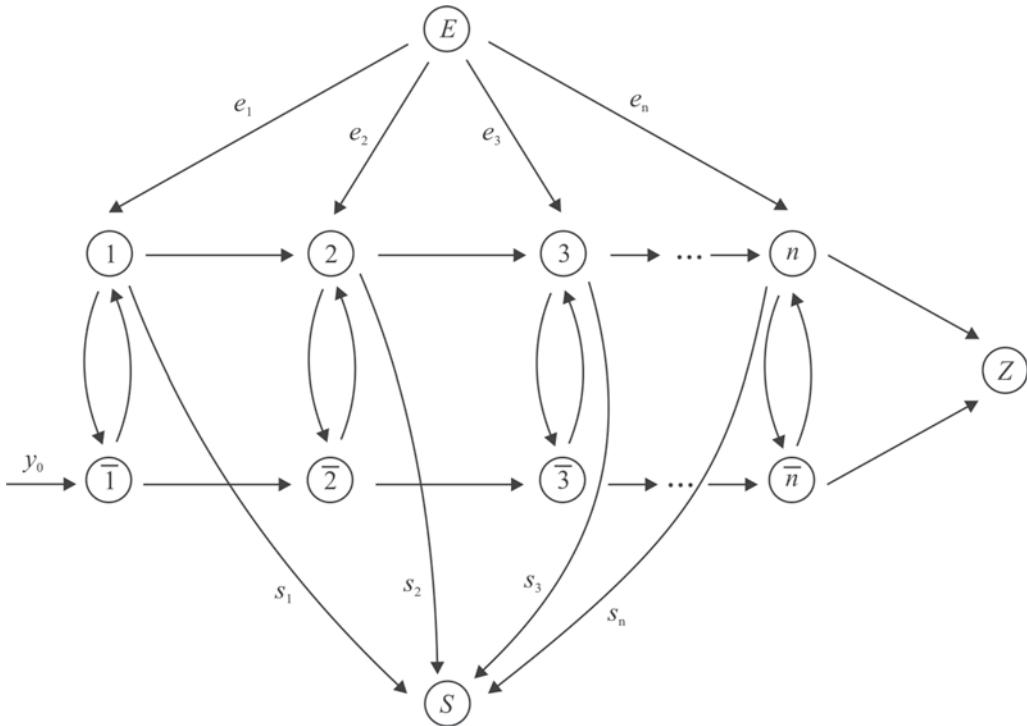
- (i) Deixar parte ou todo dinheiro no próprio caixa durante o período, com taxa de juros  $\alpha$  (por exemplo, rendimentos de aplicações financeiras de curto prazo);
- (ii) Utilizar parte ou todo dinheiro em uma aplicação financeira com menor liquidez do que a opção 1 (isto é, resgate restrito), porém com taxa de juros no período  $\beta$ ,  $\beta > \alpha$  (por exemplo, fundos de ações, poupança jurídica, títulos públicos etc.).

As conversões (aplicações e resgates) entre as opções 1 e 2 podem ser realizadas apenas no início de cada período  $t$ . Há um custo unitário de conversão  $c_{1,2}$  da opção 1 para a opção 2 (por exemplo, taxas de administração) e um custo unitário de conversão  $c_{2,1}$  da opção 2 para a opção 1 (por exemplo, impostos). No início do horizonte de planejamento, a empresa dispõe de  $y_0$  unidades monetárias investidas na opção 2 (decorrente de aplicações anteriores).

Este problema é representado em uma rede de fluxos de dinheiro<sup>10</sup> na Figura 2.7. O nó  $E$  simboliza a entrada de caixa (origem das contas a receber de clientes) e o nó  $S$  a saída de caixa (destino das contas a pagar aos fornecedores). Os nós  $1, 2, \dots, n$  e os nós  $\bar{1}, \bar{2}, \dots, \bar{n}$  simbolizam os

<sup>9</sup> Para outros problemas de otimização em finanças, veja Zenios (1993).

<sup>10</sup> Conceitos de redes introduzidos neste exemplo são mais bem formalizados no Capítulo 4.



**Figura 2.7** Rede de fluxos entre as opções 1 e 2 nos períodos 1, 2, ...,  $n$ .

inícios dos períodos para as opções de investimento 1 e 2, respectivamente, e o nó  $Z$  o final do planejamento. Como a empresa deseja maximizar o retorno do seu fluxo de caixa no final do horizonte de planejamento, o objetivo é maximizar a soma dos fluxos de dinheiro de  $n$  para  $Z$  e de  $\bar{n}$  para  $Z$ .

Definimos as variáveis de decisão:

$f_{ij}$  fluxo de dinheiro do nó  $i$  para o nó  $j$ .

Note na Figura 2.7 que o fluxo de dinheiro (aplicação ou resgate) ocorre entre nós específicos: de  $i$  para  $i+1$  (dinheiro mantido em caixa no período  $i$  com  $\alpha$  de rendimentos); de  $i$  para  $\bar{i}$  (dinheiro aplicado no período  $i$  com taxas de administração de  $c_{i,\bar{i}}$ ); de  $\bar{i}$  para  $i$  (dinheiro resgatado no período  $i$  com impostos de  $c_{\bar{i},i}$ ) etc. Por exemplo, se  $i = 1$  e  $j = 2$  então  $f_{12}$  é a quantia de dinheiro mantida em caixa no período 1, que corresponde a uma entrada de caixa no período 2 de  $(1+\alpha)f_{12}$ ; ou se  $i = \bar{1}$  e  $j = \bar{2}$  então  $f_{\bar{1},\bar{2}}$  é a quantia mantida na opção 2 no período 1, de modo que a quantia  $(1+\beta)f_{\bar{1},\bar{2}}$  estará aplicada no período 2. Deve-se notar que o total aplicado na opção 2 no período 2 é, além da quantia  $(1+\beta)f_{\bar{1},\bar{2}}$ , também a quantia  $(1-c_{1,\bar{2}})f_{2,\bar{2}}$  (em que  $f_{2,\bar{2}}$  é a quantia de dinheiro que sai do caixa no período 2 para a opção 2 e  $c_{1,\bar{2}}$  é a taxa de administração), menos a quantia resgatada no período 2, dada por:  $f_{\bar{2},2}$ . Isto nos leva a escrever a seguinte equação (balanço de fluxo de dinheiro no período 2, para a opção 2: total de entrada = total de saída):

$$(1+\beta)f_{\bar{1},\bar{2}} + (1-c_{1,\bar{2}})f_{2,\bar{2}} = f_{\bar{2},2} + f_{\bar{2},\bar{3}}$$

Note que a quantia  $f_{\bar{2},\bar{3}}$  é vista como que saindo do nó  $\bar{2}$ . Para cada um dos nós na rede de fluxos da Figura 2.7 podemos escrever uma equação de balanço de fluxo, similar à equação acima. No final do planejamento, o fluxo entrando no nó  $Z$  deve ser maximizado, o qual é dado por:  $(1+\alpha)f_{n,Z} + (1+\beta)f_{\bar{n},Z}$ . O modelo matemático completo para o problema de fluxo de caixa é dado por:

$$\begin{aligned}
 & \text{Maximizar } (1+\alpha)f_{n,Z} + (1+\beta)f_{\bar{n},Z} \\
 e_1 + (1-c_{2,1})f_{\bar{1},1} &= f_{1,2} + f_{\bar{1},\bar{1}} + s_1 && (\text{nó 1}) \\
 y_0 + (1-c_{1,2})f_{1,\bar{1}} &= f_{\bar{1},\bar{2}} + f_{\bar{1},1} && (\text{nó } \bar{1}) \\
 e_2 + (1+\alpha)f_{1,2} + (1-c_{2,1})f_{\bar{2},2} &= f_{2,3} + f_{\bar{2},\bar{2}} + s_2 && (\text{nó 2}) \\
 (1+\beta)f_{\bar{1},\bar{2}} + (1-c_{1,2})f_{2,\bar{2}} &= f_{\bar{2},\bar{3}} + f_{\bar{2},2} && (\text{nó } \bar{2}) \\
 &\dots \\
 e_n + (1+\alpha)f_{n-1,n} + (1-c_{2,1})f_{\bar{n},n} &= f_{n,Z} + f_{n,\bar{n}} + s_n && (\text{nó } n) \\
 (1+\beta)f_{\bar{n}-1,\bar{n}} + (1-c_{1,2})f_{n,\bar{n}} &= f_{\bar{n},Z} + f_{\bar{n},n} && (\text{nó } \bar{n}) \\
 f_{i,j} &\geq 0, \text{ para todo } i, j.
 \end{aligned}$$

Este modelo de otimização linear pode ser estendido para tratar mais de duas opções de investimento por período, considerar taxas de juros e custos unitários de conversão variando com cada período e incorporar a possibilidade de empréstimos ao longo do horizonte de planejamento.

**Exemplo 2.9**<sup>11</sup> Considere um horizonte de planejamento de  $n = 3$  períodos, com os seguintes fluxos de entrada e saída de caixa:  $e_1 = \$12$ ,  $e_2 = \$8$ ,  $e_3 = \$5$  e  $s_1 = \$4$ ,  $s_2 = \$10$ ,  $s_3 = \$10$  unidades monetárias. A empresa tem duas opções de investimento: (i) deixar dinheiro no caixa e aplicá-lo em fundos de investimento de curto prazo, com expectativa de rendimento de 5% do valor investido por período, ou (ii) aplicá-lo em títulos públicos com rendimento pré-fixado de 8% do valor investido por período. A transação de aplicação em títulos tem um custo unitário de 2% do valor aplicado e seu resgate tem um custo unitário de 2% do valor resgatado. As aplicações e resgates só podem ser feitos no início de cada período. A empresa já dispõe de um investimento inicial de \$1 em títulos públicos, disponível para resgate. Determine quanto aplicar e resgatar no início de cada período, de maneira a maximizar o retorno do caixa no final do horizonte de planejamento.

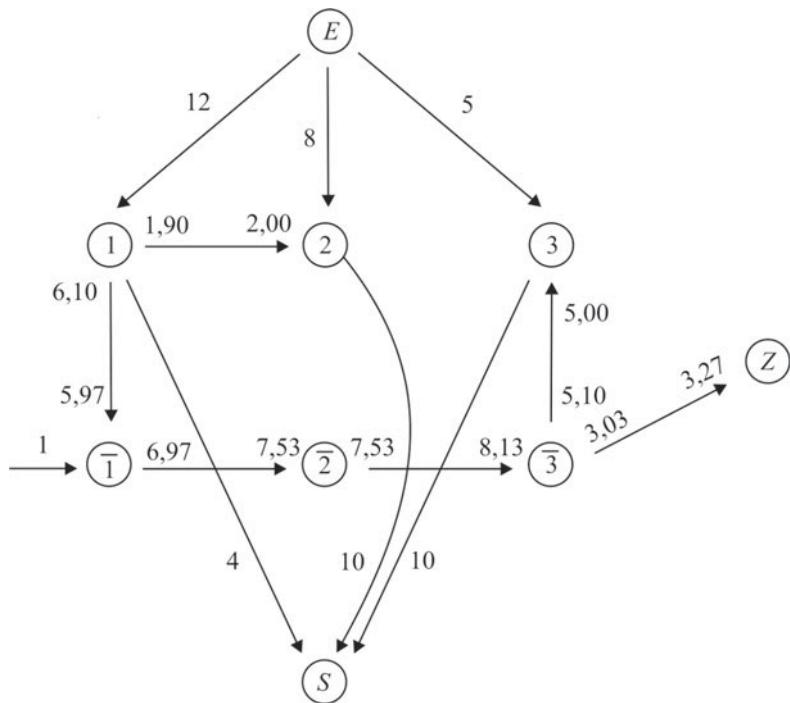
Note que  $\alpha = 0,05$ ,  $\beta = 0,08$ ,  $c_{1,2} = c_{2,1} = 0,02$  e  $y_0 = \$1$ . Uma solução ótima deste exemplo, ilustrada na Figura 2.8, consiste em: (i) aplicar \$6,10 do caixa, mais o investimento inicial de \$1, em títulos públicos no período 1; (ii) não aplicar nada do caixa em títulos, nem resgatar nada dos títulos para o caixa, deixando \$7,53 aplicados em títulos no período 2; (iii) resgatar \$5,10 dos títulos para o caixa no período 3, deixando \$3,03 aplicados em títulos no período 3. O retorno no final do período 3 é \$3,27 (aplicados em títulos).

## 2.2.6 Problemas de meio ambiente

A preocupação com a preservação do meio ambiente tem sido cada vez mais importante para melhorar a qualidade de vida. Uma maneira de preservar nosso meio ambiente é cuidar para que a poluição dos recursos naturais esteja sob controle. Um desses problemas é ilustrado a seguir. Pense numa fábrica que desvia parte da água de um rio para ser utilizada em seu processo produtivo. Durante a produção, componentes químicos poluidores A e B são adicionados à água desviada, que depois retorna ao rio. A água desviada, se não tratada, polui o rio, deixando-o com níveis intoleráveis desses poluentes. O processo produtivo não acarreta mudanças no fluxo de água. A adição dos poluentes também não afeta esse volume.

As concentrações de poluentes consideradas aceitáveis pela Companhia Estadual de Controle Ambiental são de  $a_0$  e  $b_0$  gramas por ML (milhões de litros) de água por dia para os componentes A e B, respectivamente. A vazão do rio é de  $V$  ML por dia e a fábrica precisa de pelo menos  $U$  ML de água por dia para sua produção. Existem 3 tipos de tratamento que a empresa pode utilizar para diminuir a concentração dos poluentes resultantes de seu processo de produção.

<sup>11</sup> Este exemplo é baseado em Golden *et al.* (1979).



**Figura 2.8** Solução ótima do exemplo 2.8.

Esses tratamentos têm custos diferenciados (\$/ML) e resultam, após o tratamento, em níveis de concentração diferentes (gramas por ML) de cada um dos poluentes, como indicado na Tabela 2.10 (isto é,  $a_i$  é a quantidade do componente A após o tratamento 1 de 1 ML de águapolluída).

O problema consiste em determinar a quantidade de água a ser tratada em cada tipo de tratamento, de modo que as exigências ambientais regulamentadas em lei sejam atendidas e o custo total de tratamento seja mínimo. Defina a variável de decisão  $x_i$ ,  $i = 1, 2, 3$ , como a quantidade de água (em ML) por dia a ser tratada pelo tratamento 1, 2 e 3, respectivamente. Pela própria definição, essas variáveis devem ser não-negativas. Nosso objetivo no problema é minimizar o custo de tratamento da água, ou seja, minimizar  $(c_1x_1 + c_2x_2 + c_3x_3)$ , que é a soma dos custos de cada um dos tratamentos a serem efetuados. A quantidade diária de água (em ML) desviada para a produção é simplesmente  $(x_1 + x_2 + x_3)$ , e essa quantidade deve ser no mínimo  $U$ , a quantidade mínima necessária para a fábrica funcionar em cada dia. Temos, portanto, a restrição

$$x_1 + x_2 + x_3 \geq U$$

**Tabela 2.10**  
Eficiência e custo dos tratamentos de água.

Tratamento	1	2	3
<b>A</b>	$a_1$	$a_2$	$a_3$
<b>B</b>	$b_1$	$b_2$	$b_3$
<b>Custo/ML</b>	$c_1$	$c_2$	$c_3$

Da mesma forma, a quantidade desviada não pode superar a vazão do rio. Assim, temos

$$x_1 + x_2 + x_3 \leq V$$

Após o tratamento de  $x_i$  milhões de litros de água, retornam ao rio  $a_i x_i$  e  $b_i x_i$  gramas dos poluentes A e B, respectivamente, por dia. O total de poluentes (em gramas) resultante dos tratamentos é:

$$\begin{aligned} & a_1 x_1 + a_2 x_2 + a_3 x_3 \text{ do poluente A} \\ & b_1 x_1 + b_2 x_2 + b_3 x_3 \text{ do poluente B.} \end{aligned}$$

A concentração de poluentes no rio deve ficar abaixo dos limites considerados aceitáveis. Assim, devemos ter:

$$\begin{aligned} & (a_1 x_1 + a_2 x_2 + a_3 x_3)/V \leq a_0 \\ & (b_1 x_1 + b_2 x_2 + b_3 x_3)/V \leq b_0. \end{aligned}$$

O modelo completo fica, portanto,

$$\begin{aligned} \text{Minimizar } & f(x_1, x_2, x_3) = c_1 x_1 + c_2 x_2 + c_3 x_3 \\ & x_1 + x_2 + x_3 \geq U \\ & x_1 + x_2 + x_3 \leq V \\ & (a_1 x_1 + a_2 x_2 + a_3 x_3)/V \leq a_0 \\ & (b_1 x_1 + b_2 x_2 + b_3 x_3)/V \leq b_0 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{aligned}$$

### 2.2.7 Problemas de corte e empacotamento

Várias indústrias, tais como as de papel, vidro, plástico, metalúrgica, moveleira etc., usam a seguinte estratégia na manufatura de seus produtos: objetos (peças grandes) de tamanhos padronizados são cortados em itens (peças menores) de tamanhos variados, que dependem de solicitações de clientes e que, em geral, não são padronizados. Esse processo de corte gera perdas de material indesejáveis. Surge, então, um problema de otimização que consiste em cortar os objetos para a produção dos itens nas quantidades solicitadas, de modo que a perda de material dos objetos seja mínima. De forma análoga, podemos definir o problema de empacotamento, em que itens devem ser alocados em objetos (por exemplo, contêineres) de modo que o espaço vazio dos objetos seja minimizado. A seguir, apresentamos alguns exemplos.<sup>12</sup>

#### Corte de bobinas de papel

Indústrias de papel produzem bobinas-jumbo de vários metros de largura (por exemplo, 5 metros). Essas bobinas-jumbo são cortadas em sub-bobinas menores, de tamanhos e quantidades variadas (são dezenas de tamanhos diferentes das sub-bobinas: 21 cm, 29,7 cm etc.). Essas sub-bobinas ainda podem ser cortadas na produção de retângulos (tamanho A4, por exemplo). O desperdício deve ser minimizado.

#### Corte de barras metálicas

Indústrias de estruturas metálicas produzem treliças tridimensionais (uma treliça tridimensional é uma estrutura formada essencialmente de barras lineares, muito usada na construção civil, construção de aviões etc.), cortando-se tubos ou perfis de tamanhos grandes em pedaços menores de

<sup>12</sup> Para outros estudos desta classe de problemas, veja, por exemplo, Birgin *et al.* (2005), Dyckhoff e Finke (1992), Hoto *et al.* (2006), Marques e Arenales (2002), Morabito e Arenales (2000), Morabito e Belluzzo (2006), Parada *et al.* (2000), Perin *et al.* (2003) e Yanasse *et al.* (1991, 1994).

diversos comprimentos. Deseja-se saber qual o número mínimo de tubos grandes que devem ser cortados para obter os pedaços menores nos tamanhos e quantidades desejados.

O problema de corte pode ser unidimensional, isto é, apenas uma dimensão é relevante no processo de corte, por exemplo, barras de aço, bobinas de papel, rolos de filme etc., ou bidimensionais, isto é, duas dimensões são relevantes, por exemplo, placas de madeira, tecido, chapas de aço etc., ou tridimensionais, quando três dimensões são relevantes, por exemplo, cortes de blocos de espuma para a produção de colchões e travesseiros ou o empacotamento de produtos em paletes, contêineres, caminhões, vagões ferroviários e navios, que também podem ser vistos como problemas de corte.

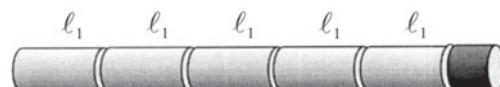
#### *Formulação matemática de um problema de corte*

Para a apresentação da formulação matemática, focalizamos o problema unidimensional, embora essa modelagem possa ser estendida para problemas com mais dimensões.

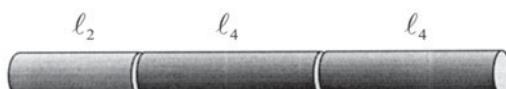
Em um problema unidimensional, deseja-se cortar barras disponíveis de um tamanho padronizado  $L$  para a produção de  $m$  tipos de itens (barras de tamanhos menores) com tamanhos  $\ell_1, \ell_2, \dots, \ell_m$  em quantidades variadas, digamos  $b_1, b_2, \dots, b_m$ , respectivamente (isto é, deve ser produzida a quantidade  $b_i$  da peça de comprimento  $\ell_i$ ).

Conforme os tamanhos e as quantidades dos itens encomendados, podemos ter várias maneiras de cortar as barras em estoque. Uma maneira particular de se cortar uma barra define o que chamamos de *padrão de corte* e, a cada padrão de corte  $j$ ,  $j = 1, 2, \dots$ , associamos um vetor  $m$ -dimensional  $\mathbf{a}_j = (a_{1j}, a_{2j}, \dots, a_{mj})$ ,<sup>13</sup> em que  $a_{ij}$  fornece o número de itens do tipo  $i$  no padrão de corte  $j$ . Para ilustrar esses conceitos de padrões de corte e vetores associados, consideramos os seguintes dados:  $L = 11$ ,  $\ell_1 = 2$ ,  $\ell_2 = 3$ ,  $\ell_3 = 3,5$ ,  $\ell_4 = 4$  ( $m = 4$ ). Podemos ter, por exemplo, os seguintes padrões de corte e seus vetores associados, ilustrados na Figura 2.9.

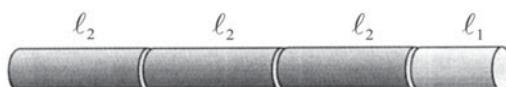
*Padrão 1:*  $\mathbf{a}_1 = (5, 0, 0, 0)^T$



*Padrão 2:*  $\mathbf{a}_2 = (0, 1, 0, 2)^T$



*Padrão 3:*  $\mathbf{a}_3 = (1, 3, 0, 0)^T$



**Figura 2.9** Alguns padrões de corte unidimensionais.

<sup>13</sup> Neste livro, vetores e matrizes são denotados em negrito.

Os padrões de corte na Figura 2.9 são apenas alguns possíveis. Vários outros padrões distintos podem ser determinados. Um vetor  $\alpha = (\alpha_1 \alpha_2 \dots \alpha_m)^T$  representa um padrão de corte se e somente se o seguinte sistema é satisfeito:

$$\begin{aligned} l_1\alpha_1 + l_2\alpha_2 + \dots + l_m\alpha_m &\leq L \\ \alpha_1 \geq 0, \alpha_2 \geq 0, \dots, \alpha_m &\geq 0 \text{ e inteiros.} \end{aligned} \quad (2.15)$$

Modelos que envolvem uma condição de integralidade como em (2.15) são tratados no Capítulo 3. Suponha que existam  $n$  soluções possíveis para esse sistema, isto é,  $n$  padrões de corte. Embora esse número  $n$  seja muito grande para problemas práticos, em geral apenas um pequeno número de padrões de corte é necessário para a resolução do problema de corte. Uma vez definidos os padrões de corte, o problema consiste em determinar quantas barras devem ser cortadas de acordo com cada padrão, de modo que a demanda de cada item seja atendida, utilizando-se o menor número possível de barras disponíveis. Define-se a variável  $x_j$  como o número de barras cortadas conforme o padrão de corte  $j$ . O problema de corte pode ser formulado por:

$$\begin{aligned} \text{Minimizar } f(x_1, x_2, \dots, x_n) &= x_1 + x_2 + \dots + x_n \\ \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix}x_1 + \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix}x_2 + \dots + \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix}x_n &= \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \\ x_j \geq 0, j = 1, \dots, n. \end{aligned} \quad (2.16)$$

em que cada coluna é uma solução do sistema (2.15).

As variáveis deste modelo devem ser necessariamente inteiras, pois representam o número de barras cortadas de acordo com um padrão de corte. Essa condição dificulta substancialmente a resolução do modelo matemático (2.16). Porém, em muitas situações práticas, essa condição de integralidade pode ser relaxada. Suponha que a demanda é dada em toneladas, isto é,  $b_i$  é a quantidade em toneladas demandada para o item de largura  $l_i$ . O lado esquerdo da restrição de demanda do modelo (2.16) fornece o número de itens cortados e, portanto, precisa ser modificado. Suponha, também, que cada bobina em estoque mede  $L$  cm de largura e seu peso é  $T$  toneladas, de modo que cada centímetro cortado pesa  $\rho = \frac{T}{L}$  tonelada/cm ( $\rho$  é chamado peso específico linear). Assim, um item de largura  $l_i$  cm cortado da bobina pesa  $\rho l_i$  toneladas. Por exemplo, se  $L = 400$  cm e  $T = 1$  tonelada, um item de largura  $l_1 = 40$  cm pesa 0,1 tonelada (100kg). Como, em um padrão de corte  $j$ , o número de itens do tipo  $i$  é  $a_{ij}$ , segue-se que a quantidade em toneladas do item tipo  $i$  produzida pelo padrão de corte  $j$  é  $\rho l_i a_{ij}$  toneladas e o modelo (2.16) deve ser alterado para:

$$\begin{aligned} \text{Minimizar } f(x_1, x_2, \dots, x_n) &= x_1 + x_2 + \dots + x_n \\ \begin{bmatrix} \rho l_1 a_{11} \\ \rho l_2 a_{21} \\ \vdots \\ \rho l_m a_{m1} \end{bmatrix}x_1 + \begin{bmatrix} \rho l_1 a_{12} \\ \rho l_2 a_{22} \\ \vdots \\ \rho l_m a_{m2} \end{bmatrix}x_2 + \dots + \begin{bmatrix} \rho l_1 a_{1n} \\ \rho l_2 a_{2n} \\ \vdots \\ \rho l_m a_{mn} \end{bmatrix}x_n &= \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \\ x_j \geq 0, j = 1, \dots, n, \end{aligned} \quad (2.17)$$

em que  $x_j$  é o número de bobinas cortadas segundo o padrão de corte  $j$ , que deve ser um valor inteiro. Porém, substituindo  $\rho = \frac{T}{L}$  e deixando  $T$  multiplicar cada coluna do modelo (2.17), obtém-se

$$\text{Minimizar } f(x_1, x_2, \dots, x_n) = x_1 + x_2 + \dots + x_n$$

$$\begin{bmatrix} \frac{\ell_1}{L} a_{11} \\ \frac{\ell_2}{L} a_{21} \\ \vdots \\ \frac{\ell_m}{L} a_{m1} \end{bmatrix} (Tx_1) + \begin{bmatrix} \frac{\ell_1}{L} a_{12} \\ \frac{\ell_2}{L} a_{22} \\ \vdots \\ \frac{\ell_m}{L} a_{m2} \end{bmatrix} (Tx_2) + \dots + \begin{bmatrix} \frac{\ell_1}{L} a_{1n} \\ \frac{\ell_2}{L} a_{2n} \\ \vdots \\ \frac{\ell_m}{L} a_{mn} \end{bmatrix} (Tx_n) = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \quad (2.18)$$

$x_j \geq 0, j = 1, \dots, n$

Podemos fazer em (2.18) a seguinte mudança de variável:

$$y_j = Tx_j \quad \text{toneladas cortadas conforme o padrão de corte } j,$$

e temos um modelo equivalente (sem perda de generalidade, eliminamos o termo comum  $1/T$  das variáveis da função objetivo):

$$\text{Minimizar } g(y_1, y_2, \dots, y_n) = y_1 + y_2 + \dots + y_n$$

$$\begin{bmatrix} \frac{\ell_1}{L} a_{11} \\ \frac{\ell_2}{L} a_{21} \\ \vdots \\ \frac{\ell_m}{L} a_{m1} \end{bmatrix} y_1 + \begin{bmatrix} \frac{\ell_1}{L} a_{12} \\ \frac{\ell_2}{L} a_{22} \\ \vdots \\ \frac{\ell_m}{L} a_{m2} \end{bmatrix} y_2 + \dots + \begin{bmatrix} \frac{\ell_1}{L} a_{1n} \\ \frac{\ell_2}{L} a_{2n} \\ \vdots \\ \frac{\ell_m}{L} a_{mn} \end{bmatrix} y_n = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \quad (2.19)$$

$y_j \geq 0, j = 1, \dots, n.$

Em (2.19), uma solução com  $y_j = 3,5$  tem significado, pois indica que 3,5 toneladas de material (em bobinas de comprimento  $L$ ) deverão ser cortadas segundo o padrão  $j$ . Supondo que as bobinas sejam produzidas pela indústria (caso da indústria de papel), essa solução pode ser traduzida na produção de, por exemplo, 3 bobinas de 1 tonelada e 1 bobina de 0,5 tonelada. Em caso de bobinas adquiridas de terceiros em pesos padronizados, a solução não-inteira ainda faz sentido se as bobinas puderem ser parcialmente desenroladas e cortadas (esse é o caso das indústrias metalúrgicas que compram e cortam bobinas de aço).

Embora os modelos matemáticos (2.16) e a modificação (2.19) tenham sido desenvolvidos para o caso unidimensional (o sistema (2.15) define um padrão de corte unidimensional), esses modelos são também adequados para problemas de corte bidimensionais, como cortes em placas de madeira, chapas de aço, placas de vidro etc. A dificuldade de tratar problemas com mais de uma dimensão está na geração dos padrões de corte (lembre-se que o sistema (2.15) modela padrões de corte unidimensionais).

**Exemplo 2.9** Uma indústria de papel produz bobinas-jumbo de  $L = 400\text{cm}$  de largura e cada uma pesa  $T = 1$  tonelada. Os jumbos devem ser cortados em bobinas menores nas larguras e quantidades apresentadas na Tabela 2.11, conforme solicitações de diversos clientes.

Na Figura 2.10, listamos 6 padrões de corte que poderiam ser utilizados para cortar os jumbos (outros padrões de corte são deixados a cargo do leitor).

Utilizando os padrões de corte na Figura 2.10 e  $\frac{\ell_1}{L} = 0,1; \frac{\ell_2}{L} = 0,1125; \frac{\ell_3}{L} = 0,1375; \frac{\ell_4}{L} = 0,15$ , o modelo (2.19) é dado por:

**Tabela 2.11**  
Dados para um problema de corte.

Larguras ( $l_i$ )	Dados da demanda	
		Quantidades ( $b_i$ )
40cm		5 ton
45cm		3,5 ton
55cm		4 ton
60cm		5 ton

Padrão de corte 1:  $\mathbf{a}_1 = (10 \ 0 \ 0 \ 0)^T$



Padrão de corte 2:  $\mathbf{a}_2 = (1 \ 8 \ 0 \ 0)^T$



Padrão de corte 3:  $\mathbf{a}_3 = (0 \ 0 \ 7 \ 0)^T$



Padrão de corte 4:  $\mathbf{a}_4 = (1 \ 0 \ 0 \ 6)^T$



Padrão de corte 5:  $\mathbf{a}_5 = (0 \ 4 \ 4 \ 0)^T$



Padrão de corte 6:  $\mathbf{a}_6 = (0 \ 0 \ 4 \ 3)^T$



**Figura 2.10** Possíveis padrões de corte para uma bobina-jumbo.

Minimizar  $g(y_1, y_2, y_3, y_4, y_5, y_6, \dots) = y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + \dots$

$$\begin{bmatrix} 10 \times 0,1 \\ 0 \\ 0 \\ 0 \end{bmatrix} y_1 + \begin{bmatrix} 1 \times 0,1 \\ 8 \times 0,1125 \\ 0 \\ 0 \end{bmatrix} y_2 + \begin{bmatrix} 0 \\ 0 \\ 7 \times 0,1375 \\ 0 \end{bmatrix} y_3 + \begin{bmatrix} 1 \times 0,1 \\ 0 \\ 0 \\ 6 \times 0,15 \end{bmatrix} y_4 + \begin{bmatrix} 0 \\ 4 \times 0,1125 \\ 4 \times 0,1375 \\ 0 \end{bmatrix} y_5 + \begin{bmatrix} 0 \\ 0 \\ 4 \times 0,1375 \\ 3 \times 0,15 \end{bmatrix} y_6 + \dots = \begin{bmatrix} 5 \\ 3,5 \\ 4 \\ 5 \end{bmatrix}$$

$y_1, y_2, y_3, y_4, y_5, y_6, \dots \geq 0$ .

■

## 2.2.8 Ajuste de curvas

O problema de ajuste de curvas tem aplicação em diversas áreas e consiste, em geral, em buscar uma lei que rege um fenômeno observável, o qual depende de um conjunto de fatores controláveis. Por exemplo, um pesquisador da área de saúde mede a pressão arterial de milhares de pacientes (fenômeno observável ou variável dependente) que deseja relacionar à idade, ao sexo, ao peso etc. (variáveis independentes). Denotando essas variáveis independentes por  $a_1, a_2, \dots, a_n$ , supondo que haja  $n$  delas, e o fenômeno observável por  $b$ , escrevemos:

$$b = a_1x_1 + a_2x_2 + \dots + a_nx_n + \varepsilon,$$

supondo que o fenômeno observável dependa linearmente das variáveis independentes, em que  $\varepsilon$  é um ruído inevitável e desconhecido que o pesquisador deve considerar em seu *modelo de regressão linear* (como é conhecida a equação anterior), e  $x_j, j = 1, 2, \dots, n$  são incógnitas a serem determinadas. Para cada observação feita (por exemplo, a pressão de cada paciente), o modelo é aplicado. Supondo que fizemos  $m$  observações, temos

$$b_i = a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n + \varepsilon_i \quad i = 1, 2, \dots, m,$$

em que  $b_i$  e  $a_{ij}$  são conhecidos (por exemplo, a pressão arterial observada e os dados do paciente  $i$ , como idade, peso, altura etc.). Considere que o modelo seja adequado e que os dados tenham sido obtidos de modo que os ruídos sejam os menores possíveis (mesmo que inevitáveis e desconhecidos). Surge, então, um problema de otimização que consiste em determinar as incógnitas  $x_1, x_2, \dots, x_n$  de modo que os ruídos sejam mínimos em magnitude, já que podem ser positivos ou negativos (por exemplo, a pressão arterial pode ter sido medida acima ou abaixo do valor real). Para isso, precisamos de uma função para medir o tamanho do ruído  $\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_m)$  nas várias observações. Algumas funções de medida do ruído são bastante estudadas. A mais popular é a soma dos quadrados dos ruídos, e o problema de otimização é:

$$\text{Minimizar } f(x_1, x_2, \dots, x_n, \varepsilon_1, \varepsilon_2, \dots, \varepsilon_m) = (\varepsilon_1)^2 + (\varepsilon_2)^2 + \dots + (\varepsilon_m)^2$$

$$b_i = a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n + \varepsilon_i \quad i = 1, 2, \dots, m$$

Este é o clássico *problema dos quadrados mínimos*, cuja solução é obtida diretamente do cálculo, bastando substituir  $\varepsilon_i = b_i - (a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n)$  na função a ser minimizada e exigir que as derivadas parciais sejam nulas. Uma outra medida para o ruído  $\varepsilon$ , de muito interesse para pesquisadores em regressão linear, é a soma dos módulos dos ruídos e o problema de otimização pode ser expresso por:

$$\text{Minimizar } f(x_1, x_2, \dots, x_n, \varepsilon_1, \varepsilon_2, \dots, \varepsilon_m) = |\varepsilon_1| + |\varepsilon_2| + \dots + |\varepsilon_m|$$

$$b_i = a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n + \varepsilon_i \quad i = 1, 2, \dots, m$$

A função anterior a ser minimizada acima é linear por partes:

$$|\varepsilon_i| = \begin{cases} \varepsilon_i & \text{se } \varepsilon_i \geq 0 \\ -\varepsilon_i & \text{se } \varepsilon_i \leq 0 \end{cases}$$

e não diferenciável em  $\varepsilon_i = 0$ . Embora isso possa apresentar uma dificuldade para o cálculo diferencial, pode ser bem formulada como um modelo de otimização linear, com a seguinte estratégia de redefinir as variáveis  $\varepsilon_i$ :

$$\varepsilon_i = \varepsilon_i^+ - \varepsilon_i^- \text{ em que } \varepsilon_i^+ \geq 0, \varepsilon_i^- \geq 0 \quad i = 1, 2, \dots, m.$$

Podemos escrever  $|\varepsilon_i| = \varepsilon_i^+ + \varepsilon_i^-$ . A rigor, deveríamos proibir que  $\varepsilon_i^+ > 0$  e  $\varepsilon_i^- > 0$  simultaneamente, já que existem infinitos valores para  $\varepsilon_i^+$  e  $\varepsilon_i^-$  que fornecem o mesmo valor de  $\varepsilon_i = \varepsilon_i^+ - \varepsilon_i^-$  (por

exemplo,  $\varepsilon_i = 10 = 10 - 0 = 15 - 5$  etc.). Porém, como minimizamos  $\varepsilon_i^+ + \varepsilon_i^-$ , qualquer solução ótima tem necessariamente  $\varepsilon_i^+ = 0$  ou  $\varepsilon_i^- = 0$ . Para ver isso, suponha que  $\varepsilon_i^+ = \varepsilon_i + \theta$  e  $\varepsilon_i^- = \theta$ ,  $\theta > 0$  (note que  $\varepsilon_i^+ - \varepsilon_i^- = \varepsilon_i$  qualquer que seja  $\theta$ , portanto, as equações do modelo não dependem de  $\theta$ ). Entretanto, a função a ser minimizada depende de  $\theta$ , uma vez que  $\varepsilon_i^+ + \varepsilon_i^- = \varepsilon_i + 2\theta$  é sempre maior que  $\varepsilon_i$  para todo  $\theta > 0$ . Portanto, nenhuma solução ótima pode ter  $\theta > 0$ . De modo semelhante, para as soluções do tipo:  $\varepsilon_i^+ = \theta$  e  $\varepsilon_i^- = \varepsilon_i + \theta$ ,  $\theta > 0$ . Desse modo, restringimo-nos às soluções em que  $\theta = 0$ . Dessa forma, o problema de minimizar a soma dos módulos dos ruídos pode ser escrita como:

$$\begin{aligned} \text{Minimizar } f(x_1, x_2, \dots, x_n, \varepsilon_1^+, \varepsilon_1^-, \varepsilon_2^+, \varepsilon_2^-, \dots, \varepsilon_m^+, \varepsilon_m^-) &= \varepsilon_1^+ + \varepsilon_1^- + \varepsilon_2^+ + \varepsilon_2^- + \dots + \varepsilon_m^+ + \varepsilon_m^- \\ a_{i1}x_1 + a_{i2}x_2 + \dots + a_nx_n + \varepsilon_i^+ - \varepsilon_i^- &= b_i \quad i = 1, 2, \dots, m \\ \varepsilon_i^+ \geq 0, \varepsilon_i^- \geq 0 & \quad i = 1, 2, \dots, m. \end{aligned}$$

De forma análoga, podemos escrever um problema de otimização linear se os ruídos são ponderados por valores positivos,  $v_i > 0$ ,  $i = 1, \dots, m$  (por exemplo, a ponderação  $v_i = \frac{1}{i}$ ,  $i = 1, \dots, m$ , procura um modelo no qual as primeiras observações são mais relevantes). A função a ser minimizada é:

$$\text{Minimizar } f(x_1, x_2, \dots, x_n, \varepsilon_1, \varepsilon_2, \dots, \varepsilon_m) = v_1|\varepsilon_1| + v_2|\varepsilon_2| + \dots + v_m|\varepsilon_m|.$$

**Exemplo 2.10** A Tabela 2.12 contém dados de um experimento em que, para cada valor de  $x$ , um valor de  $y$  foi medido. Supõe-se que as variáveis sejam linearmente relacionadas:  $y = ax + b$ .

O problema consiste em determinar os coeficientes  $a$  e  $b$  da reta  $y = ax + b$  (ver Figura 2.11) que melhor se ajusta aos dados da Tabela 2.14. Considerando que um ruído ocorre em cada medida, escrevemos:  $y_i = ax_i + b + \varepsilon_i$ ,  $i = 1, \dots, 4$ :

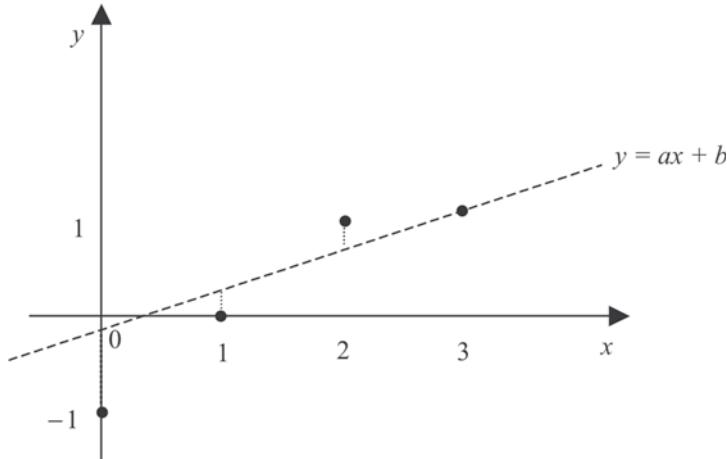
$$\begin{aligned} -1 &= a \times 0 + b + \varepsilon_1 \\ 0 &= a \times 1 + b + \varepsilon_2 \\ 1 &= a \times 2 + b + \varepsilon_3 \\ 1 &= a \times 3 + b + \varepsilon_4 \end{aligned}$$

O problema de ajuste que minimiza a soma dos ruídos em módulo pode ser escrito como:

$$\begin{aligned} \text{Minimizar } f(a, b, \varepsilon_1^+, \varepsilon_1^-, \varepsilon_2^+, \varepsilon_2^-, \varepsilon_3^+, \varepsilon_3^-, \varepsilon_4^+, \varepsilon_4^-) &= \varepsilon_1^+ + \varepsilon_1^- + \varepsilon_2^+ + \varepsilon_2^- + \varepsilon_3^+ + \varepsilon_3^- + \varepsilon_4^+ + \varepsilon_4^- \\ b + \varepsilon_1^+ - \varepsilon_1^- &= -1 \\ a + b + \varepsilon_2^+ - \varepsilon_2^- &= 0 \\ 2a + b + \varepsilon_3^+ - \varepsilon_3^- &= 1 \\ 3a + b + \varepsilon_4^+ - \varepsilon_4^- &= 1 \\ \varepsilon_1^+ \geq 0, \varepsilon_1^- \geq 0, \varepsilon_2^+ \geq 0, \varepsilon_2^- \geq 0, \varepsilon_3^+ \geq 0, \varepsilon_3^- \geq 0, \varepsilon_4^+ \geq 0, \varepsilon_4^- \geq 0. & \end{aligned}$$

**Tabela 2.12**  
Dados experimentais.

$x$	0	1	2	3
$y$	-1	0	1	1



**Figura 2.11** Representação gráfica para o ajuste de uma reta.

### 2.2.9 Controle ótimo de sistemas lineares

Considere um sistema dinâmico discreto que evolui segundo o seguinte modelo:

$$\begin{aligned}\mathbf{x}_{t+1} &= \mathbf{Ax}_t + \mathbf{Bu}_t & t = 0, \dots, T-1 \\ y_t &= \mathbf{c}^T \mathbf{x}_t + \mathbf{d}^T \mathbf{u}_t & t = 0, \dots, T-1\end{aligned}$$

O vetor de variáveis  $\mathbf{x}_t$  é chamado *estado do sistema no instante t* (o estado inicial  $\mathbf{x}_0$  é dado),  $\mathbf{u}_t$  é o vetor das *variáveis de controle* (isto é, são as variáveis para as quais podemos escolher seus valores),  $y_t$  é um escalar chamado *resposta do sistema no instante t* que depende linearmente das variáveis de estado e de controle. Assim, uma vez escolhido o valor de  $\mathbf{u}_0$ , então podemos calcular  $\mathbf{x}_1 = \mathbf{Ax}_0 + \mathbf{Bu}_0$  e  $y_0 = \mathbf{c}^T \mathbf{x}_0 + \mathbf{d}^T \mathbf{u}_0$ ; escolhido  $\mathbf{u}_1$ , calculamos  $\mathbf{x}_2 = \mathbf{Ax}_1 + \mathbf{Bu}_1$  e  $y_1 = \mathbf{c}^T \mathbf{x}_1 + \mathbf{d}^T \mathbf{u}_1$ ; e assim por diante. Podemos ter ainda restrições adicionais sobre as variáveis de estado e de controle (por exemplo:  $\underline{\mathbf{x}}_t \leq \mathbf{x}_t \leq \bar{\mathbf{x}}_t$ ,  $\underline{\mathbf{u}}_t \leq \mathbf{u}_t \leq \bar{\mathbf{u}}_t$ ). As aplicações de sistemas que evoluem conforme o modelo acima são diversas, como circuitos elétricos, sistemas de manufatura, crescimento econômico etc. Tipicamente, devemos escolher as variáveis de controle  $\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{T-1}$  para levar o sistema a um estado final  $\mathbf{x}_T$  desejável (estado-alvo):  $\mathbf{x}_T = \mathbf{x}_T^*$ . Normalmente, existem muitas escolhas possíveis para  $\mathbf{u}_t$ , de modo que podemos escolher uma que faça com que a magnitude da resposta do sistema seja tão pequena quanto possível em todos os instantes, ou seja, temos o seguinte problema de otimização:

$$\begin{aligned}&\text{Minimizar } \{\max |y_t|, t = 0, \dots, T-1\} \\ \mathbf{x}_{t+1} &= \mathbf{Ax}_t + \mathbf{Bu}_t & t = 0, \dots, T-1 \\ y_t &= \mathbf{c}^T \mathbf{x}_t + \mathbf{d}^T \mathbf{u}_t & t = 0, \dots, T-1 \\ \underline{\mathbf{x}}_t &\leq \mathbf{x}_t \leq \bar{\mathbf{x}}_t & t = 1, \dots, T \\ \underline{\mathbf{u}}_t &\leq \mathbf{u}_t \leq \bar{\mathbf{u}}_t & t = 0, \dots, T-1 \\ \mathbf{x}_0 &\text{ e } \mathbf{x}_T \text{ dados.}\end{aligned}$$

A função a ser minimizada  $\{\max |y_t|, t = 0, \dots, T-1\}$  não é uma função linear, porém, se a chamamos de:

$$z = \{\max |y_t|, t = 0, \dots, T-1\}$$

então, as relações

$$-z \leq y_t \leq z, \quad t = 0, \dots, T-1$$

são válidas. O valor mínimo de  $z$  que satisfaz todas essas desigualdades é exatamente o  $\{\max |y_t|, t = 0, \dots, T-1\}$ . Assim, para se achar o valor mínimo de  $\{\max |y_t|, t = 0, \dots, T-1\}$ , basta resolver o problema

$$\begin{array}{ll} \text{Minimizar } z \\ -z \leq y_t \leq z & t = 0, \dots, T-1, \end{array}$$

de modo que o modelo anterior pode ser descrito por:

$$\begin{array}{ll} \text{Minimizar } z \\ -z \leq y_t \leq z & t = 0, \dots, T-1 \\ \mathbf{x}_{t+1} = \mathbf{Ax}_t + \mathbf{Bu}_t & t = 0, \dots, T-1 \\ y_t = \mathbf{c}^T \mathbf{x}_t + \mathbf{d}^T \mathbf{u}_t & t = 0, \dots, T-1 \\ \underline{\mathbf{x}}_t \leq \mathbf{x}_t \leq \bar{\mathbf{x}}_t & t = 1, \dots, T-1 \\ \underline{\mathbf{u}}_t \leq \mathbf{u}_t \leq \bar{\mathbf{u}}_t & t = 0, \dots, T-1 \\ \mathbf{x}_0 \text{ e } \mathbf{x}_T \text{ dados} \end{array}$$

**Exemplo 2.11**<sup>14</sup> Suponha que um foguete viaje em linha reta e sejam  $(x_t, v_t)$  a posição e a velocidade do foguete no instante  $t$ , as quais são determinadas pela aceleração  $a_t$  no instante  $t$ . Tais variáveis estão relacionadas por:

$$\begin{aligned} v_t &= \frac{\partial x_t}{\partial t} \quad (\text{velocidade é a derivada de } x_t) \\ a_t &= \frac{\partial v_t}{\partial t} \quad (\text{aceleração é a derivada de } v_t). \end{aligned}$$

Uma aproximação para este sistema consiste em fazer  $v_t = \frac{\partial x_t}{\partial t} = \frac{x_{t+\Delta t} - x_t}{\Delta t}$ ,  $a_t = \frac{\partial v_t}{\partial t} = \frac{v_{t+\Delta t} - v_t}{\Delta t}$ . Considerando-se  $\Delta t$  fixo e denotando-se  $x_{t+1} = x_{t+\Delta t}$  e  $v_{t+1} = v_{t+\Delta t}$  (ou seja, considerando o instante inicial  $t = 0$ , então,  $t = 1$  corresponde ao instante  $\Delta t$ ,  $t = 2$  ao instante  $2\Delta t$ , ...,  $t = T$  corresponde ao instante  $T\Delta t$ ), segue que:

$$\begin{aligned} x_{t+1} &= x_t + \Delta t v_t \\ v_{t+1} &= v_t + \Delta t a_t. \end{aligned}$$

Note que as variáveis de estado são  $(x_t, v_t)$  e as de controle,  $a_t, t = 1, \dots, T-1$ . Admitimos que o consumo de combustível no instante  $t$  seja proporcional à magnitude da aceleração  $|a_t|$ . Suponha que o foguete esteja inicialmente em repouso, isto é,  $x_0 = 0$  e  $v_0 = 0$ . Desejamos que o foguete decole e aterrisse suavemente a uma distância de uma unidade da origem depois de transcorridas  $T$  unidades de tempo, isto é,  $x_T = 1$  (posição final do foguete) e  $v_T = 0$  (atterrisagem suave). Além disso, queremos controlar a aceleração de modo econômico, por exemplo, minimizar  $\sum_{t=0}^{T-1} |a_t|$ . Uma alternativa seria minimizar  $\{\max |a_t|, t = 0, \dots, T-1\}$ , como descrito no modelo geral. Ambas as funções podem ser escritas de forma linear (veja o problema de ajuste de curvas anterior). Identifique, neste exemplo, as matrizes  $\mathbf{A}$ ,  $\mathbf{B}$  e os vetores  $\mathbf{c}$  e  $\mathbf{d}$  no modelo geral (note que a resposta do sistema é  $y_t = a_t$ ). ■

## 2.2.10 Problemas lineares por partes

Os problemas de ajuste de curva e controle ótimo definidos nas Seções 2.2.8 e 2.2.9 envolvem funções a serem minimizadas que não são lineares, mas podem ser redefinidas como funções lineares.

Funções lineares por partes também aparecem em outros contextos. Por exemplo, no problema de dimensionamento de lotes, modelado na Seção 2.2.3, uma variável importante consiste no

<sup>14</sup> Este exemplo é baseado em Bertsimas e Tsitsiklis (1997).

nível de estoque de um produto:  $I_{it}$  é a quantidade do produto  $i$  disponível em estoque ao final do período  $t$ . A equação de balanceamento de estoque, junto com a não-negatividade do nível de estoque:

$$I_{it} = I_{i,t-1} + x_{it} - d_{it} \quad \text{e} \quad I_{it} \geq 0,$$

garantem que a demanda no período  $t$  deve ser satisfeita:  $I_{i,t-1} + x_{it} \geq d_{it}$ .

Se permitirmos que a variável  $I_{it}$  assuma valores negativos, isso significa que  $I_{i,t-1} + x_{it} < d_{it}$ , isto é, a demanda não é satisfeita no período  $t$ , o que pode acontecer em uma situação real. Quando  $I_{it} < 0$ , dizemos *estoque em falta* e  $I_{it} > 0$  *estoque em mãos*. No caso de estoque em falta, normalmente incorre-se em uma *penalidade* pelo não-atendimento da demanda. Considerando o estoque em mãos e o estoque em falta, temos uma função custo (ver Figura 2.12) do tipo

$$f(I_{11}, I_{12}, \dots) = f_{11}(I_{11}) + f_{12}(I_{12}) + \dots$$

em que:

$$f_{it}(I_{it}) = \begin{cases} h_{it}^+ I_{it} & \text{se } I_{it} \geq 0 \\ h_{it}^- I_{it} & \text{se } I_{it} \leq 0 \end{cases}$$

Análogo ao problema de ajuste de curvas (Seção 2.2.8), podemos redefinir a variável de estoque  $I_{it}$  como  $I_{it} = I_{it}^+ - I_{it}^-$ , com  $I_{it}^+ \geq 0$ ,  $I_{it}^- \geq 0$ , de modo que a função anterior, linear por partes, pode ser escrita como:

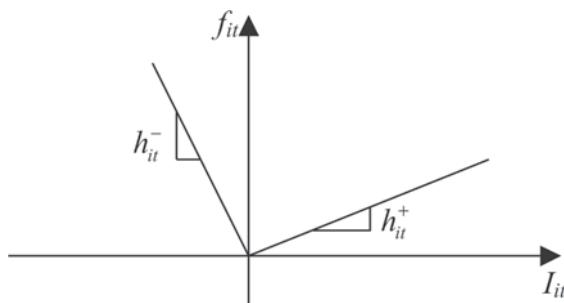
$$f(I_{11}^+, I_{11}^-, I_{12}^+, I_{12}^-, \dots) = h_{11}^+ I_{11}^+ + h_{11}^- I_{11}^- + h_{12}^+ I_{12}^+ + h_{12}^- I_{12}^- + \dots$$

Podemos também definir outros critérios a serem minimizados, como, por exemplo, podemos desejar que o valor máximo, em módulo, das variáveis que medem estoque em mãos ou em falta seja o menor possível. Isso pode ser descrito por:<sup>15</sup>

$$\text{Minimizar } \{\text{máximo } \{|I_{it}|\}, \text{ para todo } i, t\}.$$

Podemos, ainda, desejar que os estoques estejam entre limites desejáveis, mínimo e máximo, e níveis fora desses intervalos devem ser evitados. Um esboço de uma função a ser minimizada que traduz esse desejo é ilustrado na Figura 2.13.

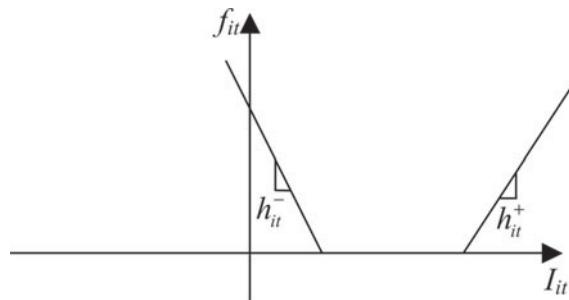
Este problema pode também ser linearizado (isto é, escrito de forma equivalente como um problema linear), como feito nos casos anteriores.<sup>16</sup>



**Figura 2.12** Função penalidade para o estoque não-nulo.

15 Para mais aplicações de problemas *minmax*, veja, por exemplo, Martinez *et al.* (1994) e Balbo *et al.* (2006).

16 Veja Marins *et al.* (1997), Oliveira e Lyra (2004) e Cavichia e Arenales (2000) para estudos de problemas com funções lineares por partes.



**Figura 2.13** Função penalidade para o estoque fora dos limites desejados.

## 2.3 HIPÓTESES DE LINEARIDADE

Nos modelos de otimização linear, como aqueles apresentados na Seção 2.2, são admitidas algumas hipóteses às quais as grandezas envolvidas precisam obedecer: aditividade, proporcionalidade e fracionamento.

- *Hipótese de aditividade*

Esta hipótese pressupõe que o todo é igual à soma das partes. Por exemplo, se em 1kg do ingrediente  $j$  encontramos 200g (ou 0,2kg) do componente  $i$  e, em 1kg do ingrediente  $k$ , encontramos 100g do mesmo componente, então a mistura de 2kg, obtida pela adição de 1kg de cada ingrediente  $j$  e  $k$ , tem 300g do componente  $i$ . Cabe alertar que, apesar de esse comportamento ser comum, são muitas as aplicações práticas em que isso não acontece ou pode ser negligenciado. Por exemplo, quando se misturam diferentes tipos de ingredientes cujos componentes reagem biológica, física ou quimicamente, formando outros componentes. Isso acontece, por exemplo, se adicionarmos o equivalente em volume a 0,1 litro de açúcar em 1 litro de água. O volume resultante da água açucarada não é de 1,1 litro. Outro exemplo é o preparo de fluidos para freios de veículos, em que os insumos reagem quimicamente, alterando características importantes como ponto de ebulição ou viscosidade.

- *Hipótese de proporcionalidade*

Esta hipótese pressupõe que, se  $a_{ij}$  é a quantidade do componente  $i$  em uma unidade do ingrediente  $j$ , então  $a_{ij}x_j$  será a quantidade do componente  $i$  em  $x_j$  unidades; se uma unidade do ingrediente  $j$  custa  $c_j$ , então  $x_j$  unidades custam  $c_jx_j$ . Por exemplo, se 1kg de um ingrediente contém 200g de um componente, então  $\frac{1}{2}kg$  deste ingrediente contém 100g do mesmo componente, assim como 3kg contêm 600g.

- *Hipótese de fracionamento*

Valores fracionários para as variáveis são aceitáveis. Por exemplo, podemos utilizar 1kg de um ingrediente em uma mistura, como também 0,25kg desse ingrediente. Porém, dependendo do tipo do problema, o arredondamento de valores pode dar uma conotação muito distorcida da prática, invalidando os resultados. Por exemplo, uma variável pode representar o número de máquinas ou veículos a serem adquiridos por uma empresa. Nesses casos, em geral, deve-se restringir

essas variáveis ao conjunto dos números inteiros, tornando o modelo de otimização inteira. O bom senso deve ditar simplificações razoáveis. Por exemplo, se a variável  $x_1$  representa o número de peças do tipo 1 que devem ser fabricadas e o resultado do modelo matemático é  $x_1 = 331,4$ , produzir 331 ou 332 peças do tipo 1 são decisões aceitáveis e coerentes na prática, então não há motivo para modelar a variável  $x_1$  como uma variável inteira. Um modelo de otimização inteira, em geral, é bem mais difícil de se resolver do que um modelo de otimização linear, conforme discutido no Capítulo 3.

Embora as hipóteses de linearidade possam sugerir que modelos de otimização linear têm utilização limitada, os exemplos de aplicações nas mais diversas áreas de conhecimento e situações práticas indicam o contrário, como ilustrado na seção anterior. Existem inúmeros outros exemplos de aplicações de modelos de otimização linear em diversas áreas das engenharias (alimentos, aeronáutica, civil, computação, controle e automação, elétrica, eletrônica, florestal, produção, química, mecânica, mecatrônica, metalúrgica, minas, naval etc.), em economia e finanças, medicina, física, ciências sociais, ecologia, esportes etc. Os modelos de otimização linear apóiam o processo de tomada de decisões e, para extraírem desses modelos informações de interesse do decisor, foram desenvolvidos diversos métodos que são apresentados nas seções seguintes deste capítulo.

## 2.4 CONCEITOS BÁSICOS

Nesta seção apresentamos terminologias que são usadas ao longo deste capítulo.

### 2.4.1 Um problema de otimização linear

**Definição 2.1** (*forma padrão*) A seguinte forma do problema de otimização linear é chamada de *forma padrão*:

$$\text{Minimizar } f(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (2.20)$$

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= b_m \end{aligned} \quad (2.21)$$

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0. \quad (2.22)$$

A função linear  $f$  em (2.20), a ser minimizada, é chamada *função objetivo*, o sistema de equações lineares em (2.21) define as *restrições do problema*, juntamente com as *condições de não negatividade* das variáveis em (2.22). O problema (2.20)–(2.22) pode ser escrito equivalentemente em notação matricial como:

$$\text{Minimizar } f(\mathbf{x}) = \mathbf{c}^T \mathbf{x}$$

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

$$\mathbf{x} \geq \mathbf{0},$$

em que

$$\bullet \quad \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \quad \text{é uma matriz } m \times n, \text{ chamada matriz dos coeficientes (ou matriz tecnológica);}$$

- $\mathbf{c}^T = (c_1 \ c_2 \ \dots \ c_n)$  é o vetor de custos,
- $\mathbf{x}^T = (x_1 \ x_2 \ \dots \ x_n)$  é o vetor das variáveis ou incógnitas,
- $\mathbf{b}^T = (b_1 \ b_2 \ \dots \ b_m)$  é o vetor dos termos independentes ou de recursos,
- $\mathbf{0}^T = (0 \ 0 \ \dots \ 0)$  é o vetor cujos elementos são todos iguais a 0.

*Observações:* Denotamos por  $\mathbf{x}^T$  o transposto do vetor  $\mathbf{x}$ . Em geral, supomos que um vetor é do tipo coluna, isto é, um vetor de  $n$  coordenadas é uma matriz  $n \times 1$ . Por comodidade, utilizamos também a notação  $(x_1, x_2, \dots, x_n)$  para o vetor  $\mathbf{x}$ .

Na forma padrão, o problema de otimização linear tem as seguintes características:

- a função objetivo deve ser minimizada;
- as restrições do problema são definidas por um sistema de equações lineares;
- as condições de não-negatividade de todas as variáveis de decisão complementam as restrições do problema.

Qualquer problema de otimização linear pode ser escrito na forma padrão. Com isso, sem perda de generalidade, desenvolvimentos teóricos, métodos e propriedades de problemas de otimização linear podem ser apresentados considerando o problema nessa forma.

**Definição 2.2** (*solução factível e região factível*) Uma solução  $(x_1, x_2, \dots, x_n)$  é dita factível se satisfizer todas as restrições (2.21) e as condições de não-negatividade (2.22). O conjunto de todas as soluções factíveis é chamado *região factível*.

**Exemplo 2.12** Analise o problema de otimização linear dado por

$$\begin{aligned} \text{Minimizar} \quad f(x_1, x_2, x_3) &= 2x_1 - x_2 + 4x_3 \\ x_1 + 2x_2 + x_3 &= 3 \\ x_2 + 2x_3 &= 4 \\ x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0. \end{aligned}$$

Em termos do modelo (2.20)–(2.22), temos:  $m = 2$  (duas restrições),  $n = 3$  (três variáveis),  $c_1 = 2$ ,  $c_2 = -1$ ,  $c_3 = 4$ ,  $a_{11} = 1$ ,  $a_{12} = 2$ ,  $a_{13} = 1$ ,  $b_1 = 3$ ,  $a_{21} = 0$  (observe que a variável  $x_1$  não aparece explicitamente na segunda equação),  $a_{22} = 1$ ,  $a_{23} = 2$ ,  $b_2 = 4$ . As variáveis deste problema  $(x_1, x_2, x_3)$  correspondem a um vetor de três coordenadas e, portanto, o espaço de possíveis soluções está contido no  $\mathbb{R}^3$ . A solução particular  $x_1 = 1$ ,  $x_2 = 0$ ,  $x_3 = 2$  é uma solução factível, pois satisfaz todas as restrições do problema. Para essa solução, a função objetivo tem valor  $f(1, 0, 2) = 10$ . Uma outra solução factível  $x_1 = 0,25$ ,  $x_2 = 0,5$ ,  $x_3 = 1,75$  (verifique que satisfaz todas as restrições do problema) tem valor da função objetivo  $f(0,25; 0,5; 1,75) = 7$ . Isso significa que esta solução é melhor do que a anterior, já que o objetivo do problema é determinar o menor valor possível para

$f(x_1, x_2, x_3)$ . A matriz dos coeficientes é  $2 \times 3$ , dada por:  $\mathbf{A} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}$ , o vetor dos custos por  $\mathbf{c} = (2 \ -1 \ 4)^T$ , o vetor de recursos por  $\mathbf{b} = (3 \ 4)^T$  e o vetor de variáveis por  $\mathbf{x} = (x_1 \ x_2 \ x_3)^T$ . ■

**Definição 2.3** (*solução ótima*) Uma solução factível que fornece o menor valor à função objetivo  $f$  é chamada *solução ótima*, denotada por  $(x_1^*, x_2^*, \dots, x_n^*)$ . Uma solução factível é ótima se:

$$f(x_1^*, x_2^*, \dots, x_n^*) \leq f(x_1, x_2, \dots, x_n),$$

para qualquer solução factível  $(x_1, x_2, \dots, x_n)$ .

**Exemplo 2.13** Examine o problema do Exemplo 2.12 juntamente com as soluções factíveis indicadas. Podemos observar que a solução factível  $x_1 = 0,25, x_2 = 0,5, x_3 = 1,75$  é melhor que a solução factível  $x_1 = 1, x_2 = 0, x_3 = 2$ . Será que existe alguma solução melhor? Considere a seguinte solução factível  $x_1 = 0, x_2 = \frac{2}{3}, x_3 = \frac{5}{3}$  (verifique que esta solução é factível). O valor da função objetivo para esta solução é  $f(0, \frac{2}{3}, \frac{5}{3}) = 6$ . Isso mostra que esta solução é melhor do que as anteriores. Mas ainda resta a questão: há outra solução melhor que esta? Esta solução é de fato ótima, como pode ser verificado adiante com a teoria deste capítulo. ■

Resolver um problema de otimização consiste em determinar uma solução ótima, isto é, determinar uma solução que satisfaça todas as restrições do problema e que atribua o melhor valor à função objetivo.

#### 2.4.2 Transformação de problemas na forma padrão

Várias outras formas de problemas de otimização linear aparecem em problemas reais, por exemplo, quando se deseja maximizar a função objetivo, ou existem restrições de desigualdade, ou ainda, quando algumas ou todas as variáveis não têm a condição de não-negatividade. A seguir, veremos como problemas de otimização linear que não estejam na forma padrão podem ser reescritos nesta forma.

##### *Problemas de maximização*

Encontrar uma solução ótima que maximize a função objetivo, corresponde a encontrar uma solução factível  $\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_n^*)$  tal que

$$f(\mathbf{x}^*) \geq f(\mathbf{x}), \text{ para toda solução } \mathbf{x} \text{ factível.}$$

Multiplicando-se essa desigualdade por  $-1$ , tem-se, de forma equivalente

$$-f(\mathbf{x}^*) \leq -f(\mathbf{x}), \text{ para toda solução } \mathbf{x} \text{ factível,}$$

ou seja, encontrar uma solução factível  $\mathbf{x}^*$  que maximize  $f(\mathbf{x})$  é equivalente a encontrar uma solução factível  $\mathbf{x}^*$  que minimize  $-f(\mathbf{x})$ , conforme a definição 2.3. Portanto, caso o problema seja de maximizar  $f(\mathbf{x})$ , podemos considerar, em seu lugar, o problema equivalente de minimizar  $-f(\mathbf{x})$ .

**Exemplo 2.14** O seguinte problema de otimização linear:

$$\begin{aligned} & \text{Maximizar } f(x_1, x_2, x_3) = 2x_1 - x_2 + 4x_3 \\ & x_1 + 2x_2 + x_3 = 3 \\ & x_2 + 2x_3 = 4 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, \end{aligned}$$

é equivalente ao problema na forma padrão:

$$\begin{aligned} & \text{Minimizar } -f(x_1, x_2, x_3) = -2x_1 + x_2 - 4x_3 \\ & x_1 + 2x_2 + x_3 = 3 \\ & x_2 + 2x_3 = 4 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{aligned}$$

■

### Restrições de desigualdade

As restrições do problema poderiam ser, originalmente, dadas por inequações, em vez de equações, como em (2.21). Nesse caso, convertemos o problema na forma padrão com o auxílio de novas variáveis. Suponha que a restrição  $i$  seja dada por

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i \quad (\text{por exemplo, } 3x_1 + 4x_2 - x_3 \leq 7).$$

A quantidade que falta para a igualdade é dada pela diferença entre o lado direito e o esquerdo da desigualdade e é desconhecida, isto é, uma incógnita:

$$x_k = b_i - (a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n) \geq 0.$$

Portanto, podemos escrever uma igualdade, com a inclusão de uma nova variável não-negativa:

$$\begin{aligned} a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n + x_k &= b_i \quad (\text{por exemplo, } 3x_1 + 4x_2 - x_3 + x_4 = 7, x_4 \geq 0) \\ x_k &\geq 0. \end{aligned}$$

Em resumo, basta somar uma variável não-negativa no lado esquerdo, que representa a folga existente na inequação original. Analogamente, se a restrição for da forma

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i \quad (\text{por exemplo, } 3x_1 + 4x_2 - x_3 \geq 7),$$

basta subtrair uma variável  $x_k$  no lado esquerdo da inequação para transformá-la em igualdade, sendo  $x_k \geq 0$ . A desigualdade  $\geq$  é, então, escrita como:

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n - x_k = b_i \quad (\text{por exemplo, } 3x_1 + 4x_2 - x_3 - x_4 = 7, x_4 \geq 0).$$

Essas variáveis adicionais, chamadas de *variáveis de folga*, são muito úteis, pois deixam todas as restrições em forma de igualdade, mantendo-se as condições de não-negatividade (no caso de desigualdade do tipo  $\geq$ , a variável introduzida também é chamada de *variável de excesso*).

**Exemplo 2.15** Escreva o seguinte problema de otimização linear na forma padrão.

$$\begin{aligned} \text{Minimizar } f(x_1, x_2, x_3) &= 2x_1 - 3x_2 + 3x_3 \\ x_1 + 2x_2 - x_3 &\geq 3 \\ -2x_1 + x_2 + x_3 &\leq -1 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{aligned}$$

Introduzindo as variáveis de folga, temos:

$$\begin{aligned} \text{Minimizar } f(x_1, x_2, x_3, x_4, x_5) &= 2x_1 - 3x_2 + 3x_3 + 0x_4 + 0x_5 \\ x_1 + 2x_2 - x_3 - x_4 &= 3 \\ -2x_1 + x_2 + x_3 + x_5 &= -1 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0. \end{aligned}$$

### Variáveis livres

No caso de existir alguma variável  $x_i$  irrestrita de sinal no problema, chamada *variável livre*, podemos substituir esta variável por outras duas e obter um problema equivalente na forma padrão. Observe que qualquer número (seja positivo, negativo ou nulo) pode ser sempre escrito como uma diferença de dois outros não-negativos, isto é, podemos escrever uma variável livre  $x_i$  como:

$$x_i = x_i^+ - x_i^-, \text{ com } x_i^+ \geq 0, x_i^- \geq 0.$$

Substituindo essa redefinição da variável livre, o problema resultante tem todas as variáveis não-negativas, conforme desejado.

**Exemplo 2.16** Reescreva o seguinte problema de otimização linear na forma padrão.

$$\begin{aligned} \text{Minimizar } f(x_1, x_2, x_3, x_4, x_5) &= 2x_1 - 3x_2 + 3x_3 \\ x_1 + 2x_2 - x_3 - x_4 &= 3 \\ -2x_1 + x_2 + x_3 + x_5 &= -1 \\ x_1 \text{ livre}, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0. \end{aligned}$$

Como a variável  $x_1$  é irrestrita em sinal, podemos escrevê-la como a diferença de duas variáveis não-negativas  $x_1 = x_1^+ - x_1^-$ ,  $x_1^+ \geq 0$ ,  $x_1^- \geq 0$ . Substituindo no problema, temos:

$$\begin{aligned} \text{Minimizar } f(x_1^+, x_1^-, x_2, x_3, x_4, x_5) &= 2x_1^+ - 2x_1^- - 3x_2 + 3x_3 \\ x_1^+ - x_1^- + 2x_2 - x_3 - x_4 &= 3 \\ -2x_1^+ + 2x_1^- + x_2 + x_3 + x_5 &= -1 \\ x_1^+ \geq 0, x_1^- \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0. \end{aligned}$$

■

Nas Seções 2.2.8-2.2.10 são apresentadas situações típicas em que variáveis livres ocorrem.

#### *Expressões alternativas na forma padrão*

O problema (2.20)–(2.22) pode ser escrito de maneira equivalente, usando-se expressões mais compactas, como a forma matricial apresentada anteriormente. Outras formas ainda podem ser utilizadas, tais como:

$$\begin{aligned} \text{Minimizar } f(x_1, x_2, \dots, x_n) &= \sum_{j=1}^n \mathbf{c}_j x_j \\ \sum_{j=1}^n \mathbf{a}_j x_j &= \mathbf{b} \\ x_j \geq 0, j &= 1, \dots, n, \end{aligned}$$

em que:

$$\mathbf{a}_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix} : j\text{-ésima coluna da matriz } \mathbf{A}, \text{ que são os coeficientes que multiplicam } x_j$$

**Exemplo 2.17** Observe o sistema linear no Exemplo 2.12. Podemos reescrevê-lo explicitando suas colunas:

$$\begin{cases} x_1 + 2x_2 + x_3 = 3 \\ x_2 + 2x_3 = 4 \end{cases} \Leftrightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} x_1 + \begin{pmatrix} 2 \\ 1 \end{pmatrix} x_2 + \begin{pmatrix} 1 \\ 2 \end{pmatrix} x_3 = \begin{pmatrix} 3 \\ 4 \end{pmatrix}.$$

■

## 2.5 RESOLUÇÃO GRÁFICA

A visualização de soluções de um problema matemático, quando possível e mesmo que limitada a um desenho esquematizado no plano, pode ser bastante útil para melhorar nossa intuição sobre um problema em estudo. Em particular, a representação gráfica de problemas de otimização linear permite-nos intuir várias propriedades teóricas e delinear um método de solução, que está formalmente desenvolvido nas seções seguintes. Como vimos na seção 2.4 (conceitos básicos), resolver um problema de otimização linear consiste em encontrar uma solução ótima, isto é, para um problema de minimização, consiste em determinar uma solução factível  $\mathbf{x}^*$  tal que  $f(\mathbf{x}^*) \leq f(\mathbf{x})$ ,

para todo  $\mathbf{x}$  factível. Consideramos exemplos de problemas com apenas duas variáveis, de modo a ilustrar a representação das soluções factíveis e da solução ótima em um plano cartesiano. Desenhemos, inicialmente, a região factível, isto é, o espaço de todas as soluções factíveis e, então, identificamos qual delas fornece o menor valor à função objetivo. Por conveniência, consideramos o problema de otimização linear na forma de desigualdades.

**Exemplo 2.18** Considere o problema de otimização linear

$$\begin{aligned} \text{Maximizar } f(x_1, x_2) &= x_1 + 2x_2 \\ x_1 + x_2 &\leq 4 \\ x_1 &\leq 2 \\ x_2 &\leq 3 \\ x_1 \geq 0, x_2 &\geq 0. \end{aligned}$$

Denominamos a região factível por:

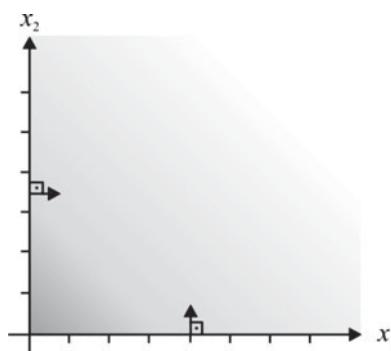
$$\mathbf{S} = \{(x_1, x_2) \text{ tal que } x_1 + x_2 \leq 4, x_1 \leq 2, x_2 \leq 3, x_1 \geq 0, x_2 \geq 0\}.$$

#### Desenhando a região factível ( $\mathbf{S}$ )

Devemos ter em mente que as soluções factíveis devem satisfazer todas as restrições. Representamos inicialmente os pontos no plano  $(x_1, x_2)$  que satisfazem as condições de não-negatividade,  $x_1 \geq 0, x_2 \geq 0$ , isto é, o primeiro quadrante do plano (ver Figura 2.14).

Para representar os pontos no plano  $(x_1, x_2)$  que também satisfazem a restrição  $x_1 + x_2 \leq 4$ , podemos proceder do seguinte modo:

- Inicialmente, identificamos os pontos que satisfazem a igualdade  $x_1 + x_2 = 4$ . Esta equação define uma reta no plano (uma reta é bem definida por 2 pontos, por exemplo,  $x_1 = 0, x_2 = 4$  e  $x_2 = 0, x_1 = 4$ ). Observe que os coeficientes da reta, o vetor  $(1, 1)^T$  é perpendicular à reta (verifique isso graficamente na Figura 2.15);
- Identificamos, a seguir, os pontos que satisfazem  $x_1 + x_2 < 4$ . Para identificar este conjunto, observamos que o vetor  $(1, 1)^T$  aponta no sentido em que a função  $x_1 + x_2$  cresce (mostramos essa afirmação a seguir). Portanto, os pontos no plano a partir da reta, do “lado” para o qual o vetor  $(1, 1)^T$  aponta, são tais que  $x_1 + x_2 > 4$ , e os pontos do lado oposto àque-



**Figura 2.14** Região definida por  $x_1 \geq 0$  e  $x_2 \geq 0$ .

le para o qual o vetor  $(1, 1)^T$  aponta são tais que  $x_1 + x_2 < 4$ . Estes últimos são os pontos que nos interessam. Uma outra maneira simples de identificar qual é o lado da reta que nos interessa consiste em verificar a factibilidade de um ponto qualquer fora da reta. Neste exemplo, a origem  $(x_1, x_2) = (0, 0)$  é um ponto bem conveniente para verificar sua factibilidade. Notamos que  $0 + 0 < 4$ , portanto, a origem  $(0, 0)$  é uma solução factível, bem como todos os pontos que estão do mesmo lado da origem.

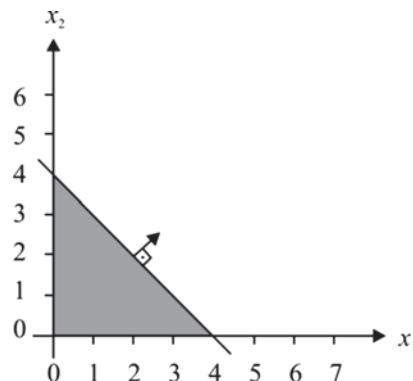
A reunião dos pontos tais que  $x_1 + x_2 = 4$  e  $x_1 + x_2 < 4$  é o que queremos considerar. A Figura 2.15 ilustra esses pontos, já considerando as condições de não-negatividade (primeiro quadrante).

De modo semelhante, desenhamos as regiões de pontos que satisfazem as restrições  $x_1 \leq 2$  e  $x_2 \leq 3$ , e as restrições de não-negatividade (Figuras 2.16 e 2.17).

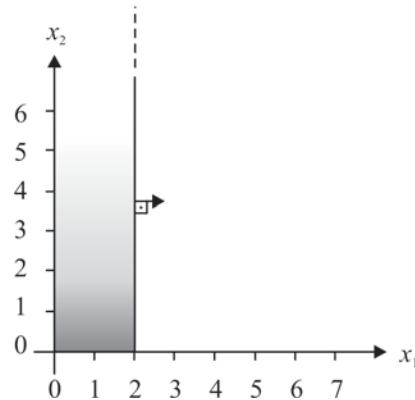
A intersecção de todas as regiões representadas nas Figuras 2.14 a 2.17 define a região factível, ilustrada na Figura 2.18.

#### Determinando a solução ótima ( $\mathbf{x}^*$ )

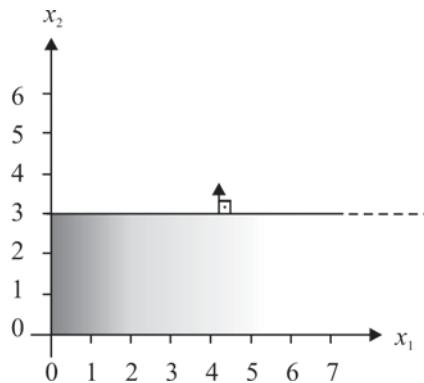
A função objetivo  $f(x_1, x_2) = x_1 + 2x_2$ , definida no conjunto S, pode assumir infinitos valores. Por exemplo, na solução factível  $\mathbf{x}' = (x'_1 \ x'_2)^T = (0 \ 0)^T$ , a função objetivo vale  $f' = f(\mathbf{x}') = 0$  e to-



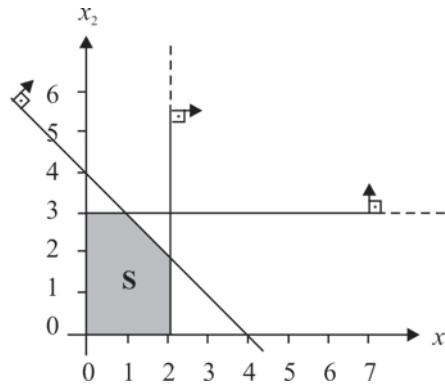
**Figura 2.15** Região definida por  $x_1 + x_2 \leq 4$ ,  $x_1 \geq 0$ ,  $x_2 \geq 0$ .



**Figura 2.16** Região definida por  $x_1 \leq 2$ .



**Figura 2.17** Região definida por  $x_2 \leq 3$ .

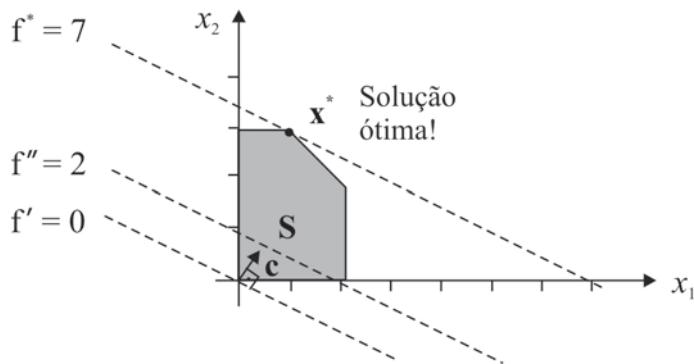


**Figura 2.18** Região factível  $S$ .

dos os pontos do plano  $(x_1, x_2)$  que atribuem este mesmo valor à função objetivo estão na reta  $x_1 + 2x_2 = 0$  (o conjunto de pontos que atribui o mesmo valor à função objetivo é chamado *curva de nível*, neste caso,  $f(x_1, x_2) = f'$ ). Esta curva de nível é representada na Figura 2.19 pela reta tracejada  $f' = 0$ . Como já observamos, ao definir a região factível, o vetor dos coeficientes  $(1 \ 2)^T$  (este vetor é chamado *gradiente de f* e denotado por  $\nabla f(x_1, x_2)$ ) é perpendicular à reta  $x_1 + 2x_2 = 0$  (uma curva de nível de  $f$ ) e aponta no sentido em que a função  $f$  cresce. Com isso, podemos visualizar que qualquer ponto de  $S$  atribui valor maior que zero à função  $f$ . Como queremos maximizar  $f$ , podemos concluir, graficamente, que a solução factível  $\mathbf{x}' = (0 \ 0)^T$  não é uma solução ótima.

Analisemos outra solução factível  $\mathbf{x}'' = (x'_1 \ x'_2)^T = (2 \ 0)^T$ , em que a função objetivo vale  $f'' = f(\mathbf{x}'') = 2$  e a curva de nível  $x_1 + 2x_2 = 2$  está representada, na Figura 2.19, por  $f'' = 2$ . Como o gradiente não se altera, essa reta é paralela à anterior,  $f' = 0$ . Novamente, o gradiente aponta no sentido em que  $f$  cresce e podemos identificar no gráfico outros pontos de  $S$  que atribuem valores maiores que 2 à função objetivo. Logo,  $\mathbf{x}''$  não é uma solução ótima.

Continuando com este procedimento (de identificar pontos que atribuem valores maiores à função objetivo), chegamos a um extremo  $\mathbf{x}^* = (x^*_1 \ x^*_2)^T = (1 \ 3)^T$ , para o qual  $f(\mathbf{x}^*) = 7$ . A curva de nível  $x_1 + 2x_2 = 7$  nos permite agora observar que todos os pontos de  $S$  atribuem valores



**Figura 2.19** Determinando a solução ótima  $\mathbf{x}^*$  (problema de maximização).

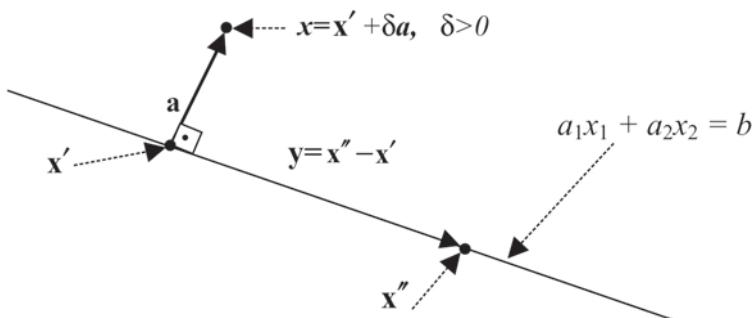
menores que 7 à função objetivo, uma vez que o gradiente aponta no sentido de crescimento de  $f$ . De outra forma, para todo  $\mathbf{x} \in \mathbf{S}$ ,  $f(\mathbf{x}) \leq 7 = f(\mathbf{x}^*)$ , o que significa que  $\mathbf{x}^*$  é uma solução ótima.

Portanto, a solução  $\mathbf{x}^*$  que satisfaz todas as restrições simultaneamente e maximiza  $f(\mathbf{x})$  existe e é única:

$$\mathbf{x}^* = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}.$$

Refazemos a seguir os passos do Exemplo 2.18 com um pouco mais de generalidade. Antes disso, mostramos as afirmações feitas na apresentação desse exemplo.

Considere uma reta qualquer,  $a_1x_1 + a_2x_2 = b$  (por exemplo, as retas  $x_1 + x_2 = 4$  ou  $x_1 + 2x_2 = 0$  que surgiram no Exemplo 2.18). afirmamos que o vetor  $\mathbf{a}^\top = (a_1 \ a_2)$  é perpendicular a essa reta. Para provar esta afirmação, considere dois pontos sobre a reta, digamos,  $\mathbf{x}' = (x'_1, x'_2)$  e  $\mathbf{x}'' = (x''_1, x''_2)$ , ou seja,  $a_1x'_1 + a_2x'_2 = b$  e  $a_1x''_1 + a_2x''_2 = b$ . O vetor  $\mathbf{y} = \mathbf{x}'' - \mathbf{x}'$  define a direção da reta (veja a Figura 2.20).



**Figura 2.20** O vetor  $\mathbf{a}$  perpendicular à direção da reta  $\mathbf{y}$ .

Provamos a seguir que o vetor  $\mathbf{a}$  é perpendicular ao vetor  $\mathbf{y}$ , ou seja,  $\mathbf{a}^T \mathbf{y} = 0$  (o produto escalar entre vetores perpendiculares é nulo). De fato,  $\mathbf{a}^T \mathbf{y}$  é igual a:

$$(a_1 \ a_2) \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = a_1(x''_1 - x'_1) + a_2(x''_2 - x'_2) = (a_1 x''_1 + a_2 x''_2) - (a_1 x'_1 + a_2 x'_2) = b - b = 0.$$

Essa prova pode ser simplificada com a notação matricial da reta  $\mathbf{a}^T \mathbf{x} = b$ , com  $\mathbf{a} = (a_1, a_2)$  e  $\mathbf{x} = (x_1, x_2)$ . Considerando  $\mathbf{x}'$  e  $\mathbf{x}''$  dois pontos da reta, isto é,  $\mathbf{a}^T \mathbf{x}' = b$ ,  $\mathbf{a}^T \mathbf{x}'' = b$  e  $\mathbf{y} = \mathbf{x}'' - \mathbf{x}'$ , então,  $\mathbf{a}^T \mathbf{y} = \mathbf{a}^T(\mathbf{x}'' - \mathbf{x}') = \mathbf{a}^T \mathbf{x}'' - \mathbf{a}^T \mathbf{x}' = b - b = 0$ . Embora tenhamos considerado os vetores no  $\mathbf{R}^2$ , as operações feitas são válidas para qualquer dimensão. A equação  $\mathbf{a}^T \mathbf{x} = b$  no  $\mathbf{R}^3$  é a equação de um plano, ou seja, o vetor  $\mathbf{a}$  é perpendicular ao plano, pois  $\mathbf{x}'$  e  $\mathbf{x}''$  são arbitrários. Em geral, quando o espaço é o  $\mathbf{R}^n$ , a equação  $\mathbf{a}^T \mathbf{x} = b$  define um conjunto chamado *hiperplano* e o vetor  $\mathbf{a}$  é perpendicular ao hiperplano.

Outra afirmação que fizemos no Exemplo 2.18 foi de que o vetor  $\mathbf{a}$  aponta para o lado do plano cujos pontos satisfazem  $\mathbf{a}^T \mathbf{x} > b$ . De fato, um ponto no lado indicado pelo vetor  $\mathbf{a}$  se escreve como um ponto da reta, digamos,  $\mathbf{x}'$ , adicionado a um múltiplo positivo do vetor  $\mathbf{a}$ . De outra forma, os pontos do lado que indica o vetor  $\mathbf{a}$  são dados por  $\mathbf{x} = \mathbf{x}' + \delta \mathbf{a}$ ,  $\delta > 0$  (Figura 2.20). Conseqüentemente,

$$\mathbf{a}^T \mathbf{x} = \mathbf{a}^T(\mathbf{x}' + \delta \mathbf{a}) = \mathbf{a}^T \mathbf{x}' + \delta \mathbf{a}^T \mathbf{a} > b,$$

pois  $\delta > 0$  e  $\mathbf{a}^T \mathbf{a} = a_1^2 + a_2^2 + \dots + a_n^2 > 0$  (Consideramos  $\mathbf{a} \neq \mathbf{0}$ ; o caso  $\mathbf{a} = \mathbf{0}$  corresponde à equação  $0x_1 + 0x_2 = b$ , que não tem sentido para nossos estudos, pois  $b = 0$  leva a uma equação trivial  $0 = 0$ , e  $b \neq 0$  leva a uma equação impossível). Portanto, quando queremos identificar os pontos que satisfazem  $\mathbf{a}^T \mathbf{x} \leq b$ , traçamos a reta  $\mathbf{a}^T \mathbf{x} = b$ , desenhamos o vetor  $\mathbf{a}$  (perpendicular à reta) e consideramos os pontos no lado oposto ao indicado pelo vetor  $\mathbf{a}$  (como identificar os pontos que satisfazem  $\mathbf{a}^T \mathbf{x} \geq b$ ?). O mesmo raciocínio foi empregado para determinar a solução ótima. Escolhemos uma solução factível qualquer, digamos,  $\mathbf{x}'$ , e seja  $f' = f(\mathbf{x}') = \mathbf{c}^T \mathbf{x}'$  o valor da função objetivo em  $\mathbf{x}'$ . Ao traçarmos a reta  $f(\mathbf{x}) = f'$  (isto é, uma curva de nível da função  $f$ , que neste caso é uma reta  $c_1 x_1 + c_2 x_2 = f'$ ), desenhamos o gradiente de  $f$ , o vetor  $\mathbf{c}$ , que aponta para pontos tais que  $f(\mathbf{x}) > f'$ .

Se desejarmos maximizar  $f(\mathbf{x})$  (como no Exemplo 2.18), procuramos pontos factíveis que estejam do lado apontado pelo gradiente  $\mathbf{c}$ , partindo-se da curva de nível  $f(\mathbf{x}) = f'$ . Uma situação limite ocorre (em certos casos, pode também não ocorrer, como veremos adiante) quando todos os pontos do lado do gradiente  $\mathbf{c}$  são infactíveis, ou seja, todos os pontos factíveis estão do lado oposto do gradiente e, portanto, atribuem valores menores à função  $f(\mathbf{x})$ . Isso é o que caracteriza uma solução ótima. Se o objetivo for minimizar  $f(\mathbf{x})$ , aplicamos o mesmo procedimento, porém buscando pontos no sentido contrário ao do vetor gradiente. Formalizamos a seguir os passos do Exemplo 2.18 de forma genérica. Considere o problema de otimização linear (minimização):

$$\begin{aligned} \text{Minimizar } f(x_1, x_2) &= c_1 x_1 + c_2 x_2 \\ a_{11} x_1 + a_{12} x_2 &\leq b_1 \\ a_{21} x_1 + a_{22} x_2 &\leq b_2 \\ &\vdots \\ a_{i1} x_1 + a_{i2} x_2 &\leq b_i \\ &\vdots \\ a_{m1} x_1 + a_{m2} x_2 &\leq b_m \\ x_1 &\geq 0, x_2 \geq 0, \end{aligned}$$

ou, em notação matricial,

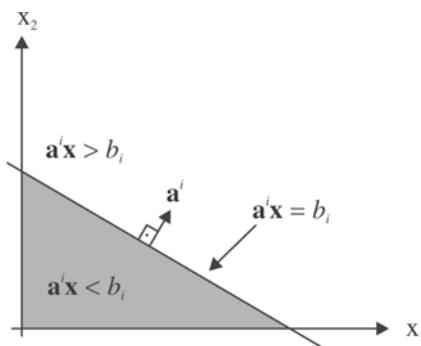
$$\begin{aligned} \text{Minimizar } f(\mathbf{x}) &= \mathbf{c}^T \mathbf{x} \\ \mathbf{Ax} &\leq \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \quad \text{com } \mathbf{x} \in \mathbb{R}^2 \end{aligned}$$

Denotamos a região factível por  $\mathbf{S}$ , isto é,  $\mathbf{S} = \{\mathbf{x} \in \mathbb{R}^2 \text{ tal que } \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ . Para desenhar o conjunto de pontos tais que  $a_{i1}x_1 + a_{i2}x_2 \leq b_i$ , traçamos, inicialmente, a reta  $a_{i1}x_1 + a_{i2}x_2 = b_i$ , que divide o espaço  $\mathbb{R}^2$  em 3 partes (em notação matricial, a restrição é  $(\mathbf{a}^i)^T \mathbf{x} \leq b_i$ , em que  $(\mathbf{a}^i)^T = (a_{i1} \ a_{i2})$  é a linha  $i$  da matriz  $\mathbf{A}$  e a equação da reta é  $(\mathbf{a}^i)^T \mathbf{x} = b_i$ ):

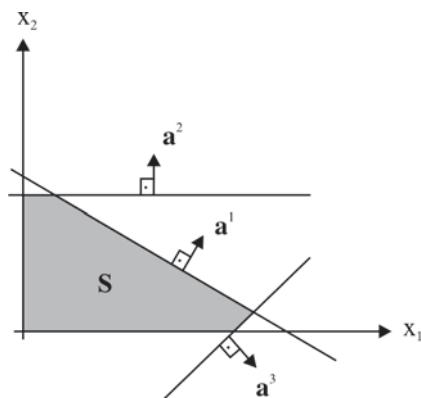
- $\mathbf{x} \in \mathbb{R}^2$  tal que  $(\mathbf{a}^i)^T \mathbf{x} = b_i$
- $\mathbf{x} \in \mathbb{R}^2$  tal que  $(\mathbf{a}^i)^T \mathbf{x} < b_i$
- $\mathbf{x} \in \mathbb{R}^2$  tal que  $(\mathbf{a}^i)^T \mathbf{x} > b_i$ .

O gradiente  $\mathbf{a}^i$  (coeficientes da equação da reta), que é perpendicular à reta  $(\mathbf{a}^i)^T \mathbf{x} = b_i$ , aponta para pontos tais que  $(\mathbf{a}^i)^T \mathbf{x} > b_i$  (veja Figura 2.21).

A região factível, que é a intersecção de todas as regiões do tipo  $\mathbf{a}^i \mathbf{x} \leq b_i$  com o primeiro quadrante (não-negatividade das variáveis), pode ser facilmente desenhada (Figura 2.22).



**Figura 2.21** As três regiões do plano: a)  $\mathbf{a}^i \mathbf{x} = b_i$     b)  $\mathbf{a}^i \mathbf{x} < b_i$     c)  $\mathbf{a}^i \mathbf{x} > b_i$ .



**Figura 2.22** Região factível de restrições do tipo  $\mathbf{a}^i \mathbf{x} \leq b_i$ ,  $i = 1, 2, \dots, m$ .

Finalmente, basta encontrar na região factível a solução  $\mathbf{x}^*$  que minimize a função objetivo  $f(\mathbf{x})$ , isto é, a solução ótima. Observando a curva de nível  $\mathbf{c}^T \mathbf{x} = f^*$  da função  $f$  em que  $f^* = f(\mathbf{x}^*)$ , observamos que todo o conjunto  $\mathbf{S}$  está do lado para onde o gradiente aponta, ou seja, para todo  $\mathbf{x} \in \mathbf{S}$ , temos  $f(\mathbf{x}) \geq f(\mathbf{x}^*)$ , que é a definição de solução ótima para um problema de minimização. A Figura 2.23 ilustra várias curvas de nível da função  $f$ .

A solução ótima  $\mathbf{x}^*$  na Figura 2.23 é uma solução factível muito especial chamada *vértice* ou *ponto extremo* (veja também a solução  $\mathbf{x}^*$  no Exemplo 2.18). Nas figuras que ilustram a região factível, é possível notar que os vértices são determinados pela intersecção de (pelo menos) duas retas que definem a fronteira da região factível (observe que  $x_i = 0$  é uma equação de reta). Desse forma, podemos intuir que os vértices são soluções de sistemas de equações lineares. Formalizaremos tais sistemas lineares mais adiante.

O leitor deve observar que, se o gradiente da função objetivo for modificado, outro vértice pode ser uma solução ótima (refaça o Exemplo 2.18, considerando a função objetivo  $f(x_1, x_2) = x_1 - 2x_2$ ), de modo a sugerir a seguinte propriedade:

*“Se um problema de otimização linear tem uma solução ótima, então existe um vértice ótimo.”*

Na próxima seção, retornamos a essa propriedade. O leitor deve ficar atento à seguinte afirmação equivocada: “*se uma solução for ótima, então ela é um vértice*” e, como exercício, pode apresentar um contra-exemplo.

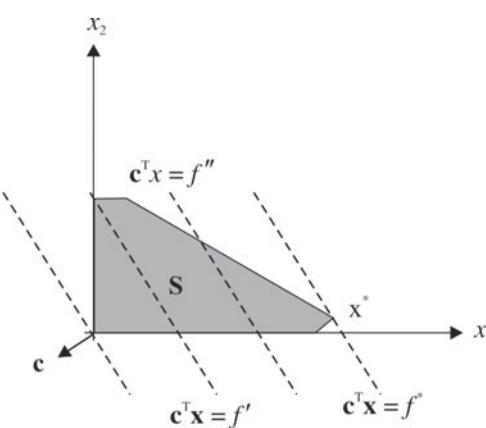
**Exemplo 2.19** Considere o problema de otimização linear:

$$\begin{aligned} &\text{Maximizar } f(x_1, x_2) = x_1 + x_2 \\ &-3x_1 + x_2 \leq 2 \\ &x_2 \leq 3 \\ &x_1 + 2x_2 \leq 9 \\ &3x_1 + x_2 \leq 18 \\ &x_1 \geq 0, x_2 \geq 0. \end{aligned}$$

■

*Desenhando a região factível ( $\mathbf{S}$ ):*

A Figura 2.24 ilustra a região factível  $\mathbf{S}$  do Exemplo 2.19.



**Figura 2.23** Curvas de nível da função objetivo,  $f' > f'' > f^*$  (problema de minimização).

*Determinando a solução ótima:*

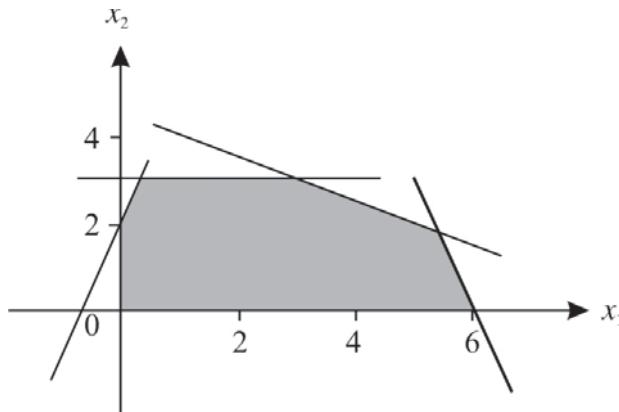
A Figura 2.25 indica a solução ótima  $\mathbf{x}^*$  do Exemplo 2.19.

Portanto, a solução  $\mathbf{x}^*$  que satisfaz todas as restrições simultaneamente e maximiza  $f(\mathbf{x})$  é única e dada pela solução do sistema:

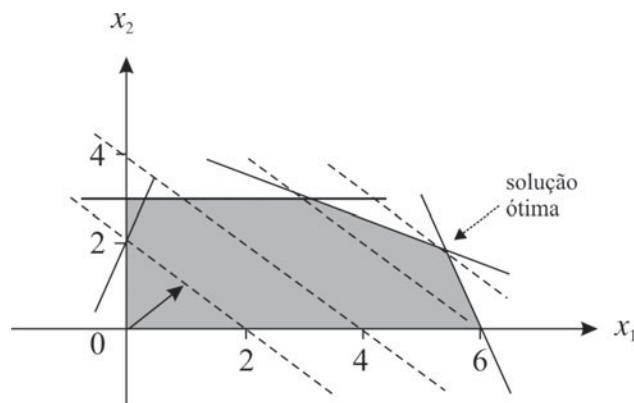
$$\begin{aligned}x_1 + 2x_2 &= 9 \\3x_1 + x_2 &= 18 \\ \mathbf{x}^* &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5,4 \\ 1,8 \end{bmatrix} \quad \text{e} \quad f(\mathbf{x}^*) = 7,2.\end{aligned}$$

O problema representado na Figura 2.25 tem a região factível limitada e apresenta uma única solução ótima. Entretanto, várias outras possibilidades podem ocorrer, o que pode ser facilmen-

---



**Figura 2.24** Gráfico do problema de otimização linear do Exemplo 2.19.



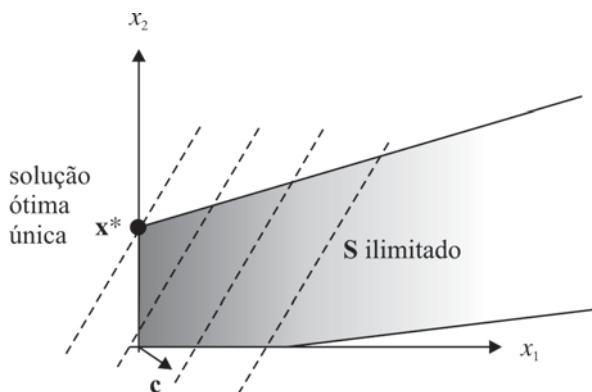
**Figura 2.25** Determinando a solução ótima (problema de maximização).

te concluído dos gráficos das Figuras 2.26 e 2.27. É verdade que, se a região factível é limitada, então existe uma solução ótima. Mas também podemos ter situações em que a região factível é ilimitada e ainda assim existir uma única solução ótima, como ilustrado na Figura 2.26.

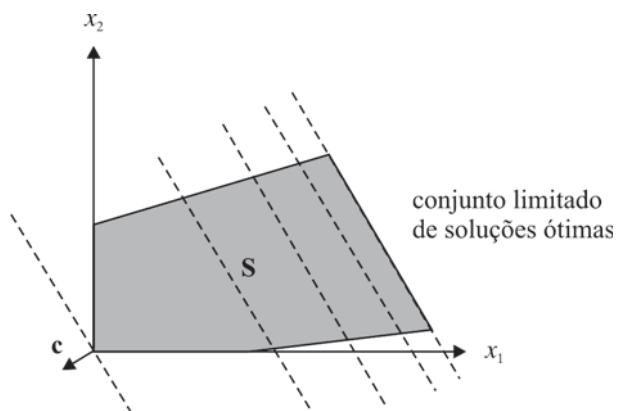
Embora os exemplos anteriores apresentem sempre uma solução ótima única, a Figura 2.27 ilustra um caso com infinitas soluções ótimas, todas as soluções sobre o segmento de reta realçado.

**Exemplo 2.20** Considere o problema de otimização linear:

$$\begin{aligned} \text{Maximizar } f(x_1, x_2) &= x_1 + 2x_2 \\ -3x_1 + x_2 &\leq 2 \\ x_2 &\leq 3 \\ x_1 + 2x_2 &\leq 9 \\ 3x_1 + x_2 &\leq 18 \\ x_1 \geq 0, x_2 \geq 0. \end{aligned}$$



**Figura 2.26** Região de factibilidade ilimitada e solução ótima única (minimização).

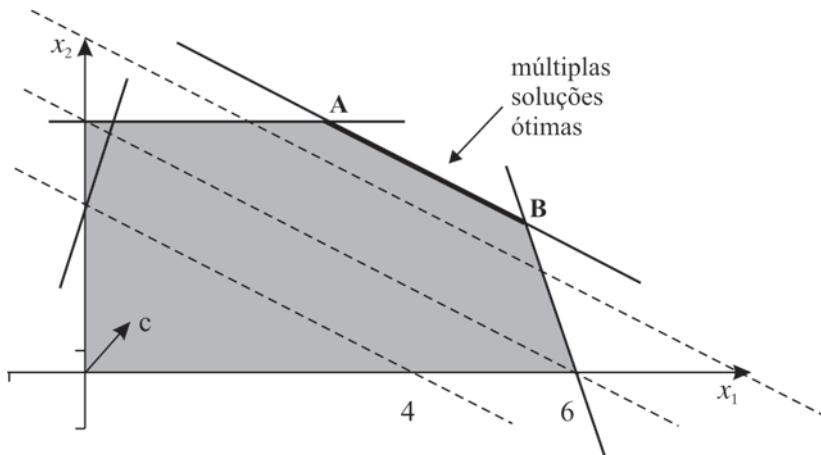


**Figura 2.27** Região factível limitada e infinitas soluções ótimas, conjunto limitado de soluções ótimas (minimização).

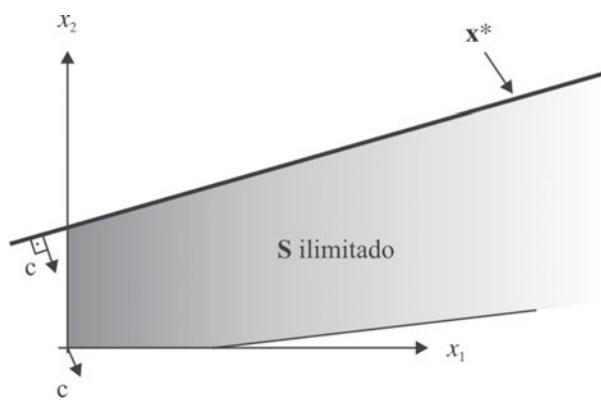
Procedendo como temos feito, identificamos a região factível e as soluções ótimas, determinadas pelo segmento de reta em destaque na Figura 2.28 que liga os pontos **A** e **B**. Note que o gradiente da função objetivo,  $\mathbf{c} = (1, 2)$ , é múltiplo do gradiente da reta  $x_1 + 2x_2 = 9$ . Assim, todos os pontos no segmento satisfazem  $f(x_1, x_2) = 9$  e são soluções ótimas, já que qualquer solução que satisfaz  $f(x_1, x_2) > 9$  é infactível. ■

Pode-se ter, também, um conjunto ilimitado de soluções ótimas, como a semi-reta indicada na Figura 2.29.

A Figura 2.30 ilustra mais uma possibilidade: um problema de otimização linear pode não ter solução ótima, apesar de ter soluções factíveis. Neste caso, dizemos que a solução ótima é ilimitada. Note, entretanto, que isso não quer dizer que sempre inexista uma solução ótima para um problema com região factível ilimitada, pois a solução ótima depende do gradiente da função objetivo (veja Figuras 2.26 e 2.29).



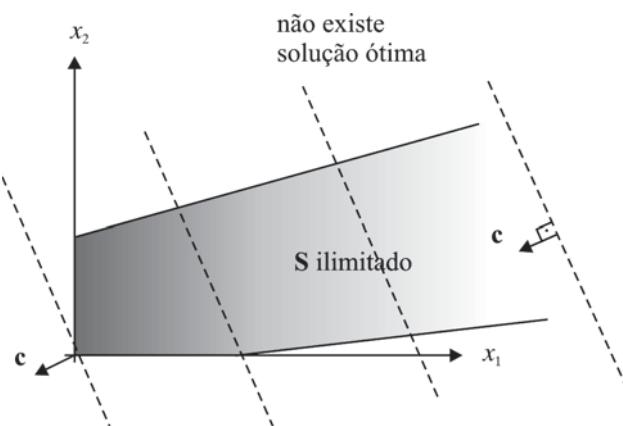
**Figura 2.28** Múltiplas soluções ótimas (maximização).



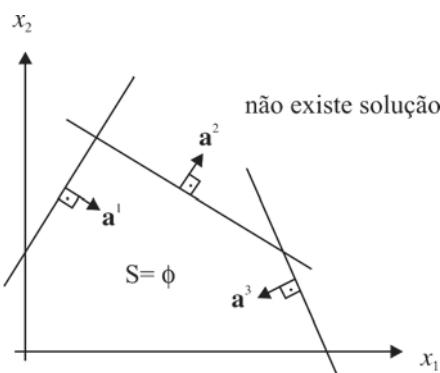
**Figura 2.29** Região factível ilimitada e infinitas soluções ótimas, conjunto ilimitado de soluções ótimas (minimização).

A inexistência de solução ótima também pode ocorrer pela inexistência de solução factível, isto é,  $S = \emptyset$ . Isso decorre do fato de as restrições serem conflitantes, como ilustra a Figura 2.31.

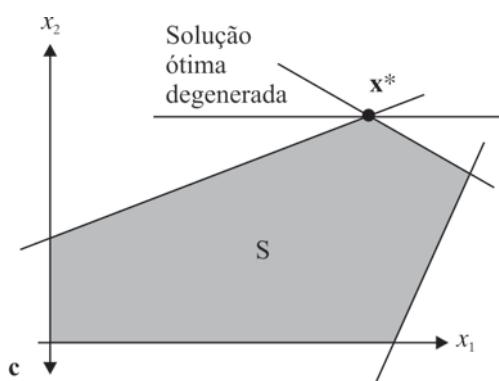
A Figura 2.32 mostra uma situação em que o mesmo vértice pode ser obtido como intersecção de retas diferentes. Essa situação produz um vértice que chamamos de *vértice degenerado*. Tais vértices, em geral, introduzem algumas dificuldades nos métodos de soluções, discutidas nas seções seguintes.



**Figura 2.30** Região factível ilimitada e não existe solução ótima (solução ótima ilimitada),  $c^T x \rightarrow -\infty$  (minimização).



**Figura 2.31** Não existe solução ótima:  $S = \emptyset$ .



**Figura 2.32** Solução ótima degenerada (minimização).

**Exemplo 2.21** Considere o seguinte problema de otimização linear:

$$\begin{aligned}
 & \text{Maximizar } f(x_1, x_2) = x_1 + 3x_2 \\
 & \quad x_2 \leq 4 \\
 & \quad x_1 + x_2 \leq 6 \\
 & \quad x_1 \leq 3 \\
 & \quad 5x_1 + x_2 \leq 18 \\
 & \quad x_1 \geq 0, \quad x_2 \leq 0.
 \end{aligned}$$

Este problema, como podemos observar na Figura 2.33, tem solução ótima não degenerada:

$$\mathbf{x}^* = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix} \quad \text{e} \quad f(\mathbf{x}^*) = 14$$

embora tenha uma solução factível degenerada:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix} \quad \text{e} \quad f(\mathbf{x}) = 12. \quad \blacksquare$$

Os casos observados na resolução gráfica de um problema de otimização linear com apenas duas variáveis são igualmente possíveis em problemas de dimensões maiores. Na próxima seção, apresentamos os subsídios teóricos necessários para tratar o caso geral. Por exemplo, já intuímos que uma solução ótima, se houver, pode ser pesquisada entre os vértices (os quais são obtidos por intersecções de retas ou pela solução de sistemas lineares). Assim, se formos capazes de sair de um vértice para outro melhor, podemos repetir isso um número finito de vezes até encontrar um vértice ótimo. A Figura 2.34 ilustra essa trajetória pela fronteira da região factível, partindo da origem até a solução ótima  $\mathbf{x}^*$ . É assim que trabalha o *método simplex*, um dos mais utilizados métodos para a resolução de problemas de otimização linear, discutido na próxima seção. Outra classe de métodos gera uma seqüência de *pontos interiores* que converge para a solução ótima  $\mathbf{x}_1, \mathbf{x}_2, \dots \rightarrow \mathbf{x}^*$  seguindo-se uma *trajetória central*, em vez de uma trajetória pela fronteira da região factível. Essa classe de métodos não é abordada neste livro, embora seja importante na resolução de problemas de otimização linear.<sup>17</sup> Outras trajetórias, seguindo soluções infactíveis, também podem ser construídas, como por exemplo, pelo método dual-simplex, apresentado na Seção 2.10.4.

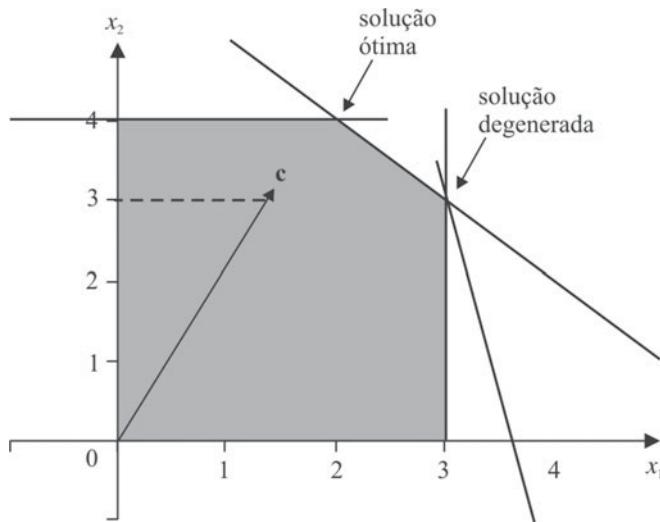
## 2.6 TEORIA BÁSICA E O MÉTODO SIMPLEX

Nesta seção, apresentamos algumas definições e propriedades fundamentais da otimização linear e um de seus métodos mais importantes: o *método simplex*. Por conveniência, a teoria é desenvolvida para o problema na forma padrão (Seção 2.4.2):

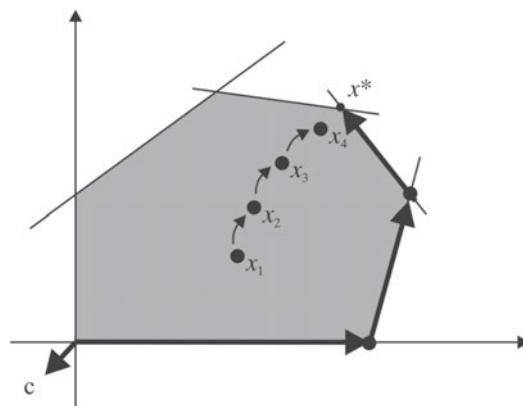
$$\begin{aligned}
 & \text{Minimizar } f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} \\
 & \mathbf{Ax} = \mathbf{b} \\
 & \mathbf{x} \geq \mathbf{0}.
 \end{aligned}$$

Embora, por conveniência, os exemplos ilustrados nesta seção tenham as restrições na forma de desigualdades (restrições de igualdade definem retas no  $\mathbb{R}^2$ ), os desenvolvimentos nas seções seguintes utilizam a forma padrão (problema de minimização e restrições de igualdade e não negatividade sobre as variáveis).

<sup>17</sup> Veja Adler *et al.* (1989), Gonzaga (1991) e Vanderbei (2001).



**Figura 2.33** Problema com solução degenerada e ótimo não degenerado.



**Figura 2.34** Trajetórias até a solução ótima (minimização).

### 2.6.1 Soluções básicas

A seguir formalizamos algumas propriedades já intuídas no capítulo anterior, quando resolvemos graficamente alguns problemas de otimização linear, e vértices da região factível foram obtidos pela resolução de sistemas de equações lineares. Iniciamos com um exemplo simples. Considere a região factível  $S \subseteq \mathbb{R}^2$ , definida pelas restrições a seguir e ilustrada na Figura 2.35:

$$\begin{aligned}
 & x_1 + x_2 \leq 6 \\
 & x_1 - x_2 \leq 4 \\
 & 3x_1 + x_2 \geq 3 \\
 & x_1 \geq 0, x_2 \geq 0.
 \end{aligned} \tag{2.23}$$

Com a definição das variáveis de folga (veja Seção 2.4.2):

$$\begin{aligned}x_3 &= 6 - (x_1 + x_2) \geq 0 \\x_4 &= 4 - (x_1 - x_2) \geq 0 \\x_5 &= (3x_1 + x_2) - 3 \geq 0,\end{aligned}\tag{2.24}$$

o problema é transformado na forma de igualdade ( $\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$ ):

$$\begin{aligned}x_1 + x_2 + x_3 &= 6 \\x_1 - x_2 + x_4 &= 4 \\3x_1 + x_2 - x_5 &= 3 \\x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0.\end{aligned}\tag{2.25}$$

O problema passa a ter cinco variáveis ( $x_1, x_2, x_3, x_4, x_5$ ), em vez de apenas as duas primeiras ( $x_1, x_2$ ). Porém, qualquer ponto em  $\mathbb{R}^2$  determina unicamente essas cinco variáveis conforme (2.25), ou seja, dado o par  $(x_1, x_2)$ , podemos determinar unicamente os valores de  $x_3, x_4$  e  $x_5$  por (2.24). Vejamos alguns exemplos numéricos representados na Figura 2.35 pelos pontos **A**, **B**, ..., **F**. Inicialmente, examinamos os pontos de **A** a **D**:

Ponto **A**:

$$\begin{aligned}x_1 &= 3 \\x_2 &= 2 \\x_3 &= 6 - (3 + 2) = 1 \\x_4 &= 4 - (3 - 2) = 3 \\x_5 &= (3 \times 3 + 2) - 3 = 8\end{aligned}$$

Ponto **B**:

$$\begin{aligned}x_1 &= 1 \\x_2 &= 3 \\x_3 &= 6 - (1 + 3) = 2 \\x_4 &= 4 - (1 - 3) = 6 \\x_5 &= (3 \times 1 + 2) - 3 = 3\end{aligned}$$

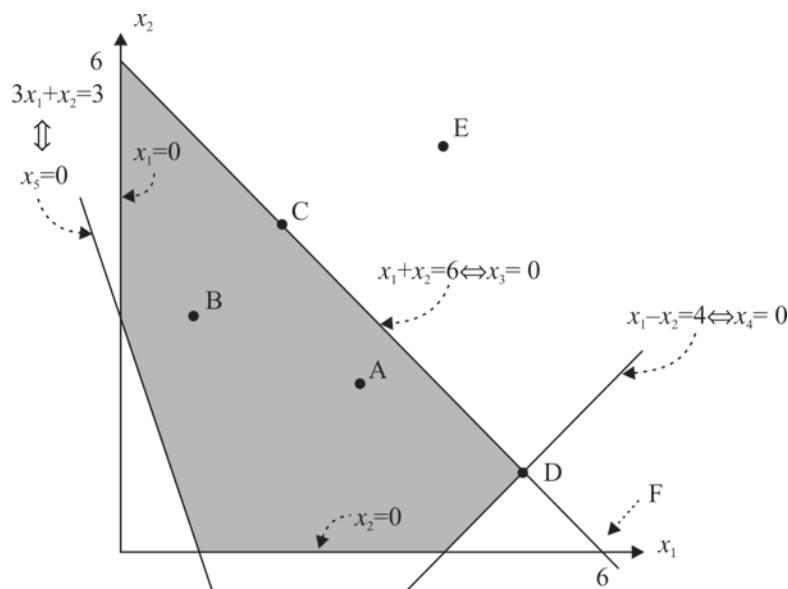
Ponto **C**:

$$\begin{aligned}x_1 &= 2 \\x_2 &= 4 \\x_3 &= 6 - (2 + 4) = 0 \\x_4 &= 4 - (2 - 4) = 6 \\x_5 &= (3 \times 2 + 2) - 3 = 7\end{aligned}$$

Ponto **D**:

$$\begin{aligned}x_1 &= 5 \\x_2 &= 1 \\x_3 &= 6 - (5 + 1) = 0 \\x_4 &= 4 - (5 - 1) = 0 \\x_5 &= (3 \times 5 + 1) - 3 = 13.\end{aligned}$$

Tais soluções são factíveis, pois o sistema  $\mathbf{Ax} = \mathbf{b}$  é satisfeito por construção (reveja como as variáveis  $x_3, x_4$  e  $x_5$  foram calculadas), além de satisfazerem as condições de não-negatividade. A factibilidade dessas soluções também pode ser visualizada na Figura 2.35, nos pontos pertencentes à região factível.



**Figura 2.35** Região factível e fronteira caracterizada por  $x_j = 0, j = 1, \dots, 5$ .

Lembre-se de que as soluções do sistema  $\mathbf{Ax} = \mathbf{b}$  têm 5 coordenadas  $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$ , porém, visualizamos na Figura 2.35 apenas as duas primeiras coordenadas  $(x_1, x_2)$ , enquanto as demais coordenadas  $(x_3, x_4, x_5)$  medem as folgas em cada restrição. Assim, os pontos **A** e **B** estão no interior da região factível, pois têm todas as coordenadas positivas, ou seja, todas as restrições estão folgadas. Por exemplo, no ponto **A**,  $x_1 + x_2 < 6$ , portanto,  $x_3 > 0$  (isto é, a restrição 1 é folgada no ponto **A**). Por outro lado, os pontos **C** e **D** estão sobre a fronteira de  $S$ , pois alguma variável se anula. Por exemplo, no ponto **C**,  $x_1 + x_2 = 6$ , portanto,  $x_3 = 0$  (a restrição 1 é ativa no ponto **C**, isto é, tem folga nula). Da mesma forma, todos os pontos sobre a reta  $x_1 + x_2 = 6$  satisfazem  $x_3 = 0$ . Em particular, no ponto **D** (o qual já chamamos de vértice da região factível), duas variáveis se anulam,  $x_3 = x_4 = 0$  (o leitor pode voltar à Figura 2.35 e determinar outras soluções no interior e na fronteira da região factível).

Examinamos agora os pontos **E** e **F**, que não pertencem à região factível (Figura 2.35). Verificamos como esta infactibilidade pode ser detectada sem o auxílio da figura (lembre-se que em problemas reais, que envolvem muitas variáveis e restrições, é, geralmente, muito difícil a representação gráfica do problema como temos feito).

Ponto **E**:

$$\begin{aligned}x_1 &= 4 \\x_2 &= 5 \\x_3 &= 6 - (4 + 5) = -3 \\x_4 &= 4 - (4 - 5) = 5 \\x_5 &= (3 \times 4 + 5) - 3 = 14\end{aligned}$$

Ponto **F**:

$$\begin{aligned}x_1 &= 6 \\x_2 &= 0 \\x_3 &= 6 - (6 + 0) = 0 \\x_4 &= 4 - (6 - 0) = -2 \\x_5 &= (3 \times 6 + 0) - 3 = 15\end{aligned}$$

Apesar de os pontos **E** e **F** satisfazerem o sistema  $\mathbf{Ax} = \mathbf{b}$  (reveja mais uma vez como foram determinadas as coordenadas), eles não são *soluções factíveis*, pois violam uma das condições de não-negatividade. Em particular, o ponto **E** satisfaz  $x_1 + x_2 > 6$ , portanto,  $x_3 < 0$ .

O leitor pode identificar na Figura 2.35 todos os pontos que satisfazem  $x_3 = 0$ ,  $x_3 > 0$  ou  $x_3 < 0$  (a reta  $x_1 + x_2 = 6$  divide o plano em três partes: pontos que satisfazem  $x_1 + x_2 = 6$ ,  $x_1 + x_2 > 6$  ou  $x_1 + x_2 < 6$ ). Além disso, a fronteira da região factível pode ser escrita como o lugar geométrico em que pelo menos uma variável se anula. No exemplo, como podemos ver na Figura 2.35, uma solução está na fronteira se e somente se  $x_j = 0$ , para algum  $j = 1, \dots, 5$ .

Na resolução gráfica de um problema de otimização linear (Seção 2.5), podemos intuir que, para encontrar uma solução ótima, basta procurar entre os vértices da região factível  $S$  (isto é, pontos como o **D** da Figura 2.35). Tais soluções em  $\mathbb{R}^2$  são determinadas pela intersecção de 2 retas, o que significa que 2 variáveis são nulas (veja a Figura 2.35). Por exemplo, o ponto **D** na Figura 2.35 é caracterizado por:

$$\text{Fixar } \begin{cases} x_3 = 0 \\ x_4 = 0 \end{cases} \text{ resulta no sistema: } \begin{cases} x_1 + x_2 = 6 \\ x_1 - x_2 = 4 \\ 3x_1 + x_2 - x_5 = 3 \end{cases}, \text{ cuja solução é: } \begin{cases} x_1 = 5 \\ x_2 = 1 \\ x_5 = 13. \end{cases} \quad (2.26)$$

Isso mostra uma maneira de se obter soluções para o sistema  $\mathbf{Ax} = \mathbf{b}$ . O sistema  $\mathbf{Ax} = \mathbf{b}$  em (2.25) tem  $m = 3$  equações e  $n = 5$  variáveis, portanto, tem 2 variáveis *independentes*, para as quais podemos atribuir quaisquer valores. Em particular, fixando-se  $x_3 = x_4 = 0$ , obtemos um sistema  $m \times m$  ( $3 \times 3$ ) nas variáveis restantes  $(x_1, x_2, x_5)$ , o qual deve ser resolvido por algum método de solução de sistemas lineares, como, por exemplo, o método de eliminação de Gauss (veja Apêndice A). Valores diferentes de zero podem ser atribuídos às variáveis  $x_3$  e  $x_4$ , obtendo diferentes soluções para o sistema, porém não são vértices (veja Exercício 2.20).

A solução obtida ao se fixar  $x_3 = 0$ ,  $x_4 = 0$  é dada por  $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5) = (5, 1, 0, 0, 13)$  e, como já havíamos observado anteriormente, é uma solução factível. Porém, ao fixarmos 2 variá-

veis quaisquer em zero, não há garantias de que obteremos uma solução factível. Veja, por exemplo, o ponto  $F$ , que pode ser obtido pela fixação das variáveis  $x_2 = 0$  e  $x_3 = 0$ , restando um sistema nas variáveis  $x_1$ ,  $x_4$  e  $x_5$ , ou seja,

$$\text{Fixar } \begin{cases} x_2 = 0 \\ x_3 = 0 \end{cases} \text{ resulta no sistema: } \begin{cases} x_1 &= 6 \\ x_1 + x_4 &= 4 \\ 3x_1 - x_5 &= 3 \end{cases}, \text{ cuja solução é: } \begin{cases} x_1 = 6 \\ x_4 = -2 \\ x_5 = 15 \end{cases}. \quad (2.27)$$

Deve ficar claro que, assim procedendo, o sistema  $\mathbf{Ax} = \mathbf{b}$  é satisfeito por construção, porém não necessariamente as condições de não-negatividade são verificadas. Este procedimento de obtenção de soluções (isto é, fixar variáveis *independentes* e resolver o sistema restante) pode ser estendido para quaisquer sistemas  $\mathbf{Ax} = \mathbf{b}$ ,  $m \times n$ . Por simplicidade de notação, rearranjamos o sistema, agrupando as  $n - m$  variáveis *independentes* que devem ser fixadas e as  $m$  variáveis restantes. Por exemplo, suponha que as variáveis  $x_3$  e  $x_4$  sejam escolhidas para serem fixadas (lembre-se que, se fixadas em zero, o ponto  $D$  é obtido), então, o sistema  $\mathbf{Ax} = \mathbf{b}$  pode ser reescrito equivalentemente por:

$$\mathbf{Ax} = \mathbf{b} \Leftrightarrow \underbrace{\begin{bmatrix} x_1 + x_2 + x_3 \\ x_1 - x_2 + x_4 \\ 3x_1 + x_2 - x_5 \end{bmatrix}}_{\mathbf{Ax}} = \begin{bmatrix} 6 \\ 4 \\ 3 \end{bmatrix} \Leftrightarrow \underbrace{\begin{bmatrix} x_1 + x_2 \\ x_1 - x_2 \\ 3x_1 + x_2 - x_5 \end{bmatrix}}_{\substack{\text{variáveis} \\ \text{restantes}}} + \underbrace{\begin{bmatrix} x_3 \\ x_4 \\ 0 \end{bmatrix}}_{\substack{\text{variáveis} \\ \text{a serem} \\ \text{fixadas}}} = \begin{bmatrix} 6 \\ 4 \\ 3 \end{bmatrix}, \quad (2.28)$$

ou, em notação matricial:

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 3 & 1 & -1 \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_5 \end{bmatrix}}_{\mathbf{x}_B} + \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\mathbf{N}} \underbrace{\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}}_{\mathbf{x}_N} = \begin{bmatrix} 6 \\ 4 \\ 3 \end{bmatrix}, \quad (2.29)$$

em que  $\mathbf{B}$  e  $\mathbf{N}$  são matrizes formadas por colunas da matriz original  $\mathbf{A}$  do sistema (2.25). Essas matrizes e os vetores correspondentes de variáveis  $\mathbf{x}_B$  e  $\mathbf{x}_N$  decorrem das regras de multiplicação de matrizes. Denotando-se:  $\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \mathbf{a}_3 \ \mathbf{a}_4 \ \mathbf{a}_5]$ , em que  $\mathbf{a}_i$  é a  $i$ -ésima coluna da matriz  $\mathbf{A}$ , temos:

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 3 & 1 & -1 \end{bmatrix} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \mathbf{a}_5] \quad \text{e} \quad \mathbf{N} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = [\mathbf{a}_3 \ \mathbf{a}_4].$$

Em um sistema  $\mathbf{Ax} = \mathbf{b}$ , cada variável multiplica sua respectiva coluna da matriz  $\mathbf{A}$ , isto é,  $\mathbf{Ax} = \mathbf{b}$  é equivalente a  $\mathbf{a}_1 x_1 + \mathbf{a}_2 x_2 + \dots + \mathbf{a}_n x_n = \mathbf{b}$  e, portanto, na reorganização do sistema (2.29) está implícita a seguinte reorganização nas variáveis:

$$\mathbf{x}_B = \begin{bmatrix} x_1 \\ x_2 \\ x_5 \end{bmatrix} \text{ e } \mathbf{x}_N = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix}.$$

Podemos, ainda, definir dois vetores de índices:

$$\begin{aligned} B &= (B_1 \ B_2 \ B_3): & B_1 &= 1, \ B_2 = 2, \ B_3 = 5, \\ N &= (N_1 \ N_2): & N_1 &= 3, \ N_2 = 4, \end{aligned}$$

de modo que as matrizes  $\mathbf{B}$  e  $\mathbf{N}$ , e as variáveis  $\mathbf{x}_B$  e  $\mathbf{x}_N$  podem ser referenciadas por:

$$\mathbf{B} = [\mathbf{a}_{B_1} \mathbf{a}_{B_2} \mathbf{a}_{B_3}] = [\mathbf{a}_1 \mathbf{a}_2 \mathbf{a}_5] \quad \text{e} \quad \mathbf{N} = [\mathbf{a}_{N_1} \mathbf{a}_{N_2}] = [\mathbf{a}_3 \mathbf{a}_4],$$

$$\mathbf{x}_B = \begin{bmatrix} x_{B_1} \\ x_{B_2} \\ x_{B_3} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_5 \end{bmatrix} \quad \text{e} \quad \mathbf{x}_N = \begin{bmatrix} x_{N_1} \\ x_{N_2} \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix}.$$

Em resumo, o sistema  $\mathbf{Ax} = \mathbf{b}$  pode ser reescrito de forma equivalente como:

$$\mathbf{Ax} = \mathbf{b} \Leftrightarrow \mathbf{Bx}_B + \mathbf{Nx}_N = \mathbf{b}$$

e fixando-se  $\mathbf{x}_N = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ , o sistema restante é  $\mathbf{Bx}_B = \mathbf{b}$ . Os Exercícios 2.21 – 2.23 sugerem manipulações com as matrizes envolvidas nas partições básicas.

A idéia de reorganizar as colunas do sistema (como feito acima) para encontrar uma solução qualquer é importante para a descrição do *método simplex*. Embora já tenhamos introduzido uma notação matricial que permite a generalização do procedimento anterior para a obtenção de soluções de um sistema  $\mathbf{Ax} = \mathbf{b}$  qualquer, repetimos esse procedimento anterior explicitando algumas hipóteses que ficaram subjacentes. Como vimos, um modo de resolver o sistema consiste em fixar valores para  $n - m$  variáveis, resultando em um sistema de  $m$  variáveis (as variáveis restantes) e  $m$  equações. Se a matriz desse sistema restante for invertível, a solução é bem determinada. Podemos agir também de modo inverso, isto é, selecionamos  $m$  colunas da matriz  $\mathbf{A}$  que formem uma matriz invertível e fixamos as demais variáveis. Sem perda de generalidade, podemos supor que a matriz  $\mathbf{A}$  tem posto completo por linhas (veja Apêndice A), ou seja,  $\text{posto}(\mathbf{A}) = m$ . Dessa forma, sempre é possível selecionar  $m$  colunas da matriz  $\mathbf{A}$  que formem uma matriz invertível.

A hipótese de que  $\text{posto}(\mathbf{A}) = m$  implica que  $m \leq n$ , ou seja, o número de equações é menor ou igual ao número de variáveis. Para o caso  $m = n$ , o sistema tem solução única, portanto, o problema de otimização é trivial de ser resolvido. Quando consideramos um problema de otimização linear na forma padrão, admitimos, sem perda de generalidade, que  $m < n$  (é comum que  $m \ll n$ ; lê-se “ $m$  muito menor que  $n$ ”). Com isso, o sistema linear  $\mathbf{Ax} = \mathbf{b}$  tem sempre infinitas soluções e o grau de liberdade é  $n - m$ , isto é, o sistema tem  $n - m$  variáveis “sobrando”, que, uma vez fixadas, o sistema passa a ter solução única.

**Definição 2.4** (*partição básica*) Observe uma reorganização nas colunas da matriz da seguinte forma:

$$\mathbf{A} = [\mathbf{B} \ \mathbf{N}]$$

em que:

- $\mathbf{B}_{m \times m}$ , chamada *matriz básica*, é formada por  $m$  colunas da matriz  $\mathbf{A}$  e é invertível, dada por  $\mathbf{B} = [\mathbf{a}_{B_1} \mathbf{a}_{B_2} \cdots \mathbf{a}_{B_m}]$ , isto é,  $B_1, B_2, \dots, B_m$  são os índices das colunas da matriz  $\mathbf{A}$  que pertencem a  $\mathbf{B}$ , chamados *índices básicos*.
- $\mathbf{N}_{m \times (n-m)}$ , chamada *matriz não-básica*, é formada pelas  $n - m$  colunas restantes de  $\mathbf{A}$  (isto é, colunas de  $\mathbf{A}$  que não estão em  $\mathbf{B}$ ), dada por  $\mathbf{N} = [\mathbf{a}_{N_1} \mathbf{a}_{N_2} \cdots \mathbf{a}_{N_{n-m}}]$ , isto é,  $N_1, N_2, \dots, N_{n-m}$  são os índices das colunas da matriz  $\mathbf{A}$  que pertencem a  $\mathbf{N}$ .

Essa partição nas colunas da matriz  $\mathbf{A}$  é chamada *partição básica* e introduz uma partição no vetor  $\mathbf{x}$ :

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix}$$

em que:

- $\mathbf{x}_B = \begin{bmatrix} x_{B_1} \\ x_{B_2} \\ \vdots \\ x_{B_m} \end{bmatrix}$ , chamado vetor das *variáveis básicas*;
- $\mathbf{x}_N = \begin{bmatrix} x_{N_1} \\ \vdots \\ x_{N_{n-m}} \end{bmatrix}$ , chamado vetor das *variáveis não-básicas*.

Considerando uma partição básica  $\mathbf{A} = [\mathbf{B} \ \mathbf{N}]$ , o sistema  $\mathbf{Ax} = \mathbf{b}$  pode ser reescrito de forma equivalente como

$$\mathbf{Ax} = \mathbf{b} \Leftrightarrow [\mathbf{BN}] \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} = \mathbf{b} \quad (2.30)$$

ou

$$\mathbf{Bx}_B + \mathbf{Nx}_N = \mathbf{b}.$$

Portanto,

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{Nx}_N \quad (\text{solução geral}). \quad (2.31)$$

A expressão (2.31) é chamada *solução geral do sistema*, pois com ela podemos determinar qualquer solução do sistema, bastando atribuir valores quaisquer às  $n - m$  variáveis não-básicas em  $\mathbf{x}_N$ , de modo que as  $m$  variáveis básicas em  $\mathbf{x}_B$  fiquem unicamente determinadas e a solução resultante satisfaça o sistema  $\mathbf{Ax} = \mathbf{b}$ . A ordem inicial das variáveis é alterada conforme a partição básica e um abuso de notação é feito, sem que signifique um erro conceitual. Por exemplo, no sistema (2.25), as variáveis são organizadas por  $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$ , enquanto, com a introdução da partição básica, no sistema (2.29) as variáveis são reorganizadas por  $\mathbf{x} = (\mathbf{x}_B, \mathbf{x}_N) = (x_1, x_2, x_5 | x_3, x_4)$  e nos referimos ao vetor de variáveis  $\mathbf{x}$ , em ambos os casos.

**Definição 2.5** (*solução básica*) Considere uma partição básica  $\mathbf{A} = [\mathbf{B} \ \mathbf{N}]$  e a seguinte solução obtida ao se fixar as  $n - m$  variáveis de  $\mathbf{x}_N$  em zero, isto é:

$$\begin{cases} \hat{\mathbf{x}}_B = \mathbf{B}^{-1}\mathbf{b} \\ \hat{\mathbf{x}}_N = \mathbf{0}. \end{cases}$$

A solução  $\hat{\mathbf{x}}$  assim obtida é chamada *solução básica*. Se  $\hat{\mathbf{x}}_B = \mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$ , ou seja, se todas as variáveis básicas são não-negativas, dizemos que  $\hat{\mathbf{x}}$  é uma *solução básica factível*. Além disso, se  $\hat{\mathbf{x}}_B = \mathbf{B}^{-1}\mathbf{b} > \mathbf{0}$  (todas as variáveis básicas são positivas), dizemos que a solução básica factível é *não-degenerada*.

Retomemos o exemplo introduzido no início desta seção, cuja região factível é representada na Figura 2.35, e analisemos o ponto **D**. Como já observado anteriormente, podemos escrever o sistema linear como:

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 3 & 1 & -1 \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_5 \end{bmatrix}}_{\mathbf{x}_B} + \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\mathbf{N}} \underbrace{\begin{bmatrix} x_3 \\ x_4 \end{bmatrix}}_{\mathbf{x}_N} = \underbrace{\begin{bmatrix} 6 \\ 4 \\ 3 \end{bmatrix}}_{\mathbf{b}}$$

Assim, fixando as variáveis não-básicas em zero:  $\hat{x}_3 = \hat{x}_4 = 0$  (isto é,  $\hat{\mathbf{x}}_N = \mathbf{0}$ ), temos:

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 3 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_5 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \\ 3 \end{bmatrix} \Leftrightarrow \mathbf{B}\hat{\mathbf{x}}_B = \mathbf{b}$$

cuja solução é:

$$\hat{\mathbf{x}}_B = \begin{bmatrix} x_1 \\ x_2 \\ x_5 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ 13 \end{bmatrix}$$

Portanto (observe a reorganização do vetor  $\mathbf{x}$ ),

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_B \\ \hat{\mathbf{x}}_N \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_5 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ 13 \\ 0 \\ 0 \end{bmatrix}$$

que resolve o sistema  $\mathbf{Ax} = \mathbf{b}$  e não fere a condição de não-negatividade, ou seja,  $\mathbf{D}$  é uma solução básica factível. Considere agora o ponto  $\mathbf{F}$ , que também é obtido pela interseção de duas retas,  $x_2 = 0$  e  $x_3 = 0$ . Isso sugere a partição básica:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 3 & 0 & -1 \end{bmatrix}}_B \underbrace{\begin{bmatrix} x_1 \\ x_4 \\ x_5 \end{bmatrix}}_{\mathbf{x}_B} + \underbrace{\begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 1 & 0 \end{bmatrix}}_N \underbrace{\begin{bmatrix} x_2 \\ x_3 \\ x_N \end{bmatrix}}_{\mathbf{x}_N} = \begin{bmatrix} 6 \\ 4 \\ 3 \end{bmatrix}_{\mathbf{b}}$$

(a partição básica agora é diferente da anterior, mas o sistema  $\mathbf{Ax} = \mathbf{b}$  é o mesmo). No ponto  $\mathbf{F}$ ,  $\hat{x}_2 = \hat{x}_3 = 0$  (isto é,  $\hat{\mathbf{x}}_N = \mathbf{0}$ ), então  $\mathbf{B}\hat{\mathbf{x}}_B = \mathbf{b}$  é dado por:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 3 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \\ 3 \end{bmatrix}$$

cuja solução é:

$$\hat{\mathbf{x}}_B = \begin{bmatrix} x_1 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 6 \\ -2 \\ 15 \end{bmatrix}$$

Vemos que  $x_4 = -2$  viola a condição de não-negatividade, portanto,  $\mathbf{F}$  não é solução factível para o problema, algo já observado na Figura 2.35 ( $x_1 - x_2 > 4$  é equivalente a  $x_4 < 0$ ).

A visualização gráfica mostra que os *vértices* de  $\mathbf{S}$  (um *vértice* de  $\mathbf{S}$  é um ponto de  $\mathbf{S}$  que não pertence ao interior de qualquer segmento de reta que liga dois outros pontos de  $\mathbf{S}$ ) correspondem às soluções básicas factíveis. De fato, essa afirmação é verdadeira para uma região factível qualquer e pode ser resumida da seguinte forma.

**Propriedade 2.1** Considere a região factível  $\mathbf{S} = \{\mathbf{x} \in \mathbf{R}^n \text{ tal que } \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ . Um ponto  $\mathbf{x} \in \mathbf{S}$  é um vértice de  $\mathbf{S}$  se e somente se  $\mathbf{x}$  for uma solução básica factível.

Uma consequência desta propriedade é que a região factível  $S$  tem um número finito de vértices, pois há um número finito de partições básicas, limitado por  $C_m^n = \frac{n!}{m!(n-m)!}$ . Apesar de esse número ser finito, pode ser muito grande para problemas práticos em que  $n$  é da ordem de centenas ou milhares, e muito maior que  $m$ . Enunciaremos agora uma propriedade fundamental da otimização linear, a qual já havíamos intuído durante a resolução gráfica desses problemas.

**Propriedade 2.2** Se um problema de otimização linear tem solução ótima, então existe um vértice ótimo.<sup>18</sup>

Em outras palavras, se existe uma solução ótima, existe uma solução básica factível ótima. Como consequência desta propriedade, basta que se procure o ótimo entre todas as soluções básicas factíveis. Isso sugere um método de solução:

- Determine todas as, digamos,  $K$  soluções básicas factíveis (vértices da região factível  $S$ ):  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K$ .
- Determine a solução ótima  $\mathbf{x}_j$  tal que  $f(\mathbf{x}_j) = \text{mínimo } \{f(\mathbf{x}_k), k = 1, 2, \dots, K\}$ .

Porém, o número  $K$  (note que  $K \leq C_m^n$ ) pode ser muito grande em problemas práticos, inviabilizando computacionalmente este procedimento. Um método mais elaborado que inicia com uma solução básica factível e pesquisa apenas outras soluções básicas melhores que a corrente é o método simplex. Este método está detalhado nas Seções 2.6.2–2.6.4. Para um estudo mais direto do mecanismo do método simplex, essas seções podem ser substituídas pela Seção 2.7, na qual o método simplex é apresentado em forma de tabelas.

### 2.6.2\* O método simplex

O método simplex encontra um vértice ótimo pesquisando apenas um subconjunto (em geral, pequeno) dos  $K$  vértices de  $S$ . Para construir um método de resolução de um problema de otimização linear, devemos responder a duas perguntas-chave:

Dada uma solução básica factível (vértice de  $S$ , portanto, candidata à solução ótima),

1. Essa solução é ótima?

2. Caso não seja ótima, como determinar uma outra solução básica factível melhor?

**Resposta à primeira pergunta:** *a solução é ótima?*

Considere uma solução básica factível

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_B \\ \hat{\mathbf{x}}_N \end{bmatrix} \text{ em que } \begin{cases} \hat{\mathbf{x}}_B = \mathbf{B}^{-1}\mathbf{b} \geq 0 \\ \hat{\mathbf{x}}_N = \mathbf{0}, \end{cases}$$

e a solução geral (2.31) usando a mesma partição básica, isto é:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} \text{ tal que } \mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N \quad (2.32)$$

(lembre-se de que, para qualquer  $\mathbf{x}_N$  escolhido,  $\mathbf{x}$  é uma solução do sistema  $\mathbf{A}\mathbf{x} = \mathbf{b}$ ).

A função objetivo  $f(\mathbf{x})$  pode ser expressa considerando a partição básica:

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{c}^T \mathbf{x} = \\ &= [\mathbf{c}_B^T \quad \mathbf{c}_N^T] \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} = \mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_N^T \mathbf{x}_N \end{aligned} \quad (2.33)$$

18 Demonstrações matemáticas das propriedades 2.1 e 2.2 podem ser encontradas em Bregalda *et al.* (1988) ou Bazaraa *et al.* (1990).

com:

$\mathbf{c}_B^T$ : coeficientes das variáveis básicas na função objetivo;

$\mathbf{c}_N^T$ : coeficientes das variáveis não-básicas na função objetivo.

Substituindo (2.32) em (2.33), isto é, restringindo  $\mathbf{x}$  ao sistema  $\mathbf{Ax} = \mathbf{b}$ , temos:

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{c}_B^T \underbrace{(\mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{Nx}_N)}_{\mathbf{x}_B} + \mathbf{c}_N^T \mathbf{x}_N \\ &= \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{b} - \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{Nx}_N + \mathbf{c}_N^T \mathbf{x}_N. \end{aligned} \quad (2.34)$$

O primeiro termo de (2.34) corresponde ao valor da função objetivo em  $\hat{\mathbf{x}}$ :

$$f(\hat{\mathbf{x}}) = \mathbf{c}_B^T \hat{\mathbf{x}}_B + \mathbf{c}_N^T \hat{\mathbf{x}}_N = \mathbf{c}_B^T (\mathbf{B}^{-1}\mathbf{b}) + \mathbf{c}_N^T (\mathbf{0}) = \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{b}.$$

Para simplificar a notação e, posteriormente, os cálculos, definimos a seguir um vetor auxiliar, que é estudado em detalhes na Seção 2.10 .

**Definição 2.6** (*vetor multiplicador simplex*) O vetor  $\boldsymbol{\lambda}_{m \times 1}$ , dado por

$$\boldsymbol{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1}$$

é chamado *vetor multiplicador simplex* (ou também, vetor de *variáveis duais*).

**Nota:** O vetor multiplicador simplex pode ser obtido pela resolução do sistema de equações lineares  $\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B$ , que é obtido ao se tomar a transposta de  $\boldsymbol{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1}$  e multiplicar ambos os termos da igualdade por  $\mathbf{B}^T$ , isto é:

$$\boldsymbol{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1} \Leftrightarrow \boldsymbol{\lambda} = (\mathbf{B}^{-1})^T \mathbf{c}_B \Leftrightarrow \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B.$$

Utilizando o vetor multiplicador simplex na expressão de  $f(\mathbf{x})$  em (2.34), segue-se que:

$$\begin{aligned} f(\mathbf{x}) &= f(\hat{\mathbf{x}}) - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{Nx}_N + \mathbf{c}_N^T \mathbf{x}_N = f(\hat{\mathbf{x}}) - \boldsymbol{\lambda}^T \mathbf{Nx}_N + \mathbf{c}_N^T \mathbf{x}_N \\ &= f(\hat{\mathbf{x}}) + (\mathbf{c}_N^T - \boldsymbol{\lambda}^T \mathbf{N}) \mathbf{x}_N \end{aligned}$$

ou, ainda, considerando que:

$$\begin{aligned} \mathbf{c}_N^T - \boldsymbol{\lambda}^T \mathbf{N} &= (c_{N_1}, c_{N_2}, \dots, c_{N_{n-m}}) - \boldsymbol{\lambda}^T (a_{N_1}, a_{N_2}, \dots, a_{N_{n-m}}) \\ &= (c_{N_1} - \boldsymbol{\lambda}^T a_{N_1}, c_{N_2} - \boldsymbol{\lambda}^T a_{N_2}, \dots, c_{N_{n-m}} - \boldsymbol{\lambda}^T a_{N_{n-m}}) \end{aligned}$$

e

$$\mathbf{x}_N = (x_{N_1}, x_{N_2}, \dots, x_{N_{n-m}})$$

obtemos:

$$f(\mathbf{x}) = f(\hat{\mathbf{x}}) + (c_{N_1} - \boldsymbol{\lambda}^T a_{N_1}) x_{N_1} + (c_{N_2} - \boldsymbol{\lambda}^T a_{N_2}) x_{N_2} + \dots + (c_{N_{n-m}} - \boldsymbol{\lambda}^T a_{N_{n-m}}) x_{N_{n-m}} \quad (2.35)$$

**Definição 2.7** (*custos relativos*) Os coeficientes  $\hat{c}_{N_j} = (c_{N_j} - \boldsymbol{\lambda}^T a_{N_j})$  das variáveis não-básicas na função objetivo descrita por (2.35) são chamados *custos relativos* ou *custos reduzidos*.

Com essa notação, (2.35) é simplesmente reescrita por:

$$f(\mathbf{x}) = f(\hat{\mathbf{x}}) + \hat{c}_{N_1} x_{N_1} + \hat{c}_{N_2} x_{N_2} + \dots + \hat{c}_{N_{n-m}} x_{N_{n-m}} \quad (2.36)$$

**Exemplo 2.22** Considere o seguinte problema de otimização linear:

$$\begin{aligned} \text{Minimizar } f(x_1, x_2) &= -2x_1 - x_2 \\ x_1 + x_2 &\leq 4 \\ x_1 &\leq 3 \\ x_2 &\leq \frac{7}{2} \\ x_1 \geq 0, x_2 \geq 0 & \end{aligned}$$

que pode ser reescrito na forma padrão:

$$\begin{aligned} \text{Minimizar } f(x_1, x_2, x_3, x_4, x_5) &= -2x_1 - x_2 + 0x_3 + 0x_4 + 0x_5 \\ x_1 + x_2 + x_3 &= 4 \\ x_1 + x_4 &= 3 \\ x_2 + x_5 &= \frac{7}{2} \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0. & \end{aligned}$$

A resolução gráfica é dada na Figura 2.36 e a solução ótima é obtida pela intersecção das retas  $x_1 + x_2 = 4$  e  $x_1 = 3$ . Portanto,  $x_1^* = 3$ ,  $x_2^* = 1$  e  $f(\mathbf{x}^*) = -7$ . Essas equações de reta decorrem das restrições anteriores na forma de igualdade com  $x_3 = 0$  e  $x_4 = 0$ , o que resulta no sistema 3x3:

$$\begin{aligned} x_1 + x_2 &= 4 \\ x_1 &= 3 \\ x_2 + x_5 &= \frac{7}{2} \end{aligned}$$

cuja solução é:  $x_1^* = 3$ ,  $x_2^* = 1$ ,  $x_5^* = \frac{5}{2}$  ( $x_3^* = x_4^* = 0$ ).

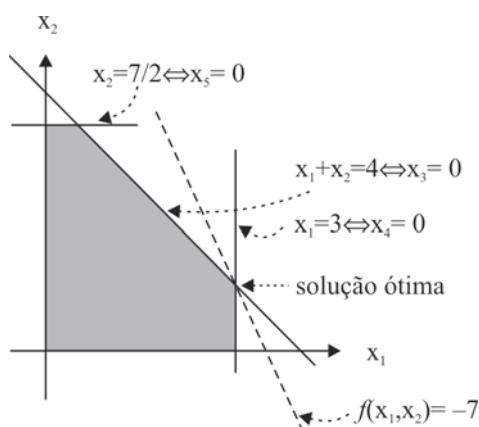
A solução ótima é uma solução básica com a seguinte partição básica:

$$B_1 = 1, B_2 = 2, B_3 = 5, N_1 = 3, N_2 = 4.$$

Com essa partição, temos:

$$\begin{aligned} \mathbf{B} &= [\mathbf{a}_{B_1} \quad \mathbf{a}_{B_2} \quad \mathbf{a}_{B_3}] = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_5] = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad \mathbf{N} = [\mathbf{a}_{N_1} \quad \mathbf{a}_{N_2}] = [\mathbf{a}_3 \quad \mathbf{a}_4] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \\ \mathbf{c}_B^T &= (c_{B_1} \quad c_{B_2} \quad c_{B_3}) = (c_1 \quad c_2 \quad c_5) = (-2 \quad -1 \quad 0), \quad \mathbf{c}_N^T = (c_{N_1} \quad c_{N_2}) = (c_3 \quad c_4) = (0 \quad 0). \end{aligned}$$

Para o cálculo dos custos relativos  $\hat{c}_{N_j} = (c_{N_j} - \lambda^T \mathbf{a}_{N_j})$ ,  $j=1, 2$  (lembre-se de que  $n - m = 2$ ), primeiramente o vetor multiplicador simplex é calculado.



**Figura 2.36** Solução gráfica do Exemplo 2.22.

- Multiplicador simplex:  $\lambda^T = \mathbf{c}_B^T \mathbf{B}^{-1}$ .

A seguir, calculamos o vetor multiplicador por dois procedimentos: (i) explicitando a inversa da matriz  $\mathbf{B}$ ; e (ii) resolvendo um sistema linear. Sempre que uma operação envolve o cálculo de uma inversa de matriz, ela pode ser substituída por um sistema linear, para o qual podem ser desenvolvidas técnicas avançadas de resolução.

$$(i) \text{ usando a inversa: } \mathbf{B}^{-1} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -1 & 0 \\ -1 & 1 & 1 \end{bmatrix}.$$

$$\lambda^T = \mathbf{c}_B^T \mathbf{B}^{-1} = (-2 \quad -1 \quad 0) \begin{bmatrix} 0 & 1 & 0 \\ 1 & -1 & 0 \\ -1 & 1 & 1 \end{bmatrix} = (-1 \quad -1 \quad 0).$$

(ii) resolvendo o sistema de equações lineares:  $\mathbf{B}^T \lambda = \mathbf{c}_B$ , isto é:

$$\begin{aligned} \lambda_1 + \lambda_2 &= -2 \\ \lambda_1 + \lambda_3 &= -1 \\ \lambda_3 &= 0 \end{aligned}$$

pelo método de eliminação de Gauss ou escalonamento (veja Apêndice A). Neste exemplo, o método consiste no primeiro passo, em subtrair a primeira equação da segunda para eliminar  $\lambda_1$  da segunda equação (o coeficiente de  $\lambda_1$  na terceira equação já é zero), e obter o sistema linear equivalente:

$$\begin{aligned} \lambda_1 + \lambda_2 &= -2 \\ -\lambda_2 + \lambda_3 &= 1 \\ \lambda_3 &= 0. \end{aligned}$$

O segundo passo do método para eliminar  $\lambda_2$  da terceira equação é desnecessário, já que o sistema é triangular superior e, portanto, pode ser resolvido por substituições da terceira equação para a primeira (em sistemas de grande porte, com matrizes esparsas, muitas operações de eliminação são desnecessárias):

$$\begin{aligned} \lambda_3 &= 0 \\ -\lambda_2 + \lambda_3 &= 1 \quad \Rightarrow \quad \lambda_2 = -1 + \lambda_3 = -1 \\ \lambda_1 + \lambda_2 &= -2 \quad \Rightarrow \quad \lambda_1 = -2 - \lambda_2 = -1. \end{aligned}$$

Podemos aplicar o método de eliminação de Gauss, sem perda de generalidade, na matriz aumentada do sistema linear  $[\mathbf{B}^T \mid \mathbf{c}_B]$ , isto é, podemos reproduzir as operações de eliminação de Gauss trabalhando apenas com os coeficientes do sistema: multiplique a primeira linha da matriz aumentada por  $-1$  e some à segunda, para obter a nova matriz aumentada que corresponde ao sistema triangular anterior.

$$\left[ \begin{array}{ccc|c} 1 & 1 & 0 & -2 \\ 1 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 \end{array} \right] \rightarrow \left[ \begin{array}{ccc|c} 1 & 1 & 0 & -2 \\ 0 & -1 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{array} \right].$$

**Nota:** Para matrizes de ordens pequenas, como no exemplo anterior, é irrelevante o uso da inversa ou a resolução do sistema linear, mas, para matrizes de porte grande, devem ser utilizadas técnicas avançadas para resolução de sistemas lineares.

Com o vetor multiplicador simplex calculado, podemos determinar facilmente os custos relativos para todas as variáveis não-básicas:

$$j=1: \hat{c}_{N_1} = \hat{c}_3 = c_3 - \boldsymbol{\lambda}^T \mathbf{a}_3 = 0 - (-1 \quad -1 \quad 0) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = 1$$

$$j=2: \hat{c}_{N_2} = \hat{c}_4 = c_4 - \boldsymbol{\lambda}^T \mathbf{a}_4 = 0 - (-1 \quad -1 \quad 0) \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = 1.$$

Assim, a função objetivo, quando restrita ao sistema  $\mathbf{Ax} = \mathbf{b}$  (ver expressão 2.36), é dada por  $f(\mathbf{x}) = -7 + x_3 + x_4$  e, portanto,  $f(\mathbf{x}) \geq -7$  para toda solução factível, uma vez que  $x_3 \geq 0$  e  $x_4 \geq 0$ . Disso, concluímos que a solução ótima é obtida com  $x_3 = x_4 = 0$ , ou seja, a solução básica é ótima. ■

Retomemos as expressões (2.35) e (2.36). Sabemos que  $x_{N_j} \geq 0$  (todas as variáveis do problema são não-negativas). Portanto, se  $(c_{N_j} - \boldsymbol{\lambda}^T \mathbf{a}_{N_j}) \geq 0, j = 1, \dots, n-m$ , então  $f(\mathbf{x}) \geq f(\hat{\mathbf{x}})$  para todo  $\mathbf{x}_N \geq \mathbf{0}$ . Assim, podemos enunciar a seguinte propriedade:

**Propriedade 2.3** (*condição de otimalidade*) Considere uma partição básica  $\mathbf{A} = [\mathbf{B} \ \mathbf{N}]$  em que a solução básica associada  $\hat{\mathbf{x}}_B = \mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$  (isto é, solução básica factível), e seja  $\boldsymbol{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1}$  o vetor multiplicador simplex. Se  $(c_{N_j} - \boldsymbol{\lambda}^T \mathbf{a}_{N_j}) \geq 0, j = 1, \dots, n-m$ , (isto é, todos os custos relativos são não-negativos), então a solução básica é ótima.

A propriedade anterior fornece uma maneira simples de se afirmar a otimalidade de uma solução básica factível: *se a condição de otimalidade for verificada, então a solução básica factível é ótima*. Se a solução básica factível for não-degenerada, a recíproca da condição de otimalidade também é verdadeira: *se uma solução básica factível não-degenerada (isto é,  $\hat{\mathbf{x}}_B = \mathbf{B}^{-1}\mathbf{b} > \mathbf{0}$ ) é uma solução ótima, então  $(c_{N_j} - \boldsymbol{\lambda}^T \mathbf{a}_{N_j}) \geq 0, j = 1, \dots, n-m$* . Porém, podemos ter uma solução ótima degenerada em mãos sem que a condição de otimalidade seja verificada (ver Exercício 2.24-f). Nesse caso, entretanto, é sempre possível identificar outra partição básica para a mesma solução degenerada que satisfaça a condição de otimalidade (soluções degeneradas podem ser representadas por diferentes partições básicas).

**Resposta à segunda pergunta:** *como determinar uma solução básica factível melhor?*

Considere uma solução básica factível e suponha que a condição de otimalidade seja violada (caso contrário, a solução é ótima), isto é, suponha que exista  $k$  tal que:

$$\hat{c}_{N_k} = c_{N_k} - \boldsymbol{\lambda}^T \mathbf{a}_{N_k} < 0,$$

ou seja, o custo relativo da variável não-básica  $x_{N_k}$  é negativo.

**Exemplo 2.23** Considere o problema de otimização linear dado no Exemplo 2.22 e a seguinte partição básica:  $B_1 = 3, B_2 = 4, B_3 = 5, N_1 = 1, N_2 = 2$ .

Com essa partição, temos:

$$\mathbf{B} = [\mathbf{a}_{B_1} \quad \mathbf{a}_{B_2} \quad \mathbf{a}_{B_3}] = [\mathbf{a}_3 \quad \mathbf{a}_4 \quad \mathbf{a}_5] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{N} = [\mathbf{a}_{N_1} \quad \mathbf{a}_{N_2}] = [\mathbf{a}_1 \quad \mathbf{a}_2] = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$\mathbf{c}_B^T = (c_{B_1} \quad c_{B_2} \quad c_{B_3}) = (c_3 \quad c_4 \quad c_5) = (0 \quad 0 \quad 0)$$

$$\mathbf{c}_N^T = (c_{N_1} \quad c_{N_2}) = (c_1 \quad c_2) = (-2 \quad -1).$$

Os custos relativos  $\hat{c}_{N_j} = (c_{N_j} - \lambda^T \mathbf{a}_{N_j})$ ,  $j = 1, 2$ , podem ser calculados:

- *Multiplicador simplex:*

$$\lambda^T = \mathbf{c}_B^T \mathbf{B}^{-1} = (0 \ 0 \ 0) \quad (\text{neste caso, a inversa de } \mathbf{B} \text{ é a própria identidade});$$

$$\bullet j = 1: \hat{c}_{N_1} = \hat{c}_1 = c_1 - \lambda^T \mathbf{a}_1 = -2 - (0 \ 0 \ 0) \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = -2;$$

$$\bullet j = 2: \hat{c}_{N_2} = \hat{c}_2 = c_2 - \lambda^T \mathbf{a}_2 = -1 - (0 \ 0 \ 0) \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = -1.$$

Com essa partição,  $\hat{c}_1$  e  $\hat{c}_2$  são negativos e, portanto, não satisfazem a condição de otimalidade. ■

Vemos a seguir como a solução básica factível pode ser perturbada de modo a diminuir o valor da função objetivo, por uma estratégia que fornece o fundamento do *método simplex*.

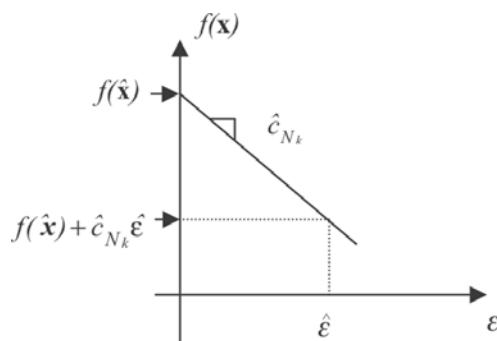
**Definição 2.8** (*estratégia simplex*) Chamamos de estratégia simplex a perturbação de uma solução básica factível que consiste em alterar as variáveis não-básicas por:

$$\begin{cases} x_{N_k} = \varepsilon \geq 0, & (\text{variável com custo relativo negativo}) \\ x_{N_j} = 0, \quad j = 1, 2, \dots, n-m, \quad i \neq k. \end{cases}$$

Em palavras, apenas uma variável não-básica,  $x_{N_k}$ , deixa de ser nula (por simplicidade, supomos que a solução é não degenerada, caso contrário  $\varepsilon > 0$  pode levar a uma solução infactível, como vemos adiante). Com isso, a função objetivo passa a valer (ver (2.36)):

$$\begin{aligned} f(\mathbf{x}) &= f(\hat{\mathbf{x}}) + \hat{c}_{N_1} \underbrace{0}_{x_{N_1}} + \cdots + \hat{c}_{N_k} \underbrace{\varepsilon}_{x_{N_k}} + \cdots + \hat{c}_{N_{n-m}} \underbrace{0}_{x_{N_{n-m}}} = \\ &= f(\hat{\mathbf{x}}) + \hat{c}_{N_k} \varepsilon < f(\hat{\mathbf{x}}) \end{aligned}$$

Note que a função objetivo decresce quando  $\varepsilon$  cresce, com a taxa negativa  $\hat{c}_{N_k}$ , ilustrada na Figura 2.37. Quanto menor o valor de  $\hat{c}_{N_k}$ , mais rápido a função objetivo decresce. Isso justifica a escolha da variável não-básica a ser perturbada com o menor custo relativo (essa escolha é conhecida na literatura como a *regra de Dantzig*).



**Figura 2.37** Variação da função objetivo com a estratégia simplex.

Como a função objetivo decresce quando  $\varepsilon$  cresce, determinamos o maior valor possível para  $\varepsilon$  que mantém factível a solução perturbada. Vejamos a seguir como determinar tal valor.

### Tamanho do passo $\varepsilon$

Com a alteração nos valores das variáveis não-básicas pela estratégia simplex, as variáveis básicas  $\mathbf{x}_B$  também devem ser alteradas, de modo que o sistema  $\mathbf{Ax} = \mathbf{b}$  seja satisfeito (reveja a solução geral (2.31)). A estratégia simplex é equivalente a alterar as variáveis não-básicas para:

$$\mathbf{x}_N = \begin{pmatrix} x_{N_1} \\ \vdots \\ x_{N_k} \\ \vdots \\ x_{N_{n-m}} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ \varepsilon \\ \vdots \\ 0 \end{pmatrix} \leftarrow k \quad (2.37)$$

portanto, as variáveis básicas são modificadas por

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N = \hat{\mathbf{x}}_B - \underbrace{\mathbf{B}^{-1}\mathbf{a}_{N_k}}_y \varepsilon = \hat{\mathbf{x}}_B - \mathbf{y}\varepsilon \quad (2.38)$$

em que  $\mathbf{y} = \mathbf{B}^{-1}\mathbf{a}_{N_k}$ . A identidade anterior decorre de definição da matriz não-básica N e da definição de  $\mathbf{x}_N$  dada em (2.37):

$$\mathbf{N}\mathbf{x}_N = \mathbf{N}(0 \dots \varepsilon \dots 0)^T = [\mathbf{a}_{N_1} \dots \mathbf{a}_{N_k} \dots \mathbf{a}_{N_{n-m}}](0 \dots \varepsilon \dots 0)^T = \mathbf{a}_{N_k} \varepsilon.$$

**Definição 2.9** (*direção simplex*) Chamamos de direção simplex o vetor  $\mathbf{y} = \mathbf{B}^{-1}\mathbf{a}_{N_k}$ , o qual fornece os coeficientes de como as variáveis básicas são alteradas pela estratégia simplex. A direção simplex é solução do sistema de equações lineares  $\mathbf{By} = \mathbf{a}_{N_k}$ .

Reescrevendo a equação vetorial (2.38) em cada uma de suas coordenadas, e considerando a não-negatividade das variáveis básicas temos:

$$x_{B_i} = \hat{x}_{B_i} - y_i \varepsilon \geq 0, \quad i = 1, \dots, m.$$

Assim,

$$\text{se } y_i \leq 0, \text{ então } x_{B_i} \geq 0, \text{ para todo } \varepsilon \geq 0$$

$$\text{se } y_i > 0, \text{ como } x_{B_i} = \hat{x}_{B_i} - y_i \varepsilon \geq 0, \text{ então, } \varepsilon \leq \frac{\hat{x}_{B_i}}{y_i}.$$

Logo, o maior valor de  $\varepsilon$  é dado por

$$\hat{\varepsilon} = \frac{\hat{x}_{B_\ell}}{y_\ell} = \min \left\{ \frac{\hat{x}_{B_i}}{y_i} \text{ tal que } y_i > 0 \right\}. \quad (2.39)$$

**Solução ótima ilimitada:** Se  $y_i \leq 0$  para todo  $i = 1, \dots, m$ , então (2.39) não se aplica, ou seja, não há limite superior para  $\varepsilon$ , o que significa que a solução perturbada pela estratégia simplex será sempre factível para todo valor de  $\varepsilon \geq 0$ . Como a função objetivo decresce com o crescimento de  $\varepsilon$ , então  $f(\mathbf{x}) \rightarrow -\infty$ , com  $\varepsilon \rightarrow \infty$ . Disso, concluímos que o problema não tem solução ótima (é comum referir-se a esta situação como *solução ótima ilimitada*); ver Figura 2.30.

**Exemplo 2.24** Considere o problema de otimização linear no Exemplo 2.22 e a partição básica:

$$(B_1, B_2, B_3) = (3, 4, 5) \quad (N_1, N_2) = (1, 2).$$

Calculamos a solução básica, verificamos sua optimidade e aplicamos a estratégia simplex para obter um vértice melhor.

- Solução básica:  $\mathbf{x}_B = (x_3, x_4, x_5)$ , a qual é obtida ao se anularem as variáveis não-básicas.

A solução do sistema  $\mathbf{Bx}_B = \mathbf{b}$  (ou,  $\hat{\mathbf{x}}_B = \mathbf{B}^{-1}\mathbf{b}$ ), e neste caso a matriz  $\mathbf{B}$  é a identidade, é dada

por  $\hat{\mathbf{x}}_B = \begin{pmatrix} 4 \\ 3 \\ \frac{7}{2} \end{pmatrix}$  e o valor da função objetivo avaliada nesta solução é

$$f(\hat{\mathbf{x}}) = c_{B_1}\hat{x}_{B_1} + c_{B_2}\hat{x}_{B_2} + c_{B_3}\hat{x}_{B_3} = 0 \times 4 + 0 \times 3 + 0 \times \frac{7}{2} = 0.$$

- Optimalidade:

- multiplicador simplex: lembre que  $\mathbf{c}_B = (c_{B_1}, c_{B_2}, c_{B_3}) = (c_3, c_4, c_5) = (0, 0, 0)$

A solução do sistema  $\mathbf{B}^T\lambda = \mathbf{c}_B$  (ou,  $\lambda^T = \mathbf{c}_B^T \mathbf{B}^{-1}$ ) é  $\lambda = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

- custos relativos: as variáveis não-básicas são:  $N_1 = 1, N_2 = 2$ ,

$$\hat{c}_1 = c_1 - \lambda^T \mathbf{a}_1 = -2 - (0 \ 0 \ 0) \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = -2 \leftarrow k = 1 \quad (x_1 \text{ é alterada pela estratégia simplex})$$

$$\hat{c}_2 = c_2 - \lambda^T \mathbf{a}_2 = -1 - (0 \ 0 \ 0) \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = -1$$

Observação: como os dois custos relativos são negativos, aumentar  $x_1$  ou  $x_2$  faz decrescer a função objetivo. A escolha do menor custo relativo, chamada *regra de Dantzig*, é bastante utilizada.

- direção simplex  $\mathbf{y} = (y_1, y_2, y_3)$  (ver a definição 2.9).

A solução do sistema  $\mathbf{By} = \mathbf{a}_{N_1}$  é  $\mathbf{y} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$ , que proporciona como as variáveis básicas se movem:

dificam:  $x_{B_i} = \hat{x}_{B_i} - y_i \varepsilon$  (ver expressão (2.38)), ou seja,

$$x_3 = 4 - \varepsilon$$

$$x_4 = 3 - \varepsilon$$

$$x_5 = \frac{7}{2}.$$

O valor máximo para  $\varepsilon$  deve ser tal que as variáveis básicas sejam não-negativas, o que define o tamanho do passo.

- Tamanho do passo:

$$\hat{\varepsilon} = \min \left\{ \frac{\hat{x}_{B_1}}{y_1}, \frac{\hat{x}_{B_2}}{y_2} \right\} = \min \left\{ \frac{4}{1}, \frac{3}{1} \right\} = 3 = \frac{\hat{x}_{B_2}}{y_2}$$

Com o valor de  $\hat{\varepsilon} = 3$ , a variável  $x_{B_2} = x_4$  se anula (isto é,  $x_4 = 3 - 3 = 0$ ). Por outro lado, a variável não-básica  $x_1$  torna-se positiva:  $x_1 = \hat{\varepsilon} = 3$  ■

Retomemos a expressão (2.39). Com o valor de  $\hat{\varepsilon} = \frac{\hat{x}_{B_\ell}}{y_\ell}$ , a variável básica  $x_{B_\ell}$  se anula e a variável não-básica  $x_{N_k}$  torna-se positiva. De fato, da expressão (2.38), temos:

- $\ell$ -ésima variável básica:  $x_{B_\ell} = \hat{x}_{B_\ell} - y_\ell \hat{\varepsilon} = \hat{x}_{B_\ell} - y_\ell \left( \frac{x_{B_\ell}}{y_\ell} \right) = 0$ ,

- $k$ -ésima variável não-básica:  $x_{N_k} = \hat{\varepsilon}$ ,

Isto é, a nova solução tem a seguinte característica:

$$(x_{B_1} \cdots \underbrace{x_{B_\ell}}_{=0} \cdots x_{B_m} | 0 \cdots \underbrace{x_{N_k}}_{=\hat{\epsilon}} \cdots),$$

ou seja,  $n - m$  variáveis são nulas:  $x_{N_1}, \dots, x_{B_\ell}, \dots, x_{N_{n-m}}$ , as quais podem ser consideradas não-básicas. Isso resulta em uma nova partição básica com os índices trocados:

$$B_\ell \leftrightarrow N_k$$

Isto é, a variável  $x_{N_k}$  torna-se básica e a variável  $x_{B_\ell}$  não-básica. As matrizes básica e não-básica são alteradas por apenas uma coluna:

$$\begin{aligned} \mathbf{B} = [\mathbf{a}_{B_1}, \dots, \mathbf{a}_{B_\ell}, \dots, \mathbf{a}_{B_m}] &\rightarrow \mathbf{B}' = [\mathbf{a}_{B_1}, \dots, \underbrace{\mathbf{a}_{N_k}}_{\ell-\text{ésima coluna}}, \dots, \mathbf{a}_{B_m}] \\ \mathbf{N} = [\mathbf{a}_{N_1}, \dots, \mathbf{a}_{N_k}, \dots, \mathbf{a}_{N_{n-m}}] &\rightarrow \mathbf{N}' = [\mathbf{a}_{N_1}, \dots, \underbrace{\mathbf{a}_{B_\ell}}_{k-\text{ésima coluna}}, \dots, \mathbf{a}_{N_{n-m}}]. \end{aligned}$$

Dizemos que  $x_{N_k}$  entra na base e  $x_{B_\ell}$  sai da base. A seguinte propriedade garante que a nova partição é básica.

**Propriedade 2.4** A matriz  $\mathbf{B}'$ , definida anteriormente, é invertível, de modo que  $\mathbf{A} = [\mathbf{B}' \ \mathbf{N}']$  é uma partição básica.

Os passos para mostrar a validade da Propriedade 2.4 são sugeridos no Exercício 2.25. A solução básica associada à nova partição é aquela obtida pela estratégia simplex:

- $x_{N_k} = \hat{\epsilon}$
- $x_{B_i} = \hat{x}_{B_i} - y_i \hat{\epsilon} \quad i = 1, \dots, m, i \neq \ell,$

que é uma solução factível, por construção.

Com isso, mostramos que a estratégia simplex produz uma nova solução básica factível (isto é, um novo vértice), para a qual a função objetivo tem um valor menor:

$$f(\mathbf{x}) = f(\hat{\mathbf{x}}) + \hat{c}_{N_k} \hat{\epsilon} < f(\hat{\mathbf{x}}),$$

e podemos repetir o procedimento, isto é: *encontrar outra solução básica melhor a partir daquela em mãos*, enquanto a condição de optimalidade não for verificada. Este procedimento basicamente consiste no método simplex.

**Exemplo 2.25** Observe o problema de otimização linear no Exemplo 2.22 e a partição básica:

$$(B_1, B_2, B_3) = (1, 2, 4) \quad (N_1, N_2) = (3, 5).$$

Com essa partição, definimos as matrizes  $\mathbf{B}$  e  $\mathbf{N}$  e os vetores  $\mathbf{c}_B$  e  $\mathbf{c}_N$ :

$$\mathbf{B} = [\mathbf{a}_{B_1} \ \mathbf{a}_{B_2} \ \mathbf{a}_{B_3}] = [\mathbf{a}_1 \ \mathbf{a}_2 \ \mathbf{a}_4] = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{N} = [\mathbf{a}_{N_1} \ \mathbf{a}_{N_2}] = [\mathbf{a}_3 \ \mathbf{a}_5] = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix},$$

$$\mathbf{c}_B^T = (c_{B_1} \ c_{B_2} \ c_{B_3}) = (c_1 \ c_2 \ c_4) = (-2 \ -1 \ 0) \quad \mathbf{c}_N^T = (c_{N_1} \ c_{N_2}) = (c_3 \ c_5) = (0 \ 0).$$

Por simplicidade de exposição, calculamos  $\mathbf{B}^{-1}$ . A inversa de  $\mathbf{B}$  é

$$\mathbf{B}^{-1} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 1 \\ -1 & 1 & 1 \end{bmatrix}.$$

- *Solução básica:*  $\mathbf{x}_B = (x_1, x_2, x_4)$  é a solução do sistema  $\mathbf{Bx}_B = \mathbf{b}$ , ou seja, podemos aplicar o método de eliminação de Gauss, operando sobre a matriz aumentada:

$$\left[ \begin{array}{ccc|c} 1 & 1 & 0 & 4 \\ 1 & 0 & 1 & 3 \\ 0 & 1 & 0 & \frac{7}{2} \end{array} \right] \text{ e obter a solução: } \hat{\mathbf{x}}_B = \begin{pmatrix} \frac{1}{2} \\ \frac{7}{2} \\ \frac{5}{2} \end{pmatrix}$$

ou, de forma equivalente,  $\hat{\mathbf{x}}_B = \mathbf{B}^{-1}\mathbf{b} = \left[ \begin{array}{ccc} 1 & 0 & -1 \\ 0 & 0 & 1 \\ -1 & 1 & 1 \end{array} \right] \begin{pmatrix} 4 \\ 3 \\ \frac{7}{2} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{7}{2} \\ \frac{5}{2} \end{pmatrix}$ .

O valor da função objetivo:  $f(\hat{\mathbf{x}}) = c_{B_1}\hat{x}_{B_1} + c_{B_2}\hat{x}_{B_2} + c_{B_3}\hat{x}_{B_3} = -2 \times \frac{1}{2} + (-1) \times \frac{7}{2} + 0 \times \frac{5}{2} = -\frac{9}{2}$ .

- *Otimalidade:* primeiro calculamos o vetor multiplicador simplex e, em seguida, os custos relativos.

(i) *multiplicador simplex:* lembre que:  $\mathbf{c}_B = (c_{B_1}, c_{B_2}, c_{B_3}) = (c_1, c_2, c_4) = (-2, -1, 0)$ .

O vetor multiplicador do simplex é dado pelo sistema  $\mathbf{B}^T \lambda = \mathbf{c}_B$  ou, de forma equivalente,

$$\lambda^T = \mathbf{c}_B^T \mathbf{B}^{-1} = [-2 \quad -1 \quad 0] \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 1 \\ -1 & 1 & 1 \end{bmatrix} = (-2 \quad 0 \quad 1).$$

(ii) *cálculo dos custos relativos:*  $N_1 = 3, N_2 = 5$

$$\hat{c}_{N_1} = \hat{c}_3 = c_3 - \lambda^T \mathbf{a}_3 = 0 - (-2 \quad 0 \quad 1) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = 2$$

$$\hat{c}_{N_2} = \hat{c}_5 = c_5 - \lambda^T \mathbf{a}_5 = -0 - (-2 \quad 0 \quad 1) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = -1 \leftarrow k = 2 \quad (x_{N_2} = x_5 \text{ entra na base}).$$

A condição de otimalidade não é verificada e a variável  $x_5$  pode ser aumentada para diminuir a função objetivo por  $f(\mathbf{x}) = f(\hat{\mathbf{x}}) + \hat{c}_{N_k} \hat{\varepsilon} = -\frac{9}{2} - \varepsilon$ .

- *Cálculo da direção simplex:*  $\mathbf{y} = \mathbf{B}^{-1} \mathbf{a}_5 = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 1 \\ -1 & 1 & 1 \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}$ .

- *Tamanho do passo:*

$$\hat{\varepsilon} = \min \left\{ \frac{\hat{x}_{B_2}}{y_2}, \frac{\hat{x}_{B_3}}{y_3} \right\} = \min \left\{ \frac{\frac{7}{2}}{1}, \frac{\frac{5}{2}}{1} \right\} = \frac{5}{2} = \frac{\hat{x}_{B_3}}{y_3} \quad (x_{B_3} = x_4 \text{ sai da base}).$$

Com o valor de  $\varepsilon = \frac{5}{2}$ , a variável  $x_4$  se anula e a variável não-básica  $x_5$  torna-se positiva e, portanto, temos a nova partição básica:

$$(B_1, B_2, B_3) = (1, 2, 5) \quad (N_1, N_2) = (3, 4).$$

Com essa partição, as novas matrizes básica e não-básica são dadas por:

$$\mathbf{B} = [\mathbf{a}_{B_1} \ \mathbf{a}_{B_2} \ \mathbf{a}_{B_3}] = [\mathbf{a}_1 \ \mathbf{a}_2 \ \mathbf{a}_5] = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad \mathbf{N} = [\mathbf{a}_{N_1} \ \mathbf{a}_{N_2}] = [\mathbf{a}_3 \ \mathbf{a}_4] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

que já verificamos, no Exemplo 2.22, que corresponde à solução ótima. ■

### 2.6.3\* O algoritmo simplex

Considere um problema de otimização linear escrito na forma padrão. Resumimos a seguir o procedimento desenvolvido na Seção 2.6.2. Note que o desenvolvimento anterior pressupõe o conhecimento de uma solução básica factível inicial. O procedimento detalhado de como determinar uma solução inicial é chamado *Fase I* e os seus detalhes são apresentados na Seção 2.8. O método simplex propriamente dito é o que chamamos de *Fase II* e seu procedimento é apresentado mais adiante.

*Fase I:*

- Determine inicialmente uma partição básica factível  $\mathbf{A} = [\mathbf{B}, \mathbf{N}]$ . A rigor, precisamos de dois vetores de índices básicos e não-básicos:

$$(B_1, B_2, \dots, B_m) \text{ e } (N_1, N_2, \dots, N_{n-m}).$$

Os vetores das variáveis básicas e não-básicas são, respectivamente:

$$\mathbf{x}_B^T = (x_{B_1} \ x_{B_2} \ \cdots \ x_{B_m}) \text{ e } \mathbf{x}_N^T = (x_{N_1} \ x_{N_2} \ \cdots \ x_{N_{n-m}}).$$

Faça iteração = 1.

*Fase II:*

{*início da iteração simplex*}

Passo 1: {cálculo da solução básica}

$$\begin{cases} \hat{\mathbf{x}}_B = \mathbf{B}^{-1}\mathbf{b} & (\text{equivalentemente, resolva o sistema } \mathbf{Bx}_B = \mathbf{b}) \\ \hat{\mathbf{x}}_N = \mathbf{0} \end{cases}$$

Passo 2: {cálculo dos custos relativos}

2.1) {vetor multiplicador simplex}

$$\boldsymbol{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1} \quad (\text{equivalentemente, resolva o sistema } \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B)$$

2.2) {custos relativos}

$$\hat{c}_{N_j} = c_{N_j} - \boldsymbol{\lambda}^T \mathbf{a}_{N_j} \quad j = 1, 2, \dots, n-m$$

2.3) {determinação da variável a entrar na base}

$$\hat{c}_{N_k} = \min\{\hat{c}_{N_j}, j=1, \dots, n-m\} \quad (\text{a variável } x_{N_k} \text{ entra na base})$$

Passo 3: {teste de optimidade}

Se  $\hat{c}_{N_k} \geq 0$ , então: *pare* {solução na iteração atual é ótima}

Passo 4: {cálculo da direção simplex}

$$\mathbf{y} = \mathbf{B}^{-1} \mathbf{a}_{N_k} \quad (\text{equivalentemente, resolva o sistema: } \mathbf{By} = \mathbf{a}_{N_k})$$

Passo 5: {determinação do passo e variável a sair da base}

Se  $\mathbf{y} \leq \mathbf{0}$ , então: *pare* {problema não tem solução ótima finita:  $f(\mathbf{x}) \rightarrow -\infty$ }

Caso contrário, determine a variável a sair da base pela razão mínima:

$$\hat{\varepsilon} = \frac{\hat{x}_{B_\ell}}{y_\ell} = \min \left\{ \frac{\hat{x}_{B_i}}{y_i} \mid \text{tal que } y_i > 0, i = 1, \dots, m \right\} \quad (\text{a variável } x_{B_\ell} \text{ sai da base})$$

Passo 6: {atualização: nova partição básica, troque a  $\ell$ -ésima coluna de  $\mathbf{B}$  pela  $k$ -ésima coluna de  $\mathbf{N}$ }:

matriz básica nova:  $\mathbf{B} = [\mathbf{a}_{B_1} \ \cdots \ \mathbf{a}_{B_{\ell-1}} \ \mathbf{a}_{N_k} \ \mathbf{a}_{B_{\ell+1}} \ \cdots \ \mathbf{a}_{B_m}]$

matriz não-básica nova:  $\mathbf{N} = [\mathbf{a}_{N_1} \ \cdots \ \mathbf{a}_{N_{k-1}} \ \mathbf{a}_{B_\ell} \ \mathbf{a}_{N_{k+1}} \ \cdots \ \mathbf{a}_{N_{n-m}}]$

iteração = iteração + 1

Retorne ao passo 1

{*fim da iteração simplex*}

### 2.6.4\* Exemplos numéricos e interpretações geométricas

**Exemplo 2.26** Considere o seguinte problema de otimização linear:

$$\begin{aligned} \text{Minimizar } f(x_1, x_2) &= -x_1 - 2x_2 \\ x_1 + x_2 &\leq 6 \\ x_1 - x_2 &\leq 4 \\ -x_1 + x_2 &\leq 4 \\ x_1 \geq 0, x_2 \geq 0. \end{aligned}$$

Introduzindo as variáveis de folga  $x_3, x_4$  e  $x_5$ , temos o problema na forma padrão, cujos coeficientes estão representados na Tabela 2.13.

**Tabela 2.13**  
Dados do problema.

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	<b>b</b>
<b>A</b>	1	1	1	0	0	6
	1	-1	0	1	0	4
	-1	1	0	0	1	4
Min $f$	-1	-2	0	0	0	

A seguir, aplicamos o algoritmo simplex, com todos os passos explicitados a cada iteração.

*Fase I:*

Uma solução básica factível inicial é facilmente obtida, pois os coeficientes das variáveis de folga formam uma matriz identidade que, aliada à não-negatividade dos termos independentes (e se fossem negativos?), fornece uma partição básica factível,

$$(B_1, B_2, B_3) = (3, 4, 5), \quad (N_1, N_2) = (1, 2),$$

ou seja, a matriz básica  $\mathbf{B} = \mathbf{I}$ . Fazendo  $\mathbf{x}_N = (x_1, x_2) = (0, 0)$ , temos trivialmente os valores das variáveis básicas (ver solução inicial da Figura 2.38).

*Fase II:*

**1ª Iteração**

**Tabela 2.14**  
Dados conforme partição na iteração 1.

	<b>Índices</b>					<b>b</b>
	<b>básicos</b>			<b>não-básicos</b>		
	$B_1=3$	$B_2=4$	$B_3=5$	$N_1=1$	$N_2=2$	
<b>[B   N]</b>	1	0	0	1	1	6
	0	1	0	1	-1	4
	0	0	1	-1	1	4
<b>[c_B   c_N]</b>	0	0	0	-1	-2	$f = 0$

- Passo 1: {cálculo da solução básica} =  $\mathbf{x}_B = (x_3, x_4, x_5)$

Resolva o sistema  $\mathbf{Bx}_B = \mathbf{b}$  e obtenha  $\hat{\mathbf{x}}_B = \begin{pmatrix} 6 \\ 4 \\ 4 \end{pmatrix}$ .

Avaliação da função objetivo:  $f(\mathbf{x}) = c_{B_1}x_{B_1} + c_{B_2}x_{B_2} + c_{B_3}x_{B_3} = 0 \times 6 + 0 \times 4 + 0 \times 4 = 0$

- Passo 2: {Cálculo dos custos relativos}

2.1) {vetor multiplicador simplex}:  $(\mathbf{c}_B = (c_{B_1}, c_{B_2}, c_{B_3})) = (c_3, c_4, c_5) = (0, 0, 0)$ .

A solução do sistema  $\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B$  é  $\boldsymbol{\lambda}^T = (0, 0, 0)$ .

2.2) {custos relativos}:  $(N_1 = 1, N_2 = 2)$

$$\hat{c}_1 = c_1 - \boldsymbol{\lambda}^T \mathbf{a}_1 = -1 - (0 \ 0 \ 0) \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} = -1,$$

$$\hat{c}_2 = c_2 - \boldsymbol{\lambda}^T \mathbf{a}_2 = -2 - (0 \ 0 \ 0) \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} = -2 \leftarrow k = 2. \text{ (a variável } x_{N_2} = x_2 \text{ entra na base)}$$

2.3) {determinação da variável a entrar na base}

Como  $\hat{c}_2 = \hat{c}_{N_2} = \min\{\hat{c}_{N_j}, j=1, 2\} = -2 < 0$ , então a variável  $x_2$  entra na base.

- Passo 3: {teste de otimalidade}

Os custos relativos mostram a função objetivo em termos das variáveis não-básicas:  $f(\mathbf{x}) = 0 - 1x_1 - 2x_2$ . Como há custos relativos negativos, a solução atual não é ótima.

- Passo 4: {cálculo da direção simplex}

Resolva o sistema  $\mathbf{By} = \mathbf{a}_2$  e obtenha  $\mathbf{y} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$ .

O vetor  $\mathbf{y}$  mostra como as variáveis básicas são alteradas:  $\mathbf{x}_B = \hat{\mathbf{x}}_B - \mathbf{y}\varepsilon$ , as quais, nesta iteração, são  $x_3, x_4$  e  $x_5$ . As variáveis não-básicas  $x_1$  e  $x_2$  se alteram conforme a estratégia simplex:

$$x_1 = 0 \quad \text{e} \quad x_2 = \varepsilon.$$

Isso mostra como a solução é alterada no plano  $(x_1, x_2)$ , neste caso, sobre o eixo  $x_2$ .

- Passo 5: {determinação do passo e variável a sair da base}

$$\hat{\varepsilon} = \min\left\{\frac{\hat{x}_{B_1}}{y_1}, \frac{\hat{x}_{B_3}}{y_3}\right\} = \min\left\{\frac{6}{1}, \frac{4}{1}\right\} = 4 = \frac{\hat{x}_{B_3}}{y_3}. \text{ (a variável } x_{B_3} = x_5 \text{ sai da base)}$$

Com esse valor de  $\hat{\varepsilon}$ ,  $x_5 = x_{B_3} = 0$  e  $(x_1, x_2) = (0, 4)$  é a nova solução obtida pelo método simplex (veja Figura 2.38).

- Passo 6: {atualização: nova partição básica, troque a  $\ell$ -ésima coluna de  $\mathbf{B}$  pela  $k$ -ésima coluna de  $\mathbf{N}$ }:

$$(B_1, B_2, B_3) = (3, 4, 2) \quad (N_1, N_2) = (1, 5),$$

$$\{\text{novo valor para a função objetivo: } f(\mathbf{x}) = f(\hat{\mathbf{x}}) + \hat{c}_{N_k} \hat{\varepsilon} = 0 - 2 \times 4 = -8\}.$$

**Tabela 2.15**  
**Dados conforme partição na iteração 2.**

	<i>Índices</i>					$b$
	<i>básicos</i>	$B_1=3$	$B_2=4$	$B_3=2$	<i>não-básicos</i>	
$[B \mid N]$						
	1	0	1	1	0	6
	0	1	-1	1	0	4
	0	0	1	-1	1	4
$[c_B \mid c_N]$	0	0	-2	-1	0	$f = -8$

## 2ª Iteração

- Solução básica:  $\mathbf{x}_B = (x_3, x_4, x_2)$

Resolva o sistema  $\mathbf{Bx}_B = \mathbf{b}$  e obtenha  $\hat{\mathbf{x}}_B = \begin{pmatrix} 2 \\ 8 \\ 4 \end{pmatrix}$ .

- Optimalidade:

i) vetor multiplicador simplex:  $((c_{B_1}, c_{B_2}, c_{B_3}) = (c_3, c_4, c_2) = (0, 0, -2))$ .

Resolva o sistema  $\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B$  e obtenha  $\boldsymbol{\lambda} = (0, 0, -2)$ .

ii) custos relativos:  $(N_1 = 1, N_2 = 5)$ ,

$$\hat{c}_1 = c_1 - \boldsymbol{\lambda}^T \mathbf{a}_1 = -1 - (0 \ 0 \ -2) \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} = -3 \leftarrow$$

$$\hat{c}_5 = c_5 - \boldsymbol{\lambda}^T \mathbf{a}_5 = 0 - (0 \ 0 \ -2) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 2.$$

Isto fornece:  $f(\mathbf{x}) = -8 - 3x_1 + 2x_2$ .

Como  $\hat{c}_1 = \hat{c}_{N_1} = \min\{\hat{c}_{N_j}, j=1, 2\} < 0$ , então a solução básica não é ótima e a variável  $x_1$  entra na base.

- Direção simplex:  $\mathbf{y} = (y_1, y_2, y_3)$ .

Resolva o sistema  $\mathbf{By} = \mathbf{a}_1$  e obtenha:  $\mathbf{y} = \begin{pmatrix} 2 \\ 0 \\ -1 \end{pmatrix}$ .

As variáveis básicas são alteradas:  $\mathbf{x}_B = \hat{\mathbf{x}}_B - \mathbf{y}\varepsilon$ , as quais, nesta iteração, são  $x_3, x_4$  e  $x_2$ . As variáveis não-básicas  $x_1$  e  $x_5$  se alteram conforme a estratégia simplex:

$$x_1 = \varepsilon \quad \text{e} \quad x_5 = 0.$$

Isso mostra como a solução é alterada no plano  $(x_1, x_2)$ , neste caso, sobre a reta  $x_5 = 0$  ou, de outra forma, sobre a reta  $x_1 + x_2 = 4$ .

- Tamanho do passo: (apenas  $y_1 > 0$ )

$$\hat{\varepsilon} = \min\left\{\frac{\hat{x}_{B_1}}{y_1}\right\} = \left\{\frac{2}{2}\right\} = 1 = \frac{\hat{x}_{B_1}}{y_1}.$$

Com este valor de  $\varepsilon$ , segue que  $x_1 = 1$  (alteração da variável não-básica) e  $x_2 = \hat{x}_{B_3} - y_3\varepsilon = 4 + \varepsilon = 5$  (alteração da variável básica) (ver Figura 2.38).

- Atualização:

$$(B_1, B_2, B_3) = (1, 4, 2) \quad (N_1, N_2) = (3, 5)$$

O novo valor para a função objetivo:  $f(\mathbf{x}) = f(\hat{\mathbf{x}}) + \hat{c}_{N_k} \hat{\varepsilon} = -8 - 3 \times 1 = -11$ .

### 3ª Iteração

**Tabela 2.16**  
**Dados conforme partição na iteração 3.**

		<i>Índices</i>				<b>b</b>
		<i>básicos</i>	<i>não-básicos</i>	<i>N<sub>1</sub></i> =3	<i>N<sub>2</sub></i> =5	
		<i>B<sub>1</sub></i> =1	<i>B<sub>2</sub></i> =4	<i>B<sub>3</sub></i> =2		
[ <b>B</b>   <b>N</b> ]	1	0	1	1	0	6
	1	1	-1	0	0	4
	-1	0	1	0	1	4
[ <b>c<sub>B</sub></b>   <b>c<sub>N</sub></b> ]	-1	0	-2	0	0	$f = -11$

- Solução básica:  $\mathbf{x}_B = (x_1, x_4, x_2)$

Resolva o sistema  $\mathbf{B}\mathbf{x}_B = \mathbf{b}$  e obtenha  $\hat{\mathbf{x}}_B = \begin{pmatrix} 1 \\ 8 \\ 5 \end{pmatrix}$ .

- Optimalidade:

(i) vetor multiplicador simplex:  $((c_{B_1}, c_{B_2}, c_{B_3}) = (c_1, c_4, c_2) = (-1, 0, -2))$ .

Resolva o sistema  $\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B$  e obtenha  $\boldsymbol{\lambda}^T = (-\frac{3}{2}, 0, -\frac{1}{2})$ .

(ii) custos relativos: ( $N_1 = 3, N_2 = 5$ ),

$$\hat{c}_3 = c_3 - \boldsymbol{\lambda}^T \mathbf{a}_3 = 0 - \left(-\frac{3}{2} \quad 0 \quad -\frac{1}{2}\right) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \frac{3}{2},$$

$$\hat{c}_5 = c_5 - \boldsymbol{\lambda}^T \mathbf{a}_5 = 0 - \left(-\frac{3}{2} \quad 0 \quad -\frac{1}{2}\right) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \frac{1}{2}.$$

Como  $\min\{\hat{c}_{N_j}, j=1,2\} = \frac{1}{2} > 0$ , segue-se que a solução atual,

$$\hat{\mathbf{x}}_B = \begin{pmatrix} x_1 \\ x_4 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 8 \\ 5 \end{pmatrix} \quad \text{e} \quad \hat{\mathbf{x}}_N = \begin{pmatrix} x_3 \\ x_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

ou

$$\hat{\mathbf{x}} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 5 \\ 0 \\ 8 \\ 0 \end{pmatrix},$$

é ótima.

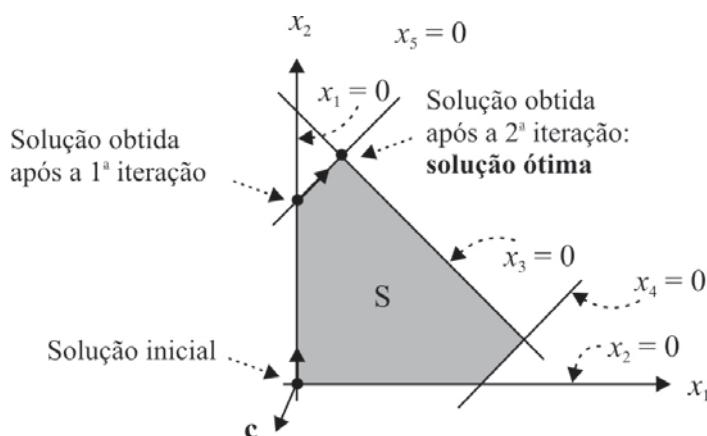
A função objetivo, escrita em termos das variáveis não-básicas, é  $f(\mathbf{x}) = f(\hat{\mathbf{x}}) + \hat{c}_{N_1}x_{N_1} + \hat{c}_{N_2}x_{N_2} = -11 + \frac{3}{2}x_3 + \frac{1}{2}x_5 \geq -11$ , para todo  $x_3 \geq 0$  e  $x_5 \geq 0$ . Além disso, qualquer outra solução factível diferente da solução básica na última iteração (em que  $x_3 = 0$  e  $x_5 = 0$ ), deve ter  $x_3 > 0$  ou  $x_5 > 0$ . Como os custos relativos das variáveis não-básicas são positivos (poderiam ser nulos na optimalidade), segue-se que:  $f(\mathbf{x}) > -11$ , para toda solução factível diferente da solução básica e, portanto, a solução ótima obtida é única. O próximo exemplo ilustra o caso com custos relativos nulos na optimalidade, condição necessária para a ocorrência de múltiplas soluções ótimas. ■

A Figura 2.38 ilustra a trajetória percorrida pelo método simplex. Uma trajetória diferente poderia ter sido percorrida se, em lugar de escolhermos  $k = 2$  na primeira iteração (segunda variável não-básica entra na base), tivéssemos escolhido  $k = 1$  (primeira variável não-básica entra na base). Essa possibilidade é deixada como exercício ao leitor.

**Exemplo 2.27** (*soluções ótimas múltiplas*) Considere o programa linear do Exemplo 2.26, com uma ligeira alteração na função objetivo:

$$\begin{aligned} \text{Minimizar } & f(x_1, x_2) = -x_1 - x_2 \\ & x_1 + x_2 \leq 6 \\ & x_1 - x_2 \leq 4 \\ & -x_1 + x_2 \leq 4 \\ & x_1 \geq 0, x_2 \geq 0. \end{aligned}$$

À semelhança do Exemplo 2.26, o leitor pode aplicar o método simplex e encontrar, na terceira iteração, a seguinte base ótima.



**Figura 2.38** Trajetória do método simplex para Exemplo 2.26.

**3<sup>a</sup> Iteração**  $(B_1, B_2, B_3) = (2, 1, 5)$ ,  $(N_1, N_2) = (4, 3)$ .

- Solução básica:  $\mathbf{x}_B = (x_2, x_1, x_5)$

Resolva o sistema  $\mathbf{Bx}_B = \mathbf{b}$ , cuja matriz aumentada é dada por: 
$$\left[ \begin{array}{ccc|c} 1 & 1 & 0 & 6 \\ -1 & 1 & 0 & 4 \\ 1 & -1 & 1 & 4 \end{array} \right]$$
, que pode ser resolvido pelo método de eliminação de Gauss, cuja solução é  $\hat{\mathbf{x}}_B = \begin{pmatrix} 1 \\ 5 \\ 8 \end{pmatrix}$  e a função objetivo vale  $f(\hat{\mathbf{x}}) = -6$ .

- Optimalidade:

i) vetor multiplicador simplex:  $((c_{B_1}, c_{B_2}, c_{B_3}) = (c_2, c_1, c_5) = (-1, -1, 0))$ .

Resolva o sistema  $\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B$ , cuja matriz aumentada é dada por 
$$\left[ \begin{array}{ccc|c} 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{array} \right]$$
, e cuja solução é  $\boldsymbol{\lambda}^T = (-1 \ 0 \ 0)$ .

ii) custos relativos:  $(N_1 = 4, N_2 = 3)$

$$\begin{aligned} \hat{c}_4 &= c_4 - \boldsymbol{\lambda}^T \mathbf{a}_4 = 0 \\ \hat{c}_3 &= c_3 - \boldsymbol{\lambda}^T \mathbf{a}_3 = 1. \end{aligned}$$

Como  $\hat{c}_j \geq 0$  para todas variáveis não-básicas, segue-se que a solução atual

$$\hat{\mathbf{x}}_B = \begin{pmatrix} x_2 \\ x_1 \\ x_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 5 \\ 8 \end{pmatrix} \quad \text{e} \quad \hat{\mathbf{x}}_N = \begin{pmatrix} x_4 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

ou

$$\hat{\mathbf{x}} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 5 \\ 1 \\ 0 \\ 0 \\ 8 \end{pmatrix},$$

é ótima.

A optimalidade na última iteração decorreu do fato de a função objetivo, em termos das variáveis não-básicas, ser  $f(\mathbf{x}) = -6 + 0x_4 + x_3 \geq -6$ , para todo  $x_4 \geq 0$  e  $x_3 \geq 0$ . Entretanto,  $f(\mathbf{x}) = -6$ , para todo  $x_4 > 0$  e  $x_3 = 0$ , ou seja, a solução básica pode ser alterada com valores não-nulos para  $x_4$  sem que a função objetivo se altere. Portanto, o problema tem múltiplas soluções ótimas, as quais podem ser determinadas pela atribuição de valores diferentes a  $x_4$ . ■

Se a solução básica ótima no Exemplo 2.27 fosse degenerada (alguma variável básica nula), então, com o aumento de  $x_4$ , poderia ocorrer que uma variável básica nula ficasse negativa, de modo que  $x_4$  não poderia assumir valores diferentes de zero. Isto é, suponha que a  $i$ -ésima variável básica seja nula, ou seja,  $\hat{x}_{B_i} = 0$ , então  $x_{B_i} = \hat{x}_{B_i} - y_i \varepsilon = -y_i \varepsilon$ . Portanto, se  $y_i > 0$  e caso  $\varepsilon > 0$ , então  $x_{B_i} < 0$  e, portanto, a solução se torna infactível. Nesse caso, o problema não apresenta múltiplas soluções ótimas, apesar de haver um custo relativo nulo na optimalidade (ver Exercício 2.24).

Em resumo, se na optimalidade os custos relativos são todos positivos, podemos afirmar que o problema tem solução ótima única. A existência de custos relativos nulos é condição necessária para múltiplas soluções ótimas, mas não é condição suficiente.

**Exemplo 2.28** (problema sem solução ótima – solução ilimitada) Considere o programa linear:

$$\begin{aligned} \text{Minimizar } f(x_1, x_2) &= -x_1 - x_2 \\ x_1 - x_2 &\leq 4 \\ -x_1 + x_2 &\leq 4 \\ x_1 \geq 0, x_2 &\geq 0. \end{aligned}$$

Introduzindo as variáveis de folga  $x_3$  e  $x_4$ , temos o problema na forma padrão, cujos coeficientes estão representados na Tabela 2.17.

Procedendo como no Exemplo 2.26, a partir da partição básica inicial:

$$(B_1, B_2) = (3, 4), \quad (N_1, N_2) = (1, 2).$$

Na segunda iteração do método simplex, obtemos:

$$\text{2ª Iteração} \quad (B_1, B_2) = (1, 4), \quad (N_1, N_2) = (3, 2).$$

- Solução básica:  $\mathbf{x}_B = (x_1, x_4)^T$ .

Resolva o sistema  $\mathbf{Bx}_B = \mathbf{b}$ , cuja matriz aumentada é  $\left[ \begin{array}{cc|c} 1 & 0 & 4 \\ -1 & 1 & 4 \end{array} \right]$ , sua solução é  $\hat{\mathbf{x}}_B = \begin{pmatrix} 4 \\ 8 \end{pmatrix}$  e

a função objetivo é  $f(\hat{\mathbf{x}}) = c_{B_1} \hat{x}_{B_1} + c_{B_2} \hat{x}_{B_2} = -1 \times 4 + 0 \times 8 = -4$

- Optimalidade:

i) vetor multiplicador simplex:  $(\mathbf{c}_B = (c_{B_1}, c_{B_2})^T = (c_1, c_4) = (-1, 0))$

Resolva o sistema  $\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B$ , cuja matriz aumentada é  $\left[ \begin{array}{cc|c} 1 & 0 & -1 \\ -1 & 1 & 0 \end{array} \right]$  e obtenha  $\boldsymbol{\lambda} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$ .

ii) custos relativos:  $(N_1 = 3, N_2 = 2)$

$$\hat{c}_3 = c_3 - \boldsymbol{\lambda}^T \mathbf{a}_3 = 1$$

$$\hat{c}_2 = c_2 - \boldsymbol{\lambda}^T \mathbf{a}_2 = -1 \leftarrow k=2 \text{ } (x_2 \text{ entra na base}).$$

A função objetivo em termos das variáveis não-básicas é  $f(\mathbf{x}) = 0 + 1x_3 - 1x_2$ .

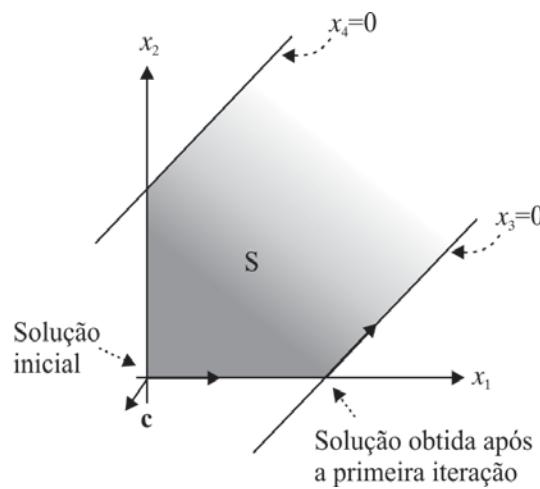
- Direção simplex

Resolva o sistema  $\mathbf{By} = \mathbf{a}_2$ , cuja matriz aumentada é  $\left[ \begin{array}{cc|c} 1 & 0 & -1 \\ -1 & 1 & 1 \end{array} \right]$  e obtenha  $\mathbf{y} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$ .

Temos então que, se aumentamos o valor da variável  $x_2$ , a função objetivo decresce (custo relativo negativo). Note que  $x_2$  pode crescer indefinidamente, já que a direção simplex não tem componentes positivos (direções desse tipo são chamadas de raios da região factível). Assim, o problema não tem solução ótima. A Figura 2.39 ilustra o problema. ■

**Tabela 2.17**  
Dados do problema do Exemplo 2.28.

	$x_1$	$x_2$	$x_3$	$x_4$	<b>b</b>
<b>A</b>	1	-1	1	0	4
	-1	1	0	1	4
Min $f$	-1	-1	0	0	



**Figura 2.39** Solução gráfica do Exemplo 2.28.

### 2.6.5\* Considerações sobre implementações do método simplex

O método simplex tem sido utilizado com sucesso na resolução de problemas práticos, os quais podem envolver dezenas, centenas ou milhares de variáveis e restrições. Com os computadores disponíveis hoje, implementações razoavelmente simples do método simplex podem resolver com sucesso problemas com um número relativamente grande de variáveis e restrições. Por exemplo, uma implementação simples pode trabalhar explicitamente com a inversa da matriz básica  $\mathbf{B}^{-1}$  e, como apenas uma coluna básica é alterada por iteração, é econômico atualizar a inversa da base para a iteração seguinte. Este método é conhecido na literatura como *método simplex revisado*. O Exercício 2.26 mostra como atualizar a base e os principais vetores envolvidos nas iterações do método simplex.

O método simplex revisado tem duas limitações importantes. A primeira é que, para atualizar a inversa da base de uma iteração para outra, são necessárias operações de pivotamento (eliminação de Gauss), que fazem acumular erros de arredondamento devido à aritmética do ponto flutuante utilizada pelos computadores (o leitor deve estar ciente de que toda operação aritmética no computador não está livre de erros de arredondamento). Dessa forma, uma coluna que entrou e saiu da base pode contribuir com erros de arredondamento em  $\mathbf{B}^{-1}$ , quando não deveria, já que não é mais básica. Para contornar essa dificuldade, pode-se incluir uma *reinversão* periódica da matriz básica. Há também outras formas de armazenar a matriz  $\mathbf{B}^{-1}$ , por exemplo, a chamada *forma produto da inversa*, que consiste em guardar uma seqüência de vetores que representam as operações elementares feitas a cada iteração do simplex (veja matriz  $\mathbf{E}$  no Exercício 2.25). A forma produto permite alguns variantes na eliminação de Gauss, como a escolha do elemento pivô maior em módulo, que faz com que os erros de arredondamento sejam, em geral, mais bem comportados.

A segunda limitação de se trabalhar com a inversa da base é que, normalmente,  $\mathbf{B}^{-1}$  inclui muitos elementos não-nulos, enquanto  $\mathbf{B}$  pode ser esparsa (poucos elementos não-nulos). Para problemas grandes, tipicamente bastante esparsos, este procedimento pode se tornar inviável computacionalmente por exigir mais tempo para resolver um problema, além da quantidade de memória requerida ultrapassar os limites disponíveis. Para contornar essa dificuldade, é possível utilizar estruturas de dados mais adequadas, que armazenem apenas elementos não-nulos, e métodos de resolução dos sistemas lineares que explorem a esparsidade das matrizes<sup>19</sup> envolvidas nas iterações do método simplex.

19 Maros (2003) detalha várias estruturas de dados para implementações eficientes do método simplex.

Uma outra forma de apresentação e implementação do método simplex, chamada *método simplex em tabelas*, está detalhada na próxima seção. Essa forma é adequada para resolver problemas com dezenas de restrições e variáveis. É bastante popular devido à facilidade de se trabalhar com exemplos pequenos, como os resolvidos na Seção 2.6.4. O método simplex em tabelas utiliza implicitamente a inversa da matriz básica e calcula todas as direções simplex por iteração (poucas, no caso de exemplos pequenos), embora apenas uma delas seja utilizada pela variável que entra na base. Entretanto, existem vantagens em se fazer isso, pois as direções simplex da iteração seguinte são mais fáceis de se calcular.

O método simplex pode ser estendido para lidar com limitantes superiores para as variáveis, isto é,  $x_j \leq u_j$ , assim como lida com limitantes inferiores (não-negatividade,  $x_i \geq 0$ ) e é conhecido como *método simplex para variáveis canalizadas*. Para isso, basta estender o conceito de solução básica, em que as variáveis não-básicas estão fixadas nos limites inferiores ou superiores, isto é, para cada  $j \in N$ , então  $x_j = 0$  ou  $x_j = u_j$ , e as variáveis básicas são calculadas segundo a solução geral (2.31). Se as variáveis básicas assim calculadas têm seus valores dentro dos limites definidos, então a solução é básica factível e corresponde a um vértice da região factível. A estratégia simplex também pode ser estendida para permitir o aumento de uma variável não-básica que esteja no limite inferior (como na forma padrão), ou a diminuição de uma variável não-básica que esteja no limite superior, isto é, para cada índice  $k \in N$ , então  $x_k = \varepsilon$ , se  $\hat{c}_k < 0$  (como na forma padrão), ou  $x_k = u_k - \varepsilon$ , se  $\hat{c}_k > 0$ , com  $\varepsilon \geq 0$ , dependendo se a variável não-básica  $x_k$  está no limite inferior ou superior. O tamanho do passo  $\varepsilon$  deve levar em conta os limitantes inferiores e superiores das variáveis básicas e da variável não-básica  $x_k$  escolhida para entrar na base. As condições de optimização também são estendidas: se, para todo  $j \in N$ ,  $\hat{c}_j \geq 0$ ; se  $x_j = 0$ , e  $\hat{c}_j \leq 0$  se  $x_j = u_j$ , então a solução básica é ótima.<sup>20</sup>

## 2.7 MÉTODO SIMPLEX EM TABELAS<sup>21</sup>

As operações realizadas no método simplex podem ser organizadas em tabelas, chamadas *tabelas simplex*. Embora não seja a forma ideal de se implementar o método simplex em um computador, essa organização pode ser interessante para manipular problemas pequenos e compreender mais rapidamente como o método funciona.

Considere um problema de otimização linear:

$$\begin{aligned} \text{Minimizar } f(\mathbf{x}) &= \mathbf{c}\mathbf{x} \\ \mathbf{Ax} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0}. \end{aligned}$$

Os coeficientes do sistema de equações lineares e da função objetivo são suficientes para descrever o problema e podem ser organizados da seguinte maneira (Tabela 2.18):

**Tabela 2.18**  
Coeficientes de um problema de otimização linear.

$x_1$	$x_2$	$\dots$	$x_n$	←	variáveis
$c_1$	$c_2$	$\dots$	$c_n$	$f$	← coeficientes da função objetivo
$a_1$	$a_2$	$\dots$	$a_n$	$b$	← coeficientes das restrições

<sup>20</sup> Para mais detalhes destes procedimentos, o leitor pode consultar Bazaraa *et al.* (1990).

<sup>21</sup> Nesta seção, o método simplex é apresentado de forma mais resumida e alternativa às Seções 2.6.2–2.6.4.

em que  $\mathbf{a}_j$  é a  $j$ -ésima coluna da matriz  $\mathbf{A}$ . Usamos um exemplo simples para introduzir as tabelas simplex. Observe o problema de otimização linear do exemplo 2.26:

$$\begin{aligned} \text{Minimizar } f(x_1, x_2) &= -x_1 - 2x_2 \\ x_1 + x_2 &\leq 6 \\ x_1 - x_2 &\leq 4 \\ -x_1 + x_2 &\leq 4 \\ x_1 \geq 0, x_2 \geq 0. \end{aligned}$$

Com a introdução das variáveis de folga  $x_3$ ,  $x_4$  e  $x_5$  (ver Seção 2.4.2), os coeficientes do sistema de equações lineares  $\mathbf{Ax} = \mathbf{b}$  e da função objetivo  $f(\mathbf{x}) = \mathbf{c}^T \mathbf{x}$  estão apresentados na Tabela 2.19.

As colunas das variáveis  $x_3$ ,  $x_4$ ,  $x_5$  formam uma matriz identidade. Elas são chamadas de variáveis básicas. Em uma tabela simplex, a matriz dos coeficientes das variáveis básicas é sempre a matriz identidade. Isso significa que cada variável básica está expressa em termos das demais variáveis, no caso,  $x_1$ ,  $x_2$ , as quais são chamadas de variáveis não-básicas. Assim, basta atribuir valores às variáveis não-básicas  $x_1$ ,  $x_2$  para que tenhamos uma solução que satisfaça o sistema  $\mathbf{Ax} = \mathbf{b}$ . Escrevemos os comentários anteriores em termos matemáticos, usando o exemplo numérico (verifique isso na Tabela 2.19):

$$\begin{aligned} x_3 &= b_1 - a_{11}x_1 - a_{12}x_2 = 6 - x_1 - x_2 \\ x_4 &= b_2 - a_{21}x_1 - a_{22}x_2 = 4 - x_1 + x_2 \\ x_5 &= b_3 - a_{31}x_1 - a_{32}x_2 = 4 + x_1 - x_2. \end{aligned} \tag{2.40}$$

Se fixarmos, por exemplo, as variáveis não-básicas  $x_1$ ,  $x_2$  em seus limites  $x_1 = 0$ ,  $x_2 = 0$  (lembre-se de que  $x_1 \geq 0$ ,  $x_2 \geq 0$ , portanto, zero é o limite inferior), então as demais variáveis têm os valores  $x_3 = 6$ ,  $x_4 = 4$ ,  $x_5 = 4$  e produzem uma solução factível para o problema. Esta é uma solução básica factível e corresponde a um vértice da região factível (ver Figura 2.38). O valor da função objetivo para esta solução é  $f = 0$ . Como as variáveis não-básicas  $x_1$  e  $x_2$  estão em seus limites inferiores, a única possibilidade de examinar soluções alternativas consiste em aumentar os valores de  $x_1$ ,  $x_2$  (diminuir os valores de  $x_1$  ou  $x_2$  significa fazê-los negativos, levando a uma solução infactível). A função objetivo está escrita em termos das variáveis não-básicas  $x_1$ ,  $x_2$  (em uma tabela simplex como a da Tabela 2.19, a função objetivo sempre é escrita em termos das variáveis não-básicas):

$$f = -x_1 - 2x_2.$$

Dessa forma, ao examinarmos soluções alternativas (aumentando  $x_1$  ou  $x_2$ ), vemos que a função objetivo decresce e, portanto, a solução básica ( $x_1 = 0$ ,  $x_2 = 0$ ,  $x_3 = 6$ ,  $x_4 = 4$ ,  $x_5 = 4$ ) não é uma solução ótima (para um problema de minimização). Por exemplo, aumentando o valor de  $x_2$  e mantendo o valor de  $x_1 = 0$ , a função objetivo decresce à taxa de variação  $-2$ :

$$f = -2x_2,$$

**Tabela 2.19**  
**Tabela simplex inicial – Dados do problema.**

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	b
<b>-1</b>	<b>-2</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>f</b>
1	1	1	0	0	6
1	-1	0	1	0	4
-1	1	0	0	1	4

e quanto maior o valor de  $x_2$ , menor será o valor de  $f$ . Essa estratégia de alterar apenas uma variável não-básica (neste caso,  $x_2$ ) é chamada *estratégia simplex*. De (2.40), notamos que, se  $x_2$  crescer (mantendo-se  $x_1 = 0$ ), os valores das variáveis básicas ( $x_3, x_4, x_5$ ) são alterados e podem aumentar ou diminuir, de modo que deve-se tomar certo cuidado para preservar a não-negatividade das variáveis (caso contrário, a solução deixa de ser factível). A seguir, escrevemos as expressões (2.40) para  $x_1 = 0$ :

$$\begin{aligned}x_3 &= b_1 - a_{12}x_2 = 6 - x_2 \geq 0 & (a_{12} > 0) \\x_4 &= b_2 - a_{22}x_2 = 4 + x_2 \geq 0 & (a_{22} < 0) \\x_5 &= b_3 - a_{32}x_2 = 4 - x_2 \geq 0 & (a_{32} > 0).\end{aligned}$$

Das desigualdades anteriores, concluímos que apenas as variáveis básicas  $x_3, x_5$  limitam o crescimento de  $x_2$ , isto é,

- como  $a_{12} > 0$ , então  $b_1 - a_{12}x_2 \geq 0$  implica  $x_2 \leq \frac{b_1}{a_{12}}$ , portanto,  $x_2 \leq 6$ ,
- como  $a_{32} > 0$ , então  $b_3 - a_{32}x_2 \geq 0$  implica  $x_2 \leq \frac{b_3}{a_{32}}$ , portanto,  $x_2 \leq 4$ .

Note que, como  $a_{22} < 0$ , então a variável  $x_4$  cresce com  $x_2$ , portanto, não limita o aumento em  $x_2$ .

**Solução ótima ilimitada:** Se ocorresse o caso  $a_{ij} \leq 0$ ,  $i = 1, 2, 3$ , então a variável  $x_2$  poderia crescer indefinidamente e  $f \rightarrow -\infty$ , e assim o problema não teria solução ótima finita.

Portanto, o maior valor possível para  $x_2$  é 4 ( $x_2 = \min\{\frac{b_1}{a_{12}}, \frac{b_3}{a_{32}}\} = \frac{b_3}{a_{32}} = 4$ ) e, com este valor, segue-se que  $x_5 = 4 - x_2 = 0$ , que é a variável na terceira equação em (2.40) que se anula. Isso produz uma nova solução:

variáveis não-básicas:  $x_1 = 0, x_2 = 4$ ,

variáveis básicas:  $x_3 = 2, x_4 = 8, x_5 = 0$ ,

e o valor da função objetivo é  $f = 0 - 2x_2 = -8$ .

Se redefinirmos como variáveis não-básicas aquelas cujos valores são nulos (isto é, estão em seus limites inferiores) e, como variáveis básicas, aquelas com valores positivos (isto é, fora de seus limites), então:

variáveis não-básicas:  $x_1 = 0, x_5 = 0$ ,

variáveis básicas:  $x_3 = 2, x_4 = 8, x_2 = 4$ .

$$f = -8.$$

Dizemos que a variável não-básica  $x_2$  “entrou na base” no lugar da variável básica  $x_5$ , que “saiu da base”. Portanto,  $x_2$  passa a ser a variável básica da terceira equação em (2.40). Note que tínhamos uma solução básica factível melhor, cujo valor da função objetivo valia  $f = 0$ , e agora temos outra solução básica factível, cujo valor da função objetivo vale  $f = -8$ . As colunas das novas variáveis básicas não formam a matriz identidade, portanto, a tabela simplex deve ser *atualizada*, como vemos a seguir. Este procedimento, que parte de uma solução básica factível e encontra outra melhor, consiste em uma iteração do *método simplex*. As operações realizadas em uma iteração do método simplex são descritas a seguir. Dada uma tabela simplex:

- Encontre uma variável não-básica que tenha o coeficiente negativo na função objetivo, digamos,  $x_k$  ( $x_k$  entra na base).
- Percorra a coluna na tabela simplex da variável  $x_k$  e, para cada coeficiente positivo ( $a_{ik} > 0$ ), calcule a razão  $\frac{b_i}{a_{ik}}$  (os valores que anulam a variável básica na linha  $i$ ) e determine  $\frac{b_\ell}{a_{\ell k}} = \min\{\frac{b_i}{a_{ik}} \mid a_{ik} > 0, i = 1, \dots, m\}$ . Com  $x_k = \frac{b_\ell}{a_{\ell k}}$ , a variável básica na linha  $\ell$  se anula (isto é, a variável básica da linha  $\ell$  sai da base). Se  $a_{ik} \leq 0, i = 1, \dots, m$ , então  $f \rightarrow -\infty$  e o problema não tem solução ótima finita. Nesse caso, pare.
- Redefina as variáveis básicas e não-básicas e reconstrua a tabela simplex para essa nova solução básica (este terceiro passo é explicado a seguir).

Para obter a tabela simplex com a redefinição das variáveis básicas e não-básicas, precisamos “pivotar” a tabela anterior para que os coeficientes das variáveis básicas formem a matriz identidade, e a função objetivo esteja em termos das novas variáveis não-básicas (pivotar significa fazer operações do tipo: multiplicar uma equação por um número e adicionar o resultado a outra – esta operação também é chamada de *eliminação de Gauss*, e não altera a solução do sistema. Veja Apêndice A). Analisando a Tabela 2.19 (tabela simplex inicial), notamos que a coluna da nova variável básica  $x_2$  deve ser transformada na terceira coluna da matriz identidade (a variável  $x_5$  é básica na terceira equação).

Reproduzimos a seguir a tabela simplex inicial (Tabela 2.20), exibindo as variáveis básicas à esquerda da tabela (VB) e qual variável não-básica entra na base e qual variável básica sai da base. As colunas das variáveis  $x_3, x_4, x_5$  formam uma matriz identidade, e dizemos que  $x_3$  é a variável básica na primeira equação,  $x_4$  na segunda e  $x_5$  na terceira.

Tomando o elemento na posição (3,2) da Tabela 2.20 como pivô, efetuamos operações elementares e restauramos a tabela simplex com as novas variáveis básicas em que  $x_2$  é a nova variável básica na terceira equação (Tabela 2.21). Note a matriz identidade nas novas colunas de  $x_3, x_4, x_2$ .

A linha da tabela simplex da Iteração 1 (Tabela 2.21) relativa à função objetivo, significa  $f + 8 = -3x_1 + 2x_5$ , ou, equivalentemente,  $f = -8 - 3x_1 + 2x_5$ . Os coeficientes das variáveis não-básicas na função objetivo são chamados custos relativos. Vemos que aumentar o valor da variável  $x_1$ , mantendo-se  $x_5 = 0$ , faz decrescer a função objetivo à taxa de  $-3$  (Tabela 2.21). As equações do sistema nos dizem (com  $x_5 = 0$ ):

$$\begin{aligned} x_3 &= 2 - 2x_1 \geq 0 \\ x_4 &= 8 + 0x_1 \geq 0 \\ x_5 &= 4 + x_1 \geq 0. \end{aligned}$$

**Tabela 2.20**  
**Tabela simplex inicial – VB:  $x_3, x_4, x_5$ .**

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	<b>b</b>
VB	-1	-2 ↓	0	0	0	$f$
$x_3$	1	1	1	0	0	6
$x_4$	1	-1	0	1	0	4
$\leftarrow x_5$	-1	1	0	0	1	4

**Tabela 2.21**  
**Tabela simplex iteração 1 – VB:  $x_3, x_4, x_2$ .**

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	<b>b</b>
VB	-3 ↓	0	0	0	2	$f+8$
$\leftarrow x_3$	2	0	1	0	-1	2
$x_4$	0	0	0	1	1	8
$x_2$	-1	1	0	0	1	4

portanto, o valor máximo para  $x_1$  é dado por:  $\frac{b_1}{a_{11}} = \frac{2}{2} = 1$ . Com este valor para  $x_1$ , a variável básica na primeira equação  $x_3 = 2 - 2x_1 = 0$ , isto é, a variável básica  $x_3$  se anula, enquanto a variável não-básica  $x_1$  torna-se positiva. Isso fornece uma nova solução básica:

$$\text{variáveis não-básicas: } x_1 = 1, x_5 = 0$$

$$\text{variáveis básicas: } x_3 = 0, x_4 = 8, x_2 = 4.$$

Novamente, redefinimos as variáveis não básicas como aquelas cujos valores são zero (isto é, estão em seus limites) e as variáveis básicas como aquelas cujos valores são positivos, então:

$$\text{variáveis não-básicas: } x_3 = 0, x_5 = 0,$$

$$\text{variáveis básicas: } x_1 = 1, x_4 = 8, x_2 = 5.$$

Como a variável  $x_1$  entrou na base (primeira coluna) e a variável  $x_3$  saiu da base (primeira equação), podemos reescrever a nova tabela simplex pivotando sobre o elemento (1,1) da Tabela 2.21, transformando a coluna  $x_1$  na primeira coluna da matriz identidade e  $x_1$  é a variável básica na primeira equação.

A tabela simplex iteração 2 (Tabela 2.22) significa:

$$\begin{aligned} f &= -11 + \frac{3}{2}x_3 + \frac{3}{2}x_5 \\ x_1 &= 1 - \frac{1}{2}x_3 + \frac{1}{2}x_5 \\ x_4 &= 8 + 0x_3 - x_5 \\ x_2 &= 5 - \frac{1}{2}x_3 - \frac{1}{2}x_5 \end{aligned}$$

e com  $(x_3, x_5) = (0, 0)$ , então  $(x_1, x_4, x_2) = (1, 8, 5)$  (solução básica) e  $f = -11$ . Como qualquer outra solução factível é obtida atribuindo-se valores positivos a  $x_3$  ou  $x_5$  e, como  $x_3$  e  $x_5$  têm custos relativos positivos (Tabela 2.22), a função objetivo cresce, ou seja,  $f(\mathbf{x}) \geq -11$  para qualquer solução factível  $\mathbf{x}$ , o que significa que a solução básica atual é ótima. Quando todos os custos relativos são não-negativos (maiores ou iguais a zero), dizemos que a condição de otimalidade foi verificada.

### O algoritmo simplex (em tabelas)

Considere um problema de otimização linear escrito na forma padrão.

*Fase I:* Determine uma tabela simplex inicial, isto é,

- a matriz dos coeficientes contém uma matriz identidade  $m \times m$  ( $m$  é o número de equações) e o vetor independente  $\mathbf{b} \geq \mathbf{0}$ ;
- a função objetivo é escrita em termos das variáveis não-básicas, isto é, os coeficientes das variáveis básicas são nulos.

**Tabela 2.22**  
**Tabela simplex iteração 2 – VB:  $x_1, x_4, x_2$ .**

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$\mathbf{b}$
VB	0	0	$\frac{3}{2}$	0	$\frac{3}{2}$	$f+11$
$x_1$	1	0	$\frac{1}{2}$	0	$-\frac{1}{2}$	1
$x_4$	0	0	0	1	1	8
$x_2$	0	1	$\frac{1}{2}$	0	$\frac{1}{2}$	5

Faça iteração = 0.

*Fase II:*

1. Determine o menor dos custos relativos:  $c_k = \min \{c_j \text{ para toda variável não-básica}\}$ .
2. Se  $c_k \geq 0$ , então pare (a solução básica na iteração é ótima). Se não, a variável  $x_k$  entra na base.
3. Se  $a_{ik} \leq 0$ ,  $i = 1, \dots, m$ , então pare (não existe solução ótima finita). Se não, determine:  $\frac{b_\ell}{a_{ik}} = \min \left\{ \frac{b_i}{a_{ik}} \mid a_{ik} > 0, i=1, \dots, m \right\}$  (a variável básica da linha  $\ell$  sai da base).
4. Atualize a tabela simplex (pivotamento no elemento  $(\ell, k)$ ). A variável  $x_k$  passa a ser a variável básica na linha  $\ell$ . Faça iteração = iteração + 1 e retorne ao passo 1.

**Exemplo 2.29** (solução ilimitada) Considere o problema de otimização linear do Exemplo 2.28:

$$\begin{aligned} \text{Minimizar } f(x_1, x_2) &= -x_1 - x_2 \\ x_1 - x_2 &\leq 4 \\ -x_1 + x_2 &\leq 4 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Introduzindo as variáveis de folga, temos uma matriz identidade formada pelas colunas das variáveis  $x_3$  e  $x_4$ , e como o vetor de recursos  $\mathbf{b}$  é não-negativo, temos uma tabela simplex inicial (Tabela 2.23).

A variável  $x_1$  entra na base e a variável  $x_3$  sai da base, resultando na Tabela 2.26.

Como todos os coeficientes na coluna de  $x_2$  são menores ou iguais a zero (ver Tabela 2.24), então a variável  $x_2$  pode crescer indefinidamente sem que nenhuma variável básica se torne negativa, ou seja,  $f \rightarrow -\infty$ , e o problema não tem solução ótima finita. ■

**Tabela 2.23**  
**Tabela simplex inicial – VB:  $x_3, x_4$ .**

	$x_1$	$x_2$	$x_3$	$x_4$	$\mathbf{b}$
VB	-1↓	-1	0	0	$f$
$\leftarrow x_3$	1	-1	1	0	4
$x_4$	-1	1	0	1	4

**Tabela 2.24**  
**Tabela simplex iteração 1 – VB:  $x_1, x_4$ .**

	$x_1$	$x_2$	$x_3$	$x_4$	$\mathbf{b}$
VB	0	-2↓	1	0	$f+4$
$x_1$	1	-1	1	0	4
$x_4$	0	0	1	1	8

## 2.8 DETERMINAÇÃO DE UMA SOLUÇÃO BÁSICA FACTÍVEL INICIAL

Para que o método simplex possa ser aplicado, precisamos de uma solução básica factível inicial (fase I do algoritmo simplex). Algumas classes de problemas de otimização linear oferecem naturalmente essa solução básica factível, por exemplo:

$$\begin{aligned} \text{Minimizar } f(\mathbf{x}) &= \mathbf{c}^T \mathbf{x} \\ \mathbf{Ax} &\leq \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0}, \end{aligned}$$

em que  $\mathbf{b} \geq \mathbf{0}$ .

Após a introdução das variáveis de folga, digamos,  $\mathbf{x}_f$ , temos:

$$\begin{aligned} \text{Minimizar } f(\mathbf{x}) &= \mathbf{c}^T \mathbf{x} \\ \mathbf{Ax} + \mathbf{x}_f &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0}, \mathbf{x}_f \geq \mathbf{0}, \end{aligned}$$

de modo que a matriz dos coeficientes das restrições agora é dada por  $[\mathbf{A} \quad \mathbf{I}]$  e uma partição básica factível é dada por:

- $\mathbf{B} = \mathbf{I}$ : as variáveis básicas são as variáveis de folga  $\mathbf{x}_B = \mathbf{x}_f$ ,
- $\mathbf{N} = \mathbf{A}$ : as variáveis não-básicas são as variáveis originais  $\mathbf{x}_N = \mathbf{x}$ ,

e a solução básica factível é dada por:

$$\begin{cases} \mathbf{x}_B = \mathbf{x}_f = \mathbf{b} \geq \mathbf{0} \\ \mathbf{x}_N = \mathbf{x} = \mathbf{0}. \end{cases}$$

Porém, em geral, uma solução básica factível não está disponível e deve ser encontrada. Considere o problema de otimização linear:

$$\begin{aligned} \text{Minimizar } f(\mathbf{x}) &= \mathbf{c}^T \mathbf{x} \\ \mathbf{Ax} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0}, \end{aligned}$$

em que  $\mathbf{A}$  não tem uma submatriz identidade. Sem perda de generalidade, suponha que  $\mathbf{b} \geq \mathbf{0}$ , pois se  $b_i < 0$ , basta multiplicar a  $i$ -ésima restrição por  $-1$ . A questão é como encontrar uma partição nas colunas da matriz  $\mathbf{A}$ ,  $\mathbf{A} = [\mathbf{B} \quad \mathbf{N}]$ , tal que exista  $\mathbf{B}^{-1}$  e  $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$ , isto é, uma partição básica factível.

Para ter uma idéia dessa dificuldade, suponha, por exemplo, que  $\mathbf{A}$  seja uma matriz  $10 \times 20$  ( $m = 10$  e  $n = 20$ ). Precisamos identificar 10 colunas de  $\mathbf{A}$  que sejam linearmente independentes para formar a matriz  $\mathbf{B}$ , e a solução do sistema  $\mathbf{B} \mathbf{x}_B = \mathbf{b}$  deve satisfazer  $\mathbf{x}_B \geq \mathbf{0}$ .

Podemos fazer uma busca ao acaso, isto é, um algoritmo exaustivo simples do tipo:

- Selecione 10 colunas e resolva o sistema resultante.
- Se a solução do sistema for única e não-negativa, então há sucesso. Podemos usar esta seleção como uma partição básica factível e iniciar o método simplex.
- Se não ( $\mathbf{B}$  não é invertível ou a solução única tem coordenadas negativas), selecione outras 10 colunas até que o sucesso seja obtido ou todas as possibilidades tenham sido investigadas.

Entretanto, esse procedimento simples para determinar uma partição básica inicial pode envolver a resolução de  $C_{10}^{20} = \frac{20!}{10!(20-10)!} = 184.756$  sistemas de equações lineares  $10 \times 10$ . Para problemas grandes, esse procedimento torna-se inaceitável do ponto de vista computacional, embora matematicamente correto, já que o número de combinações é finito.

### Método das duas fases

Notamos que as variáveis de folga foram úteis para a seguinte classe de problemas:

$$\begin{array}{l} \mathbf{Ax} \leq \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{array} \quad \text{equivalente a} \quad \begin{array}{l} \mathbf{Ax} + \mathbf{x}_f = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0}, \mathbf{x}_f \geq \mathbf{0}. \end{array}$$

No entanto, se o problema está na forma de igualdade, não há variáveis de folga. Assim, introduzimos novas variáveis como se fossem de folga:

$$\begin{array}{l} \mathbf{Ax} + \mathbf{y} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}. \end{array}$$

Essas novas variáveis são chamadas *variáveis artificiais* e não fazem parte do problema original, portanto, devem ser eliminadas. Para eliminar as variáveis artificiais, utilizamos um *novo objetivo*, que resulta no *problema artificial*:

$$\begin{aligned} & \text{Minimizar } f_a(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^m y_i \\ & \mathbf{Ax} + \mathbf{y} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}. \end{aligned}$$

A função objetivo do problema artificial penaliza as variáveis artificiais de modo que a solução ótima desse problema deve (se possível) ter  $y_i = 0, i = 1, \dots, m$ . Os coeficientes na função objetivo das variáveis artificiais podem ser quaisquer números positivos (por simplicidade, escolhemos todos iguais a 1). Caso haja colunas da matriz identidade na matriz  $\mathbf{A}$  (geralmente introduzidas por variáveis de folga de algumas restrições de desigualdade), estas podem ser utilizadas na construção da base inicial, sendo complementadas com variáveis artificiais. O exemplo a seguir esclarece essa observação.

**Exemplo 2.30** Considere o seguinte problema de otimização linear:

$$\begin{aligned} & \text{Minimizar } f(\mathbf{x}) = x_1 - x_2 + 2x_3 \\ & x_1 + x_2 + x_3 = 3 \\ & 2x_1 - x_2 + 3x_3 \leq 4 \\ & x_i \geq 0, i = 1, 2, 3 \end{aligned}$$

o qual pode ser escrito na forma padrão:

$$\begin{aligned} & \text{Minimizar } f(\mathbf{x}) = x_1 - x_2 + 2x_3 + 0x_4 \\ & x_1 + x_2 + x_3 = 3 \\ & 2x_1 - x_2 + 3x_3 + x_4 = 4 \\ & x_i \geq 0, i = 1, \dots, 4 \end{aligned}$$

Por exemplo, os seguintes problemas podem ser utilizados como problemas articiais:

**Caso A:** Uma variável artificial é introduzida para cada restrição.

$$\begin{aligned} & \text{Minimizar } f_a(x_1, \dots, x_6) = x_5 + x_6 \\ & x_1 + x_2 + x_3 + x_5 = 3 \\ & 2x_1 - x_2 + 3x_3 + x_4 + x_6 = 4 \\ & x_i \geq 0, i = 1, \dots, 6 \end{aligned}$$

em que  $x_5$  e  $x_6$  são variáveis artificiais (por conveniência, denotamos as variáveis por  $x_5$  e  $x_6$ , em vez de  $y_1$  e  $y_2$ ). Uma base factível é formada pelas colunas das variáveis artificiais,  $x_5$  e  $x_6$ .

**Caso B:** Apenas uma variável artificial é introduzida – uma coluna da matriz identidade já está disponível.

$$\begin{aligned} \text{Minimizar } f_a(x_1, \dots, x_5) &= x_5 \\ x_1 + x_2 + x_3 + x_5 &= 3 \\ 2x_1 - x_2 + 3x_3 + x_4 &= 4 \\ x_i &\geq 0, i = 1, \dots, 5, \end{aligned}$$

em que a variável  $x_5$  é a variável artificial (idem comentário anterior sobre  $x_5$  e  $x_6$ ). Uma base factível é formada pelas colunas das variáveis  $x_4$  (variável de folga) e  $x_5$  (variável artificial). ■

Tanto no caso A como no caso B no Exemplo 2.30, as matrizes básicas são identidades, ou permutações desta, dependendo da ordem em que as variáveis básicas são definidas. Por exemplo, podemos definir, no caso B, as variáveis básicas por  $\mathbf{x}_B = (x_4, x_5)$  ou  $\mathbf{x}_B = (x_5, x_4)$ . O problema artificial tem sempre uma partição básica factível óbvia (por simplicidade de exposição, consideramos que uma variável artificial seja incluída em cada restrição, como no caso A do Exemplo 2.30):

- $\mathbf{B} = \mathbf{I}$  : variáveis básicas  $\mathbf{x}_B = \mathbf{y}$ ,
- $\mathbf{N} = \mathbf{A}$ : variáveis não-básicas  $\mathbf{x}_N = \mathbf{x}$ ,

portanto, podemos aplicar o método simplex, que promove as trocas das colunas das variáveis artificiais por colunas das variáveis originais. Ao final, as variáveis artificiais são não-básicas (nulas, conforme o novo objetivo), e uma base com as colunas originais de  $\mathbf{A}$  é obtida (algumas dificuldades serão discutidas nos exemplos, pois nem sempre essa troca é possível). Se o problema original tem solução factível, digamos,  $\hat{\mathbf{x}}$ , então:

$$\mathbf{A}\hat{\mathbf{x}} = \mathbf{b}$$

$$\hat{\mathbf{x}} \geq \mathbf{0},$$

e esta solução é ótima para o problema artificial, pois o par  $(\hat{\mathbf{x}}, \mathbf{0})$ , ou seja,  $\hat{\mathbf{y}} = \mathbf{0}$ , é factível para o problema artificial:

$$\mathbf{A}\hat{\mathbf{x}} + \hat{\mathbf{y}} = \mathbf{b}$$

$$\hat{\mathbf{x}} \geq \mathbf{0}, \hat{\mathbf{y}} \geq \mathbf{0},$$

e  $f_a(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = 0$ , que é o mínimo possível para o objetivo artificial. Isso mostra que o problema artificial tem, em geral, infinitas soluções ótimas (qualquer solução factível do problema original). Exceções ocorrem quando o problema original tem uma única solução factível ou não tem solução factível.

Uma vez encontrada uma solução básica em que todas as variáveis artificiais são não-básicas, temos uma base formada por colunas originais e, portanto, podemos aplicar o método simplex para resolver o problema original a partir dessa base. Esse procedimento é chamado de método das duas fases: Fase I – resolve o problema artificial e Fase II – resolve o problema original, a partir da base factível obtida na Fase I. Se o problema original é infactível, então o problema artificial tem solução ótima com  $\mathbf{y} \neq \mathbf{0}$ .

**Exemplo 2.31** Considere o problema de otimização linear definido no Exemplo 2.30 e o problema artificial definido no caso B, em que apenas uma variável artificial é introduzida. Problema artificial:

$$\begin{aligned}
 & \text{Minimizar} \quad f_a(x_1, \dots, x_5) = x_5 \\
 & x_1 + x_2 + x_3 + x_5 = 3 \\
 & 2x_1 - x_2 + 3x_3 + x_4 = 4 \\
 & x_i \geq 0, \quad i = 1, \dots, 5
 \end{aligned}$$

Para resolver o problema artificial, aplicamos o método simplex.

*Fase I:* Partição básica factível inicial:  $B_1 = 4$ ,  $B_2 = 5$ ,  $N_1 = 1$ ,  $N_2 = 2$  e  $N_3 = 3$ .

$$\mathbf{B} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{e} \quad \mathbf{N} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & -1 & 3 \end{bmatrix}.$$

### 1<sup>a</sup> Iteração

- *Solução básica:*

Resolva o sistema  $\mathbf{B}\hat{\mathbf{x}}_B = \mathbf{b}$ , cuja matriz aumentada é  $\left[ \begin{array}{cc|c} 0 & 1 & 3 \\ 1 & 0 & 4 \end{array} \right]$  e obtenha a solução:  $\hat{\mathbf{x}}_B = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$ .

- *Teste de otimalidade:*

i) *vetor multiplicador:* Resolva o sistema  $\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B$ , cuja matriz aumentada é  $\left[ \begin{array}{cc|c} 0 & 1 & 0 \\ 1 & 0 & 1 \end{array} \right]$  e ob-tenha  $\boldsymbol{\lambda} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ ,

ii) *custos relativos:*

$$\begin{aligned}
 N_1 = 1: \hat{c}_1 &= c_1 - \boldsymbol{\lambda}^T \mathbf{a}_1 = -1 \leftarrow x_{N_1} = x_1 \text{ entra na base} \\
 N_2 = 2: \hat{c}_2 &= c_2 - \boldsymbol{\lambda}^T \mathbf{a}_2 = -1 \\
 &\vdots \hat{c}_3 = c_3 - \boldsymbol{\lambda}^T \mathbf{a}_3 = -1.
 \end{aligned}$$

- *Direção simplex:* Resolva o sistema  $\mathbf{By} = \mathbf{a}_1$ , cuja matriz aumentada é  $\left[ \begin{array}{cc|c} 0 & 1 & 1 \\ 1 & 0 & 2 \end{array} \right]$  e obtenha  $\mathbf{y} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ .

- *Tamanho do passo:*

$$\varepsilon = \min \left\{ \frac{x_{B_i}}{y_i} \mid y_i > 0 \right\} = \min \left\{ \frac{4}{2}, \frac{3}{1} \right\} = 2 = \frac{x_{B_1}}{y_1} \quad (x_{B_1} = x_4 \text{ sai da base}).$$

- *Atualização*

$$B_1 = 1 \quad B_2 = 5 \quad N_1 = 4 \quad N_2 = 2 \quad N_3 = 3.$$

$$\mathbf{B} = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix} \quad \text{e} \quad \mathbf{N} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & -1 & 3 \end{bmatrix}.$$

**2<sup>a</sup> Iteração:**  $B_1 = 1$ ,  $B_2 = 5$ ,  $N_1 = 4$ ,  $N_2 = 2$  e  $N_3 = 3$ .

- *Solução básica:* Resolva o sistema  $\mathbf{B}\hat{\mathbf{x}}_B = \mathbf{b}$ :  $\left[ \begin{array}{cc|c} 1 & 1 & 3 \\ 2 & 0 & 4 \end{array} \right]$  e obtenha  $\hat{\mathbf{x}}_B = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ .

- *Teste de otimalidade*

i) *vetor multiplicador:* Resolva o sistema  $\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B$ :  $\left[ \begin{array}{cc|c} 1 & 2 & 0 \\ 1 & 0 & 1 \end{array} \right]$  e obtenha  $\boldsymbol{\lambda} = \begin{bmatrix} 1 \\ -1/2 \end{bmatrix}$ ,

ii) custos relativos

$$N_1 = 4 : \hat{c}_4 = c_4 - \lambda^T \mathbf{a}_4 = \frac{1}{2}$$

$$N_2 = 2 : \hat{c}_2 = c_2 - \lambda^T \mathbf{a}_2 = -\frac{3}{2} \leftarrow x_{N_2} = x_2 \text{ entra na base}$$

$$N_3 = 3 : \hat{c}_3 = c_3 - \lambda^T \mathbf{a}_3 = \frac{1}{3}$$

- *Direção simplex:* Resolva o sistema  $\mathbf{By} = \mathbf{a}_2$ : 
$$\begin{array}{cc|c} 1 & 1 & 1 \\ 2 & 0 & -1 \end{array}$$
 e obtenha  $\mathbf{y} = \begin{bmatrix} -\frac{1}{2} \\ \frac{3}{2} \end{bmatrix}$ .

- *Tamanho do passo*

$$\varepsilon = \min \left\{ \frac{x_{B_i}}{y_i} \mid y_i > 0 \right\} = \min \left\{ \frac{1}{\frac{3}{2}}, \frac{2}{3} \right\} = \frac{2}{3} = \frac{x_{B_2}}{y_2} \quad (x_{B_2} = x_5 \text{ sai da base}).$$

- *Atualização:*

$$B_1 = 1, B_2 = 2, N_1 = 4, N_2 = 5 \text{ e } N_3 = 3.$$

$$B = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}.$$

Base formada por variáveis do problema original, a variável artificial  $x_5$  torna-se não básica. Fim da Fase I.

*Fase II:* Aplicar o método simplex a partir da base obtida na Fase I. A variável artificial (segunda variável não-básica,  $N_2 = 5$ ) é descartada e os índices não-básicos são redefinidos:  $N_1 = 4, N_3 = 3$ . ■

### Método do M–grande

Um objetivo alternativo para o problema artificial consiste em considerar a função objetivo original e penalizar exageradamente as variáveis artificiais de modo que estas sejam nulas na solução ótima, caso o problema original tenha solução factível. Seja  $M$  um número suficientemente grande, então um objetivo alternativo para o problema artificial é:

$$\text{Minimizar } f_a(\mathbf{x}, \mathbf{y}) = \mathbf{c}^T \mathbf{x} + M(y_1 + y_2 + \dots + y_m)$$

Este procedimento é chamado método do *M–grande* e, em certo sentido, tem o desempenho similar ao método das duas fases, pois o método simplex (ou qualquer outro método que seja utilizado) deve orientar as trocas de bases de modo a substituir as colunas artificiais por colunas legítimas, já que as variáveis artificiais têm um custo muito alto.

**Exemplo 2.32** Considere o problema de otimização linear no Exemplo 2.30. O método do *M–grande* consiste em resolver o problema (caso B):

$$\begin{aligned} \text{Minimizar } f_a(x_1, \dots, x_5) &= x_1 - x_2 + 2x_3 + 1000x_5 \\ x_1 + x_2 + x_3 + x_5 &= 3 \\ 2x_1 - x_2 + 3x_3 + x_4 &= 4 \\ x_i &\geq 0, i = 1, \dots, 5. \end{aligned}$$

Por conveniência de notação, a variável artificial  $y_1$  foi denotada por  $x_5$ . O valor  $M = 1000$  é arbitrário. O leitor pode, como exercício, resolver esse problema com o método simplex. ■

## 2.9 PROGRAMAÇÃO DE METAS

Nas seções anteriores, estudamos a resolução de problemas de otimização linear que envolvem apenas uma função objetivo. Entretanto, em alguns exemplos apresentados anteriormente, é possível notar que a função objetivo consiste em uma composição de objetivos, como no exemplo da

Seção 1.3 do Capítulo 1, em que se deseja minimizar o custo de produção, dado por  $\sum_{i,t} K_i(x_{it})$ , e o custo de estocagem, dado por  $\sum_{i,t} h_i I_{it}$ . Se otimizarmos somente o custo de produção, possivelmente encontraremos uma solução que acumula a produção dos itens em poucos períodos, de modo a diminuir o custo com as preparações de máquinas, fazendo com que os níveis de estoque sejam altos. Por outro lado, se otimizarmos apenas o custo de estoque, encontraremos uma solução em que a produção dos itens ocorre nos períodos próximos das demandas, diminuindo os níveis de estoque, mas fazendo com que ocorra produção em vários períodos, aumentando o custo com as preparações. Dizemos, então, que esses objetivos são conflitantes e geralmente é impossível determinar uma única solução que otimize simultaneamente ambos os objetivos. Em casos como o desse exemplo, entretanto, um objetivo único é aceitável, já que ambos os objetivos têm a mesma unidade monetária e se deseja minimizar o custo total ou maximizar a contribuição ao lucro.

Em geral, os problemas podem ter objetivos conflitantes medidos em unidades diferentes, por exemplo, a água de um reservatório pode ser destinada à geração de energia elétrica, à irrigação de lavouras, ao abastecimento da população ou à recreação. Faz-se necessário um meio-termo, já que a melhor solução para a empresa de geração de energia elétrica pode comprometer a irrigação de lavouras, o abastecimento e a recreação, e vice-versa. Da mesma forma, o alto lucro desejado por um investidor financeiro é comumente conflitante com o baixo risco também almejado, de modo que deve-se buscar uma solução de compromisso que garanta lucro e risco em níveis aceitáveis. Estes são exemplos de problemas de *otimização multiobjetivo*, e nestes casos é necessário analisar o compromisso (em inglês, *trade-off*) entre os objetivos envolvidos. O exemplo a seguir ilustra isso.

**Exemplo 2.33**<sup>22</sup> Considere um banco com um capital próprio de \$20 milhões, mais \$150 milhões de depósitos em contas-correntes e \$80 milhões de depósitos em fundos de investimentos. O banco tem várias opções de investimento e deseja maximizar o lucro, mas os riscos devem ser minimizados. A Tabela 2.25 fornece as opções de investimentos com suas taxas de retorno, além de outras informações relacionadas ao risco.

Para modelar o investimento bancário, considere as variáveis:

$$x_j : \text{quantia (\$ milhões) investida na opção } j, j = 1, \dots, 8.$$

O principal objetivo do banco é maximizar seu lucro, o qual é expresso por:

$$\text{Maximizar } f_1(\mathbf{x}) = 0,04x_2 + 0,045x_3 + 0,055x_4 + 0,07x_5 + 0,105x_6 + 0,085x_7 + 0,092x_8.$$

Entretanto, o risco deve ser minimizado e expressar o risco não é tão simples quanto o lucro anterior. Duas medidas (funções) de risco são comumente utilizadas. A primeira medida é dada pela razão entre o capital requerido para a solvência e o capital do banco. Um valor baixo para esta medida indica menos risco, de modo que é conveniente:

$$\text{Minimizar } f_2(\mathbf{x}) = \frac{1}{20}(0,005x_2 + 0,04x_3 + 0,05x_4 + 0,075x_5 + 0,1x_6 + 0,1x_7 + 0,1x_8).$$

Uma outra medida para o risco é dada pela razão entre a quantia investida em opções de alto risco e o capital do banco. Uma instituição financeira segura tem essa razão baixa, de modo que é conveniente:

$$\text{Minimizar } f_3(\mathbf{x}) = \frac{1}{20}(x_6 + x_7 + x_8).$$

Algumas restrições de investimento são imediatas:

- $x_j \geq 0, j = 1, \dots, 8$
- $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 = 250$  (o capital total  $20 + 150 + 80$  é distribuído entre as opções).

<sup>22</sup> Baseado em um exemplo apresentado em Rardin (1998).

**Tabela 2.25**  
**Opções de investimento.**

<i>Tipo de investimento</i>	<i>Taxa de retorno(%)</i>	<i>Parte líquida(%)</i>	<i>Capital requerido(%)</i>	<i>Risco(?)</i>
1: Caixa	0,0	100,0	0,0	Não
2: Curto prazo	4,0	99,5	0,5	Não
3: Governo – 1 a 5 anos	4,5	96,0	4,0	Não
4: Governo – 5 a 10 anos	5,5	90,0	5,0	Não
5: Governo – mais de 10 anos	7,0	85,0	7,5	Não
6: Empréstimos pessoais	10,5	0,0	10,0	Sim
7: Financiamento de imóveis	8,5	0,0	10,0	Sim
8: Empréstimos comerciais	9,2	0,0	10,0	Sim

Outras restrições decorrem de leis que regem o mercado financeiro e políticas de investimento do banco, tais como:

- (caixa) A reserva disponível em caixa deve ser de pelo menos 14% dos depósitos em contas-correntes, mais 4% dos depósitos em fundos de investimento ( $0,14(150) + 0,04(80) = 24,2$ ):

$$x_1 \geq 24,2$$

- (liquidez) A porção de investimentos considerados líquidos deve ser de pelo menos 47% dos depósitos em contas-correntes, mais 36% dos depósitos em fundos de investimento ( $0,47(150) + 0,36(80) = 99,3$ ):

$$x_1 + 0,995x_2 + 0,96x_3 + 0,9x_4 + 0,85x_5 \geq 99,3$$

- (diversificação) Pelo menos 5% do capital total devem ser aplicados em cada opção ( $0,05(20 + 150 + 80) = 12,5$ ):

$$x_j \geq 12,5, \quad j = 1, \dots, 8$$

- (comercial) Pelo menos 30% do capital total deve ser utilizado para empréstimos comerciais ( $0,3(20 + 150 + 80) = 75$ ):

$$x_8 \geq 75$$

Em resumo, temos um modelo de otimização linear dado por:

$$\text{Maximizar } f_1(\mathbf{x}) = 0,04x_2 + 0,045x_3 + 0,055x_4 + 0,07x_5 + 0,105x_6 + 0,085x_7 + 0,092x_8$$

$$\text{Minimizar } f_2(\mathbf{x}) = \frac{1}{20}(0,005x_2 + 0,04x_3 + 0,05x_4 + 0,075x_5 + 0,1x_6 + 0,1x_7 + 0,1x_8)$$

$$\text{Minimizar } f_3(\mathbf{x}) = \frac{1}{20}(x_6 + x_7 + x_8)$$

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 = 250$$

$$x_1 \geq 24,2$$

$$x_1 + 0,995x_2 + 0,96x_3 + 0,9x_4 + 0,85x_5 \geq 99,3$$

$$x_j \geq 12,5, \quad j = 1, \dots, 8$$

$$x_8 \geq 75$$

$$x_j \geq 0, \quad j = 1, \dots, 8.$$

■

Há vários métodos para resolver um problema de otimização multiobjetivo como o exemplo de investimento bancário. Um dos mais populares é a *programação de metas*, que, para encontrar uma solução de compromisso entre os objetivos conflitantes, estabelece *a priori* um conjunto de valores (metas) aceitáveis para cada função objetivo. Considere o seguinte problema de otimização linear multiobjetivo (por simplicidade, consideramos três objetivos):

$$\begin{aligned} \text{Maximizar } & f_1(\mathbf{x}) = \mathbf{c}^T \mathbf{x} \\ \text{Minimizar } & f_2(\mathbf{x}) = \mathbf{d}^T \mathbf{x} \\ \text{Minimizar } & f_3(\mathbf{x}) = \mathbf{e}^T \mathbf{x} \\ & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

ou seja, buscamos soluções que façam com que o valor de  $f_1(\mathbf{x})$  seja grande e  $f_2(\mathbf{x})$  e  $f_3(\mathbf{x})$  sejam pequenos. Supomos que o decisor (pessoa responsável pela implementação da solução) fique satisfeito com qualquer solução que atenda:

- $f_1(\mathbf{x}) \geq \sigma_1$  : valor mínimo para  $f_1(\mathbf{x})$  que se deseja maximizar,
- $f_2(\mathbf{x}) \leq \sigma_2$  : valor máximo para  $f_2(\mathbf{x})$  que se deseja minimizar,
- $f_3(\mathbf{x}) = \sigma_3$  : valor ideal para  $f_3(\mathbf{x})$ ,

em que  $\sigma_i$  são chamadas *metas*. As desigualdades e a igualdade anteriores são definidas pelo decisor. Dessa forma, o problema de otimização linear é reformulado como um problema de encontrar uma solução factível para o sistema:

$$\begin{aligned} f_1(\mathbf{x}) &\geq \sigma_1 \\ f_2(\mathbf{x}) &\leq \sigma_2 \\ f_3(\mathbf{x}) &= \sigma_3 \\ \mathbf{Ax} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0}. \end{aligned}$$

Tal problema, entretanto, pode ser infactível, indicando que o decisor deve abrir mão de uma ou mais metas, ou seja, as restrições-meta podem ser violadas. Supomos que as restrições originais do problema  $\mathbf{Ax} = \mathbf{b}$ ,  $\mathbf{x} \geq \mathbf{0}$ , sejam factíveis, caso contrário o problema multiobjetivo não tem qualquer solução. Para formalizar essa flexibilização das restrições-meta, introduzimos novas variáveis nas restrições-meta:

$$\begin{aligned} f_1(\mathbf{x}) + y_1 &\geq \sigma_1 \\ f_2(\mathbf{x}) - y_2 &\leq \sigma_2 \\ f_3(\mathbf{x}) + y_3^+ - y_3^- &= \sigma_3, \end{aligned}$$

em que  $y_1 \geq 0$ ,  $y_2 \geq 0$ ,  $y_3^+ \geq 0$ ,  $y_3^- \geq 0$ . Tais variáveis têm um papel semelhante ao das variáveis artificiais na Seção 2.8, uma vez que fazem com que o sistema acima seja factível. Elas também são chamadas variáveis de desvio porque representam os desvios acima ou abaixo dos valores das metas (lado direito das restrições). Por exemplo, se uma solução factível  $\mathbf{x}$  é tal que  $f_1(\mathbf{x}) < \sigma_1$ , então, com  $y_1 = \sigma_1 - f_1(\mathbf{x}) > 0$ , a restrição-meta flexibilizada é satisfeita.

Dois métodos têm sido utilizados para otimizar um modelo de programação de metas: (i) o *método dos pesos* e (ii) o *método hierárquico* (em inglês, *preemptive*). Ambos baseiam-se na estratégia de converter múltiplos objetivos em um único objetivo. O método dos pesos considera que as soluções de interesse do decisor são aquelas em que as variáveis de desvio ou flexibilização são o mais próximo possível de nulas, portanto, o seguinte problema deve ser resolvido:

$$\begin{aligned}
\text{Minimizar} \quad & \varphi(\mathbf{x}, \mathbf{y}) = w_1 y_1 + w_2 y_2 + w_3 (y_3^+ + y_3^-) \\
& f_1(\mathbf{x}) + y_1 \geq \sigma_1 \\
& f_2(\mathbf{x}) - y_2 \leq \sigma_2 \\
& f_3(\mathbf{x}) + y_3^+ - y_3^- = \sigma_3 \\
& \mathbf{Ax} = \mathbf{b} \\
& \mathbf{x} \geq \mathbf{0}, y_1 \geq 0, y_2 \geq 0, y_3^+ \geq 0, y_3^- \geq 0,
\end{aligned}$$

em que  $w_1, w_2, w_3$  são os pesos (positivos) das metas, respectivamente, representando a importância relativa entre elas para o decisor. Por exemplo, se  $w_i = 1, i = 1, 2, 3$ , então todas as metas têm a mesma importância para o decisor, ou se  $w_1 > w_2 > w_3$  então a meta 1 é mais importante que a meta 2, e esta é mais importante que a meta 3.

**Exemplo 2.34** Analise o problema de investimento bancário no Exemplo 2.33. Suponha que em lugar de buscar o lucro máximo, o investidor busque a meta de pelo menos \$18,5 milhões de lucro, isto é,  $f_1(\mathbf{x}) \geq 18,5$ . Além disso, não é desejável que as duas medidas de risco superem os valores 0,8 e 7, respectivamente, isto é,  $f_2(\mathbf{x}) \leq 0,8$  e  $f_3(\mathbf{x}) \leq 7$ . Suponha, também, que todas essas metas têm a mesma importância para o decisor. Introduzindo as variáveis de desvio ou flexibilização e utilizando peso igual a 1 para cada meta, temos o seguinte problema de otimização linear (único objetivo):

$$\begin{aligned}
\text{Minimizar} \quad & \varphi(\mathbf{x}, \mathbf{y}) = y_1 + y_2 + y_3 \\
& 0,04x_2 + 0,045x_3 + 0,055x_4 + 0,07x_5 + 0,105x_6 + 0,085x_7 + 0,092x_8 + y_1 \geq 18,5 \\
& \frac{1}{20}(0,005x_2 + 0,04x_3 + 0,05x_4 + 0,075x_5 + 0,1x_6 + 0,1x_7 + 0,1x_8) - y_2 \leq 0,8 \\
& \frac{1}{20}(x_6 + x_7 + x_8) - y_3 \leq 7 \\
& x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 = 250 \\
& x_1 \geq 24,2 \\
& x_1 + 0,995x_2 + 0,96x_3 + 0,9x_4 + 0,85x_5 \geq 99,3 \\
& x_j \geq 12,5, j = 1, \dots, 8 \\
& x_8 \geq 75 \\
& x_j \geq 0, j = 1, \dots, 8, y_1 \geq 0, y_2 \geq 0, y_3 \geq 0.
\end{aligned}$$

Resolvendo esse modelo pelo algoritmo simplex (Seção 2.6.2), obtemos a seguinte solução:  $y_1 = 0, y_2 = 0,128$  e  $y_3 = 0$ , o que resulta em atingir a meta 1 (lucro de 18,5), em ter um desvio de 0,128 na meta 2 (razão entre o capital requerido para a solvência e o capital do banco igual a 0,928) e em atingir também a meta 3 (razão entre a quantia investida em opções de alto risco e o capital do banco igual a 7). O leitor pode verificar que esta solução muda se os pesos relativos das metas forem diferentes, por exemplo, se  $w_1 = 1, w_2 = 10$  e  $w_3 = 1$ , isto é, a meta 2 é mais importante, então apenas a meta 3 é atingida, mas fazendo diminuir o desvio de  $y_2 = 0,128$  para  $y_2 = 0,015$ . ■

No método hierárquico, os objetivos são tomados em ordem de prioridade, um após o outro. Por exemplo, suponha que o objetivo 1 é o mais prioritário para o decisor, o objetivo 2 é o segundo mais prioritário, o objetivo 3 é o terceiro mais prioritário, e assim por diante. Então, primeiro minimizamos o desvio  $y_1$  do primeiro objetivo, depois minimizamos o desvio  $y_2$  do segundo objetivo sujeito à condição de que o desvio do primeiro objetivo seja mínimo (isto é, o valor  $y_1$  é fixado no valor mínimo obtido anteriormente), depois minimizamos o desvio  $y_3$  do terceiro objetivo sujeito à condição de que os desvios do primeiro e do segundo objetivos sejam mínimos (isto é, os valores  $y_1$  e  $y_2$  são fixados nos valores mínimos já obtidos anteriormente), e assim por diante. O exemplo a seguir ilustra o método hierárquico.

**Exemplo 2.35** Considere novamente o problema de investimento bancário no Exemplo 2.33. Suponha que o decisor estabeleça a seguinte prioridade entre as metas: a meta 1 (lucro) é mais importante que a meta 2 (primeira medida de risco), que é mais importante que a meta 3 (segunda medida de risco). Aplicando o método hierárquico para resolver o problema, temos que (inicialmente resolvemos o modelo de investimento bancário considerando apenas a meta 1):

*Meta 1:* Minimizar  $\varphi(\mathbf{x}, \mathbf{y}) = y_1$

$$\begin{aligned} 0,04x_2 + 0,045x_3 + 0,055x_4 + 0,07x_5 + 0,105x_6 + 0,085x_7 + 0,092x_8 + y_1 &\geq 18,5 \\ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 &= 250 \\ x_1 &\geq 24,2 \\ x_1 + 0,995x_2 + 0,96x_3 + 0,9x_4 + 0,85x_5 &\geq 99,3 \\ x_j &\geq 12,5, j = 1, \dots, 8 \\ x_8 &\geq 75 \\ x_j &\geq 0, j = 1, \dots, 8, y_1 \geq 0. \end{aligned}$$

A solução deste modelo resulta em  $y_1 = 0$  e lucro igual a 18,5. Em seguida, resolvemos o modelo impondo este lucro de 18,5 (isto é, simplesmente incluindo a restrição  $y_1 = 0$  no modelo) e considerando como objetivo o atendimento apenas da meta 2:

*Meta 2:* Minimizar  $\varphi(\mathbf{x}, \mathbf{y}) = y_2$

$$\begin{aligned} 0,04x_2 + 0,045x_3 + 0,055x_4 + 0,07x_5 + 0,105x_6 + 0,085x_7 + 0,092x_8 + y_1 &\geq 18,5 \\ \frac{1}{20}(0,005x_2 + 0,04x_3 + 0,05x_4 + 0,075x_5 + 0,1x_6 + 0,1x_7 + 0,1x_8) - y_2 &\leq 0,8 \\ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 &= 250 \\ x_1 &\geq 24,2 \\ x_1 + 0,995x_2 + 0,96x_3 + 0,9x_4 + 0,85x_5 &\geq 99,3 \\ x_j &\geq 12,5, j = 1, \dots, 8 \\ x_8 &\geq 75 \\ y_1 &= 0 \\ x_j &\geq 0, j = 1, \dots, 8, y_1 \geq 0, y_2 \geq 0. \end{aligned}$$

Alternativamente a incluir a restrição  $y_1 = 0$  no modelo da meta 2 anterior, poderíamos ter reescrito o modelo fixando o valor da variável  $y_1$  em zero, o que reduziria o tamanho do modelo. Esta observação também é válida para o modelo da meta 3 adiante. Optamos pela forma acima para evidenciar a fixação da variável de desvio  $y_1$ .

A solução do modelo da meta 2 anterior resulta em  $y_2 = 0,119$  e a razão entre o capital requerido para a solvência e o capital do banco igual a 0,919 (meta 2). Finalmente, resolvemos o modelo impondo o lucro de 18,5 e o risco de 0,919 (isto é, incluindo as restrições  $y_1 = 0$  e  $y_2 = 0,119$  no modelo), e considerando apenas a meta 3:

*Meta 3:* Minimizar  $\varphi(\mathbf{x}, \mathbf{y}) = y_3$

$$\begin{aligned} 0,04x_2 + 0,045x_3 + 0,055x_4 + 0,07x_5 + 0,105x_6 + 0,085x_7 + 0,092x_8 + y_1 &\geq 18,5 \\ \frac{1}{20}(0,005x_2 + 0,04x_3 + 0,05x_4 + 0,075x_5 + 0,1x_6 + 0,1x_7 + 0,1x_8) - y_2 &\leq 0,8 \\ \frac{1}{20}(x_6 + x_7 + x_8) - y_3 &\leq 7 \\ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 &= 250 \\ x_1 &\geq 24,2 \end{aligned}$$

$$\begin{aligned}
 & x_1 + 0,995x_2 + 0,96x_3 + 0,9x_4 + 0,85x_5 \geq 99,3 \\
 & x_j \geq 12,5, \quad j = 1, \dots, 8 \\
 & x_8 \geq 75 \\
 & y_1 = 0 \\
 & y_2 = 0,119 \\
 & x_j \geq 0, \quad j = 1, \dots, 8, \quad y_1 \geq 0, \quad y_2 \geq 0, \quad y_3 \geq 0.
 \end{aligned}$$

A solução deste modelo resulta em  $y_3 = 0,158$  e a razão entre a quantia investida em opções de alto risco e o capital do banco igual a 7,158 (meta 3). Note que a solução obtida pelo método hierárquico é diferente da obtida com o método dos pesos no Exemplo 2.34 anterior. No entanto, convém salientar que a solução do método hierárquico também pode ser obtida pelo método dos pesos se usarmos uma ponderação exagerada para os objetivos. Verifique no Exemplo 2.34 que essa solução é obtida pelo método dos pesos se ponderarmos as três metas com, por exemplo,  $w_1 = 10.000$ ,  $w_2 = 100$  e  $w_3 = 1$ , respectivamente. ■

Os métodos de programação de metas procuram uma solução que satisfaz as metas do problema sem se preocupar com a otimização do problema. Em outras palavras, se houver várias soluções que satisfazem todas as metas, esses métodos não determinam a melhor delas do ponto de vista da otimização dos objetivos originais do problema,  $f_1$ ,  $f_2$  e  $f_3$ . Isso pode ser contornado adicionando-se às funções objetivos dos modelos de programação de metas múltiplos positivos e pequenos dos objetivos originais a serem minimizados ou subtraíndo-se múltiplos positivos e pequenos, se originalmente desejava-se maximizar. Por exemplo, no método dos pesos, é conveniente considerar: minimizar  $\phi(\mathbf{x}, \mathbf{y}) = w_1y_1 + w_2y_2 + w_3(y_1^- + y_3^-) - \epsilon_1f_1(\mathbf{x}) + \epsilon_2f_2(\mathbf{x}) + \epsilon_3f_3(\mathbf{x})$ . Embora nossa discussão tenha focado apenas problemas de otimização linear, a programação de metas também pode ser aplicada a problemas de otimização discreta e otimização não-linear. No caso de otimização linear, o método simplex pode ser especializado para resolver os modelos de programação de metas.<sup>23</sup>

Um exemplo de aplicação prática da programação de metas ocorre na programação de mistura (*blendagem*) e distribuição (transporte e armazenagem) do suco concentrado congelado de laranja.<sup>24</sup> O objetivo prioritário é minimizar os custos de mistura e distribuição do suco desde as fábricas de suco até os portos marítimos (para exportação), enquanto o segundo objetivo está associado a especificações técnicas do suco e refere-se a minimizar o desvio do *ratio* do suco (relação *brix/acidez* que fornece o grau de maturação e qualidade do suco) em relação aos *ratios* máximo e mínimo especificados pelo cliente. Por exemplo, se o intervalo do *ratio* especificado pelo cliente é [14, 15], deseja-se produzir um suco com *ratio* o mais perto possível de 14,5.

Existem diversos outros métodos e abordagens para tratar problemas de otimização multicritério, entre eles, THOR, ELECTRE, DEA etc.<sup>25</sup>

## 2.10 DUALIDADE

Estudamos nas seções anteriores problemas de otimização linear que, com simplificações, modelam situações práticas e suas variáveis significam alguma decisão a ser tomada, como, por exemplo, quantidades de cada ingrediente em uma mistura, níveis de estoque de produtos em um determinado período, número de barras a serem cortadas segundo um padrão de corte etc. Por outro

<sup>23</sup> Para mais detalhes das observações deste parágrafo, veja, por exemplo, Rardin (1998) e Taha (2003).

<sup>24</sup> Veja, por exemplo, Munhoz e Morabito (2001).

<sup>25</sup> Aplicações e desenvolvimentos de métodos e abordagens multicritério podem ser encontradas, por exemplo, em Almeida (2001), Alcantara *et al.* (2003), Avellar *et al.* (2005), Balestieri e Correia (1987), Borenstein e Betencourt (2005), Enslin *et al.* (1999), Ferreira (1996), Gomes (2005), Lins *et al.* (2003), Novaes (2001), Rangel *et al.* (2003), Ribeiro *et al.* (2001), Soares de Mello *et al.* (2006) e Zambon *et al.* (2005).

lado, os valores dessas variáveis (isto é, uma solução do problema de otimização) dependem dos dados do problema como, por exemplo, o estoque disponível dos ingredientes, capacidades de máquinas, a demanda dos itens etc. e, embora sejam dados (isto é, valores fixados), em geral é conveniente um decisor examinar como as possíveis variações nos dados interferem na solução do problema.

Questões do seguinte tipo podem ser de interesse: em um problema de planejamento da produção, se o estoque de uma matéria-prima aumentasse, como o custo de produção se alteraria? (isso poderia sugerir políticas de compras de matérias-primas). Em um problema de distribuição de água em redes urbanas, se a capacidade de um reservatório de água fosse ampliada, como o consumo de energia para bombeamento de água seria afetado? (isso poderia sugerir investimentos em infra-estrutura). Em um problema de corte de material, se a demanda por um tamanho de um item fosse maior ou menor, como a perda de material seria alterada? (isso poderia sugerir políticas de descontos nos preços dos itens). Essas observações correspondem a analisar o modelo matemático sob outro ponto de vista e introduzem um novo modelo de otimização linear, chamado *problema dual*, em correspondência ao original, que é chamado *problema primal*. A teoria da dualidade, estudada nesta seção, fornece alguns subsídios para responder às questões levantadas e, em adição, uma abordagem alternativa para se resolver um problema de otimização linear, o *método dual simplex*.

Para o leitor que deseja um resumo sobre dualidade, apresentamos, no final das próximas duas seções, os resumos das seções, que podem ser lidos diretamente sem a necessidade dos desenvolvimentos a seguir.

### 2.10.1 Relaxação lagrangiana e o problema dual

Considere um problema de otimização linear na forma padrão, que denominamos *problema primal*:

$$\begin{aligned} \text{Minimizar } f(\mathbf{x}) &= \mathbf{c}^T \mathbf{x} \\ \mathbf{Ax} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0}, \end{aligned}$$

em que  $\mathbf{A}$  é uma matriz  $m \times n$ . Utilizamos o seguinte exemplo para ilustrar os conceitos nesta seção.

**Exemplo 2.36** Considere o problema de corte da Seção 2.2.7. Deseja-se cortar bobinas de aço, sendo que cada bobina tem largura de  $L = 1$  metro e pesa 1 tonelada, para a produção de 108 toneladas de sub-bobinas de 0,4m e 120 toneladas de 0,3m. O peso total das bobinas cortadas deve ser mínimo. Um modelo matemático (ver (2.19) na Seção 2.2.7) pode ser escrito após, em primeiro lugar, determinar os possíveis padrões de corte, os quais são representados pelos vetores associados:

$$\mathbf{p}_1 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \quad \mathbf{p}_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad \mathbf{p}_3 = \begin{pmatrix} 0 \\ 3 \end{pmatrix},$$

ou seja, o primeiro padrão de corte produz 2 sub-bobinas de 0,4m e nenhuma de 0,3m, o segundo padrão produz 1 sub-bobina de 0,4m e 2 sub-bobinas de 0,3m, e o terceiro padrão produz somente 3 sub-bobinas de 0,3m. Seja  $x_j$  a quantidade (em toneladas) de bobinas cortadas segundo o padrão de corte  $j$ , então (por (2.19)) temos:

$$\text{Minimizar } f(\mathbf{x}) = x_1 + x_2 + x_3$$

$$\left( \begin{pmatrix} \frac{4}{5} \\ 0 \end{pmatrix} \right) x_1 + \left( \begin{pmatrix} \frac{2}{5} \\ \frac{3}{5} \end{pmatrix} \right) x_2 + \left( \begin{pmatrix} 0 \\ \frac{9}{10} \end{pmatrix} \right) x_3 = \begin{pmatrix} 108 \\ 120 \end{pmatrix} \quad \text{ou} \quad \begin{cases} \frac{4}{5}x_1 + \frac{2}{5}x_2 = 108 \\ \frac{3}{5}x_2 + \frac{9}{10}x_3 = 120 \end{cases}$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$$

A base ótima é  $\mathbf{B} = [\mathbf{a}_1 \ \mathbf{a}_2]$  e a solução básica ótima obtida pela resolução do sistema  $\mathbf{Bx}_B = \mathbf{b}$ ,  $\mathbf{x}_B \geq \mathbf{0}$ , é  $\mathbf{x}^* = (35 \ 200 \ 0)^T$  e  $f(\mathbf{x}^*) = 1 \times 35 + 1 \times 200 + 1 \times 0 = 235$ , ou seja, são necessárias 235 toneladas de bobinas cortadas para atender a demanda. Considere a seguinte questão: se houver um aumento na demanda de uma das sub-bobinas, qual é o impacto disso sobre a necessidade de cortar mais bobinas? ■

Se considerarmos que o “vetor de recursos”  $\mathbf{b}$  é passível de perturbações, então a restrição  $\mathbf{b} - \mathbf{Ax} = \mathbf{0}$  não precisa ser satisfeita exatamente, e podemos analisá-la como um vetor  $\mathbf{y} = \mathbf{b} - \mathbf{Ax}$  (no Exemplo 2.36, imagine que a demanda das subbobinas possa sofrer alterações – isso é comum em função das incertezas na previsão da demanda). O vetor  $\mathbf{y}$  pode ser visto como uma perturbação no vetor  $\mathbf{b}$ , de modo que a solução do problema agora pode ser  $\mathbf{Ax} = \mathbf{b} - \mathbf{y}$  (o sinal de  $\mathbf{y}$  nesta perturbação é irrelevante, alguns livros de otimização linear adotam  $+\mathbf{y}$ , ao invés de  $-\mathbf{y}$ , bastando definir  $\mathbf{y} = \mathbf{Ax} - \mathbf{b}$ ).

Suponha que  $\lambda_i$  seja a penalização ou o “custo” unitário de perturbar o “recurso”  $i$ . Assim,  $\lambda_i y_i$  é o custo adicional de perturbar o recurso  $i$  em  $y_i$  unidades. Isso sugere um novo problema associado, chamado *problema lagrangiano*, isto é, para cada  $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)$ , resolva:

$$\begin{aligned} & \text{Minimizar } f(\mathbf{x}) + \lambda_1 y_1 + \lambda_2 y_2 + \dots + \lambda_m y_m \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

em que  $\mathbf{y} = \mathbf{b} - \mathbf{Ax}$ .

**Definição 2.10** (*função lagrangiana e função dual*) A função objetivo do problema lagrangiano é chamada *função lagrangiana*, e é dada por:

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \lambda_1 y_1 + \lambda_2 y_2 + \dots + \lambda_m y_m$$

em que  $\mathbf{y} = \mathbf{b} - \mathbf{Ax}$ , ou seja,

$$\begin{aligned} L(\mathbf{x}, \boldsymbol{\lambda}) &= \mathbf{c}^T \mathbf{x} + \boldsymbol{\lambda}^T \mathbf{y} \\ &= \mathbf{c}^T \mathbf{x} + \boldsymbol{\lambda}^T (\mathbf{b} - \mathbf{Ax}) \\ &= (\mathbf{c}^T - \boldsymbol{\lambda}^T \mathbf{A}) \mathbf{x} + \boldsymbol{\lambda}^T \mathbf{b}. \end{aligned}$$

Seja  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$ , em que  $\mathbf{a}_j$  é a  $j$ -ésima coluna da matriz  $\mathbf{A}$  e  $\mathbf{c} = (c_1, c_2, \dots, c_n)$ , então:

$$(\mathbf{c}^T - \boldsymbol{\lambda}^T \mathbf{A}) = (c_1 - \boldsymbol{\lambda}^T \mathbf{a}_1, c_2 - \boldsymbol{\lambda}^T \mathbf{a}_2, \dots, c_n - \boldsymbol{\lambda}^T \mathbf{a}_n)$$

de modo que a função lagrangiana pode ser escrita como:

$$L(x_1, \dots, x_n, \boldsymbol{\lambda}) = (c_1 - \boldsymbol{\lambda}^T \mathbf{a}_1) x_1 + (c_2 - \boldsymbol{\lambda}^T \mathbf{a}_2) x_2 + \dots + (c_n - \boldsymbol{\lambda}^T \mathbf{a}_n) x_n + \boldsymbol{\lambda}^T \mathbf{b}$$

A função dual é definida por:

$$\begin{aligned} \min_{\mathbf{x} \geq 0} g(\boldsymbol{\lambda}) &= \{ L(x_1, \dots, x_n, \boldsymbol{\lambda}) \} \\ &= \min_{\mathbf{x} \geq 0} \{(c_1 - \boldsymbol{\lambda}^T \mathbf{a}_1) x_1 + (c_2 - \boldsymbol{\lambda}^T \mathbf{a}_2) x_2 + \dots + (c_n - \boldsymbol{\lambda}^T \mathbf{a}_n) x_n + \boldsymbol{\lambda}^T \mathbf{b}\} \\ &= \min_{x_1 \geq 0} \{(c_1 - \boldsymbol{\lambda}^T \mathbf{a}_1) x_1\} + \min_{x_2 \geq 0} \{(c_2 - \boldsymbol{\lambda}^T \mathbf{a}_2) x_2\} + \dots + \min_{x_n \geq 0} \{(c_n - \boldsymbol{\lambda}^T \mathbf{a}_n) x_n\} + \boldsymbol{\lambda}^T \mathbf{b} \end{aligned}$$

A decomposição acima na soma de  $n$  subproblemas menores é válida, pois as variáveis  $x_1, x_2, \dots, x_n$  são independentes entre si.

**Exemplo 2.37** Considere o problema de corte no Exemplo 2.36. A função dual é dada por:

$$\begin{aligned} g(\boldsymbol{\lambda}) = & \quad \text{Minimizar } L(\mathbf{x}, \boldsymbol{\lambda}) = x_1 + x_2 + x_3 + \lambda_1 y_1 + \lambda_2 y_2 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{aligned}$$

em que  $y_1 = 108 - \frac{4}{5}x_1 - \frac{2}{5}x_2$  e  $y_2 = 120 - \frac{3}{5}x_2 - \frac{9}{10}x_3$ .

Substituindo  $y_1$  e  $y_2$  na função lagrangiana:

$$\begin{aligned} g(\boldsymbol{\lambda}) &= \text{Minimizar } L(\mathbf{x}, \boldsymbol{\lambda}) = [(1 - \frac{4}{5}\lambda_1)x_1 + (1 - \frac{2}{5}\lambda_1 - \frac{3}{5}\lambda_2)x_2 + (1 - \frac{9}{10}\lambda_2)x_3] + (108\lambda_1 + 120\lambda_2) \\ &\quad x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \\ &= \min_{x_1 \geq 0} (1 - \frac{4}{5}\lambda_1)x_1 + \min_{x_2 \geq 0} (1 - \frac{2}{5}\lambda_1 - \frac{3}{5}\lambda_2)x_2 + \min_{x_3 \geq 0} (1 - \frac{9}{10}\lambda_2)x_3 + \\ &\quad (108\lambda_1 + 120\lambda_2) \end{aligned}$$

Como as restrições são simplesmente  $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$ , o problema de minimização para a determinação de  $g(\boldsymbol{\lambda})$  é simples de ser resolvido. Se algum coeficiente das variáveis for negativo, então  $g(\boldsymbol{\lambda}) = -\infty$ . Por exemplo, se  $\lambda_1 = 5, \lambda_2 = 0$ , então o coeficiente de  $x_1$  é  $1 - \frac{4}{5}\lambda_1 = -3$ , e a minimização faz com que  $x_1 = \infty$ , de modo que  $g(\boldsymbol{\lambda}) = -\infty$ . Por outro lado, se  $\lambda_1 = \frac{5}{4}, \lambda_2 = \frac{5}{6}$ , então os coeficientes de  $x_1$  e  $x_2$  são nulos (isto é,  $1 - \frac{4}{5}\lambda_1 = 0$  e  $1 - \frac{2}{5}\lambda_1 - \frac{3}{5}\lambda_2 = 0$ ), de modo que as variáveis  $x_1$  e  $x_2$  podem assumir quaisquer valores não-negativos sem que a função dual  $g(\boldsymbol{\lambda})$  se altere, enquanto o coeficiente de  $x_3$  é:  $1 - \frac{9}{10}\lambda_2 = \frac{1}{4} > 0$ , o que implica  $x_3 = 0$  para minimizar  $g(\boldsymbol{\lambda})$ . A primeira parte de  $L(\mathbf{x}, \boldsymbol{\lambda})$ , que depende de  $x_1, x_2, x_3$ , é sempre  $-\infty$  ou zero. Se for zero (como ocorre se  $\lambda_1 = \frac{5}{4}, \lambda_2 = \frac{5}{6}$ ), então  $L(\mathbf{x}, \boldsymbol{\lambda})$  é igual à segunda parte, ou seja,  $L(\mathbf{x}, \boldsymbol{\lambda}) = 108\lambda_1 + 120\lambda_2$ . Essa observação é sempre válida e é examinada a seguir. ■

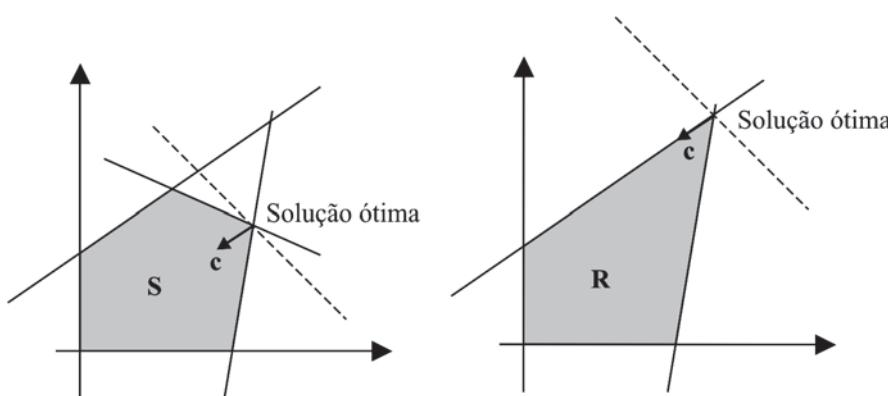
A definição da função dual fornece uma desigualdade fundamental da teoria da dualidade, a qual é baseada em uma estratégia de se obter limitantes inferiores (para os problemas de minimização). Esta estratégia é muito comum em otimização e é chamada relaxação. Suponha que um conjunto  $\mathbf{R}$  contenha  $\mathbf{S}$  ( $\mathbf{R}$  é uma relaxação de  $\mathbf{S}$ ), conforme ilustrado na Figura 2.40. Então o mínimo de uma função  $f$  em  $\mathbf{R}$  (conjunto maior) deve ser menor ou igual ao mínimo de  $f$  em  $\mathbf{S}$ . Isso porque, se  $\mathbf{x}^0 \in \mathbf{S}$  é tal que  $f(\mathbf{x}^0) = \text{mínimo}\{f(\mathbf{x}), \mathbf{x} \in \mathbf{S}\}$ , e como  $\mathbf{x}^0 \in \mathbf{R}$  (já que  $\mathbf{R} \supseteq \mathbf{S}$ ), então  $\mathbf{x}^0$  também pode ser um mínimo de  $f$  em  $\mathbf{R}$  ou um outro ponto de  $\mathbf{R}$  pode ser ainda melhor (Figura 2.40), ou seja,

$$\text{Minimizar}\{f(\mathbf{x}), \mathbf{x} \in \mathbf{R}\} \leq \text{Minimizar}\{f(\mathbf{x}), \mathbf{x} \in \mathbf{S}\}, \text{ em que } \mathbf{R} \supseteq \mathbf{S}.$$

Pela definição da função dual e definindo  $\mathbf{R} = \{\mathbf{x} \in \mathbf{R}^n \text{ tal que } \mathbf{x} \geq \mathbf{0}\} \supseteq \mathbf{S} = \{\mathbf{x} \in \mathbf{R}^n \text{ tal que } \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ , temos:

$$\begin{aligned} g(\boldsymbol{\lambda}) &= \min_{\mathbf{x} \geq \mathbf{0}} \mathbf{c}^T \mathbf{x} + \boldsymbol{\lambda}^T (\mathbf{b} - \mathbf{Ax}) \\ &\leq \min_{\{\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}} \mathbf{c}^T \mathbf{x} + \boldsymbol{\lambda}^T (\mathbf{b} - \mathbf{Ax}) && \{\text{o termo } \mathbf{b} - \mathbf{Ax} \text{ se anula}\} \\ &= \min \{\mathbf{c}^T \mathbf{x}, \text{ sujeito a: } \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\} && \{\text{problema primal}\} \\ &\leq f(\mathbf{x}), \text{ para todo } \mathbf{x} \text{ tal que } \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Em resumo, a seguinte propriedade é válida:



**Figura 2.40** Soluções ótimas em  $\mathbf{S}$  e  $\mathbf{R}$ .

**Propriedade 2.5** Para todo  $\lambda \in \mathbb{R}^m$  e para todo  $\mathbf{x}$  tal que  $\mathbf{Ax} = \mathbf{b}$ ,  $\mathbf{x} \geq \mathbf{0}$ , então  $g(\lambda) \leq f(\mathbf{x})$ .

Em outras palavras, a função dual  $g(\lambda)$  fornece um limitante inferior para a função objetivo primal  $f(\mathbf{x})$ , para todo  $\mathbf{x}$  factível. Isso nos sugere procurar o  $\lambda$  que ofereça o maior dos limitantes inferiores, o que define o problema dual.

**Definição 2.11** (*problema dual*) O maior limitante inferior para  $f(\mathbf{x})$ , obtido pela função dual, define o *problema dual lagrangiano*, ou simplesmente *problema dual*, dado por:

$$\begin{aligned} & \text{Maximizar } g(\lambda) \\ & \lambda \in \mathbb{R}^m. \end{aligned}$$

As variáveis  $(\lambda_1, \lambda_2, \dots, \lambda_m)$  são chamadas *variáveis duais*.

**Nota:** A desigualdade na propriedade 2.5,  $g(\lambda) \leq f(\mathbf{x})$ , é fundamental para escrever o problema dual, e as restrições de igualdade no problema primal fazem com que a desigualdade seja válida independentemente do sinal das variáveis duais  $\lambda_i$ , uma vez que  $\mathbf{b} - \mathbf{Ax} = \mathbf{0}$  para toda solução factível primal. Veremos adiante como as restrições de desigualdade no problema primal influenciam o sinal das variáveis duais. Por enquanto, devemos ter em mente que restrições de igualdade no primal levam a variáveis duais irrestritas de sinal.

Como já observado no Exemplo 2.37, para cada escolha das variáveis duais, o problema lagrangiano é facilmente resolvido como uma soma de  $n$  subproblemas simples e, se  $c_i - \lambda^T \mathbf{a}_i < 0$ , então,  $\min_{x_i \geq 0} (c_i - \lambda^T \mathbf{a}_i)x_i = -\infty$ . De outra forma, se existe  $i$  tal que  $c_i - \lambda^T \mathbf{a}_i < 0$ , então  $g(\lambda) = -\infty$ , que consiste em um limitante inferior inócuo (já que  $-\infty$  é um limitante inferior para qualquer número real). Para que limitantes inferiores finitos possam ser construídos, devemos escolher  $\lambda$  de modo que  $c_i - \lambda^T \mathbf{a}_i \geq 0$ ,  $i = 1, \dots, n$  e, neste caso,  $\min_{x_i \geq 0} (c_i - \lambda^T \mathbf{a}_i)x_i = 0$ ,  $i = 1, \dots, n$ , pois a solução do problema lagrangiano é dada por:

- se  $c_i - \lambda^T \mathbf{a}_i > 0$ , então  $x_i = 0$  e  $(c_i - \lambda^T \mathbf{a}_i)x_i = 0$ ;
- se  $c_i - \lambda^T \mathbf{a}_i = 0$ , então  $x_i$  é qualquer valor não-negativo e  $(c_i - \lambda^T \mathbf{a}_i)x_i = 0$ .

**Nota:** A não-negatividade da variável do problema primal, isto é,  $x_i \geq 0$ , fez surgir a restrição sobre as variáveis duais  $c_i - \lambda^T \mathbf{a}_i \geq 0$ , ou seja, o problema primal de minimização e o sinal da variável primal contribuíram para que restrições sobre as variáveis duais fossem do tipo  $c_i - \lambda^T \mathbf{a}_i \geq 0$ . No Exercício 2.27, pede-se que outros casos sejam estudados.

Portanto, como já observado no Exemplo 2.37, se escolhemos  $\lambda$  tal que  $c_i - \lambda^T \mathbf{a}_i \geq 0$ ,  $i = 1, \dots, n$ , então  $g(\lambda) = \lambda^T \mathbf{b}$ . As  $n$  desigualdades podem ser escritas em notação matricial:

$$\lambda^T \mathbf{a}_1 \leq c_1, \lambda^T \mathbf{a}_2 \leq c_2, \dots, \lambda^T \mathbf{a}_n \leq c_n \Leftrightarrow (\lambda^T \mathbf{a}_1 \ \lambda^T \mathbf{a}_2 \ \dots \ \lambda^T \mathbf{a}_n) \leq (c_1 \ c_2 \ \dots \ c_n) \Leftrightarrow \lambda^T \mathbf{A} \leq \mathbf{c}^T.$$

Tomando-se a transposta de ambos os lados da última desigualdade matricial, segue-se que:

$$\mathbf{A}^T \lambda \leq \mathbf{c}.$$

**Exemplo 2.38** Considerando o Exemplo 2.36, para que a função  $g(\lambda)$  seja finita, devemos escolher  $\lambda$  tal que:

$$\mathbf{A}^T \lambda \leq \mathbf{c} \Leftrightarrow \begin{cases} \frac{4}{5}\lambda_1 + 0\lambda_2 \leq 1 \\ \frac{2}{5}\lambda_1 + \frac{3}{5}\lambda_2 \leq 1 \\ 0\lambda_1 + \frac{9}{10}\lambda_2 \leq 1. \end{cases}$$

■

Em resumo, se escolhemos as variáveis duais tais que  $\mathbf{A}^T \boldsymbol{\lambda} \leq \mathbf{c}$ , então a função dual é dada por  $g(\boldsymbol{\lambda}) = \boldsymbol{\lambda}^T \mathbf{b}$  (note que  $\boldsymbol{\lambda}^T \mathbf{b} = \lambda_1 b_1 + \lambda_2 b_2 + \dots + \lambda_m b_m = \mathbf{b}^T \boldsymbol{\lambda}$ ), e podemos reescrever o problema dual na definição 2.11 conforme a seguir.

**Propriedade 2.6** (*problema dual*) Considere o seguinte problema primal:

$$\begin{aligned} & \text{Minimizar } f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} \\ & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

então o problema dual é dado pelo seguinte problema de otimização linear:

$$\begin{aligned} & \text{Maximizar } g(\boldsymbol{\lambda}) = \mathbf{b}^T \boldsymbol{\lambda} \\ & \mathbf{A}^T \boldsymbol{\lambda} \leq \mathbf{c}. \end{aligned}$$

Os coeficientes da função objetivo dual são dados pelo vetor de recursos  $\mathbf{b}$  do problema primal; o termo independente das restrições duais é dado pelo vetor custo  $\mathbf{c}$  do problema primal, e a matriz de restrições do problema dual é a transposta da matriz das restrições do problema primal,  $\mathbf{A}^T$ .

**Definição 2.12** (*restrições duais e solução dual factível*) O conjunto de restrições  $\mathbf{A}^T \boldsymbol{\lambda} \leq \mathbf{c}$  é chamado de *restrições duais*, e todo vetor  $\boldsymbol{\lambda}$  que satisfaça as restrições duais é chamado de *solução dual factível*.

**Exemplo 2.39** O problema dual do problema primal no Exemplo 2.36 é dado por:

$$\begin{aligned} & \text{Maximizar } g(\boldsymbol{\lambda}) = 108\lambda_1 + 120\lambda_2 \\ & \begin{cases} \frac{4}{5}\lambda_1 + 0\lambda_2 \leq 1 \\ \frac{2}{5}\lambda_1 + \frac{3}{5}\lambda_2 \leq 1 \\ 0\lambda_1 + \frac{9}{10}\lambda_2 \leq 1 \end{cases} \end{aligned}$$

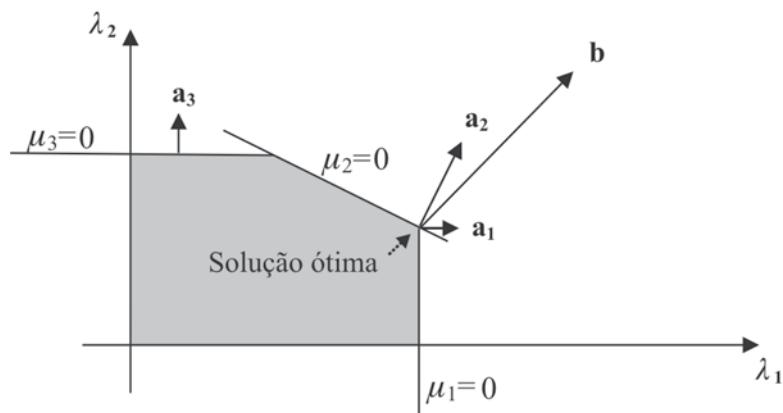
Introduzindo-se as variáveis de folga  $\mu$ , segue-se que:

$$\begin{aligned} & \text{Maximizar } g(\boldsymbol{\lambda}) = 108\lambda_1 + 120\lambda_2 \\ & \begin{cases} \frac{4}{5}\lambda_1 + 0\lambda_2 + \mu_1 = 1 \\ \frac{2}{5}\lambda_1 + \frac{3}{5}\lambda_2 + \mu_2 = 1 \\ 0\lambda_1 + \frac{9}{10}\lambda_2 + \mu_3 = 1 \end{cases} \\ & \mu_1 \geq 0, \mu_2 \geq 0, \mu_3 \geq 0 \end{aligned}$$

A resolução gráfica do problema dual (ver Figura 2.41) é simples, já que o problema dual possui apenas 2 variáveis,  $\lambda_1$  e  $\lambda_2$ , devido ao fato de o problema primal ter 2 equações. A interpretação dessas variáveis é discutida em detalhes na Seção 2.10.3 (Exemplo 2.45).

A solução ótima é obtida pela intersecção das retas  $\mu_1 = 0$  e  $\mu_2 = 0$ . À semelhança da resolução gráfica na Seção 2.5, a fronteira da região factível é definida por  $\mu_j = 0$ ,  $j = 1, 2, 3$ . Porém, as equações  $\lambda_i = 0$  não definem mais uma fronteira, pois são irrestritas de sinal (ver Figura 2.41).

Fixando-se  $\mu_1 = 0$  e  $\mu_2 = 0$  no sistema anterior, temos um sistema resultante nas variáveis  $\lambda_1$ ,  $\lambda_2$  e  $\mu_3$ , cuja solução é  $\lambda_1^* = \frac{5}{4}$ ,  $\lambda_2^* = \frac{5}{6}$ . O valor da função objetivo dual nesta solução é  $g(\boldsymbol{\lambda}^*) = 108 \times \frac{5}{4} + 120 \times \frac{5}{6} = 235$ . Reveja o Exemplo 2.36, em que  $f(\mathbf{x}^*) = 235$ . Isso antecipa uma propriedade válida,  $f(\mathbf{x}^*) = g(\boldsymbol{\lambda}^*)$ , ou seja, as soluções ótimas do primal e do dual têm o mesmo valor. Note



**Figura 2.41** Resolução gráfica do problema dual.

também que o vetor  $\mathbf{b}$  está no cone gerado pelos vetores  $\mathbf{a}_1$  e  $\mathbf{a}_2$ , isto é,  $\mathbf{b} = \mathbf{a}_1x_1 + \mathbf{a}_2x_2$  (Figura 2.41), com  $x_1 \geq 0$  e  $x_2 \geq 0$ , que é equivalente a  $\mathbf{B}\mathbf{x}_B = \mathbf{b}$ ,  $\mathbf{x}_B \geq \mathbf{0}$ . ■

Uma vez que o problema dual também é um problema de otimização linear, podemos nos perguntar sobre seu dual. A seguinte propriedade é válida.

**Propriedade 2.7** O dual do problema dual é o problema primal.

Uma das maneiras de se mostrar a validade da propriedade 2.7 consiste em escrever o problema dual na forma padrão. O problema dual na forma torna-se (definindo-se  $\boldsymbol{\lambda} = \boldsymbol{\lambda}^+ - \boldsymbol{\lambda}^-$ , com  $\boldsymbol{\lambda}^+ \geq \mathbf{0}$  e  $\boldsymbol{\lambda}^- \geq \mathbf{0}$ ):

$$\begin{aligned} \text{Minimizar } & -g(\boldsymbol{\lambda}) = -\mathbf{b}^T \boldsymbol{\lambda}^+ + \mathbf{b}^T \boldsymbol{\lambda}^- \\ & \mathbf{A}^T \boldsymbol{\lambda}^+ - \mathbf{A}^T \boldsymbol{\lambda}^- + \boldsymbol{\mu} = \mathbf{c} \\ & \boldsymbol{\lambda}^+ \geq \mathbf{0}, \boldsymbol{\lambda}^- \geq \mathbf{0}, \boldsymbol{\mu} \geq \mathbf{0}, \end{aligned}$$

em que  $\boldsymbol{\mu}$  são as variáveis de folga.

Podemos escrever o problema dual deste problema usando a propriedade 2.6. O vetor custo (isto é, os coeficientes da função objetivo), neste caso, é  $(-\mathbf{b}^T, \mathbf{b}^T, \mathbf{0}^T)$ ; o termo independente é  $\mathbf{c}$ ; e a matriz de restrições é dada por  $[\mathbf{A}^T \ -\mathbf{A}^T \ \mathbf{I}]$ . Portanto, o dual do problema é dado por:

$$\begin{aligned} \text{Maximizar } & h(\mathbf{y}) = \mathbf{c}^T \mathbf{y} \\ & [\mathbf{A}^T \ -\mathbf{A}^T \ \mathbf{I}]^T \mathbf{y} \leq (-\mathbf{b}^T, \mathbf{b}^T, \mathbf{0}^T)^T \end{aligned}$$

em que  $\mathbf{y}$  são as variáveis duais, ou:

$$\begin{aligned} \text{Maximizar } & h(\mathbf{y}) = \mathbf{c}^T \mathbf{y} \\ & \mathbf{A}\mathbf{y} \leq -\mathbf{b} \\ & -\mathbf{A}\mathbf{y} \leq \mathbf{b} \\ & \mathbf{y} \leq \mathbf{0} \end{aligned}$$

ou, ainda,

$$\begin{aligned}
 & \text{Maximizar } h(\mathbf{y}) = \mathbf{c}^T \mathbf{y} \\
 & \mathbf{A}\mathbf{y} = -\mathbf{b} \\
 & \mathbf{y} \leq \mathbf{0}.
 \end{aligned}$$

Fazendo a mudança de variável  $\mathbf{y} = -\mathbf{x}$ , obtemos

$$\begin{aligned}
 & \text{Minimizar } f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} \\
 & \mathbf{A}\mathbf{x} = \mathbf{b} \\
 & \mathbf{x} \geq \mathbf{0},
 \end{aligned}$$

que é o problema primal, comprovando a propriedade 2.7.

A combinação das propriedades 2.6 e 2.7 permite-nos escrever problemas duais de vários tipos. Entretanto, para tornar mais clara a construção de problemas duais de diversos tipos, repetimos a estratégia usada para mostrar a propriedade 2.7, transformando um problema qualquer na forma padrão. Analisemos inicialmente o efeito de restrições de desigualdades no primal sobre o dual. Já sabemos que restrições de igualdade no primal levam a variáveis duais livres de sinal. Considere o problema primal dado por:

$$\begin{aligned}
 & \text{Minimizar } f(\mathbf{x}) = c_1 x_1 + c_2 x_2 \\
 & a_{11}x_1 + a_{12}x_2 = b_1 \\
 & a_{21}x_1 + a_{22}x_2 \leq b_2 \\
 & x_1 \geq 0, x_2 \geq 0.
 \end{aligned}$$

Podemos determinar o problema dual do problema anterior usando a propriedade 2.6, após a inclusão da variável de folga na segunda restrição, escrevendo-o na forma padrão:

$$\begin{aligned}
 & \text{Minimizar } f(\mathbf{x}) = c_1 x_1 + c_2 x_2 + 0x_3 \\
 & a_{11}x_1 + a_{12}x_2 + 0x_3 = b_1 \\
 & a_{21}x_1 + a_{22}x_2 + 1x_3 = b_2 \\
 & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0.
 \end{aligned}$$

O problema dual (propriedade 2.6) é dado por:

$$\begin{array}{ll}
 \text{Maximizar } g(\lambda_1, \lambda_2) = b_1 \lambda_1 + b_2 \lambda_2 & \text{Maximizar } g(\lambda_1, \lambda_2) = b_1 \lambda_1 + b_2 \lambda_2 \\
 a_{11}\lambda_1 + a_{21}\lambda_2 \leq c_1 & a_{11}\lambda_1 + a_{21}\lambda_2 \leq c_1 \\
 a_{12}\lambda_1 + a_{22}\lambda_2 \leq c_2 & a_{12}\lambda_1 + a_{22}\lambda_2 \leq c_2 \\
 0\lambda_1 + 1\lambda_2 \leq 0, & \lambda_2 \leq 0.
 \end{array}$$

Vemos que uma restrição do tipo  $\leq$  no problema primal (minimização) leva à condição de não-positividade sobre a variável dual associada àquela restrição (isto é,  $\lambda_2 \leq 0$ ). O mesmo procedimento (isto é, transformar um problema qualquer na forma padrão e usar a propriedade 2.6) permite-nos construir o problema dual de qualquer problema de otimização linear. Vejamos mais dois casos. Consideremos agora uma restrição do tipo  $\geq$  no problema primal (minimização) e estudemos o efeito sobre o sinal da variável dual.

$$\begin{array}{ll}
 \text{Minimizar } f(\mathbf{x}) = c_1 x_1 + c_2 x_2 & \text{Minimizar } f(\mathbf{x}) = c_1 x_1 + c_2 x_2 + 0x_3 \\
 a_{11}x_1 + a_{12}x_2 = b_1 & a_{11}x_1 + a_{12}x_2 + 0x_3 = b_1 \\
 a_{21}x_1 + a_{22}x_2 \geq b_2 & a_{21}x_1 + a_{22}x_2 - 1x_3 = b_2 \\
 x_1 \geq 0, x_2 \geq 0, & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0,
 \end{array}$$

cujo problema dual (propriedade 2.6, considerando a forma padrão) é dado por:

$$\begin{array}{ll} \text{Maximizar } g(\lambda_1, \lambda_2) = b_1\lambda_1 + b_2\lambda_2 & \text{Maximizar } g(\lambda_1, \lambda_2) = b_1\lambda_1 + b_2\lambda_2 \\ a_{11}\lambda_1 + a_{21}\lambda_2 \leq c_1 & a_{11}\lambda_1 + a_{21}\lambda_2 \leq c_1 \\ a_{12}\lambda_1 + a_{22}\lambda_2 \leq c_2 & a_{12}\lambda_1 + a_{22}\lambda_2 \leq c_2 \\ 0\lambda_1 - 1\lambda_2 \leq 0, & \lambda_2 \geq 0. \end{array} \Leftrightarrow$$

Portanto, uma restrição do tipo  $\geq$  no problema primal (minimização) leva à condição de não-negatividade sobre a variável dual associada àquela restrição ( $\lambda_2 \geq 0$ ). Nos problemas anteriores, envolvendo desigualdades nas restrições do problema primal, consideramos sempre a minimização da função objetivo primal. Em resumo, temos:

- (Minimização primal) se a restrição  $i$  for do tipo  $\leq$ , então  $\lambda_i \leq 0$
- (Minimização primal) se a restrição  $i$  for do tipo  $\geq$ , então  $\lambda_i \geq 0$
- (Minimização primal) se a restrição  $i$  for do tipo  $=$ , então  $\lambda_i$  é livre.

Entretanto, o sinal da variável dual muda caso o problema primal seja de maximização. Considere o problema primal (maximização) com uma restrição do tipo  $\leq$  e reescrito na forma padrão:

$$\begin{array}{ll} \text{Maximizar } f(x_1, x_2) = c_1x_1 + c_2x_2 & \Leftrightarrow \text{Minimizar } -f(\mathbf{x}) = -c_1x_1 - c_2x_2 - 0x_3 \\ a_{12}x_1 + a_{21}x_2 = b_1 & a_{11}x_1 + a_{12}x_2 + 0x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 \leq b_2 & a_{21}x_1 + a_{22}x_2 + 1x_3 = b_2 \\ x_1 \geq 0, x_2 \geq 0, & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{array}$$

cujo problema dual (propriedade 2.6) é dado por:

$$\begin{array}{ll} \text{Maximizar } g(\lambda_1, \lambda_2) = b_1\lambda_1 + b_2\lambda_2 & \\ a_{11}\lambda_1 + a_{21}\lambda_2 \leq -c_1 & \\ a_{12}\lambda_1 + a_{22}\lambda_2 \leq -c_2 & \\ \lambda_2 \leq 0. & \end{array}$$

Embora o problema anterior seja o dual do problema precedente, seus dados são alterados por sinais negativos no vetor  $\mathbf{c}$  e podemos reescrevê-lo de modo que os negativos sejam eliminados. Para isso, multiplicamos as restrições e a função objetivo por  $-1$ , invertendo as desigualdades de  $\leq$  para  $\geq$  e a maximização para minimização e, finalmente, trocamos as variáveis  $\lambda_i \leftarrow -\lambda_i$  e  $g \leftarrow -g$ :

$$\begin{array}{ll} \text{Minimizar } -g(\lambda_1, \lambda_2) = b_1(-\lambda_1) + b_2(-\lambda_2) & \Leftrightarrow \text{Minimizar } g(\lambda_1, \lambda_2) = b_1\lambda_1 + b_2\lambda_2 \\ a_{11}(-\lambda_1) + a_{21}(-\lambda_2) \geq c_1 & a_{11}\lambda_1 + a_{21}\lambda_2 \geq c_1 \\ a_{12}(-\lambda_1) + a_{22}(-\lambda_2) \geq c_2 & a_{12}\lambda_1 + a_{22}\lambda_2 \geq c_2 \\ (-\lambda_2) \geq 0, & \lambda_2 \geq 0. \end{array}$$

Concluímos que o dual de um problema primal de maximização é um problema de minimização. Além disso, se as restrições do primal são do tipo  $\leq$ , então a variável dual associada é não-negativa ( $\lambda_2 \geq 0$ ). Como antes, a variável dual ( $\lambda_i$ ) associada a uma restrição de igualdade é irrestrita de sinal (não depende de o problema ser de maximização ou minimização). Os exemplos anteriores mostram que as desigualdades nas restrições do problema primal são determinantes dos sinais das variáveis duais, as quais devem ser combinadas com a minimização ou maximização do problema primal. A seguir, apresentamos um resumo da seção e uma tabela com a qual é possível determinar o problema dual para qualquer problema primal.

### Resumo da seção

Para um problema de otimização linear em sua forma padrão, chamado *problema primal*, podemos definir um outro problema de otimização linear, chamado *problema dual*:

Problema primal:

$$\text{Minimizar } f(\mathbf{x}) = \mathbf{c}^T \mathbf{x}$$

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

$$\mathbf{x} \geq \mathbf{0},$$

Problema dual:

$$\text{Maximizar } g(\boldsymbol{\lambda}) = \mathbf{b}^T \boldsymbol{\lambda}$$

$$\mathbf{A}^T \boldsymbol{\lambda} \leq \mathbf{c}.$$

Isto é, o problema dual é construído da seguinte forma (primal na forma padrão: minimização, restrições de igualdades e variáveis não-negativas):

- Problema dual é maximização.
- O número de variáveis duais  $\boldsymbol{\lambda}$  é igual ao número de restrições do primal.
- O número de restrições duais é igual ao número de variáveis  $\mathbf{x}$  do primal.
- Os coeficientes da função objetivo dual são os coeficientes do vetor de recursos  $\mathbf{b}$  do primal.
- A matriz dos coeficientes das restrições duais é a transposta da matriz dos coeficientes do primal,  $\mathbf{A}^T$ .
- As restrições duais são do tipo  $\leq$ .
- O vetor de recursos dual é formado pelos coeficientes  $\mathbf{c}$  da função objetivo primal (também chamado gradiente do objetivo primal).

Caso o problema primal não esteja na forma padrão (por exemplo, restrições de desigualdade), a Tabela 2.26 fornece as regras de como construir o dual de qualquer problema de otimização linear.

Assim, se tivermos um problema primal de minimização (siga a coluna à esquerda da Tabela 2.26), para cada restrição  $i$  do tipo ‘=’, temos uma variável dual  $\lambda_i$  livre; para cada restrição do tipo ‘≤’, temos uma variável dual  $\lambda_i \leq 0$  e, para cada restrição do tipo ‘≥’, uma variável dual  $\lambda_i \geq 0$ . Além disso, para uma variável primal  $x_i \geq 0$ , temos a  $i$ -ésima restrição dual do tipo ‘≤’ e assim por diante. Os exemplos a seguir ilustram como utilizar a Tabela 2.26 para escrever um problema dual.

Conforme discutido na propriedade 2.7, podemos calcular o problema dual do dual (já que o dual é um problema de otimização linear e podemos aplicar a Tabela 2.26) e encontramos precisamente o problema primal. Isto é, o dual do dual é o primal.

**Tabela 2.26**  
**Regras para construção do problema dual.**

Primal (dual)	Dual (primal)		
Minimização	Maximização		
Vetor de recursos	Gradiente do objetivo		
Gradiente do objetivo	Vetor de recursos		
Restrição	=	Livre	Variável
	≤	≤	
	≥	≥	
Variável	≥	≤	Restrição
	≤	≥	
	Livre	=	

**Exemplo 2.40** Considere os seguintes problemas primais. Utilizando a Tabela 2.26, construímos os problemas duais respectivos.

a) Primal:

$$\begin{aligned} \text{Minimizar } f(\mathbf{x}) &= x_1 + 2x_2 \\ -2x_1 + x_2 &\leq 3 \\ 3x_1 + 4x_2 &\leq 5 \\ x_1 - x_2 &\leq 2 \\ x_1 \geq 0, x_2 &\geq 0. \end{aligned}$$

Como o problema primal é do tipo *minimização*, deve-se seguir a coluna à esquerda da Tabela 2.26 para escrever as características do dual à direita.

Dual: (problema de maximização com duas restrições e três variáveis)

$$\begin{aligned} \text{Maximizar } g(\boldsymbol{\lambda}) &= 3\lambda_1 + 5\lambda_2 + 2\lambda_3 && (\text{devido ao primal minimização}) \\ -2\lambda_1 + 3\lambda_2 + \lambda_3 &\leq 1 && (\text{devido às variáveis primais } x_j \geq 0) \\ \lambda_1 + 4\lambda_2 - \lambda_3 &\leq 2 \\ \lambda_1 \leq 0, \lambda_2 \leq 0, \lambda_3 \leq 0. &&& (\text{devido às restrições primais } \leq) \end{aligned}$$

b) Primal:

$$\begin{aligned} \text{Maximizar } f(\mathbf{x}) &= x_1 + 2x_2 \\ -2x_1 + x_2 &\leq 3 \\ 3x_1 + 4x_2 &\leq 5 \\ x_1 - x_2 &\leq 2 \\ x_1 \geq 0, x_2 &\geq 0. \end{aligned}$$

Como o problema primal é do tipo *maximização*, deve-se seguir a coluna à direita da Tabela 2.26 para escrever as características do dual à esquerda.

Dual: (problema de minimização com duas restrições e três variáveis)

$$\begin{aligned} \text{Minimizar } g(\boldsymbol{\lambda}) &= 3\lambda_1 + 5\lambda_2 + 2\lambda_3 && (\text{devido ao primal maximização}) \\ -2\lambda_1 + 3\lambda_2 + \lambda_3 &\geq 1 && (\text{devido às variáveis primais } x_j \geq 0) \\ \lambda_1 + 4\lambda_2 - \lambda_3 &\geq 2 \\ \lambda_1 \geq 0, \lambda_2 \geq 0, \lambda_3 \geq 0. &&& (\text{devido às restrições primais } \geq) \end{aligned}$$

c) Primal:

$$\begin{aligned} \text{Maximizar } f(\mathbf{x}) &= x_1 + 2x_2 \\ -2x_1 + x_2 + x_3 &\geq 3 \\ 3x_1 + 4x_2 &\leq 5 \\ x_1 \geq 0, x_2 &\text{ livre, } x_3 \leq 0. \end{aligned}$$

Dual: (problema de minimização com três restrições e duas variáveis)

$$\begin{aligned} \text{Minimizar } g(\boldsymbol{\lambda}) &= 3\lambda_1 + 5\lambda_2 \\ -2\lambda_1 + 3\lambda_2 &\geq 1 && (\text{devido a } x_1 \geq 0) \\ \lambda_1 + 4\lambda_2 &= 2 && (\text{devido a } x_2 \text{ livre}) \\ \lambda_1 &\leq 0 && (\text{devido a } x_3 \leq 0) \\ \lambda_1 \leq 0, \lambda_2 &\geq 0. && (\text{devido à 1ª restrição } \geq \text{ e à 2ª restrição } \leq). \end{aligned}$$

O leitor pode aplicar a Tabela 2.26 para construir o problema dual de cada um dos problemas duais anteriores em (a), (b) e (c), para concluir que os problemas primais são reconstruídos. ■

### 2.10.2 Relações primais-duais

Nesta seção, vemos como os problemas primal e dual estão relacionados e como obter a solução de um problema a partir da solução do outro. Por simplicidade de exposição, definimos o conjunto das soluções factíveis do problema primal por:

$$\mathbf{P} = \{\mathbf{x} \in \mathbb{R}^n \text{ tal que } \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$$

e o conjunto das soluções factíveis do problema dual por:

$$\mathbf{D} = \{\boldsymbol{\lambda} \in \mathbb{R}^m \text{ tal que } \mathbf{A}^T \boldsymbol{\lambda} \leq \mathbf{c}\}$$

Assim, a propriedade 2.5 pode ser reescrita como:

$$g(\boldsymbol{\lambda}) \leq f(\mathbf{x}), \forall \boldsymbol{\lambda} \in \mathbf{D}, \forall \mathbf{x} \in \mathbf{P}$$

e várias outras consequências dessa propriedade são mostradas a seguir. Podemos ter  $\mathbf{P} \neq \emptyset$  ou  $\mathbf{P} = \emptyset$ . No caso de  $\mathbf{P} = \emptyset$ , o problema não tem solução factível. No caso de  $\mathbf{P} \neq \emptyset$ , o problema pode ter solução ótima (que pode ser única ou múltipla) ou a função objetivo é ilimitada.

**Propriedade 2.8** Suponha que  $\mathbf{P} \neq \emptyset$  (isto é, existe solução factível primal). O problema primal não tem solução ótima se e somente se  $\mathbf{D} = \emptyset$  (de outra forma, no caso de minimização,  $f(\mathbf{x}) \rightarrow -\infty$  se e somente se não existir solução factível dual).

**Exemplo 2.41** Considere o par de problemas primal-dual:

Primal:

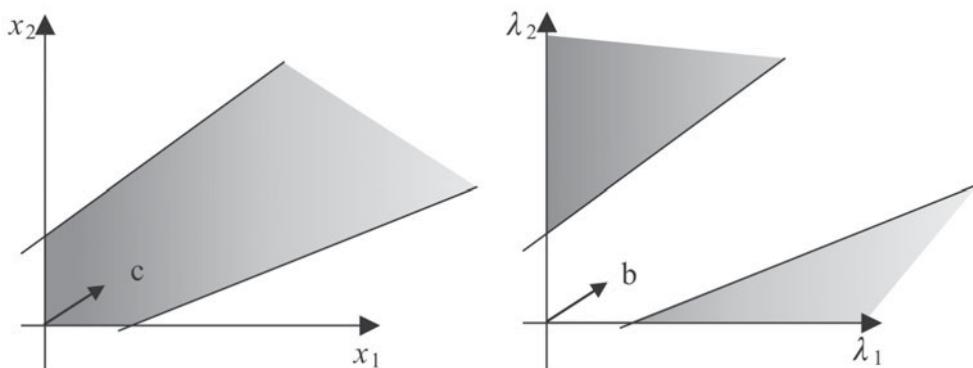
$$\begin{aligned} \text{Maximizar } f(\mathbf{x}) &= x_1 + x_2 \\ &-x_1 + x_2 \leq 1 \\ &x_1 - 2x_2 \leq 1 \\ &x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

Dual:

$$\begin{aligned} \text{Minimizar } g(\boldsymbol{\lambda}) &= \lambda_1 + \lambda_2 \\ -\lambda_1 + \lambda_2 &\geq 1 \\ \lambda_1 - 2\lambda_2 &\geq 1 \\ \lambda_1 &\geq 0, \lambda_2 \geq 0 \end{aligned}$$

A resolução gráfica (Figura 2.42) mostra que o primal não tem solução ótima, apesar de  $\mathbf{P} \neq \emptyset$  ( $f(\mathbf{x}) \rightarrow \infty$ , problema de maximização), e o dual é infactível, isto é,  $\mathbf{D} = \emptyset$ .

A validade da condição necessária na propriedade 2.8 (isto é, se  $f(\mathbf{x}) \rightarrow -\infty$ , então  $\mathbf{D} = \emptyset$ ) segue diretamente da propriedade 2.5, pois, se existe  $\boldsymbol{\lambda} \in \mathbf{D}$ , então  $g(\boldsymbol{\lambda})$  é um limitante inferior para



**Figura 2.42** (a) Primal sem solução ótima, (b) Dual infactível.

$f(\mathbf{x})$ , impedindo que  $f(\mathbf{x}) \rightarrow -\infty$ . Entretanto, a prova da condição suficiente (isto é,  $f(\mathbf{x}) \rightarrow -\infty$  se  $\mathbf{D} = \emptyset$ ) não é trivial.<sup>26</sup> O mesmo raciocínio é válido para a condição necessária da propriedade seguinte, enquanto a prova da condição suficiente também não é trivial.

**Propriedade 2.9** Suponha que  $\mathbf{D} \neq \emptyset$  (isto é, existe solução factível dual). O problema dual não tem solução ótima se e somente se  $\mathbf{P} = \emptyset$  (isto é,  $g(\boldsymbol{\lambda}) \rightarrow \infty$  se e somente se não existir solução factível primal).

Observe a simetria desses resultados (revisite o Exemplo 2.41 e troque primal por dual). As propriedades 2.8 e 2.9 pressupõem que um dos conjuntos de soluções é factível ( $\mathbf{P} \neq \emptyset$  ou  $\mathbf{D} \neq \emptyset$ ). Entretanto, há a possibilidade de ambos serem infactíveis:  $\mathbf{P} = \emptyset$  e  $\mathbf{D} = \emptyset$ , como pode ser visto no exemplo a seguir.

**Exemplo 2.42** (primal e dual infactíveis) Considere o problema:

$$\begin{array}{ll} \textbf{Primal:} & \text{Minimizar } f(x_1) = -x_1 \\ & 0x_1 = 1 \quad (\text{equação impossível: } \mathbf{P} = \emptyset) \\ & x_1 \geq 0, \\ \textbf{Dual:} & \text{Maximizar } g(\lambda_1) = \lambda_1 \\ & 0\lambda_1 \leq -1. \quad (\text{desigualdade impossível: } \mathbf{D} = \emptyset) \end{array}$$

Ambos os problemas, primal e dual, são infactíveis. ■

Das propriedades 2.8 e 2.9 e dos exemplos anteriores, podemos concluir que, se um dos problemas (primal ou dual) for infactível, então o outro não tem solução ótima (seja porque também é infactível ou porque a função objetivo é ilimitada). Podemos, ainda, escrever a seguinte propriedade.

**Propriedade 2.10** O problema primal tem solução ótima se e somente se o dual tiver solução ótima.

De fato, suponha que o problema primal tenha solução ótima. Assim,  $\mathbf{P} \neq \emptyset, f(\mathbf{x}) > -\infty$ . Da propriedade 2.8, segue-se que  $\mathbf{D} \neq \emptyset$  (pois, se  $\mathbf{D} = \emptyset$ , então  $f(\mathbf{x}) \rightarrow -\infty$ , mas é um absurdo, já que  $f(\mathbf{x}) > -\infty$ ). Assim, como  $\mathbf{P} \neq \emptyset$  e  $\mathbf{D} \neq \emptyset$ , da propriedade 2.9, segue-se que  $g(\boldsymbol{\lambda}) < \infty$  (pois se  $g(\boldsymbol{\lambda}) \rightarrow \infty$ , então  $\mathbf{P} = \emptyset$ , o que resulta em um absurdo, uma vez que  $\mathbf{P} \neq \emptyset$ ), ou seja, o problema dual não é infactível, nem sua função objetivo é ilimitada, de modo que tem solução ótima (a possibilidade restante). Raciocinando de forma análoga, se supusermos que o problema dual tem solução ótima, então o problema primal terá solução ótima.

As propriedades que relacionam os problemas primal e dual até agora não informam como seus objetivos se relacionam nas respectivas soluções ótimas ou como estas estão relacionadas. Uma nova propriedade, que decorre diretamente da propriedade 2.5, fornece uma nova condição de otimalidade.

**Propriedade 2.11** Sejam  $\mathbf{x}^* \in \mathbf{P}$  e  $\boldsymbol{\lambda}^* \in \mathbf{D}$  (isto é,  $\mathbf{x}^*$  é uma solução factível primal e  $\boldsymbol{\lambda}^*$  uma solução factível dual). Se  $f(\mathbf{x}^*) = g(\boldsymbol{\lambda}^*)$ , então  $\mathbf{x}^*$  é solução ótima primal e  $\boldsymbol{\lambda}^*$  é solução ótima dual.

A validade da propriedade 2.11 decorre da propriedade 2.5:  $g(\boldsymbol{\lambda}) \leq f(\mathbf{x})$ , para todo  $\boldsymbol{\lambda} \in \mathbf{D}$  e para todo  $\mathbf{x} \in \mathbf{P}$ , pois é impossível diminuir  $f(\mathbf{x})$  abaixo de  $f(\mathbf{x}^*)$ . Se fosse possível, então existiria  $\mathbf{x} \in \mathbf{P}$  tal que  $f(\mathbf{x}) < f(\mathbf{x}^*) = g(\boldsymbol{\lambda}^*)$ , o que contraria a propriedade 2.5. A recíproca da propriedade 2.11 é também verdadeira, como vemos adiante.

<sup>26</sup> A prova da condição suficiente pode ser encontrada em Bazaraa *et al.* (1990).

Analisemos um pouco mais a condição de otimalidade da propriedade 2.11. Sejam  $\mathbf{x} \in \mathbf{P}$  e  $\boldsymbol{\lambda} \in \mathbf{D}$  e suponha que  $f(\mathbf{x}) = g(\boldsymbol{\lambda})$ , isto é,  $\mathbf{c}^T \mathbf{x} = \boldsymbol{\lambda}^T \mathbf{b}$ . Substituindo  $\mathbf{A}\mathbf{x} = \mathbf{b}$ , segue-se que  $\mathbf{c}^T \mathbf{x} = \boldsymbol{\lambda}^T \mathbf{A}\mathbf{x}$ , o que implica  $(\mathbf{c}^T - \boldsymbol{\lambda}^T \mathbf{A})\mathbf{x} = 0$ . O vetor  $\mathbf{c}^T - \boldsymbol{\lambda}^T \mathbf{A}$  é o vetor das variáveis de folga do problema dual:

$$\mathbf{A}^T \boldsymbol{\lambda} \leq \mathbf{c} \Leftrightarrow \begin{cases} \mathbf{a}_1^T \boldsymbol{\lambda} \leq c_1 \\ \mathbf{a}_2^T \boldsymbol{\lambda} \leq c_2 \\ \vdots \\ \mathbf{a}_n^T \boldsymbol{\lambda} \leq c_n \end{cases} \Leftrightarrow \begin{cases} \mathbf{a}_1^T \boldsymbol{\lambda} + \mu_1 = c_1 \\ \mathbf{a}_2^T \boldsymbol{\lambda} + \mu_2 = c_2 \\ \vdots \\ \mathbf{a}_n^T \boldsymbol{\lambda} + \mu_n = c_n \end{cases} \quad \mu_1 \geq 0, \mu_2 \geq 0, \dots, \mu_n \geq 0.$$

Como  $\mathbf{a}_j^T \boldsymbol{\lambda} = \boldsymbol{\lambda}^T \mathbf{a}_j$  e  $\mathbf{c}^T - \boldsymbol{\lambda}^T \mathbf{A} = (c_1 - \boldsymbol{\lambda}^T \mathbf{a}_1 \ c_2 - \boldsymbol{\lambda}^T \mathbf{a}_2 \ \dots \ c_n - \boldsymbol{\lambda}^T \mathbf{a}_n)$ , segue-se que:

$$\begin{aligned} (\mathbf{c}^T - \boldsymbol{\lambda}^T \mathbf{A})\mathbf{x} = 0 &\Leftrightarrow (c_1 - \boldsymbol{\lambda}^T \mathbf{a}_1)x_1 + (c_2 - \boldsymbol{\lambda}^T \mathbf{a}_2)x_2 + \dots + (c_n - \boldsymbol{\lambda}^T \mathbf{a}_n)x_n = 0 \\ &\Leftrightarrow \mu_1 x_1 + \mu_2 x_2 + \dots + \mu_n x_n = 0, \text{ em que } \mu_j = c_j - \boldsymbol{\lambda}^T \mathbf{a}_j \geq 0. \end{aligned}$$

Como  $x_j \geq 0, j = 1, 2, \dots, n$ , cada uma das parcelas da soma acima é não-negativa, do que se conclui que todas são nulas:

$$\mu_1 x_1 = 0, \mu_2 x_2 = 0, \dots, \mu_n x_n = 0.$$

Essa relação entre soluções ótimas primal e dual é chamada de *folgas complementares*, é satisfeita para toda solução ótima primal e dual e pode ser enunciada como a propriedade a seguir. (A condição necessária decorre da propriedade 2.14).

**Propriedade 2.12** (*folgas complementares*) As soluções  $\mathbf{x} \in \mathbf{R}^n$  e  $\boldsymbol{\lambda} \in \mathbf{R}^m$  são ótimas, primal e dual respectivamente, se e somente se:

$\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$	( $\mathbf{x}$ é factível primal)
$\mathbf{A}^T \boldsymbol{\lambda} + \boldsymbol{\mu} = \mathbf{c}, \boldsymbol{\mu} \geq \mathbf{0}$	( $\boldsymbol{\lambda}$ é factível dual)
$\mu_j x_j = 0, j = 1, \dots, n$	(folgas complementares)

A propriedade 2.12 permite determinar a solução ótima de um dos problemas (primal ou dual) quando a solução ótima do outro é conhecida (ver o Exemplo 2.43). Além disso, nos diz que a resolução de um problema de otimização linear pode ser obtida pela resolução de um sistema de equações não-lineares (note que as folgas complementares são equações não-lineares). Isso sugere que um método de solução para a otimização linear pode ser construído a partir de um método de resolução do sistema não-linear da propriedade 2.12. Um método muito eficiente para a resolução de sistemas não-lineares é o método de Newton,<sup>27</sup> o qual pode ser adaptado para resolver o sistema não-linear da propriedade 2.12, evitando-se as desigualdades  $\mathbf{x} \geq \mathbf{0}$  e  $\boldsymbol{\mu} \geq \mathbf{0}$  mantendo-as estritamente positivas (tais soluções factíveis são chamadas pontos interiores). Esse é um dos princípios de um dos métodos mais importantes de pontos interiores para otimização linear.<sup>28</sup>

**Exemplo 2.43** Considere novamente o problema no Exemplo 2.36 e seu dual:

**Primal:** Minimizar  $f(\mathbf{x}) = x_1 + x_2 + x_3$

$$\begin{cases} \frac{4}{5}x_1 + \frac{2}{5}x_2 &= 108 \\ \frac{3}{5}x_2 + \frac{9}{10}x_3 &= 120 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, \end{cases}$$

**Dual:** Maximizar  $g(\boldsymbol{\lambda}) = 108\lambda_1 + 120\lambda_2$

$$\begin{cases} \frac{4}{5}\lambda_1 + 0\lambda_2 + \mu_1 &= 1 \\ \frac{2}{5}\lambda_1 + \frac{3}{5}\lambda_2 + \mu_2 &= 1 \\ 0\lambda_1 + \frac{9}{10}\lambda_2 + \mu_3 &= 1 \\ \mu_1 \geq 0, \mu_2 \geq 0, \mu_3 \geq 0. \end{cases}$$

<sup>27</sup> Veja, por exemplo, Vanderbei (2001) e outros métodos para resolução de sistemas de equações não-lineares em Ruggiero *et al.* (1992).

<sup>28</sup> Para estudos em métodos de pontos interiores, veja Adler *et al.* (1989), Gonzaga (1991) e Vanderbei (2001).

Seja a solução factível primal  $x_1 = 35$ ,  $x_2 = 200$ ,  $x_3 = 0$ . Busquemos uma solução dual que verifique as folgas complementares  $\mu_1 x_1 = 0$ ,  $\mu_2 x_2 = 0$ ,  $\mu_3 x_3 = 0$ , o que implica  $\mu_1 = 0$ ,  $\mu_2 = 0$ , já que  $x_1 > 0$  e  $x_2 > 0$ . As variáveis positivas do primal (como ocorre tipicamente com as variáveis básicas) fazem com que as restrições duais sejam satisfeitas com igualdade (dizemos que são restrições ativas), resultando em um sistema de equações lineares:

$$\begin{aligned}\frac{4}{5} \lambda_1 &= 1 & (\mu_1 = 0) \\ \frac{2}{5} \lambda_1 + \frac{3}{5} \lambda_2 &= 1 & (\mu_2 = 0)\end{aligned}$$

cuja solução é dada por  $\lambda_1 = \frac{5}{4}$ ,  $\lambda_2 = \frac{5}{6}$ . Como a terceira restrição dual,  $\frac{9}{10} \lambda_2 \leq 1$ , é satisfeita com folga nesta solução ( $\mu_3 = \frac{1}{4}$ ), então ela é factível dual (ver a resolução gráfica do problema dual no Exemplo 2.39). Em resumo, as soluções  $x_1 = 35$ ,  $x_2 = 200$ ,  $x_3 = 0$  e  $\lambda_1 = \frac{5}{4}$ ,  $\lambda_2 = \frac{5}{6}$  são soluções factíveis primal e dual, respectivamente, e satisfazem as folgas complementares. Portanto, são soluções ótimas primal e dual, respectivamente. Como já observado no Exemplo 2.39,  $f(x_1, x_2, x_3) = x_1 + x_2 + x_3 = 35 + 200 + 0 = g(\lambda_1, \lambda_2) = 108 \times \frac{5}{4} + 120 \times \frac{5}{6} = 235$ . O Exercício 2.30 propõe o caminho inverso, isto é, dada uma solução dual óptima, encontre uma solução primal óptima. ■

Substituindo  $\boldsymbol{\mu}^T = \mathbf{c}^T - \boldsymbol{\lambda}^T \mathbf{A}$  nas folgas complementares  $\boldsymbol{\mu}^T \mathbf{x} = 0$ , obtemos  $(\mathbf{c}^T - \boldsymbol{\lambda}^T \mathbf{A}) \mathbf{x} = 0$  ou, equivalentemente,  $\mathbf{c}^T \mathbf{x} = \boldsymbol{\lambda}^T \mathbf{A} \mathbf{x}$ . Como  $\mathbf{A} \mathbf{x} = \mathbf{b}$ , para  $\mathbf{x}$  factível, segue-se que  $f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} = \boldsymbol{\lambda}^T \mathbf{b} = g(\boldsymbol{\lambda})$ . Portanto, a propriedade 2.12 pode ser reescrita como:

**Propriedade 2.13** (*dualidade forte*) As soluções  $\mathbf{x}^* \in \mathbf{P}$  e  $\boldsymbol{\lambda}^* \in \mathbf{D}$  são ótimas, primal e dual respectivamente, se e somente se  $f(\mathbf{x}^*) = g(\boldsymbol{\lambda}^*)$ .

As folgas complementares (propriedade 2.12) fornecem um caminho para se calcular uma solução óptima de um dos problemas quando já se conhece a solução óptima do outro (ver Exemplo 2.43 e Exercício 2.30). A propriedade a seguir nos dá, diretamente, a solução óptima do problema dual quando uma solução óptima básica primal é encontrada.

**Propriedade 2.14** O vetor multiplicador simplex na solução óptima primal é uma solução óptima dual.

**Exemplo 2.44** Considere novamente o problema no Exemplo 2.36:

$$\begin{aligned}\text{Minimizar } f(\mathbf{x}) &= x_1 + x_2 + x_3 \\ \begin{cases} \frac{4}{5} x_1 + \frac{2}{5} x_2 &= 108 \\ \frac{3}{5} x_2 + \frac{9}{10} x_3 &= 120 \end{cases} \\ x_1 &\geq 0, x_2 \geq 0, x_3 \geq 0\end{aligned}$$

e sua solução óptima  $(x_1 \ x_2 \ x_3)^T = (35 \ 200 \ 0)^T$ , a qual é básica:

$$\mathbf{x}_B = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 108 \\ 200 \end{pmatrix}, \mathbf{x}_N = (x_3) = (0)$$

e a matriz básica:

O vetor multiplicador simplex é dado por:

$$\boldsymbol{\lambda}^T = \mathbf{c}_B \mathbf{B}^{-1} = (1 \ 1) \begin{bmatrix} \frac{5}{4} & -\frac{5}{6} \\ 0 & \frac{5}{3} \end{bmatrix} = \left( \frac{5}{4} \quad \frac{5}{6} \right)$$

que coincide com a solução ótima calculada no Exemplo 2.39. ■

A validade da propriedade 2.14 vem do fato de que o vetor multiplicador  $\boldsymbol{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1}$  é uma solução dual factível e os objetivos primal e dual coincidem (considerando a solução básica primal:  $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$ ,  $\mathbf{x}_N = \mathbf{0}$ ). De fato, os objetivos coincidem:

$$g(\boldsymbol{\lambda}) = \boldsymbol{\lambda}^T \mathbf{b} = \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b} = \mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_N^T \mathbf{x}_N = f(\mathbf{x}).$$

A igualdade entre os objetivos ocorre independentemente de o vetor multiplicador fornecer uma solução dual factível (isso ocorre em todas as iterações do método simplex). Mostramos agora que a condição de otimalidade (do problema primal) garante que o vetor multiplicador seja uma solução dual factível. Pela definição do vetor multiplicador,  $\boldsymbol{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1}$ , temos que:

$$\boldsymbol{\lambda}^T \mathbf{B} = \mathbf{c}_B^T \Leftrightarrow \boldsymbol{\lambda}^T \mathbf{a}_i = c_i \text{ para todo } i \text{ básico.}$$

Isto é, as restrições duais associadas às variáveis básicas são satisfeitas com igualdade (restrições ativas em  $\boldsymbol{\lambda}$ ). Considerando as condições de otimalidade satisfeitas (supondo uma base ótima obtida pelo método simplex), segue-se que os custos relativos

$$\mathbf{c}_j - \boldsymbol{\lambda}^T \mathbf{a}_j \geq 0 \text{ para todo } j \text{ não-básico,}$$

e, portanto,  $\boldsymbol{\lambda}^T \mathbf{a}_j \leq c_j$  para todo  $j$  não-básico, ou seja, as restrições duais associadas às variáveis não-básicas também são satisfeitas. Assim, todas as restrições duais são satisfeitas, portanto,  $\boldsymbol{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1}$  consiste em uma solução factível dual.

### *Resumo da seção*

Os problemas primal e dual, definidos na Seção 2.10.1 (ver resumo da seção anterior), mantêm relações estreitas que podem ser resumidas por:

- O problema primal tem solução ótima se e somente se o problema dual também tiver solução ótima.
- O mínimo da função objetivo primal é igual ao máximo da função objetivo dual, isto é, se  $\mathbf{x}^*$  é uma solução ótima primal e  $\boldsymbol{\lambda}^*$  é uma solução ótima dual, então  $f(\mathbf{x}^*) = g(\boldsymbol{\lambda}^*)$  e vice-versa.
- Se  $f(\mathbf{x}) \rightarrow -\infty$  (primal ilimitado), então o problema dual é infactível.
- Se  $g(\boldsymbol{\lambda}) \rightarrow \infty$  (dual ilimitado), então o problema primal é infactível.
- Ambos os problemas, primal e dual, podem ser infactíveis.
- O vetor multiplicador simplex na solução básica ótima, dado por  $\boldsymbol{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1}$ , é uma solução ótima do problema dual.

### **2.10.3 Análise de sensibilidade (pós-otimização)**

Nesta seção, estudamos o efeito de se perturbar os dados de um problema de otimização linear, particularmente no vetor de recursos  $\mathbf{b}$  e no vetor de custos  $\mathbf{c}$ .

#### *Perturbação no vetor de recursos*

Considere o problema primal na forma padrão:

$$\begin{aligned} F(\mathbf{b}) = \text{Minimizar } f(\mathbf{x}) &= \mathbf{c}^T \mathbf{x} \\ \mathbf{Ax} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0}, \end{aligned}$$

em que  $F(\mathbf{b})$  é o valor mínimo da função objetivo para um vetor de recursos  $\mathbf{b}$ . Se o vetor de recursos for alterado, então a solução ótima também se altera e, por conseguinte,  $F(\mathbf{b})$  também é modificado. Estudaremos, a seguir, como as variáveis duais fornecem subsídios para estudar a modificação em  $F(\mathbf{b})$ . Suponha que o problema primal tenha sido resolvido, obtendo-se a solução básica primal ótima:

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}, \mathbf{x}_N = \mathbf{0}$$

e a solução básica dual ótima (vetor multiplicador simplex):

$$\boldsymbol{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1}.$$

Portanto,

$$F(\mathbf{b}) = \mathbf{c}_B^T \mathbf{x}_B = \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{b} = \boldsymbol{\lambda}^T \mathbf{b} = b_1\lambda_1 + b_2\lambda_2 + \dots + b_m\lambda_m$$

é a expressão da função objetivo para um vetor de recursos  $\mathbf{b}$  específico. Se a função  $F$  for diferenciável (é o que acontece se pequenas variações em  $\mathbf{b}$  não fazem mudar a base ótima, de modo que  $\boldsymbol{\lambda}$  não se altera), então

$$\frac{\partial F(\mathbf{b})}{\partial b_i} = \lambda_i, i = 1, \dots, m,$$

ou seja, o vetor multiplicador simplex (variáveis duais) fornece as derivadas parciais da função  $F$  em  $\mathbf{b}$ , isto é, a taxa de variação do valor da função objetivo com relação ao vetor de recursos.

Note que a unidade de  $\lambda_i b_i$  deve ser a mesma da função objetivo. Por exemplo, se a função objetivo representa custo (unidade monetária \$), então  $\lambda_i b_i$  também deve ser uma grandeza na mesma unidade monetária. Portanto, a unidade de  $\lambda_i$  deve ser \$(/unidade do recurso  $i$ ), que pode ser interpretado como o *preço* de uma unidade do recurso  $i$ . Em resumo, a unidade de  $\lambda_i$  é: (unidade da função objetivo)/(unidade do recurso  $i$ ). Se fizermos uma pequena perturbação em  $b_i$  (sem que a solução básica atual deixe de ser factível), digamos,  $b_i + \delta_i$  então:

$$\begin{aligned} F(b_1, \dots, b_i + \delta_i, \dots, b_m) &= b_1\lambda_1 + \dots + (b_i + \delta_i)\lambda_i + \dots + b_m\lambda_m \\ &= F(\mathbf{b}) + \delta_i\lambda_i \end{aligned}$$

ou seja, a função objetivo avaliada em  $\mathbf{b}$  é modificada por  $\delta_i\lambda_i$ .

**Exemplo 2.45** Considere o problema de corte de estoque no Exemplo 2.36 e seu problema dual:

**Primal:** Minimizar  $f(\mathbf{x}) = x_1 + x_2 + x_3$       **Dual:** Maximizar  $g(\boldsymbol{\lambda}) = 108\lambda_1 + 120\lambda_2$

$$\begin{array}{ll} \left\{ \begin{array}{l} \frac{4}{5}x_1 + \frac{2}{5}x_2 = 108 \\ \frac{3}{5}x_2 + \frac{9}{10}x_3 = 120 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{array} \right. & \left\{ \begin{array}{l} \frac{4}{5}\lambda_1 + 0\lambda_2 + \mu_1 = 1 \\ \frac{2}{5}\lambda_1 + \frac{3}{5}\lambda_2 + \mu_2 = 1 \\ 0\lambda_1 + \frac{9}{10}\lambda_2 + \mu_3 = 1 \\ \mu_1 \geq 0, \mu_2 \geq 0, \mu_3 \geq 0 \end{array} \right. \end{array}$$

com solução ótima primal,  $x_1 = 35$ ,  $x_2 = 200$ ,  $x_3 = 0$  e solução ótima dual  $\boldsymbol{\lambda} = \left(\frac{5}{4}, \frac{5}{6}\right)^T$ . A unidade da função objetivo é o *peso* das bobinas cortadas. Assim,  $F(108, 120) = 235$  significa que 235 toneladas de bobinas são necessárias para atender a demanda de  $b_1 = 108$  e  $b_2 = 120$  toneladas de sub-bobinas dos tipos 1 e 2 (lembre-se de que  $\mathbf{b} = (108, 120)^T$ ). Além disso,  $\bar{F}(108 + \delta_1, 120) = 235 + \frac{5}{4}\delta_1$  e  $F(108, 120 + \delta_2) = 235 + \frac{5}{6}\delta_2$ . Portanto, cada tonelada a mais da sub-bobina do tipo 1 (comprimento 0,4m) necessita de  $\frac{5}{4}$  toneladas a mais de bobina para ser cortada, ou seja,  $\frac{5}{2}$  da necessidade caso a alteração fosse da sub-bobina de tipo 2 (comprimento 0,3m). Dessa forma, se possível, é mais interessante estimular a demanda de sub-bobinas do tipo 2 do que do tipo 1.

Esta análise é válida para as demandas  $b_1 = 108$  e  $b_2 = 120$ ; para outros valores de demanda, a análise muda. Deve ficar claro para o leitor que esses valores representam *tendências* e são válidos apenas para pequenas perturbações na demanda.

Podemos determinar também, por exemplo, qual seria o valor de  $\delta_2$  que mantém essa tendência, ou seja, não provoca mudança na solução ótima, com  $\delta_1 = 0$ , isto é, mantendo-se fixa a demanda do item 1:

Como  $\mathbf{B}^{-1} = \begin{bmatrix} \frac{5}{4} & -\frac{5}{6} \\ 0 & \frac{5}{3} \end{bmatrix}$ , então:

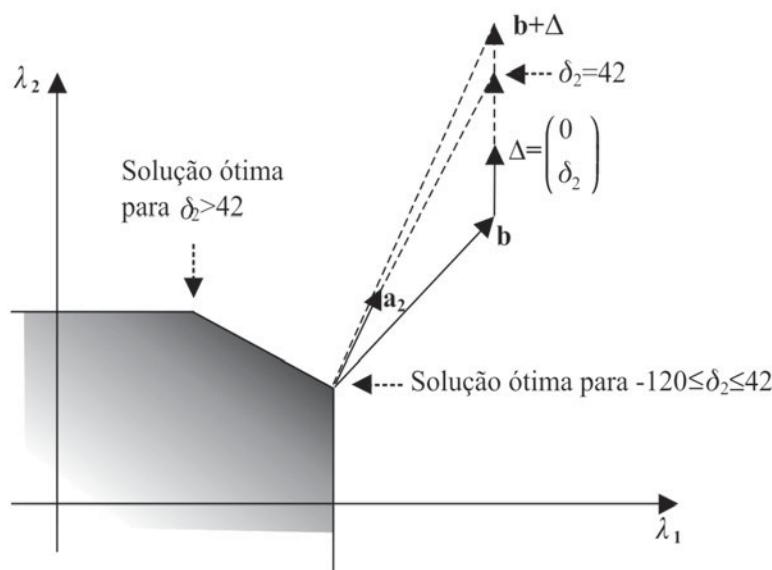
$$\mathbf{x}'_{\mathbf{B}} = \mathbf{B}^{-1}(\mathbf{b} + \Delta) = \mathbf{x}_{\mathbf{B}} + \mathbf{B}^{-1}\Delta = \begin{pmatrix} 35 \\ 200 \end{pmatrix} + \begin{bmatrix} \frac{5}{4} & -\frac{5}{6} \\ 0 & \frac{5}{3} \end{bmatrix} \begin{pmatrix} 0 \\ \delta_2 \end{pmatrix} = \begin{pmatrix} 35 - \frac{5}{6}\delta_2 \\ 200 + \frac{5}{3}\delta_2 \end{pmatrix} \quad (2.41)$$

Portanto, para que a não-negatividade seja mantida, isto é:

$$x'_1 = 35 - \frac{5}{6}\delta_2 \geq 0 \quad \text{e} \quad x'_2 = 200 + \frac{5}{3}\delta_2 \geq 0$$

devemos ter  $-120 \leq \delta_2 \leq 42$ . Para  $\delta_2 = 42$ , a demanda do item 2 é de  $120 + \delta_2 = 162$  toneladas, a solução ótima é  $x'_1 = 0$ ,  $x'_2 = 270$  e  $x'_3 = 0$  e o valor da função objetivo é  $f' = 270$  toneladas. Em relação à solução anterior, note que 35 toneladas a mais devem ser cortadas, porém, para atender a um acréscimo de 42 toneladas na demanda (isto ocorre devido ao fato de o desperdício no padrão de corte 2 ser nulo e, agora, somente ele ser necessário para a produção dos itens demandados). Para valores além deste limite,  $\delta_2 > 42$  em (2.41), a solução básica deixa de ser factível, em particular,  $x'_1 < 0$  (portanto, deixa de ser ótima), de modo que as tendências se alteram. De modo semelhante, para  $\delta_2 < -120$ , a solução deixa de ser ótima (em particular, a demanda  $120 + \delta_2 < 0$ , portanto, o problema primal torna-se infactível ou o dual ilimitado). A Figura 2.43 ilustra perturbações em  $b_2$  no espaço dual.

Os vetores  $\mathbf{a}_2$  e  $\mathbf{b} + \Delta$  se alinham com  $\delta_2 = 42$  (o leitor pode verificar que  $270\mathbf{a}_2 = \mathbf{b} + \Delta$ ) quando o problema dual tem múltiplas soluções ótimas (a solução primal é degenerada, pois  $x'_1 = 0$ ). Para  $\delta_2 > 42$ , a solução dual  $\lambda = (\frac{5}{4} \frac{5}{6})^T$  deixa de ser ótima, a base ótima passa a ser (a primeira restrição deixa de ser ativa e a terceira torna-se ativa):  $\mathbf{B} = [\mathbf{a}_2 \mathbf{a}_3] = \begin{bmatrix} \frac{2}{5} & 0 \\ \frac{3}{5} & \frac{9}{10} \end{bmatrix}$  e a solução ótima dual



**Figura 2.43** Representação da variação do vetor de recursos no espaço dual.

é  $\lambda' = \left(\frac{5}{6} \quad \frac{10}{9}\right)^T$ , o que significa que, se  $\delta_2 > 42$ , então a tendência de mudança na função objetivo é  $\frac{10}{9}$  por tonelada de sub-bobina do tipo 2 demandada, ou seja, para cada tonelada demandada a mais da sub-bobina do tipo 2 há uma necessidade de se cortar  $\frac{10}{9}$  tonelada a mais em bobinas (isso porque a nova solução introduz um desperdício). O leitor pode esboçar o vetor  $\mathbf{b} + \Delta$  para valores em que  $\delta_2 < -120$  (para  $\delta_2 = -120$ , note que os vetores  $\mathbf{a}_3$  e  $\mathbf{b} + \Delta$  ficam alinhados). Neste caso,  $g(\lambda) \rightarrow \infty$  (o problema primal torna-se infactível, já que  $b_2 = 120 + \delta_2 < 0$ ). Podemos representar em um gráfico (Figura 2.44) o valor da função objetivo em termos de  $\delta_2$ .

É comum que os pacotes comerciais de otimização linear forneçam os intervalos de variação dos valores de  $b_i$ ,  $i = 1, 2, \dots, m$ , que não alteram a otimalidade da base corrente. Esses intervalos são dados por:

$$[b_i + \underline{\delta}_i, b_i + \bar{\delta}_i]$$

em que  $\underline{\delta}_i$  e  $\bar{\delta}_i$  são os limites mínimo e máximo para  $\delta_i$ . No Exemplo 2.45, os limites determinados para  $\delta_2$  são:  $\underline{\delta}_2 = -120$ ,  $\bar{\delta}_2 = 42$ , de modo que  $b_2$  pode ter qualquer valor no intervalo  $[0 \ 162]$  sem que a base deixe de ser ótima. Além disso, o valor de  $\lambda_2 = \frac{5}{6}$  fornece a taxa de variação na função objetivo com a variação de  $b_2$ . Essa informação é útil para o decisior, que pode analisar até que ponto um recurso pode ser alterado sem que as atividades relevantes para uma solução particular sejam alteradas e qual o impacto da alteração na função objetivo. No exemplo de corte de peças, as atividades relevantes para a demanda  $b_1 = 108$  e  $b_2 = 120$  eram o corte de bobinas segundo os padrões 1 e 2. Dessa forma, mantendo-se  $b_1 = 108$ , qualquer que seja  $b_2$  no intervalo  $[0 \ 162]$ , esses padrões permanecem utilizados. Se  $b_2 > 162$ , então as atividades (padrões de corte) utilizadas são outras.

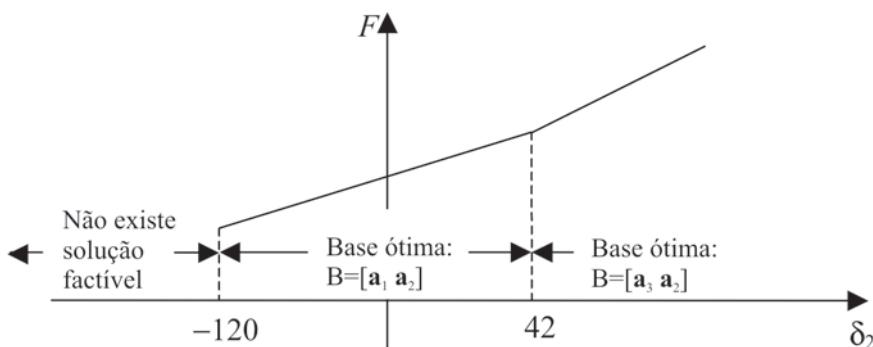
## *Perturbação nos custos*

Considere que uma solução básica ótima tenha sido encontrada para um problema de otimização linear. Deseja-se agora analisar o efeito da alteração de algum coeficiente da função objetivo, digamos, alterar  $c_k$  para  $c'_k = c_k + \delta$ . Essa alteração não torna infactível a solução ótima já encontrada, porém esta pode deixar de ser uma solução ótima. A variável  $x_k$  pode ser básica ou não-básica. Analisemos os dois casos.

- A variável  $x_k$  é não-básica

Neste caso, o vetor  $\mathbf{c}_B$  não é alterado, de modo que o vetor multiplicador simplex  $\lambda^T = \mathbf{c}_B^T \mathbf{B}^{-1}$  associado à partição básica ótima não se altera e apenas o custo relativo da variável  $x_k$ , que denotamos por  $\hat{c}_k$ , é alterado:

$$\hat{c}'_k = c'_k - \boldsymbol{\lambda}^T \mathbf{a}_k = (c_k + \delta) - \boldsymbol{\lambda}^T \mathbf{a}_k = \hat{c}_k + \delta.$$



**Figura 2.44** Mudança na função objetivo com a alteração de  $b_1$ .

Isto é, o novo custo relativo é simplesmente alterado pela adição de  $\delta$  (a quantidade que faz variar o custo da variável  $x_k$ ) ao antigo custo relativo. Assim, se  $\hat{c}'_k = \hat{c}_k + \delta \geq 0$ , então a alteração no custo não altera a solução ótima. Se  $\hat{c}'_k = \hat{c}_k + \delta < 0$ , então a solução deixa ser ótima. O método simplex pode ser usado a partir da antiga solução ótima e a variável  $x_k$  entra na base. Como  $\hat{c}_k \geq 0$  (pois a solução básica é ótima para o problema sem a alteração), vemos que necessariamente  $\delta < 0$  para que a solução ótima se altere (isto é, o custo da variável  $x_k$  deve diminuir para que  $x_k$  entre na base). Mais especificamente, se  $\delta < -\hat{c}_k$ , então a variável  $x_k$  deve ser aumentada para reduzir ainda mais a função objetivo. Essa análise pode ser empregada para estudo de viabilidade econômica de um novo produto, como no Exemplo 2.46.

**Exemplo 2.46** Considere o Exemplo 2.6, na Seção 2.2, do fabricante de geladeiras. Um novo modelo superluxo está em estudo e deseja-se saber qual é o lucro unitário mínimo desse produto (definido como o preço de venda menos o custo do produto) que torna sua produção viável economicamente em relação aos demais produtos. O modelo matemático com os produtos iniciais em sua forma padrão e sua solução ótima são dados a seguir (ver também Exercício 2.10).

$$\text{Maximizar } f(x_{luxo}, x_{básico}) = 100x_{luxo} + 50x_{básico}$$

$$\begin{aligned} 10x_{luxo} + 8x_{básico} + x_1 &= 25.000 \\ x_{luxo} + x_{básico} + x_2 &= 4.500 \\ x_{luxo} + x_3 &= 1.500 \\ x_{básico} + x_4 &= 6.000 \\ x_{luxo} \geq 0, x_{básico} \geq 0, x_j \geq 0, j &= 1, \dots, 4, \end{aligned}$$

cuja base ótima e as soluções básicas ótimas, primal e dual, são dadas por:

$$\mathbf{B} = [\mathbf{a}_{luxo} \ \mathbf{a}_{básico} \ \mathbf{a}_2 \ \mathbf{a}_4] = \begin{bmatrix} 10 & 8 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \quad \mathbf{x}_B = \begin{pmatrix} x_{luxo} \\ x_{básico} \\ x_2 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1500 \\ 1250 \\ 1750 \\ 4750 \end{pmatrix}, \quad \boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{pmatrix} = \begin{pmatrix} \frac{25}{4} \\ 0 \\ \frac{75}{2} \\ 0 \end{pmatrix}.$$

O modelo novo superluxo requer 16 homens-hora para ser montado, e compartilha a linha de montagem com os demais modelos. Se definirmos a variável  $x_{sluxo}$  como a quantidade produzida do novo modelo e  $l_{sluxo}$ , o lucro unitário, então o modelo inicial é alterado para (uma nova coluna é inserida):

$$\text{Maximizar } f(x_{luxo}, x_{básico}, x_{luxo}) = 100x_{luxo} + 50x_{básico} + l_{sluxo}x_{sluxo}$$

$$\begin{aligned} 10x_{luxo} + 8x_{básico} + 16x_{sluxo} + x_1 &= 25.000 \\ x_{luxo} + x_{básico} + x_{sluxo} + x_2 &= 4.500 \\ x_{luxo} + x_3 &= 1.500 \\ x_{básico} + x_4 &= 6.000 \\ x_{luxo} \geq 0, x_{básico} \geq 0, x_j \geq 0, j &= 1, \dots, 4 \end{aligned}$$

A variável  $x_{sluxo}$  é economicamente viável se pertencer à base ótima deste novo problema, ou seja, se fizer parte do *mix* de produção. Caso contrário, será não-básica e seu nível de produção será nulo. Para isso, devemos calcular seu custo relativo, que deve ser não-negativo (problema de maximização):

Custo relativo da variável  $x_{sluxo}$ :

$$l_{sluxo} - \boldsymbol{\lambda}^T \mathbf{a}_{sluxo} = l_{sluxo} - \left( \frac{25}{4} \quad 0 \quad \frac{75}{2} \quad 0 \right) \begin{pmatrix} 16 \\ 1 \\ 0 \\ 0 \end{pmatrix} = l_{sluxo} - 100 \geq 0.$$

Portanto, se  $l_{sluxo} \geq 100$ , então o modelo novo superluxo é viável economicamente (em relação aos demais produtos) e contribuirá para o lucro da empresa se for produzido. Caso contrário, a empresa precisa aumentar seu preço no mercado ou reduzir seu custo unitário de produção, de modo que seu lucro unitário resulte em  $l_{sluxo} \geq 100$ . Note que, para  $l_{sluxo} = 100$ , o custo relativo é nulo, indicando que existem soluções ótimas alternativas. ■

Analisamos agora o caso em que o custo de uma variável básica é alterado.

- A variável  $x_k$  é básica

Como  $x_k$  é básica, suponha que seja a  $r$ -ésima variável básica:  $x_{B_r} = x_k$ . Neste caso,  $c'_{B_r} = c_{B_r} + \delta$  e o vetor multiplicador simplex associado à base encontrada se altera por:

$$\boldsymbol{\lambda}'^T = \mathbf{c}_B^T \mathbf{B}^{-1} = \mathbf{c}_B^T \mathbf{B}^{-1} + \delta \mathbf{e}_r^T \mathbf{B}^{-1} = \boldsymbol{\lambda}^T + \delta \mathbf{e}_r^T \mathbf{B}^{-1},$$

em que  $\mathbf{e}_r$  é a  $r$ -ésima coluna da matriz identidade  $m \times m$ . Portanto, os custos relativos de todas as variáveis devem ser recalculados como segue:

$$\hat{c}'_j = c_j - \boldsymbol{\lambda}'^T \mathbf{a}_j = c_j - (\boldsymbol{\lambda}^T + \delta \mathbf{e}_r^T \mathbf{B}^{-1}) \mathbf{a}_j = (c_j - \boldsymbol{\lambda}^T \mathbf{a}_j) - \underbrace{\delta \mathbf{e}_r^T \mathbf{B}^{-1} \mathbf{a}_j}_{\mathbf{y}_j} = \hat{c}_j - \delta \mathbf{e}_r^T \mathbf{y}_j.$$

Como  $\mathbf{e}_r$  é a  $r$ -ésima coluna da matriz identidade, então o produto  $\mathbf{e}_r^T \mathbf{y}_j$  fornece o  $r$ -ésimo elemento do vetor  $\mathbf{y}_j$ , denotado por  $y_{rj}$ , de modo que o novo custo relativo é dado por:

$$\hat{c}'_j = \hat{c}_j - \delta y_{rj}.$$

Neste caso, todos os custos relativos se alteram.

#### 2.10.4\* O método dual simplex

O método dual simplex é uma aplicação do método simplex ao problema dual. Um dos principais usos deste método ocorre em reotimizações de problemas de otimização linear, quando, após o problema inicial ter sido resolvido, novas restrições são adicionadas (problema novo). Se a solução ótima do problema original (já determinada) satisfaz as novas restrições, então ela também é ótima para o problema novo. Caso contrário, a solução ótima do problema original é infactível para o problema novo, e uma reotimização deste problema usando o método simplex (Seção 2.6) necessita da introdução de variáveis artificiais (Fase I). Entretanto, do ponto de vista do problema dual, novas restrições ao problema primal implicam novas variáveis duais, de modo que a solução dual não perde a factibilidade, porém pode deixar de ser ótima. Mas a solução ótima do problema dual original é, em geral, uma boa solução inicial para o problema dual novo. No Capítulo 3 – otimização discreta –, vemos como introduzir novas restrições para obter uma solução ótima inteira. Neste caso, o uso do método dual simplex para resolver o novo problema é uma boa alternativa a ser considerada.

Utilizamos um exemplo simples e uma abordagem geométrica para deduzir o método dual simplex. Considere o par de problemas primal – dual:

**Primal:**

$$\begin{aligned} \text{Minimizar } f(\mathbf{x}) &= 2x_1 + 3x_2 + 2x_3 + x_4 \\ -x_1 + x_2 + 2x_3 - x_4 &= 2 \\ 2x_1 + x_2 - x_3 &= 1 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0 & \end{aligned}$$

**Dual:**

$$\begin{aligned} \text{Maximizar } g(\boldsymbol{\lambda}) &= 2\lambda_1 + \lambda_2 \\ -\lambda_1 + 2\lambda_2 &\leq 2 \\ \lambda_1 + \lambda_2 &\leq 3 \\ 2\lambda_1 - \lambda_2 &\leq 2 \\ -\lambda_1 &\leq 1 \end{aligned}$$

As colunas da matriz de restrições do problema primal são denotadas por  $\mathbf{a}_j, j = 1, 2, 3, 4$ , as quais definem os coeficientes das restrições do problema dual,  $\mathbf{a}_j^T \boldsymbol{\lambda} \leq c_j, j = 1, 2, 3, 4$ . Na Figura 2.45, apresentamos a resolução gráfica do problema dual.

Analisemos inicialmente o vértice ótimo, obtido pela intersecção das retas:

$$\begin{aligned} \lambda_1 + \lambda_2 &= 3 & (\mathbf{a}_2^T \boldsymbol{\lambda} = c_2) \\ 2\lambda_1 - \lambda_2 &= 2 & (\mathbf{a}_3^T \boldsymbol{\lambda} = c_3), \end{aligned}$$

cuja solução é:  $\lambda_1^* = \frac{5}{3}$ ,  $\lambda_2^* = \frac{4}{3}$  e o valor da função objetivo dual é:

$$g(\boldsymbol{\lambda}^*) = \mathbf{b}^T \boldsymbol{\lambda}^* = 2\lambda_1^* + \lambda_2^* = 2 \times \frac{5}{3} + \frac{4}{3} = \frac{14}{3}.$$

As demais restrições duais são satisfeitas com desigualdades. Dizemos que as restrições duais 2 e 3 são restrições *ativas* e as restrições duais 1 e 4 são *inativas*. A solução ótima primal também pode ser obtida ao se observar que o vetor  $\mathbf{b}$  é uma combinação positiva das colunas  $\mathbf{a}_2$  e  $\mathbf{a}_3$  (ver Figura 2.46).

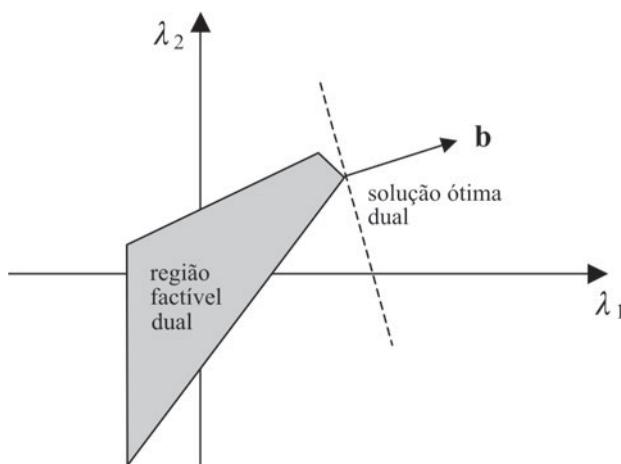
O vetor  $\mathbf{b}$  está no cone gerado pelos vetores  $\mathbf{a}_2$  e  $\mathbf{a}_3$ , o que significa que  $\mathbf{b}$  é uma combinação positiva dos vetores  $\mathbf{a}_2$  e  $\mathbf{a}_3$ . De outra forma, o sistema linear  $\mathbf{a}_2 x_2 + \mathbf{a}_3 x_3 = \mathbf{b}$ , ou seja,

$$\begin{bmatrix} 1 & 2 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

tem solução não-negativa  $x_2^* = \frac{4}{3}$ ,  $x_3^* = \frac{1}{3}$  (que fornecem os múltiplos de  $\mathbf{a}_2$  e  $\mathbf{a}_3$  na Figura 2.46). O valor da função objetivo primal, considerando  $x_1^* = x_4^* = 0$ , é dado por:

$$f(\mathbf{x}^*) = 2x_1^* + 3x_2^* + 2x_3^* + x_4^* = 2 \times 0 + 3 \times \frac{4}{3} + 2 \times \frac{1}{3} + 1 \times 0 = \frac{14}{3},$$

o que coincide com  $g(\boldsymbol{\lambda}^*)$  (veja a propriedade 2.13).



**Figura 2.45** Solução gráfica do problema dual.

Esta solução primal fornece a partição básica ótima do problema primal:

$$\mathbf{B} = [\mathbf{a}_2 \mathbf{a}_3] \quad \text{e} \quad \mathbf{N} = [\mathbf{a}_1 \mathbf{a}_4]$$

em que as colunas básicas são formadas pelos coeficientes das restrições ativas no dual. Note também que as restrições ativas (satisfetidas com a igualdade na solução ótima):

$$\begin{aligned}\lambda_1 + \lambda_2 &= 3 & (\mathbf{a}_2^T \boldsymbol{\lambda} = c_2) \\ 2\lambda_1 - \lambda_2 &= 2 & (\mathbf{a}_3^T \boldsymbol{\lambda} = c_3),\end{aligned}$$

podem ser escritas na forma matricial como:

$$\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B$$

cuja solução  $\hat{\boldsymbol{\lambda}} = (\mathbf{B}^{-1})^T \mathbf{c}_B$  é o vetor multiplicador simplex associado à partição básica ótima (ver a propriedade 2.14). Este exemplo simples nos sugere que:

- O vértice dual ótimo  $\hat{\boldsymbol{\lambda}}$  é obtido de um conjunto de restrições duais ativas  $\mathbf{a}_i^T \hat{\boldsymbol{\lambda}} = c_i$ , em que  $\mathbf{a}_i$  é a  $i$ -ésima coluna da matriz  $\mathbf{A}$ .
- As colunas de  $\mathbf{A}$ , associadas às restrições duais ativas, definem uma base ótima, a qual permite determinar a solução ótima primal.

Analisemos agora um outro vértice dual factível, porém não ótimo, obtido pelas seguintes restrições ativas:

$$\begin{aligned}-\lambda_1 + 2\lambda_2 &= 2 & (\mathbf{a}_1^T \boldsymbol{\lambda} = c_1) \\ \lambda_1 + \lambda_2 &= 3 & (\mathbf{a}_2^T \boldsymbol{\lambda} = c_2),\end{aligned}$$

cuja solução é  $\hat{\lambda}_1 = \frac{4}{3}$ ,  $\hat{\lambda}_2 = \frac{5}{3}$  e o valor da função objetivo dual é

$$g(\hat{\boldsymbol{\lambda}}) = \mathbf{b}^T \hat{\boldsymbol{\lambda}} = 2\hat{\lambda}_1 + \hat{\lambda}_2 = 2 \times \frac{4}{3} + \frac{5}{3} = \frac{13}{3}.$$

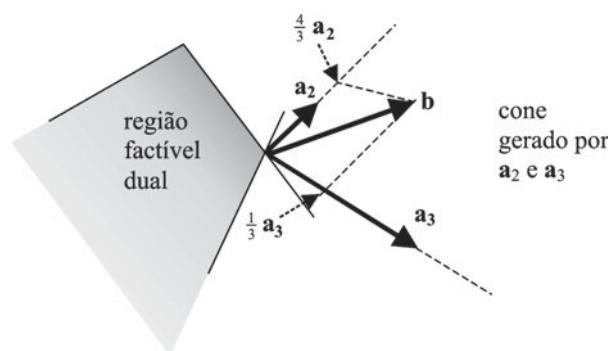
Note na Figura 2.47 que o vetor  $\mathbf{b}$  não está no cone gerado por  $\mathbf{a}_1$  e  $\mathbf{a}_2$ , isto é,  $\mathbf{b}$  não é uma combinação linear não-negativa de  $\mathbf{a}_1$  e  $\mathbf{a}_2$ . Uma das variáveis do sistema linear  $\mathbf{a}_1 x_1 + \mathbf{a}_2 x_2 = \mathbf{b}$  será negativa (na Figura 2.47 há indicações de que a variável  $x_1$  deve ser negativa). De fato, o sistema

$$\begin{bmatrix} -1 & 1 \\ 2 & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

fornecerá a solução  $\hat{x}_1 = -\frac{1}{3}$ ,  $\hat{x}_2 = \frac{5}{3}$ . Diferentemente da situação anterior, agora a partição básica:

$$\mathbf{B} = [\mathbf{a}_1 \mathbf{a}_2] \quad \text{e} \quad \mathbf{N} = [\mathbf{a}_3 \mathbf{a}_4]$$

não é primal factível (isto é, a solução do sistema  $\mathbf{Bx}_B = \mathbf{b}$  é  $\hat{\mathbf{x}}_B = \mathbf{B}^{-1}\mathbf{b} \not\geq 0$ ).



**Figura 2.46** O vetor  $\mathbf{b}$  como combinação positiva dos gradientes das restrições ativas.

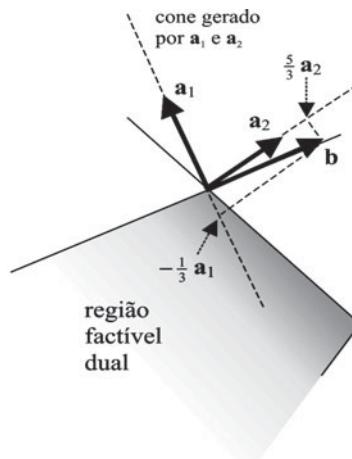
Note também que o vértice  $\hat{\lambda}$  é a solução do sistema  $\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B$ . Não sendo um vértice ótimo dual, devemos procurar outro vértice que seja melhor (isto é, o valor da função objetivo dual é maior). Para isso, perturbamos  $\hat{\lambda}$  em uma direção  $\eta$  de modo que leve a um outro vértice melhor. Pela Figura 2.48, podem ser observadas outras direções que levam a outros vértices.

As direções  $\eta_1$  e  $\eta_2$  (chamadas *direções dual simplex*) permitem perturbar a solução  $\hat{\lambda}$  ao longo das arestas da região factível dual. A Figura 2.48 ilustra a perturbação de  $\hat{\lambda}$  na direção  $\eta_1$ :

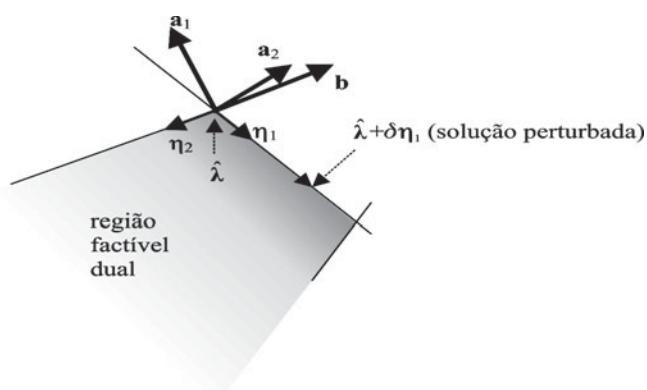
$$\boldsymbol{\lambda} = \hat{\lambda} + \delta\eta_1, \delta \geq 0$$

Para determinar a direção  $\eta_1$  (como também a direção  $\eta_2$ ), usamos uma estratégia para perturbar  $\hat{\lambda}$  chamada *estratégia dual simplex*, dada por:

- A restrição 1 deixa de ser ativa (apenas uma restrição torna-se inativa).
- A restrição 2 permanece ativa (exceto uma, todas as demais permanecem ativas).



**Figura 2.47** O vetor  $b$  não é combinação positiva dos gradientes das restrições ativas.



**Figura 2.48** Direção dual simplex e solução perturbada.

Essa estratégia pode ser escrita como:

$$\begin{cases} \mathbf{a}_1^T \boldsymbol{\lambda} = c_1 - \delta, & \delta \geq 0, \\ \mathbf{a}_2^T \boldsymbol{\lambda} = c_2. \end{cases}$$

Em notação matricial:

$$\begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \end{bmatrix} \boldsymbol{\lambda} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} - \begin{bmatrix} \delta \\ 0 \end{bmatrix}$$

e, denotando  $\mathbf{B} = [\mathbf{a}_1 \ \mathbf{a}_2]$ ,  $\mathbf{c}_B = (c_1 \ c_2)^T$ ,  $\mathbf{e}_1 = (1 \ 0)^T$ , o sistema anterior é reescrito como:

$$\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B - \delta \mathbf{e}_1$$

de modo que:

$$\boldsymbol{\lambda} = (\mathbf{B}^T)^{-1} \mathbf{c}_B - \delta (\mathbf{B}^T)^{-1} \mathbf{e}_1 = \hat{\boldsymbol{\lambda}} + \delta \boldsymbol{\eta}_1$$

em que  $\boldsymbol{\eta}_1 = -(\mathbf{B}^{-1})^T \mathbf{e}_1$  é chamada *direção dual simplex*.

A direção dual simplex  $\boldsymbol{\eta}_1$  é a primeira coluna de  $-(\mathbf{B}^{-1})^T$  ou, dito de outra forma, é a transposta da primeira linha de  $\mathbf{B}^{-1}$  multiplicada por  $-1$ . Analogamente, a direção dual simplex  $\boldsymbol{\eta}_2 = -(\mathbf{B}^{-1})^T \mathbf{e}_2$  pode ser obtida pela estratégia dual simplex de manter ativa a primeira restrição e inativa a segunda. Calculamos  $\boldsymbol{\eta}_i$ ,  $i = 1, 2$ , para o exemplo anterior:

$$\mathbf{B} = [\mathbf{a}_1 \ \mathbf{a}_2] = \begin{bmatrix} -1 & 1 \\ 2 & 1 \end{bmatrix} \quad \text{e} \quad \mathbf{B}^{-1} = \begin{bmatrix} -\frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix}$$

Portanto,  $(-1) \times$  (transposta da primeira linha de  $\mathbf{B}^{-1}$ ):

$$\boldsymbol{\eta}_1 = \begin{bmatrix} \frac{1}{3} \\ -\frac{1}{3} \end{bmatrix} \quad \text{e} \quad \boldsymbol{\eta}_2 = \begin{bmatrix} -\frac{2}{3} \\ -\frac{1}{3} \end{bmatrix}.$$

**Propriedade 2.15** Considere uma partição básica  $\mathbf{A} = [\mathbf{B} \ \mathbf{N}]$ , a solução básica primal associada  $\hat{\mathbf{x}}_B = \mathbf{B}^{-1} \mathbf{b}$  e a direção  $\boldsymbol{\eta}_i$  definida por  $\boldsymbol{\eta}_i = -(\mathbf{B}^{-1})^T \mathbf{e}_i$ . Então,  $\mathbf{b}^T \boldsymbol{\eta}_i = -\hat{x}_{B_i}$ .

A validade da propriedade vem da álgebra matricial:

$$\mathbf{b}^T \boldsymbol{\eta}_i = \mathbf{b}^T (-(\mathbf{B}^{-1})^T \mathbf{e}_i) = -(\mathbf{B}^{-1} \mathbf{b})^T \mathbf{e}_i = -\hat{\mathbf{x}}_B^T \mathbf{e}_i = -\hat{x}_{B_i}.$$

A propriedade 2.15 pode ser verificada no exemplo anterior:

$$\mathbf{b}^T \boldsymbol{\eta}_1 = (2 \quad 1) \begin{pmatrix} \frac{1}{3} \\ -\frac{1}{3} \end{pmatrix} = \frac{1}{3} = -\hat{x}_{B_1}$$

Consideremos agora a nova solução dual  $\boldsymbol{\lambda} = \hat{\boldsymbol{\lambda}} + \delta \boldsymbol{\eta}_1$ ,  $\delta \geq 0$ , e avaliemos a função objetivo dual:

$$\begin{aligned} g(\boldsymbol{\lambda}) &= g(\hat{\boldsymbol{\lambda}} + \delta \boldsymbol{\eta}_1) = \mathbf{b}^T (\hat{\boldsymbol{\lambda}} + \delta \boldsymbol{\eta}_1) = \mathbf{b}^T \hat{\boldsymbol{\lambda}} + \delta \mathbf{b}^T \boldsymbol{\eta}_1 \\ &= g(\hat{\boldsymbol{\lambda}}) - \delta \hat{x}_{B_1}. \end{aligned}$$

Para o exemplo anterior, em que  $g(\hat{\boldsymbol{\lambda}}) = \frac{13}{3}$ ,  $\hat{x}_1 = -\frac{1}{3}$  segue-se que:

$$g(\boldsymbol{\lambda}) = g(\hat{\boldsymbol{\lambda}}) - \delta x_{B_1} = \frac{13}{3} + \frac{1}{3} \delta.$$

Vemos que a função objetivo dual cresce quando  $\delta$  cresce, ou seja, perturbar a solução factível dual  $\hat{\boldsymbol{\lambda}}$  na direção  $\boldsymbol{\eta}_1$  faz crescer a função objetivo dual e a taxa de crescimento é  $\mathbf{b}^T \boldsymbol{\eta}_1 = -x_{B_1}$ . Adiante reescrevemos estas considerações como uma propriedade geral. Antes, façamos uma revisão do que este exemplo simples nos sugere:

- Um vértice dual factível  $\hat{\lambda}$  é obtido por um conjunto de restrições duais ativas, as quais definem uma partição na matriz  $\mathbf{A}$ :  $\mathbf{A} = [\mathbf{B} \ \mathbf{N}]$ , em que as colunas em  $\mathbf{B}$  definem as restrições duais ativas.
- Se a solução básica primal  $\hat{\mathbf{x}}_B = \mathbf{B}^{-1}\mathbf{b}$ , associada à partição básica for não-negativa (isto é, o vetor  $\mathbf{b}$  está no cone gerado pelas colunas básicas, que, em outras palavras, significa que a solução básica primal associada é uma solução factível para o problema primal), então a solução  $\hat{\lambda}$  é ótima dual.
- Se não, isto é, se  $\hat{\mathbf{x}}_B$  tiver algum componente negativo, digamos,  $\hat{x}_{B_\ell} < 0$ , então uma estratégia simples, chamada estratégia dual simplex, permite perturbar a solução  $\hat{\lambda}$  para encontrar uma nova solução melhor.
- A estratégia dual simplex leva a uma direção que é a transposta da  $\ell$ -ésima linha de  $\mathbf{B}^{-1}$  multiplicado por  $-1$ , que denotamos por  $\boldsymbol{\eta}_\ell$ , que fornece uma forma de perturbar a solução dual  $\hat{\lambda}$ , ou seja,  $\boldsymbol{\lambda} = \hat{\lambda} + \delta\boldsymbol{\eta}_\ell$ , que faz a função objetivo dual crescer à taxa de  $-\hat{x}_{B_\ell}$ .

Estudamos, a seguir, como determinar o passo  $\delta$  e obter um novo vértice com valor maior para a função objetivo dual. Antes, generalizamos o procedimento que o exemplo anterior sugere.

Considere uma partição básica sobre as colunas da matriz  $\mathbf{A}$ :  $\mathbf{A} = [\mathbf{B} \ \mathbf{N}]$ . Cada coluna da matriz  $\mathbf{A}$  fornece uma restrição do problema dual e, portanto, a partição básica fornece uma organização nas restrições duais (lembre-se que a partição básica também fornece uma organização nas variáveis primais):

$$\mathbf{A}^T \boldsymbol{\lambda} \leq \mathbf{c} \Leftrightarrow \begin{cases} \mathbf{B}^T \boldsymbol{\lambda} \leq \mathbf{c}_B & \leftarrow \text{restrições básicas} \\ \mathbf{N}^T \boldsymbol{\lambda} \leq \mathbf{c}_N & \leftarrow \text{restrições não-básicas} \end{cases}$$

As restrições básicas são as restrições ativas no exemplo anterior, responsáveis pela determinação do vértice  $\hat{\lambda}$  (reveja as Figuras 2.46 e 2.47).

**Definição 2.13** (*solução básica dual*) Considere uma partição básica  $\mathbf{A} = [\mathbf{B} \ \mathbf{N}]$ . Chamamos de solução básica dual, associada à partição básica, a solução do sistema  $\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B$  (a qual denotamos por  $\hat{\lambda} = (\mathbf{B}^{-1})^T \mathbf{c}_B$ , que algumas vezes preferimos sua forma transposta  $\hat{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1}$ ). Se as demais restrições duais (isto é, as restrições não-básicas) são satisfeitas, ou seja,  $\mathbf{N}^T \hat{\lambda} \leq \mathbf{c}_N$ , então  $\hat{\lambda}$  é uma solução básica dual factível.

A notação introduzida na Seção 2.6.1 é utilizada quando necessária e define os conjuntos de índices básicos e não-básicos:

$$\mathbf{B} = \quad , \quad \mathbf{N} = [\mathbf{a}_{N_1} \ \mathbf{a}_{N_2} \dots \mathbf{a}_{N_{n-m}}], \quad \mathbf{c}_B = (c_{B_1} \ c_{B_2} \dots c_{B_m})^T, \quad \mathbf{c}_N = (c_{N_1} \ c_{N_2} \dots c_{N_{n-m}})^T$$

### Observações:

- (i) Uma solução básica dual factível corresponde a um vértice da região factível dual, como sugere o exemplo anterior (esta propriedade é válida em geral).
- (ii) A solução básica dual  $\hat{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1}$  é o conhecido vetor multiplicador simplex associado a uma partição básica e os custos relativos das variáveis não-básicas são:  $\hat{c}_{N_j} = c_{N_j} - \hat{\lambda}^T \mathbf{a}_{N_j}$ .
- (iii) A factibilidade dual:  $\mathbf{N}^T \hat{\lambda} \leq \mathbf{c}_N$  (isto é,  $\mathbf{a}_{N_j}^T \hat{\lambda} \leq c_{N_j}$ ,  $j = 1, \dots, n-m$ ) é equivalente a:  $\hat{c}_{N_j} = c_{N_j} - \hat{\lambda}^T \mathbf{a}_{N_j} \geq 0$ ,  $j = 1, \dots, n-m$ , ou seja, os custos relativos são não-negativos. Portanto, a condição de otimalidade (ver propriedade 2.3) é satisfeita.
- (iv) O método simplex (Seção 2.6) trabalha com a solução básica primal factível, isto é,  $\hat{\mathbf{x}}_B = \mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$  e admite algum custo relativo negativo  $\hat{c}_{N_j} = c_{N_j} - \hat{\lambda}^T \mathbf{a}_{N_j} < 0$ , o que significa que a restrição dual  $\mathbf{a}_{N_j}^T \hat{\lambda} \leq c_{N_j}$  está violada. Quando todos os custos relativos são não-negativos (ver (iii)), a solução básica dual é factível e o método simplex pára: a solução ótima primal é encontrada.

(v) A factibilidade da solução básica dual é verificada pelos custos relativos não-negativos (veja (iii)), mas não exige que a solução básica primal  $\hat{\mathbf{x}}_B = \mathbf{B}^{-1}\mathbf{b}$  seja factível. Ou seja, é possível que  $\hat{x}_{B_\ell} < 0$ . Neste caso (já sugerido pelo exemplo anterior), a solução dual  $\hat{\lambda}$  não é ótima e é possível aumentar a função objetivo dual.

**Propriedade 2.16** (*condição de otimalidade dual*) Considere uma partição básica  $\mathbf{A} = [\mathbf{B} \ \mathbf{N}]$  em que a solução básica dual associada  $\hat{\lambda} = (\mathbf{B}^{-1})^T \mathbf{c}_B$  é factível. Se  $\hat{\mathbf{x}}_B = \mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$  (isto é, a solução básica primal associada é factível), então  $\hat{\lambda}$  e  $\hat{\mathbf{x}}_B$  resolvem o problema dual e primal, respectivamente.

A propriedade 2.16 é uma maneira alternativa e conveniente de escrever a condição de otimalidade primal (propriedade 2.3). Estamos prontos para descrever o método dual simplex. Considere uma partição básica  $\mathbf{A} = [\mathbf{B} \ \mathbf{N}]$  em que a solução básica dual associada  $\hat{\lambda} = (\mathbf{B}^{-1})^T \mathbf{c}_B$  é factível (isto é, as restrições básicas são ativas, ou seja,  $\mathbf{a}_{B_i}^T \hat{\lambda} = c_{B_i}$ ,  $i = 1, \dots, m$  e as restrições não-básicas são inativas, ou seja,  $\mathbf{a}_{N_j}^T \hat{\lambda} \leq c_{N_j}$ ,  $j = 1, \dots, n-m$ ). Suponha que a solução básica primal  $\hat{\mathbf{x}}_B = \mathbf{B}^{-1}\mathbf{b} \not\geq \mathbf{0}$ , isto é, não é factível. Existe, portanto, pelo menos um índice básico  $\ell$  tal que  $\hat{x}_{B_\ell} < 0$  (caso contrário,  $\hat{\mathbf{x}}_B \geq \mathbf{0}$  e é uma solução ótima). Uma estratégia de perturbar  $\hat{\lambda}$  que faz aumentar a função objetivo dual é dada a seguir.

*Estratégia dual simplex (considere  $\hat{x}_{B_\ell} < 0$ ):*

$$\begin{cases} \mathbf{a}_{B_\ell}^T \lambda = c_{B_\ell} - \delta, & \delta \geq 0 \\ \mathbf{a}_{B_i}^T \lambda = c_{B_i} \quad i = 1, \dots, m, \quad i \neq \ell, \end{cases} \quad (2.42)$$

isto é, uma nova solução dual é obtida (a qual depende de  $\delta$ ), mantendo-se ativas todas as restrições duais, exceto a  $\ell$ -ésima restrição que se torna inativa,  $\mathbf{a}_{B_\ell}^T \lambda = c_{B_\ell} - \delta$ , que é menor ou igual a  $c_{B_\ell}$  para todo  $\delta \geq 0$ .

A estratégia dual simplex pode ser expressa matricialmente por:

$$\mathbf{B}^T \lambda = \mathbf{c}_B - \delta \mathbf{e}_\ell,$$

em que  $\mathbf{e}_\ell$  é a  $\ell$ -ésima coluna da matriz identidade e do que se segue:

$$\lambda = (\mathbf{B}^T)^{-1} \mathbf{c}_B - \delta (\mathbf{B}^T)^{-1} \mathbf{e}_\ell.$$

Definindo

$$\eta_\ell = -(\mathbf{B}^{-1})^T \mathbf{e}_\ell,$$

chamada *direção dual simplex*, temos:

$$\lambda = \hat{\lambda} + \delta \eta_\ell. \quad (2.43)$$

Note que a direção dual simplex  $\eta_\ell$  pode ser obtida pela resolução do sistema de equações lineares:

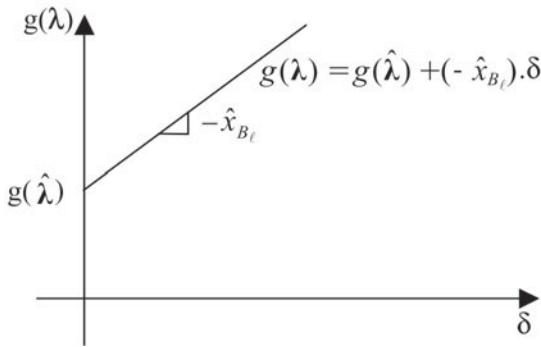
$$\mathbf{B}^T \eta_\ell = -\mathbf{e}_\ell. \quad (2.44)$$

A função objetivo para essa solução perturbada vale:

$$\begin{aligned} g(\lambda) &= g(\hat{\lambda} + \delta \eta_\ell) = \mathbf{b}^T (\hat{\lambda} + \delta \eta_\ell) = \mathbf{b}^T \hat{\lambda} + \delta \mathbf{b}^T \eta_\ell \\ &= g(\hat{\lambda}) - \delta \hat{x}_{B_\ell} \end{aligned} \quad (2.45)$$

de modo que a função objetivo dual cresce linearmente à taxa de  $-\hat{x}_{B_\ell} > 0$ . A Figura 2.49 ilustra a variação da função objetivo quando a solução básica dual é perturbada pela estratégia dual simplex.

Portanto, quanto maior o valor de  $\delta$ , maior o valor da função dual. Determinemos agora até que ponto podemos aumentar o valor de  $\delta$ .



**Figura 2.49** Variação da função objetivo dual com a estratégia dual simplex.

### Tamanho do passo

Todas as restrições duais são satisfeitas e devem continuar satisfeitas. Dividimos a análise em dois casos: restrições básicas e restrições não-básicas.

- Restrições duais básicas:  $\mathbf{B}^T \boldsymbol{\lambda} \leq \mathbf{c}_B$

Por construção, a estratégia dual simplex não viola as restrições básicas:

$$\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B - \delta \mathbf{e}_k \leq \mathbf{c}_B, \text{ para todo } \delta \geq 0,$$

e, portanto, o passo  $\delta$  não é limitado pelas restrições básicas.

- Restrições duais não-básicas:  $\mathbf{N}^T \boldsymbol{\lambda} \leq \mathbf{c}_N$

Analisemos cada uma das restrições não-básicas:  $\mathbf{a}_{N_j}^T \boldsymbol{\lambda} \leq c_{N_j}, j = 1, \dots, n-m$ . Substituindo (2.43) nas restrições não-básicas, segue-se que:

$$\boldsymbol{\lambda}^T \mathbf{a}_{N_j} \leq c_{N_j} \Leftrightarrow (\hat{\boldsymbol{\lambda}} + \delta \boldsymbol{\eta}_\ell)^T \mathbf{a}_{N_j} \leq c_{N_j} \Leftrightarrow \hat{\boldsymbol{\lambda}}^T \mathbf{a}_{N_j} + \delta \boldsymbol{\eta}_\ell^T \mathbf{a}_{N_j} \leq c_{N_j}. \quad (2.46)$$

Como, por hipótese, a solução  $\hat{\boldsymbol{\lambda}}$  é dual factível, então  $\hat{\boldsymbol{\lambda}}^T \mathbf{a}_{N_j} \leq c_{N_j}$ . Assim, se  $\boldsymbol{\eta}_\ell^T \mathbf{a}_{N_j} \leq 0, j = 1, \dots, n-m$ , então a desigualdade em (2.46) será sempre satisfeita para todo  $\delta \geq 0$  e a função dual cresce indefinidamente (o problema dual não tem solução ótima). Neste caso, o problema primal é infactível (ver propriedade 2.9).

Por outro lado, se  $\boldsymbol{\eta}_\ell^T \mathbf{a}_{N_j} > 0$ , então  $\delta$  é limitado superiormente e segue de (2.46) que:

$$\delta \leq \frac{c_{N_j} - \hat{\boldsymbol{\lambda}}^T \mathbf{a}_{N_j}}{\boldsymbol{\eta}_\ell^T \mathbf{a}_{N_j}} = \frac{\hat{c}_{N_j}}{\boldsymbol{\eta}_\ell^T \mathbf{a}_{N_j}}.$$

O numerador anterior  $\hat{c}_{N_j}$  é o custo relativo da variável  $x_{N_j}$ . Portanto, o maior valor possível para o passo  $\delta$  é:

$$\hat{\delta} = \frac{\hat{c}_{N_k}}{\boldsymbol{\eta}_\ell^T \mathbf{a}_{N_k}} = \min_{j=1, \dots, n-m} \left\{ \frac{\hat{c}_{N_j}}{\boldsymbol{\eta}_\ell^T \mathbf{a}_{N_j}} \text{ tal que } \boldsymbol{\eta}_\ell^T \mathbf{a}_{N_j} > 0 \right\}. \quad (2.47)$$

Com este valor para o passo, a  $k$ -ésima restrição não-básica (inativa em  $\hat{\boldsymbol{\lambda}}$ ) torna-se ativa na nova solução  $\boldsymbol{\lambda}' = \hat{\boldsymbol{\lambda}} + \hat{\delta} \boldsymbol{\eta}_\ell$ , juntamente com todas as restrições básicas, exceto a  $\ell$ -ésima, que deixou de ser ativa:  $\mathbf{a}_{B_\ell}^T \boldsymbol{\lambda}' = c_{B_\ell} - \hat{\delta}$ , conforme a estratégia dual simplex (2.42), isto é:

$$\begin{cases} \mathbf{a}_{N_k}^T \boldsymbol{\lambda}' = c_{N_k} \\ \mathbf{a}_{B_i}^T \boldsymbol{\lambda}' = c_{B_i}, i = 1, \dots, m, i \neq \ell \end{cases} \quad (2.48)$$

Com isso,  $m$  restrições duais são ativas em  $\lambda' = \hat{\lambda} + \hat{\delta}\eta_\ell$  (satisfitas com igualdade, reveja as Figuras 2.46 e 2.47). Reescrevendo o sistema (2.48) em forma matricial, segue que (substituímos a nova restrição dual ativa na posição daquela que deixou de ser ativa):

$$\begin{bmatrix} \mathbf{a}_{B_1}^T \\ \vdots \\ \mathbf{a}_{N_k}^T \\ \vdots \\ \mathbf{a}_{B_m}^T \end{bmatrix} \lambda' = \begin{pmatrix} c_{B_1} \\ \vdots \\ c_{N_k} \\ \vdots \\ c_{B_m} \end{pmatrix} \leftarrow \ell$$

$(\mathbf{B}')^T$        $\mathbf{c}_{\mathbf{B}'}$

Assim, definindo-se a matriz  $\mathbf{B}'$  como sendo a matriz  $\mathbf{B}$ , exceto pela coluna  $\ell$  (isto é,  $\mathbf{a}_{B_\ell}$ ), que é trocada pela coluna  $\mathbf{a}_{N_k}$ , e  $\mathbf{c}_{\mathbf{B}'}$  como sendo  $\mathbf{c}_B$  exceto pelo componente  $\ell$  (isto é,  $c_{B_\ell}$ ), que é trocado por  $c_{N_k}$ :

$$\mathbf{B}' = [\mathbf{a}_{B_1} \cdots \underset{\ell-\text{ésima coluna}}{\uparrow} \mathbf{a}_{N_k} \cdots \mathbf{a}_{B_m}] \quad \text{e} \quad \mathbf{c}_{\mathbf{B}'} = \begin{pmatrix} c_{B_1} \\ \vdots \\ c_{N_k} \\ \vdots \\ c_{B_m} \end{pmatrix} \leftarrow \ell\text{-ésimo componente}$$

e o sistema (2.48) é reescrito como  $(\mathbf{B}')^T \lambda' = \mathbf{c}_{\mathbf{B}'}$ . A matriz  $\mathbf{B}'$  é invertível, de modo que  $\mathbf{A} = [\mathbf{B}' \mathbf{N}']$  é uma partição básica, em que  $\mathbf{N}'$  coincide com  $\mathbf{N}$ , exceto pela  $k$ -ésima coluna (isto é,  $\mathbf{a}_{N_k}$ ) que é trocada por  $\mathbf{a}_{B_\ell}$ . Esta é a propriedade 2.4 e os passos para mostrar sua validade estão no exercício 2.25. Portanto, temos uma nova solução básica dual (novo vértice dual):  $\lambda' = \hat{\lambda} + \hat{\delta}\eta_\ell$ , para o qual a função objetivo dual é maior ou igual à solução  $\hat{\lambda}$  (ver 2.45):

$$g(\lambda') = g(\hat{\lambda}) - \hat{\delta}\hat{x}_{B_\ell} \geq g(\hat{\lambda}).$$

A desigualdade anterior será estrita se  $\hat{\delta} > 0$  em (2.47). Com a nova partição básica, podemos repetir o procedimento anterior, isto é, calculamos a solução básica primal  $\mathbf{B}' \mathbf{x}_{\mathbf{B}'} = \mathbf{b}$ , e, se  $\mathbf{x}_{\mathbf{B}'} \geq 0$ , então aplicamos a estratégia dual simplex para obter uma nova solução básica melhor e, assim, sucessivamente, até que a solução básica primal seja factível, ou a infactibilidade primal seja detectada (isto é, passo dual infinito). Este procedimento consiste no método dual simplex, que é resumido a seguir.

### 2.10.5\* O algoritmo dual simplex

Considere um problema de otimização linear escrito na forma padrão. Resumimos agora o procedimento desenvolvido na seção anterior, o qual pressupõe uma solução básica dual factível. A questão de como determinar uma solução básica factível inicial será discutida nos exemplos da próxima seção. O método dual simplex, propriamente dito, é o que chamamos a seguir de Fase II.

*Fase I:*

- Determine inicialmente uma partição básica dual factível:  $\mathbf{A} = [\mathbf{B}, \mathbf{N}]$ . A rigor, precisamos de dois vetores, de índices básicos e não-básicos:

$$(B_1, B_2, \dots, B_m) \text{ e } (N_1, N_2, \dots, N_{n-m}).$$

Faça iteração = 1.

Fase II:

**{início da iteração dual simplex}**

Passo 1: {cálculo da solução básica dual e custos relativos}

$$\hat{\lambda} = \mathbf{c}_B^T \mathbf{B}^{-1} \quad (\text{ou equivalentemente, resolva o sistema: } \mathbf{B}^T \hat{\lambda} = \mathbf{c}_B)$$

$$\hat{c}_{N_j} = c_{N_j} - \hat{\lambda}^T \mathbf{a}_{N_j} \quad j = 1, \dots, n-m$$

Passo 2: {teste de otimalidade}

2.1) {cálculo da solução básica primal}

$$\hat{x}_B = \mathbf{B}^{-1} \mathbf{b} \quad (\text{ou equivalentemente, resolva o sistema: } \mathbf{B} \hat{x}_B = \mathbf{b})$$

2.2) {determinação da variável a sair da base}

$$\hat{x}_{B_\ell} = \min\{\hat{x}_{B_i}, i = 1, \dots, m\}$$

Se  $\hat{x}_{B_\ell} \geq 0$ , então pare {a solução na iteração atual é ótima}

Passo 3: {cálculo da direção dual simplex}

$$\eta_\ell = -(\mathbf{B}^{-1})^T \mathbf{e}_\ell \quad (\text{ou equivalentemente, resolva o sistema: } \mathbf{B}^T \eta_\ell = -\mathbf{e}_\ell)$$

Passo 4: {determinação do passo e variável a entrar na base}

Se  $\eta_\ell^T \mathbf{a}_{N_j} \leq 0, j = 1, \dots, n-m$ , então pare. {o problema primal é infactível}

caso contrário, determine

$$\hat{\delta} = \frac{\hat{c}_{N_k}}{\eta_\ell^T \mathbf{a}_{N_k}} = \min_{j=1, \dots, n-m} \left\{ \frac{\hat{c}_{N_j}}{\eta_\ell^T \mathbf{a}_{N_j}} \text{ tal que } \eta_\ell^T \mathbf{a}_{N_j} > 0 \right\}$$

(a variável  $x_{N_k}$  entra na base)

Passo 5: {atualização: nova partição básica, troque a  $\ell$ -ésima coluna de  $\mathbf{B}$  pela  $k$ -ésima coluna de  $\mathbf{N}$ }

nova matriz básica:  $\mathbf{B} = [\mathbf{a}_{B_1} \dots, \mathbf{a}_{B_{\ell-1}}, \mathbf{a}_{N_k}, \mathbf{a}_{B_{\ell+1}} \dots, \mathbf{a}_{B_m}]$

nova matriz não-básica:  $\mathbf{N} = [\mathbf{a}_{N_1} \dots, \mathbf{a}_{N_{k-1}}, \mathbf{a}_{B_\ell}, \mathbf{a}_{N_{k+1}} \dots, \mathbf{a}_{N_{n-m}}]$

iteração = iteração + 1

Retorne ao passo 1

**{fim da iteração dual simplex}**

**Exemplo 2.47** Considere o seguinte problema de otimização linear e a forma padrão equivalente com a introdução das variáveis de folga. Resolvemos este problema usando o método dual simplex.

**Primal:**

$$\text{Minimizar } f(\mathbf{x}) = x_1 + x_2$$

$$2x_1 + x_2 \geq 4$$

$$x_1 + 3x_2 \geq 3$$

$$x_1 \geq 0, x_2 \geq 0$$

$$\text{Minimizar } f(\mathbf{x}) = x_1 + x_2$$

$$2x_1 + x_2 - x_3 = 4$$

$$x_1 + 3x_2 - x_4 = 3$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$$

**Dual:**

$$\text{Maximizar } g(\lambda) = 4\lambda_1 + 3\lambda_2$$

$$2\lambda_1 + \lambda_2 \leq 1$$

$$\lambda_1 + 3\lambda_2 \leq 1$$

$$\lambda_1 \geq 0, \lambda_2 \geq 0$$

Note que  $\lambda = (0 \ 0)^T$  é uma solução dual factível e as restrições duais decorrentes das colunas das variáveis de folga são ativas, isto é,  $-\lambda_1 \leq 0$ ,  $-\lambda_2 \leq 0$  são satisfeitas com igualdade. Isso define uma partição básica dual factível, na qual as variáveis básicas são as variáveis de folga do problema primal. Esta propriedade é válida para uma classe de problemas de otimização linear que têm duas características: i) os coeficientes da função objetivo, a ser minimizada, são positivos (ver o lado direito das restrições duais); e ii) as restrições são de desigualdade, de modo que as variáveis de folga introduzem uma matriz básica (ver Exercício 2.33). Desse modo, a Fase I, que consiste em encontrar uma partição básica dual factível para iniciar o método dual simplex, é contornada, bastando escolher como variáveis básicas (que definem as restrições ativas) as variáveis de folga  $x_3$  e  $x_4$ .

**1<sup>a</sup> Iteração:** partição básica:  $B_1 = 3, B_2 = 4, N_1 = 1, N_2 = 2$

$$\mathbf{B} = [\mathbf{a}_3 \ \mathbf{a}_4] = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \quad \mathbf{N} = [\mathbf{a}_1 \ \mathbf{a}_2] = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \quad \mathbf{c}_B = (0 \ 0)^T \quad \mathbf{c}_N = (1 \ 1)^T$$

*Passo 1 {Cálculo da solução básica dual e custos relativos}*

$$\mathbf{B}^T \lambda = \mathbf{c}_B, \text{ cuja solução é: } \hat{\lambda} = (0 \ 0)^T$$

$$\hat{c}_1 = c_1 - \hat{\lambda}^T \mathbf{a}_1 = 1 - (0 \ 0) \begin{pmatrix} 2 \\ 1 \end{pmatrix} = 1$$

$$\hat{c}_2 = c_2 - \hat{\lambda}^T \mathbf{a}_2 = 1 - (0 \ 0) \begin{pmatrix} 1 \\ 3 \end{pmatrix} = 1$$

*Passo 2 {teste de optimidade}*

2.1. {cálculo da solução básica primal}

$$\mathbf{Bx}_B = \mathbf{b}, \text{ cuja solução é: } \hat{\mathbf{x}}_B = \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} -4 \\ -3 \end{pmatrix}$$

2.2. {determinação da variável a sair da base}

a variável básica  $x_{B_1} = x_3$  é a mais negativa e é escolhida para sair da base ( $\ell=1$ ).

*Passo 3 {cálculo da direção dual simplex}*

$$\mathbf{B}^T \boldsymbol{\eta}_1 = -\mathbf{e}_1, \text{ cuja solução é: } \boldsymbol{\eta}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

*Passo 4 {determinação do passo e variável a entrar na base}*

Calculamos inicialmente:  $\boldsymbol{\eta}_\ell^T \mathbf{a}_{N_j}$ ,  $j = 1, \dots, n - m$

$$N_1 = 1: \boldsymbol{\eta}_1^T \mathbf{a}_1 = (1 \ 0) \begin{pmatrix} 2 \\ 1 \end{pmatrix} = 2$$

$$N_2 = 2: \boldsymbol{\eta}_1^T \mathbf{a}_2 = (1 \ 0) \begin{pmatrix} 1 \\ 3 \end{pmatrix} = 1$$

$$\hat{\delta} = \frac{\hat{c}_1}{\boldsymbol{\eta}_1^T \mathbf{a}_1} = \text{mínimo} \left\{ \frac{1}{2}, \frac{1}{1} \right\} = \frac{1}{2} \quad (\text{a variável } x_{N_1} = x_1 \text{ entra na base})$$

*Passo 5 {atualização: nova partição básica, troque a  $\ell$ -ésima coluna de  $\mathbf{B}$  pela  $k$ -ésima coluna de  $\mathbf{N}$ }:*  
 $x_3$  sai da base e  $x_1$  entra na base.

$$B_1 = 1, B_2 = 4, \quad N_1 = 3, N_2 = 2$$

**2<sup>a</sup> Iteração:** partição básica:  $B_1 = 1, B_2 = 4, N_1 = 3, N_2 = 2$

$$\mathbf{B} = [\mathbf{a}_1 \mathbf{a}_4] = \begin{bmatrix} 2 & 0 \\ 1 & -1 \end{bmatrix} \quad \mathbf{N} = [\mathbf{a}_3 \mathbf{a}_2] = \begin{bmatrix} -1 & 1 \\ 0 & 3 \end{bmatrix} \quad \mathbf{c}_B = (1 \ 0)^T \quad \mathbf{c}_N = (0 \ 1)^T$$

Passo 1 {solução básica dual e custos relativos}

$$\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B: \left[ \begin{array}{cc|c} 2 & 1 & 1 \\ 0 & -1 & 0 \end{array} \right], \text{ cuja solução é: } \hat{\boldsymbol{\lambda}} = \left( \begin{array}{c} \frac{1}{2} \\ 0 \end{array} \right)^T$$

$$\hat{c}_3 = c_3 - \hat{\boldsymbol{\lambda}}^T \mathbf{a}_3 = 0 - \left( \begin{array}{c} \frac{1}{2} \\ 0 \end{array} \right) \left( \begin{array}{c} -1 \\ 0 \end{array} \right) = \frac{1}{2}$$

$$\hat{c}_2 = c_2 - \hat{\boldsymbol{\lambda}}^T \mathbf{a}_2 = 1 - \left( \begin{array}{c} \frac{1}{2} \\ 0 \end{array} \right) \left( \begin{array}{c} 1 \\ 3 \end{array} \right) = \frac{1}{2}$$

Passo 2 {teste de optimidade}

2.1. {solução básica primal}

$$\mathbf{Bx}_B = \mathbf{b}: \left[ \begin{array}{cc|c} 2 & 0 & 4 \\ 1 & -1 & 3 \end{array} \right], \text{ cuja solução é: } \hat{\mathbf{x}}_B = \begin{pmatrix} x_1 \\ x_4 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

2.2. a variável básica  $x_{B_2} = x_4$  é a mais negativa e é escolhida para sair da base ( $\ell=2$ )

Passo 3 {direção dual simplex}

$$\mathbf{B}^T \boldsymbol{\eta}_2 = -\mathbf{e}_2: \left[ \begin{array}{cc|c} 2 & 1 & 0 \\ 0 & -1 & -1 \end{array} \right], \text{ cuja solução é: } \boldsymbol{\eta}_2 = \begin{pmatrix} -\frac{1}{2} \\ 1 \end{pmatrix}$$

Passo 4 {passo e variável a entrar na base}

calculamos inicialmente:  $\boldsymbol{\eta}_t^T \mathbf{a}_{N_j}, j = 1, \dots, n-m$

$$N_1 = 3: \boldsymbol{\eta}_2^T \mathbf{a}_3 = \left( -\frac{1}{2} \ 1 \right) \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \frac{1}{2}$$

$$N_2 = 2: \boldsymbol{\eta}_2^T \mathbf{a}_2 = \left( -\frac{1}{2} \ 1 \right) \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \frac{5}{2}$$

$$\hat{\delta} = \frac{\hat{c}_2}{\boldsymbol{\eta}_2^T \mathbf{a}_2} = \text{mínimo} \left\{ \frac{\frac{1}{2}}{\frac{1}{2}}, \frac{\frac{1}{2}}{\frac{5}{2}} \right\} = \frac{1}{5} \quad (\text{a variável } x_{N_2} = x_2 \text{ entra na base})$$

Passo 5 {atualização:  $x_3$  sai da base e  $x_1$  entra na base}

$$B_1 = 1, B_2 = 2, \quad N_1 = 3, N_2 = 4$$

**3<sup>a</sup> Iteração:** partição básica:  $B_1 = 1, B_2 = 2, N_1 = 3, N_2 = 4$

$$\mathbf{B} = [\mathbf{a}_1 \mathbf{a}_2] = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \quad \mathbf{N} = [\mathbf{a}_3 \mathbf{a}_4] = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \quad \mathbf{c}_B = (1 \ 1)^T \quad \mathbf{c}_N = (0 \ 0)^T$$

Passo 1 {solução básica dual e custos relativos}

$$\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B: \left[ \begin{array}{cc|c} 2 & 1 & 1 \\ 1 & 3 & 1 \end{array} \right], \text{ cuja solução é: } \hat{\boldsymbol{\lambda}} = \left( \begin{array}{c} \frac{2}{5} \ \frac{1}{5} \end{array} \right)^T$$

$$\hat{c}_3 = c_3 - \hat{\lambda}^T \mathbf{a}_3 = 0 - \begin{pmatrix} \frac{2}{5} & \frac{1}{5} \end{pmatrix} \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \frac{2}{5}$$

$$\hat{c}_4 = c_4 - \hat{\lambda}^T \mathbf{a}_4 = 0 - \begin{pmatrix} \frac{2}{5} & \frac{1}{5} \end{pmatrix} \begin{pmatrix} 0 \\ -1 \end{pmatrix} = \frac{1}{5}$$

Passo 2 {teste de optimidade}

### 2.1. {solução básica primal}

$$\mathbf{Bx}_B = \mathbf{b}: \left[ \begin{array}{cc|c} 2 & 1 & 4 \\ 1 & 3 & 3 \end{array} \right], \text{ cuja solução é: } \hat{\mathbf{x}}_B = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \frac{9}{5} \\ \frac{2}{5} \end{pmatrix}$$

2.2. como a solução básica primal é factível, então a solução é ótima. Pare.

Solução ótima primal:

$$\begin{pmatrix} x_1^* \\ x_2^* \\ x_3^* \\ x_4^* \end{pmatrix} = \begin{pmatrix} \frac{9}{5} \\ \frac{2}{5} \\ 0 \\ 0 \end{pmatrix}$$

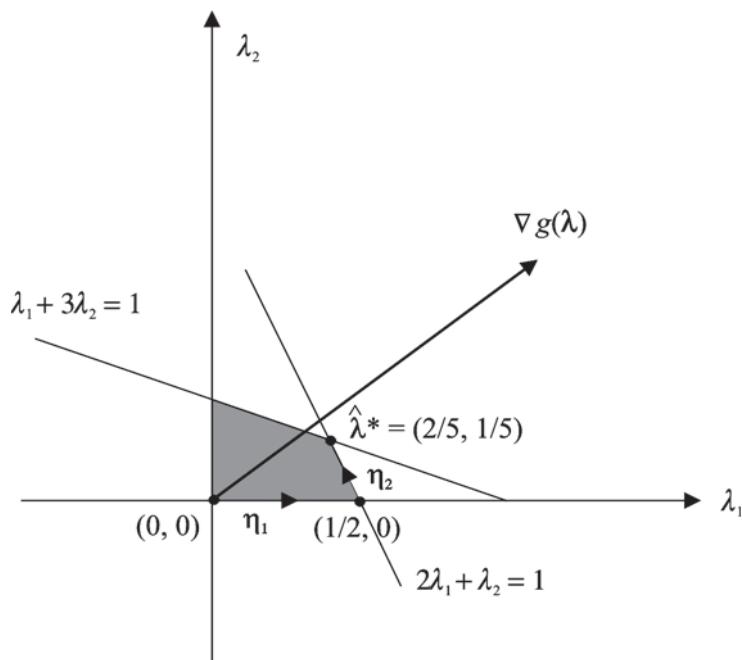
$$f(\mathbf{x}^*) = x_1^* + x_2^* = \frac{9}{5} + \frac{2}{5} = \frac{11}{5}$$

Solução ótima dual:

$$\begin{pmatrix} \lambda_1^* \\ \lambda_2^* \end{pmatrix} = \begin{pmatrix} \frac{2}{5} \\ \frac{1}{5} \end{pmatrix}$$

$$g(\lambda^*) = 4\lambda_1^* + 3\lambda_2^* = 4 \times \frac{2}{5} + 3 \times \frac{1}{5} = \frac{11}{5}$$

As soluções percorridas no espaço de soluções do problema dual podem ser visualizadas na Figura 2.50.



**Figura 2.50** A trajetória do método dual simplex.

O caminho percorrido foi da solução  $\lambda^1 = [0 \ 0]^T$  para a solução factível  $\lambda^2 = (\frac{1}{2} \ 0)^T$ , até chegar à solução factível ótima  $\lambda^* = (\frac{2}{5} \ \frac{1}{5})^T$ . Ver o Exercício 2.32 para a trajetória no plano primal.

### 2.10.6\* Reotimização após a inclusão de novas restrições

Uma das principais aplicações do método dual simplex consiste em resolver problemas cujas soluções ótimas se tornaram infactíveis depois que novas restrições são incluídas. No Capítulo 3 de otimização discreta, veremos uma situação em que novas restrições são geradas para eliminar as soluções que não sejam inteiras. Considere que o problema de otimização linear e seu dual (denominados problemas originais):

$$\begin{aligned} \text{Minimizar } f(\mathbf{x}) &= \mathbf{c}^T \mathbf{x} \\ \mathbf{Ax} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

$$\begin{aligned} \text{Maximizar } g(\boldsymbol{\lambda}) &= \mathbf{b}^T \boldsymbol{\lambda} \\ \mathbf{A}^T \boldsymbol{\lambda} &\leq \mathbf{c}, \end{aligned}$$

tenham sido resolvidos e  $\mathbf{A} = [\mathbf{B} \ \mathbf{N}]$  é uma partição básica ótima, à qual estão associadas as soluções básicas factíveis primal e dual:

- solução básica primal:  $\hat{\mathbf{x}}_B = \mathbf{B}^{-1}\mathbf{b}$ ,  $\hat{\mathbf{x}}_N = \mathbf{0}$ ,
- solução básica dual:  $\hat{\boldsymbol{\lambda}} = (\mathbf{B}^{-1})^T \mathbf{c}_B$ .

A solução primal é factível, portanto:  $\hat{\mathbf{x}}_B \geq \mathbf{0}$ , e a solução básica dual é factível, assim  $\hat{\mathbf{c}}_N = \mathbf{c}_N - \mathbf{N}^T \hat{\boldsymbol{\lambda}} \geq \mathbf{0}$  ou, de forma equivalente,  $\mathbf{N}^T \hat{\boldsymbol{\lambda}} \leq \mathbf{c}_N$ . Além disso, como essas soluções são ótimas, então  $f(\hat{\mathbf{x}}) = g(\hat{\boldsymbol{\lambda}})$ .

Suponha que uma nova restrição seja incluída no modelo:

$$\mathbf{u}^T \mathbf{x} \leq v$$

de modo que o problema primal torna-se mais restrito e seu dual mais relaxado, no sentido que vemos a seguir (denominados problemas modificados):

$$\text{Minimizar } f(\mathbf{x}) = \mathbf{c}^T \mathbf{x}$$

$$\text{Maximizar } g(\boldsymbol{\lambda}, \lambda_{m+1}) = \mathbf{b}^T \boldsymbol{\lambda} + v \lambda_{m+1}$$

$$\mathbf{Ax} + \mathbf{0}x_{m+1} = \mathbf{b}$$

$$\mathbf{A}^T \boldsymbol{\lambda} + \mathbf{u} \lambda_{m+1} \leq \mathbf{c}$$

$$\mathbf{u}^T \mathbf{x} + x_{m+1} = v$$

$$\mathbf{0}^T \boldsymbol{\lambda} + \lambda_{m+1} \leq 0$$

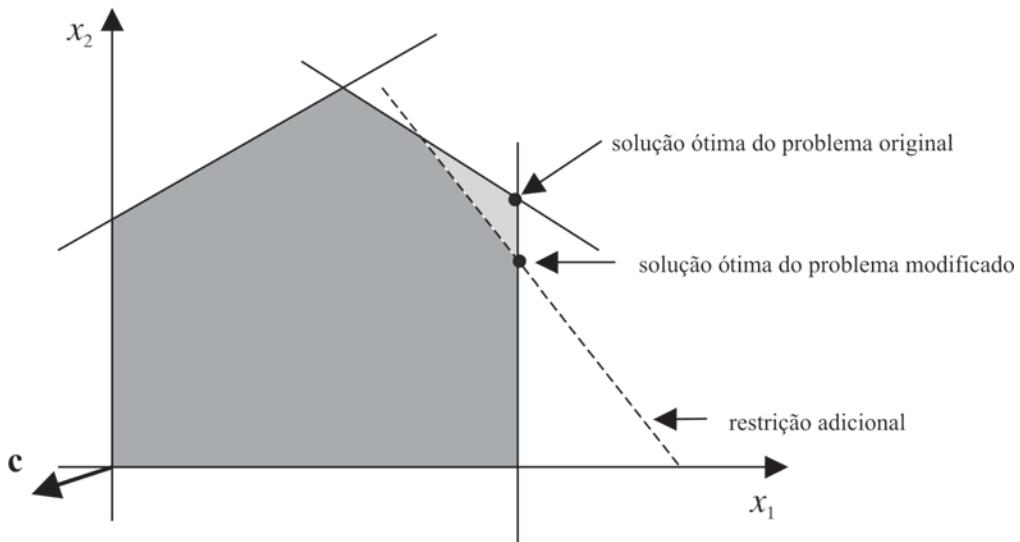
$$\mathbf{x} \geq \mathbf{0}, x_{m+1} \geq 0$$

em que  $x_{m+1}$  é a variável de folga e  $\lambda_{m+1}$  é a variável dual associada à nova restrição.

Com a inclusão da nova restrição, o problema primal tem  $m + 1$  restrições (ver Figura 2.51), de modo que uma base para o problema primal modificado deve ter dimensão  $(m + 1) \times (m + 1)$ . Determinamos a seguir uma base dual factível e, para isso, identificamos uma solução dual factível e  $m + 1$  restrições ativas.

A solução dual ótima do problema original  $\hat{\boldsymbol{\lambda}}$ , e  $\lambda_{m+1} = 0$ , fornecem uma solução dual factível para o problema dual modificado. De fato, basta observar que  $\hat{\boldsymbol{\lambda}}$  satisfaz as restrições duais do problema original  $\mathbf{A}^T \hat{\boldsymbol{\lambda}} \leq \mathbf{c}$ . Na verdade, qualquer solução dual factível para o problema dual original, e  $\lambda_{m+1} = 0$ , fornecem uma solução dual factível para o dual modificado. Nesse sentido, o problema dual modificado é uma relaxação do problema dual original. Além disso, a solução  $(\hat{\boldsymbol{\lambda}}, 0)$  do problema dual modificado satisfaz com igualdade as restrições básicas duais (restrições ativas)  $\mathbf{B}^T \hat{\boldsymbol{\lambda}} = \mathbf{c}_B$ , bem como a nova restrição dual que surge devido à coluna da variável de folga no primal modificado,  $\lambda_{m+1} = 0$ . Com isso, identificamos  $m + 1$  restrições duais ativas em  $(\hat{\boldsymbol{\lambda}}, 0)$ , isto é,  $(\hat{\boldsymbol{\lambda}}, 0)$  é a solução do sistema:

$$\begin{bmatrix} \mathbf{B}^T & \mathbf{u}_B \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \lambda_{m+1} \end{bmatrix} = \begin{bmatrix} \mathbf{c}_B \\ 0 \end{bmatrix}$$



**Figura 2.51** Inclusão de restrição adicional.

(adotamos a mesma notação anterior,  $\mathbf{u}^T = (\mathbf{u}_B^T \mathbf{u}_N^T)$ , em que  $\mathbf{u}_B^T$  é o vetor dos coeficientes de  $\mathbf{x}_B$  na nova restrição, como também os coeficientes de  $\lambda_{m+1}$  nas restrições duais básicas do problema modificado). Essas restrições ativas definem uma partição básica para o problema primal modificado:

$$\begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{u}_B & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_B \\ x_{m+1} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ v \end{bmatrix},$$

cujas soluções são dadas por:

$$\hat{\mathbf{x}}_B = \mathbf{B}^{-1} \mathbf{b}, \quad \hat{x}_{m+1} = v - \mathbf{u}_B^T \hat{\mathbf{x}}_B.$$

Como  $\hat{\mathbf{x}}_B \geq \mathbf{0}$  (é a solução ótima do problema original) e, portanto, se  $\hat{x}_{m+1} = v - \mathbf{u}_B^T \hat{\mathbf{x}}_B \geq 0$ , então a solução anterior é a solução factível do problema modificado, com  $\hat{\mathbf{x}}_N = \mathbf{0}$ . Note que a não-negatividade da variável de folga da nova restrição significa que a solução ótima do problema primal original ( $\hat{\mathbf{x}}_B, \hat{\mathbf{x}}_N$ ) não viola a nova restrição:  $\mathbf{u}_B^T \hat{\mathbf{x}}_B + \mathbf{u}_N^T \hat{\mathbf{x}}_N \leq v$ . Neste caso, temos uma solução primal factível e outra dual factível, e as funções objetivas primal e dual têm o mesmo valor  $f(\hat{\mathbf{x}}, x_{m+1}) = g(\hat{\lambda}, \hat{\lambda}_{m+1})$  e, consequentemente, são soluções ótimas.

Por outro lado, se  $\hat{x}_{m+1} < 0$ , então a solução básica primal não é factível (a nova restrição é violada:  $\mathbf{u}_B^T \hat{\mathbf{x}}_B + \mathbf{u}_N^T \hat{\mathbf{x}}_N > v$ ) e podemos aplicar o método dual simplex a partir da solução básica dual factível. Tipicamente, poucas iterações do método dual simplex são necessárias para se obter a solução ótima do problema modificado.

**Exemplo 2.48** Considere o problema de otimização linear, bem como sua forma padrão:

$$\begin{aligned} \text{Minimizar } f(\mathbf{x}) &= -5x_1 + x_2 \\ 7x_1 - 5x_2 &\leq 13 \\ 3x_1 + 2x_2 &\leq 17 \\ x_1, x_2 &\geq 0 \end{aligned}$$

$$\begin{aligned} \text{Minimizar } f(\mathbf{x}) &= -5x_1 + x_2 + 0x_3 + 0x_4 \\ 7x_1 - 5x_2 + 1x_3 + 0x_4 &= 13 \\ 3x_1 + 2x_2 + 0x_3 + 1x_4 &= 17 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}$$

cujas soluções ótimas podem ser obtidas na Figura 2.52.

A partição básica ótima é dada por (graficamente:  $x_1 > 0, x_2 > 0, x_3 = 0, x_4 = 0$ ):

$$\mathbf{B} = [\mathbf{a}_1 \ \mathbf{a}_2] = \begin{bmatrix} 7 & -5 \\ 3 & 2 \end{bmatrix} \quad \text{e} \quad \mathbf{N} = [\mathbf{a}_3 \ \mathbf{a}_4] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

e as soluções básicas ótimas, primal e dual, são:

$$\mathbf{Bx}_B = \mathbf{b}: \quad \begin{bmatrix} 7 & -5 & 13 \\ 3 & 2 & 17 \end{bmatrix}, \text{ cuja solução é: } \hat{\mathbf{x}}_B = \begin{pmatrix} \frac{111}{29} \\ \frac{80}{29} \end{pmatrix}$$

$$\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B: \quad \begin{bmatrix} 7 & 3 & -5 \\ -5 & 2 & 1 \end{bmatrix}, \text{ cuja solução é: } \hat{\boldsymbol{\lambda}} = \begin{pmatrix} -\frac{13}{29} \\ -\frac{18}{29} \end{pmatrix}.$$

Os custos relativos das variáveis não-básicas ( $N_1 = 3, N_2 = 4$ ) são:

- $\hat{c}_3 = c_3 - \hat{\boldsymbol{\lambda}}^T \mathbf{a}_3 = 0 - (-\frac{13}{29} \quad -\frac{18}{29}) \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{13}{29}$
- $\hat{c}_4 = c_4 - \hat{\boldsymbol{\lambda}}^T \mathbf{a}_4 = 0 - (-\frac{13}{29} \quad -\frac{18}{29}) \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{18}{29}$

Suponha agora que o problema modificado seja obtido pela inclusão da restrição:  $x_1 \leq 3$ , isto é,

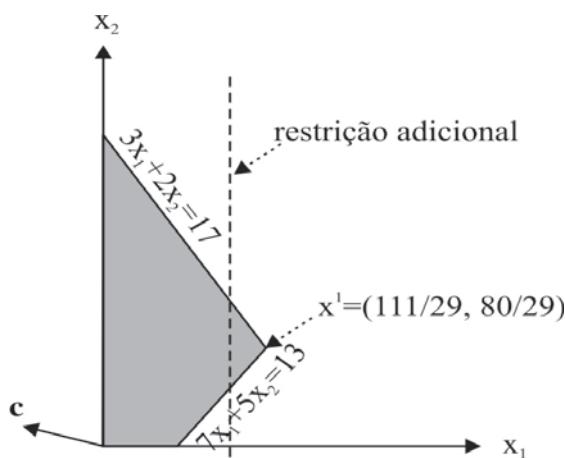
$$x_1 + 0x_2 + 0x_3 + 0x_4 \leq 3$$

de modo que o vetor  $\mathbf{u}$  é dado por:  $\mathbf{u}^T = (1 \ 0 \ 0 \ 0)$ , bem como a partição é  $\mathbf{u}_B^T = (1 \ 0)$  e  $\mathbf{u}_N^T = (0 \ 0)$ . O problema primal modificado (variáveis de folga incluídas) é:

$$\begin{aligned} \text{Minimizar} \quad f(\mathbf{x}) &= -5x_1 + x_2 + 0x_3 + 0x_4 + 0x_5 \\ 7x_1 - 5x_2 + 1x_3 + 0x_4 + 0x_5 &= 13 \\ 3x_1 + 2x_2 + 0x_3 + 1x_4 + 0x_5 &= 17 \\ x_1 + 0x_2 + 0x_3 + 0x_4 + 1x_5 &= 3 \leftarrow \text{restrição adicional} \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 & \end{aligned}$$

A partição básica dual factível tem os índices básicos dados por:

$$B_1 = 1, B_2 = 2 \text{ e } B_3 = 5$$



**Figura 2.52** Inclusão de restrição adicional  $x_1 \leq 3$ .

isto é, os índices básicos da solução anterior e o índice da variável de folga da restrição adicional. A matriz básica é dada por:

$$\begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{u}_B & 1 \end{bmatrix} = \begin{bmatrix} 7 & -5 & 0 \\ 3 & 2 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

As soluções básicas, primal e dual, são obtidas diretamente da solução anterior:

- Primal:  $\begin{pmatrix} \hat{\mathbf{x}}_B \\ x_{n+1} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_5 \end{pmatrix} = \begin{pmatrix} \frac{111}{29} \\ \frac{80}{29} \\ -\frac{24}{29} \end{pmatrix}$ , em que  $\hat{\mathbf{x}}_B$  é o vetor das variáveis básicas na solução ótima primal anterior e  $x_5$  é obtido diretamente da restrição adicional,  $x_5 = 3 - x_1$ . As variáveis não-básicas são as mesmas da solução anterior:  $\mathbf{x}_N = (x_3 \ x_4)^T$ .
- Dual:  $(\hat{\lambda}^T \ 0) = (-\frac{13}{29} \ -\frac{18}{29} \ 0)$ , em que  $\hat{\lambda}$  é a solução ótima dual anterior, aumentada de  $\lambda_3 = 0$ . Os custos relativos são os mesmos da solução anterior.

Aplicamos agora o método dual simplex a partir desta partição básica dual factível.

*Método dual simplex:*

**1ª Iteração:** partição básica:  $B_1 = 1, B_2 = 2, B_3 = 5, N_1 = 3, N_2 = 4$

$$\mathbf{B} = \begin{bmatrix} 7 & -5 & 0 \\ 3 & 2 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

*Passo 1 {solução básica dual e custos relativos}*

$$\begin{aligned} (\hat{\lambda}^T \ 0) &= (-\frac{13}{29} \ -\frac{18}{29} \ 0) \\ \hat{c}_3 &= c_3 - \hat{\lambda}^T \mathbf{a}_3 = 0 - \left(-\frac{13}{29} \ -\frac{18}{29} \ 0\right) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \frac{13}{29} \\ \hat{c}_4 &= c_4 - \hat{\lambda}^T \mathbf{a}_4 = 0 - \left(-\frac{13}{29} \ -\frac{18}{29} \ 0\right) \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \frac{18}{29} \end{aligned}$$

*Passo 2 {teste de optimidade}*

2.1. {solução básica primal}

$$\hat{\mathbf{x}}_B = \begin{pmatrix} x_1 \\ x_2 \\ x_5 \end{pmatrix} = \begin{pmatrix} \frac{111}{29} \\ \frac{80}{29} \\ -\frac{24}{29} \end{pmatrix}$$

2.2. a variável básica  $x_{B_3} = x_5$  é negativa e é escolhida para sair da base ( $\ell=3$ ).

*Passo 3 {direção dual simplex}*

$$\mathbf{B}^T \mathbf{n}_\ell = -\mathbf{e}_\ell : [\mathbf{B}^T | -\mathbf{e}_\ell] = \left[ \begin{array}{ccc|c} 7 & 3 & 1 & 0 \\ -5 & 2 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{array} \right], \text{ cuja solução é: } \mathbf{n}_\ell = \begin{pmatrix} \frac{2}{29} \\ \frac{5}{29} \\ -1 \end{pmatrix}$$

*Passo 4 {tamanho do passo e variável a entrar na base}*

calculamos inicialmente:  $\mathbf{n}_\ell^T \mathbf{a}_{N_j}, j = 1, \dots, n-m$

$$N_1 = 3: \mathbf{\eta}_3^T \mathbf{a}_3 = \left( \frac{2}{29} \quad \frac{5}{29} \quad -1 \right) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \frac{2}{29}$$

$$N_2 = 4: \mathbf{\eta}_3^T \mathbf{a}_4 = \left( \frac{2}{29} \quad \frac{5}{29} \quad -1 \right) \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \frac{5}{29}$$

$$\hat{\delta} = \frac{\hat{c}_4}{\mathbf{\eta}_3^T \mathbf{a}_4} = \min_{j=1, \dots, n-m} \left\{ \frac{\frac{13}{29}}{\frac{2}{29}}, \frac{\frac{18}{29}}{\frac{5}{29}} \right\} = \frac{18}{5} \quad (\text{a variável } x_{N_2} = x_4 \text{ entra na base})$$

*Passo 5 {atualização:  $x_5$  sai da base e  $x_4$  entra na base}*

$$\begin{array}{ll} \text{matriz básica:} & \mathbf{B} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \mathbf{a}_4] \\ \text{matriz não-básica:} & \mathbf{N} = [\mathbf{a}_3 \ \mathbf{a}_5] \end{array}$$

**2ª Iteração:** partição básica:  $B_1 = 1, B_2 = 2, B_3 = 4, N_1 = 3, N_2 = 5$

$$\mathbf{B} = \begin{bmatrix} 7 & -5 & 0 \\ 3 & 2 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

*Passo 1 {solução básica dual e custos relativos}*

$$\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B: \left[ \begin{array}{ccc|c} 7 & 3 & 1 & -5 \\ -5 & 2 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{array} \right], \text{ cuja solução é: } \hat{\boldsymbol{\lambda}}^T = \left( -\frac{1}{5} \quad 0 \quad -\frac{18}{5} \right)$$

$$\hat{c}_3 = c_3 - \hat{\boldsymbol{\lambda}}^T \mathbf{a}_3 = 0 - \left( -\frac{1}{5} \quad 0 \quad -\frac{18}{5} \right) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{5}$$

$$\hat{c}_5 = c_5 - \hat{\boldsymbol{\lambda}}^T \mathbf{a}_5 = 0 - \left( -\frac{1}{5} \quad 0 \quad -\frac{18}{5} \right) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \frac{18}{5}$$

*Passo 2 {teste de otimalidade}*

2.1. {solução básica primal}

$$\mathbf{Bx}_B = \mathbf{b}: \left[ \begin{array}{ccc|c} 7 & -5 & 0 & 13 \\ 3 & 2 & 1 & 17 \\ 1 & 0 & 0 & 3 \end{array} \right], \text{ cuja solução é: } \hat{\mathbf{x}}_B = \begin{pmatrix} x_1 \\ x_2 \\ x_4 \end{pmatrix} = \begin{pmatrix} 3 \\ \frac{8}{5} \\ \frac{24}{5} \end{pmatrix}$$

2.2. a solução básica primal é factível, portanto, ótima. Pare.

■

## 2.11 OTIMIZAÇÃO LINEAR EM TEORIA DE JOGOS

A teoria de jogos trata de situações de tomada de decisões nas quais “jogadores” (adversários) têm objetivos conflitantes e o resultado depende da combinação de estratégias escolhidas pelos joga-

dores. Além de jogos de salão, exemplos de aplicação incluem campanhas de marketing e políticas de preços de produtos (os jogadores são empresas que disputam mercados para vender seus produtos), campanhas de eleições políticas (os jogadores são candidatos que disputam eleitores para receber mais votos), programação de programas de televisão (os jogadores são redes de televisão que disputam espectadores para obter maior audiência), planejamento de estratégias militares de guerra (os jogadores são exércitos adversários). A teoria de jogos, em geral, admite que todos os jogadores são racionais e igualmente informados e tentam agir da melhor maneira possível para obter vantagens em relação a seus oponentes. Essas situações são diferentes daquelas em que o tomador de decisões está disputando um jogo com um adversário passivo, a natureza, que escolhe suas estratégias de maneira aleatória.

Um caso especial é o jogo em que dois jogadores  $A$  e  $B$  têm um número finito de estratégias: as estratégias  $A_1, A_2, \dots, A_m$  para o jogador  $A$  e as estratégias  $B_1, B_2, \dots, B_n$  para o jogador  $B$ . Para cada par  $(A_i, B_j)$  de estratégias escolhidas pelos jogadores  $A$  e  $B$ , o jogador  $A$  ganha  $a_{ij}$  unidades do jogador  $B$  (isto é, o jogador  $B$  perde  $a_{ij}$  unidades para o jogador  $A$ ). Ou seja, se o jogador  $A$  escolher a estratégia  $A_i$  e o jogador  $B$  escolher a estratégia  $B_j$ , então seus ganhos são  $a_{ij}$  e  $-a_{ij}$ , respectivamente. Tal jogo é conhecido como *jogo de duas pessoas soma-zero*, porque o ganho do jogador  $A$  é igual à perda do jogador  $B$  e vice-versa. Supõe-se que os dois jogadores conhecem todas as estratégias  $A_1, A_2, \dots, A_m$  e  $B_1, B_2, \dots, B_n$  e a tabela de ganhos  $\{a_{ij}\}$ , e que cada jogador escolhe simultaneamente uma estratégia sem saber a escolha do outro.

A solução ótima do jogo consiste em escolher uma estratégia para cada jogador, tal que qualquer mudança nas estratégias escolhidas não melhore os ganhos dos dois jogadores (ponto de equilíbrio). Essa escolha pode ter a forma de uma estratégia pura, por exemplo, a estratégia  $A_i$  para o jogador  $A$  e a estratégia  $B_j$  para o jogador  $B$ , ou a forma de uma estratégia mista (uma combinação de estratégias) para cada jogador de acordo com probabilidades predeterminadas, por exemplo:

- A estratégia  $A_1$  com probabilidade  $x_1$ , a estratégia  $A_2$  com probabilidade  $x_2$ , ..., a estratégia  $A_m$  com probabilidade  $x_m$ , para o jogador  $A$  ( $x_1 + x_2 + \dots + x_m = 1$ ).
- A estratégia  $B_1$  com probabilidade  $y_1$ , a estratégia  $B_2$  com probabilidade  $y_2$ , ..., a estratégia  $B_n$  com probabilidade  $y_n$ , para o jogador  $B$  ( $y_1 + y_2 + \dots + y_n = 1$ ).

As estratégias puras  $A_i$  e  $B_j$  acima podem ser vistas como casos particulares de estratégias mistas quando  $x_i = 1$  e  $y_j = 1$ , respectivamente (os demais  $x$  e  $y$  são iguais a 0). No caso de estratégias mistas, quando disputando o jogo, cada jogador sorteia uma de suas estratégias puras para jogar, de acordo com as probabilidades  $x$ 's e  $y$ 's.

**Exemplo 2.49**<sup>29</sup> Duas empresas  $A$  e  $B$  fabricam um produto que compete no mesmo mercado. As duas empresas podem fazer propagandas de vendas no rádio (estratégias  $A_1$  e  $B_1$ ), no jornal (estratégias  $A_2$  e  $B_2$ ) e na televisão (estratégias  $A_3$  e  $B_3$ ). Neste exemplo,  $m = n = 3$ , mas poderiam ser diferentes. A Tabela 2.27 apresenta a fatia do mercado (valores  $\{a_{ij}\}$  em percentuais) que as empresas esperam ganhar ou perder, uma da outra, para cada par  $(A_i, B_j)$  de estratégias escolhidas (valores positivos de  $a_{ij}$  indicam ganhos para a empresa  $A$  e perdas para a empresa  $B$ , e vice-versa).

Note, na Tabela 2.27, que, se a empresa  $A$  escolhe a estratégia pura  $A_1$ , então, independentemente da escolha que a empresa  $B$  fizer ( $B_1, B_2$  ou  $B_3$ ), o pior que pode acontecer à empresa  $A$  é perder 3% do mercado para a empresa  $B$  (isto é,  $\min\{3, -1, -3\} = -3$ , de acordo com a linha  $A_1$  da tabela). De modo semelhante, se a empresa  $A$  escolhe as estratégias puras  $A_2$  ou  $A_3$ , o pior que pode acontecer é ela perder 2% ou 6% do mercado para a empresa  $B$ , respectivamente (isto é,  $\min\{-2, 4, 1\} = -2$  e  $\min\{-5, -6, 2\} = -6$ , de acordo com as linhas  $A_2$  e  $A_3$  da tabela). Logo, o

<sup>29</sup> Este exemplo é baseado em um exemplo apresentado em Taha (2003).

**Tabela 2.27**  
**Fatia de mercado que as empresas A e B esperam ganhar ou perder.**

	$B_1$	$B_2$	$B_3$
$A_1$	3	-1	-3
$A_2$	-2	4	-1
$A_3$	-5	-6	2

melhor dos piores (critério maxmin) para a empresa A é escolher a estratégia pura  $A_2$  com perda de 2%, isto é:

$$\max \{\min \{3, -1, -3\}, \min \{-2, 4, 1\}, \min \{-5, -6, 2\}\} = \max \{-3, -2, -6\} = -2$$

Seguindo o mesmo raciocínio, se a empresa B escolhe a estratégia pura  $B_1$ , então, independentemente da escolha que a empresa A fizer ( $A_1, A_2$  ou  $A_3$ ), o pior que pode acontecer à empresa B é perder 3% do mercado para a empresa A (isto é,  $\max \{3, -2, -5\} = 3$ , de acordo com a coluna  $B_1$  da Tabela 2.27). De modo semelhante, se a empresa B escolhe as estratégias puras  $B_2$  ou  $B_3$ , ela perde 4% ou 2% do mercado para a empresa A, respectivamente (isto é,  $\max \{-1, 4, -6\} = 4$  e  $\max \{-3, -1, 2\} = 2$ , de acordo com as colunas  $B_2$  e  $B_3$  da tabela). Logo, o melhor dos piores (critério minmax) para a empresa B é escolher a estratégia pura  $B_3$  com perda de 2%, isto é:

$$\min \{\max \{3, -2, -5\}, \max \{-1, 4, -6\}, \max \{-3, -1, 2\}\} = \min \{3, 4, 2\} = 2$$

Como os valores obtidos pela melhor estratégia pura para a empresa A (-2 pelo critério maxmin) e pela melhor estratégia pura para a empresa B (2 pelo critério minmax) são diferentes, a solução ótima do jogo é uma estratégia mista para cada empresa, com valor entre -2 e 2 (se estes valores fossem iguais, este seria o valor do jogo, correspondendo a um ponto de sela). A melhor estratégia mista para a empresa A (com probabilidades  $x_1, x_2$  e  $x_3$ , a serem determinadas, de escolher as estratégias  $A_1, A_2$  e  $A_3$ , respectivamente) é escolher o melhor dos piores (critério maxmin), independentemente da escolha da empresa B. Para isso, deve-se procurar valores de  $x_1, x_2$  e  $x_3$  que maximizem o mínimo entre:

$$\begin{aligned} & 3x_1 - 2x_2 - 5x_3 && (\text{se a empresa } B \text{ escolher a estratégia } B_1) \\ & -1x_1 + 4x_2 - 6x_3 && (\text{se a empresa } B \text{ escolher a estratégia } B_2) \\ & -3x_1 - 1x_2 + 2x_3 && (\text{se a empresa } B \text{ escolher a estratégia } B_3) \end{aligned}$$

ou seja,

$$\max_{(x_1, x_2, x_3)} \{ \min \{3x_1 - 2x_2 - 5x_3, -1x_1 + 4x_2 - 6x_3, -3x_1 - 1x_2 + 2x_3\} \}$$

O primeiro termo  $3x_1 - 2x_2 - 5x_3$  corresponde ao ganho esperado da empresa A se a empresa B escolher a estratégia  $B_1$ , isto é, a empresa A ganha 3% com a probabilidade  $x_1$ , -2% com a probabilidade  $x_2$  e -5% com a probabilidade  $x_3$  (ver coluna  $B_1$  da tabela). Similarmente para os demais termos da expressão acima.

A empresa B também pode adotar uma estratégia mista (com probabilidades  $y_1, y_2$  e  $y_3$ , a serem determinadas, de escolher as estratégias  $B_1, B_2$  e  $B_3$ , respectivamente) para escolher o melhor dos piores (critério minmax), independentemente da escolha da empresa A. Para isso, deve procurar valores de  $y_1, y_2$  e  $y_3$  que minimizem o máximo entre:

$$\begin{aligned} & 3y_1 - 1y_2 - 3y_3 && (\text{se a empresa } A \text{ escolher a estratégia } A_1) \\ & -2y_1 + 4y_2 - 1y_3 && (\text{se a empresa } A \text{ escolher a estratégia } A_2) \\ & -5y_1 - 6y_2 + 2y_3 && (\text{se a empresa } A \text{ escolher a estratégia } A_3) \end{aligned}$$

ou seja,

$$\min_{(y_1, y_2, y_3)} \{ \max \{3y_1 - 1y_2 - 3y_3, -2y_1 + 4y_2 - 1y_3, -5y_1 - 6y_2 + 2y_3\} \}.$$

A interpretação desses termos é similar aos anteriores. Por exemplo, o primeiro termo  $3y_1 - 1y_2 - 3y_3$  indica que, se a empresa  $A$  escolhe a estratégia  $A_1$ , a empresa  $B$  perde 3% com a probabilidade  $y_1$ , -1% com a probabilidade  $y_2$  e -3% com a probabilidade  $y_3$  (ver linha  $A_1$  da tabela). ■

Jogos de duas pessoas soma-zero com estratégia mista podem ser modelados por otimização linear. De fato, G. Dantzig mencionou, em 1963, que J. von Neumann (pai da teoria de jogos e ganhador do Prêmio Nobel pela teoria dos jogos estratégicos), quando introduzido ao método simplex, em 1947, imediatamente reconheceu a relação do ponto de equilíbrio da teoria de jogos com o conceito de dualidade (Seção 2.10) em programação linear.<sup>30</sup> As probabilidades ótimas  $x_1^*, x_2^*, \dots, x_m^*$  para a estratégia mista do jogador  $A$  podem ser determinadas resolvendo-se o seguinte problema maxmin:

$$\begin{aligned} \text{Max}_{(x_1, x_2, \dots, x_m)} & \left\{ \text{Min} \left\{ \sum_{i=1}^m a_{i1}x_i, \sum_{i=1}^m a_{i2}x_i, \dots, \sum_{i=1}^m a_{in}x_i \right\} \right\} \\ & x_1 + x_2 + \dots + x_m = 1 \\ & x_i \geq 0, i = 1, \dots, m. \end{aligned}$$

Seja  $z = \min \left\{ \sum_{i=1}^m a_{i1}x_i, \sum_{i=1}^m a_{i2}x_i, \dots, \sum_{i=1}^m a_{in}x_i \right\}$ , o que implica:

$$z \leq \sum_{i=1}^m a_{i1}x_i, z \leq \sum_{i=1}^m a_{i2}x_i, \dots, z \leq \sum_{i=1}^m a_{in}x_i.$$

O modelo anterior pode ser resscrito como (ver Seção 2.2.9):

$$(A) \quad \text{Maximizar } z$$

$$\begin{aligned} \sum_{i=1}^m a_{i1}x_i & \geq z \\ \sum_{i=1}^m a_{i2}x_i & \geq z \\ & \vdots \\ \sum_{i=1}^m a_{in}x_i & \geq z \\ x_1 + x_2 + \dots + x_m & = 1 \\ x_i & \geq 0, i = 1, \dots, m, z \text{ irrestrito.} \end{aligned}$$

De modo semelhante, as probabilidades ótimas  $y_1^*, y_2^*, \dots, y_n^*$  para a estratégia mista do jogador  $B$  podem ser determinadas resolvendo-se o seguinte problema minmax:

$$\begin{aligned} \text{Min}_{(y_1, y_2, \dots, y_n)} & \left\{ \text{Max} \left\{ \sum_{j=1}^n a_{1j}y_j, \sum_{j=1}^n a_{2j}y_j, \dots, \sum_{j=1}^n a_{mj}y_j \right\} \right\} \\ & y_1 + y_2 + \dots + y_n = 1 \\ & y_j \geq 0, j = 1, \dots, n, \end{aligned}$$

que, utilizando o mesmo procedimento anterior, pode ser resscrito por:

$$(B) \quad \text{Minimizar } w$$

$$\begin{aligned} \sum_{j=1}^n a_{1j}y_j & \leq w \\ \sum_{j=1}^n a_{2j}y_j & \leq w \\ & \vdots \\ \sum_{j=1}^n a_{mj}y_j & \leq w \\ y_1 + y_2 + \dots + y_n & = 1 \\ y_j & \geq 0, j = 1, \dots, n, w \text{ irrestrito.} \end{aligned}$$

<sup>30</sup> Ver, por exemplo, Dantzig (1963), Shimizu (2001) e Taha (2003).

O problema (B) é o dual do problema (A) (verifique isso aplicando a Tabela 2.26 da seção 2.10), o que implica que a solução ótima de um produz a solução ótima do outro, e ambas têm o mesmo valor  $z^* = w^*$  (ver propriedade 2.13 da Seção 2.10).

**Exemplo 2.49 (continuação)** A solução ótima do jogo anterior das duas empresas soma-zero com estratégia mista pode ser obtida resolvendo-se os modelos de otimização linear (A) ou (B). Ou seja,

$$(A) \quad \text{Maximizar } z$$

$$\begin{aligned} 3x_1 - 2x_2 - 5x_3 &\geq z \\ -1x_1 + 4x_2 - 6x_3 &\geq z \\ -3x_1 - 1x_2 + 2x_3 &\geq z \\ x_1 + x_2 + \dots + x_m &= 1 \\ x_i &\geq 0, i = 1, 2, 3, z \text{ irrestrito.} \end{aligned}$$

$$(B) \quad \text{Miminizar } w$$

$$\begin{aligned} 3y_1 - 1y_2 - 3y_3 &\leq w \\ -2y_1 + 4y_2 - 1y_3 &\leq w \\ -5y_1 - 6y_2 + 2y_3 &\leq w \\ y_1 + y_2 + \dots + y_n &= 1 \\ y_j &\geq 0, j = 1, 2, 3, z \text{ irrestrito} \end{aligned}$$

As soluções ótimas desses modelos resultam em:  $z^* = w^* = -0,908$ ,  $(x_1^*, x_2^*, x_3^*) = (0,394, 0,312, 0,294)$  e  $(y_1^*, y_2^*, y_3^*) = (0,321, 0,083, 0,596)$ . Ou seja, a empresa A deve escolher a estratégia  $A_1$  com chance (39,4%) um pouco maior que as estratégias  $A_2$  e  $A_3$  (31,2% e 29,4%, respectivamente), enquanto a empresa B deve escolher a estratégia  $B_3$  com grande chance (59,6%) e a estratégia  $B_2$  com pequena chance (8,3%). Note que o valor ótimo do jogo  $z^*$  (para ambas as empresas A e B) está entre -2 e 2, conforme mencionado anteriormente. Pode-se mostrar que este valor corresponde ao valor esperado do jogo, ou seja,  $z^* = \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_i^* y_j^*$ , em que  $x_i^* y_j^*$  é a probabilidade de ocorrer o resultado  $a_{ij}$ , isto é, de a empresa A escolher a estratégia  $A_i$  e a empresa B escolher a estratégia  $B_j$ . ■

## 2.12 EXERCÍCIOS<sup>31</sup>

**Exercício 2.1** Considere o problema da mistura definido na Seção 2.2. Suponha que as frações dos componentes na mistura sejam limitadas inferior e superiormente por:  $b_i^{\min}$  e  $b_i^{\max}$ ,  $i = 1, \dots, m$ , respectivamente. Estenda o modelo matemático para o problema da mistura, considerando estes limitantes.

**Exercício 2.2** Considere a extensão do problema da mistura no Exercício 2.1. Suponha que se deseja produzir uma quantidade  $Q$  da mistura (por exemplo, 360kg) e os ingredientes estão disponíveis em estoque nas quantidades  $E_j$ ,  $j = 1, \dots, n$ . Estenda o modelo matemático da mistura. (Sugestões: considere a variável  $x_j$  como sendo a quantidade do ingrediente  $j$  em  $Q$  unidades da

<sup>31</sup> Vários exemplos e exercícios neste capítulo foram baseados em outros livros, os quais podem ser consultados como complemento para outros exemplos e exercícios de diversos graus de dificuldade, tais como Bazaraa *et al.* (1990), Bregalda *et al.* (1988), Dantzig (1963), Goldbarg e Luna (2000), Hillier e Lieberman (2004), Johnson e Montgomery (1974), Murty (1995), Puccini e Pizzolato (1987), Taha (2003), Wagner (1986), Winston (2004), Zenios (1993), dentre outros.

mistura. O total dos ingredientes agora deve somar  $Q$  e  $b_i^{\min} \times Q$  é a quantidade mínima do componente  $i$  em  $Q$  unidades da mistura.)

**Exercício 2.3** Uma fundição tem de produzir 10 toneladas de um tipo de liga metálica e, para isso, tem disponível: lingotes de ferro, grafite e sucata. Dois componentes são relevantes para a liga: carbono e silício. A Tabela 2.28 fornece a fração desses elementos nos ingredientes disponíveis, seus custos unitários, suas disponibilidades em estoque, bem como a composição da liga (isto é, porcentagens mínimas e máximas de cada componente na liga). Escreva um modelo de otimização linear para determinar as quantidades dos ingredientes para compor a liga metálica, de modo que as especificações técnicas sejam satisfeitas e o custo seja mínimo. Transforme o modelo na forma padrão.

**Exercício 2.4** Considere a extensão do problema da mistura no Exercício 2.2. Suponha que  $K$  misturas devem ser produzidas a partir dos mesmos ingredientes (por exemplo, rações para caninos, felinos, galináceos etc.), nas quantidades  $Q_k$ ,  $k = 1, \dots, K$ . Considere  $b_{ik}^{\min}$  e  $b_{ik}^{\max}$  como as frações mínimas e máximas do componente  $i$  na mistura  $k$ ,  $i = 1, \dots, m$ ,  $k = 1, \dots, K$ . Os custos unitários dos ingredientes são  $c_j$ ,  $j = 1, \dots, n$ . Estenda o modelo matemático do Exercício 2.2 para várias misturas, de modo que cada mistura  $k$  atenda as especificações técnicas e o custo total seja mínimo. (Sugestão: seja  $x_{jk}$  a quantidade do ingrediente  $j$  na mistura  $k$  e note que  $x_{j1} + x_{j2} + \dots + x_{jK}$  é o total do ingrediente  $j$  utilizado em todas as misturas, que não pode superar a quantidade em estoque do ingrediente  $j$ .)

**Exercício 2.5** Considere o problema do Exercício 2.3. Suponha agora que duas ligas metálicas devem ser preparadas e os mesmos ingredientes são utilizados em ambas. A liga especificada no Exercício 2.3 é referida como liga 1 e devem ser produzidas  $Q_1 = 10$  toneladas desta liga. Da outra liga, referida como liga 2, devem ser produzidas  $Q_2 = 6$  toneladas e sua composição é dada na Tabela 2.29.

Escreva um modelo de otimização linear para a produção das duas ligas. Transforme o modelo obtido na forma padrão.

**Exercício 2.6** Considere o problema da mistura na Seção 2.2. Suponha que os ingredientes estejam divididos em três classes:  $C_1$ ,  $C_2$  e  $C_3$ , isto é, o conjunto de todos os ingredientes:  $\{1, 2, \dots, n\} = C_1 \cup C_2 \cup C_3$  (por exemplo, uma dieta alimentar é constituída de carnes, que constituem a clas-

**Tabela 2.28**  
Composição dos ingredientes.

Composição (%)	Ingredientes			Liga	
	Lingotes	Grafite	Sucata	Composição mínima	Composição máxima
Carbono	0,0050	0,90	0,090	0,00	0,095
Silício	0,14	-	0,27	0,19	0,20
Custos (R\$/ton)	90	180	25		
Estoque (ton)	5	5	12		

**Tabela 2.29**  
**Composição da liga 2.**

Composição	Liga 2	
	Composição mínima	Composição máxima
Carbono	0,00	0,010
Silício	0,12	0,14

se 1; cereais, a classe 2; verduras e legumes, a classe 3). A mistura deve ter: de 20% a 30% de ingredientes da classe  $C_1$ , de 50% a 60% de  $C_2$  e de 30% a 40% de  $C_3$ . Escreva um modelo matemático em que as composições especificadas sejam atendidas (ver o modelo da Seção 2.2) e as novas restrições de classe sejam verificadas. (Sugestões: o modelo pode ser escrito considerando que  $x_j$  é a quantidade de ingredientes da classe 1 ou supondo que os ingredientes estejam organizados por classe:  $C_1 = \{1, 2, \dots, j_1\}$ ,  $C_2 = \{j_1 + 1, \dots, j_2\}$ , ... e  $\sum_{j=1}^{j_1} x_j$  é a quantidade de ingredientes da classe 1.)

**Exercício 2.7\*** Considere o problema da mistura no Exercício 2.1. Em algumas aplicações, é indesejável que um número grande de ingredientes seja utilizado, por questões operacionais. Suponha que no máximo três ingredientes devam ser utilizados na mistura. Uma abordagem heurística para considerar essa nova restrição pode ser a seguinte. Suponha que o problema da mistura seja resolvido, sem a restrição de três ingredientes, e a solução envolve 4 ingredientes, digamos,  $x_1 > 0, x_2 > 0, x_3 > 0, x_4 > 0$  (as demais variáveis têm valores nulos, ou seja, os demais ingredientes não são escolhidos para constituir a mistura). Este é um problema relaxado, pois não inclui todas as restrições do problema. Como se deseja usar apenas três ingredientes, podemos resolver problemas menores combinando apenas três dos ingredientes que aparecem na solução relaxada: 1–2–3, 1–2–4, 1–3–4 ou 2–3–4, isto é, resolvendo problemas com menos variáveis ( $x_1, x_2, x_3$  ou  $x_1, x_2, x_4$  ou  $x_2, x_3, x_4$  ou  $x_1, x_3, x_4$ ) e escolhendo a melhor solução. (Esta abordagem heurística pode funcionar bem, mas a melhor solução pode envolver outro ingrediente que não foi usado na solução relaxada, pois as quatro alternativas podem nem ter solução factível.) Formule um modelo matemático para incluir a nova restrição (Sugestão: defina  $y_j$  como uma variável que assume apenas valores 0 ou 1, de modo que:  $y_j = 1$  se  $x_j > 0$  e  $y_j = 0$ , caso contrário. As restrições  $x_j \leq y_j$  descrevem essas restrições lógicas. A soma dessas variáveis fornece o número de ingredientes usados na mistura. Tais modelos serão estudados no Capítulo 3 – “Otimização discreta”.)

**Exercício 2.8\*** Considere a extensão do problema da mistura no Exercício 2.2. Suponha, em adição, que a demanda ocorra em períodos de um horizonte de planejamento, digamos,  $Q_1, Q_2, \dots, Q_T$  são as quantidades (em toneladas) da mistura que devem estar disponíveis ao final dos períodos 1, 2, ...,  $T$  (por exemplo,  $Q_1$  é a quantidade de uma ração que deve estar disponível no final da semana 1). A capacidade do misturador no período  $t$ , em quantidade de mistura, é limitada por  $C_t$  (os períodos podem corresponder a intervalos de tempo variados e, portanto, a capacidade pode variar por período. Por exemplo, os períodos 1 e 2 podem ser um dia de trabalho e o período 3 pode ser 4 dias. Isso é comum na prática, em que os períodos iniciais são detalhados e os períodos finais agregados, pois os dados estão sujeitos a alterações e somente as decisões correspondentes aos primeiros períodos são de fato implementadas. Depois de implementar as decisões dos primeiros períodos, diz-se que o horizonte é rolado, com novos dados incluídos, e o

problema é resolvido novamente. Essa estratégia é conhecida como *horizonte rolante*.<sup>32)</sup> A demanda em um período pode ser maior do que a capacidade de produção, de modo é necessário antecipar a produção e manter o produto em estoque (veja os modelos dinâmicos de planejamento da produção da Seção 2.2, pois a solução lote-por-lote, isto é, produzir exatamente a demanda em cada período, pode ser infactível) e os custos de produção podem variar de um período para outro (custos de energia elétrica, hora extra etc.). Considere  $c_t$  (\$/tonelada) o custo de produzir uma tonelada da mistura no período  $t$  e  $d_t$  (\$/tonelada) o custo de estocar uma tonelada da mistura no final do período  $t$ . Seja  $E_{jt}$  o estoque do ingrediente  $j$ ,  $j = 1, \dots, n$ , disponível para o período 1. Considere que haja reposição do estoque do ingrediente  $j$  ao longo do horizonte de planejamento (compras efetuadas anteriormente), digamos,  $E_{jt}$  é a quantidade de ingrediente  $j$  a ser adicionada ao estoque do ingrediente  $j$  no período  $t$ . Escreva um modelo matemático para determinar como devem ser as produções da mistura em cada período, de modo que a demanda seja atendida em cada período, bem como as misturas atendam as especificações técnicas, isto é, com a composição dentro dos limites definidos, e o custo de produção e estocagem sejam minimizados. (Sugestão: defina a variável  $x_{jt}$  como a quantidade do ingrediente  $j$  no total de mistura produzida no período  $t$ . A quantidade  $z_t = x_{1t} + \dots + x_{nt}$  é o total produzido da mistura no período  $t$  e não é necessariamente igual a  $Q$ . Defina, também, a variável  $I_t$  como a quantidade de mistura em estoque no final do período  $t$ .) Proponha uma extensão desse modelo para o caso de várias misturas (Exercício 2.4).

**Exercício 2.9** Devido à grande permeabilidade, areias são usadas na constituição de filtros de Estações de Tratamento de Águas de abastecimento (ETA) como meio filtrante, por interceptar as impurezas existentes na água afluente. Essas areias são dispostas em camadas, que devem obedecer às composições granulométricas estabelecidas por norma técnica, por exemplo, nas quantidades dadas na Tabela 2.30:

**Tabela 2.30**  
**Composição de um filtro.**

Faixa granulométrica (mm)	Volume de areia ( $m^3$ )
0,42 – 0,59	16
0,59 – 0,71	16
0,71 – 0,84	16
0,84 – 1,00	64
1,00 – 1,19	40
1,19 – 1,41	8

Para a construção das unidades de filtração de uma ETA, areias são exploradas de diferentes portos, com composições granulométricas distintas. Os custos totais de dragagem, transporte, seleção e preparo para a utilização da areia são conhecidos por unidade de volume ( $m^3$ ) para cada porto. Na Tabela 2.31, as composições granulométricas de dois tipos de areia são fornecidas, além dos custos totais. Por exemplo, 17% da areia proveniente do porto 1 tem os diâmetros de seus grãos entre 0,42mm e 0,59mm, 16% entre 0,59 e 0,71 etc. e custa \$ 25,00 por  $m^3$ . Assim, de 100  $m^3$  da área do ponto 1, pode-se extrair  $0,17 \times 100 = 17 m^3$  de areia na faixa granulométrica

<sup>32</sup> Araújo e Arenales (2004) utilizam a técnica de horizonte rolante na programação da produção em uma fundição de médio porte.

**Tabela 2.31**  
**Composição das areias disponíveis**

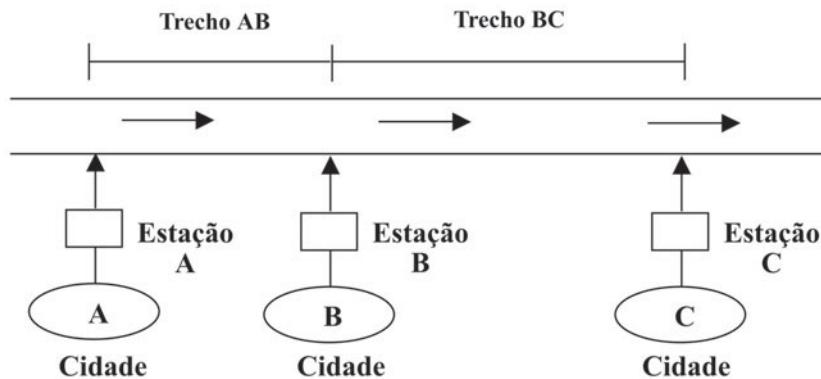
Faixa granulométrica (mm)	Volume de areia/m <sup>3</sup>	
	porto - 1	porto - 2
0,42 - 0,59	0,17	0,13
0,59 - 0,71	0,16	0,11
0,71 - 0,84	0,18	0,14
0,84 - 1,00	0,10	0,09
1,00 - 1,19	0,09	0,12
1,19 - 1,41	0,05	0,07
custo (\$/m <sup>3</sup> )	25,00	19,00

0,42–0,59, que é suficiente para a construção do filtro, que necessita 16 m<sup>3</sup> desta faixa, mas essa quantidade é insuficiente para se obter a quantidade de areia necessária na faixa 0,84–1,00.

- a) Escreva um modelo matemático de otimização linear para determinar a combinação das diferentes areias de modo a atender às especificações da norma, com o mínimo custo possível.
- b) Resolva o problema graficamente.
- c) Suponha que o excesso de areia de cada faixa granulométrica possa ser vendido para terceiros, com preços específicos por faixa (por exemplo, P<sub>1</sub> é o preço da faixa 0,42–0,59 etc.). Como este é um negócio secundário, mesmo que rentável, as quantidades em excesso são limitadas, digamos, U<sub>1</sub> é o excesso máximo aceitável para a faixa 0,42–0,59 etc. Reformule o problema considerando os ganhos com as vendas e as limitações de excesso em cada faixa granulométrica.

**Exercício 2.10** Considere o modelo no Exemplo 2.6 na Seção 2.2, para determinar o *mix* de produção do fabricante de geladeiras. (a) Represente a região factível em um plano cartesiano e determine a solução ótima. (b) Escreva o problema na forma padrão e identifique as fronteiras no gráfico do item (a) pelas variáveis que se anulam. (c) Identifique no gráfico do item (a) a base ótima (identifique as variáveis que se anulam e considere-as não-básicas). (d) Aplique o método simplex, considerando a base inicial formada pelas colunas das variáveis de folga e determine a solução ótima.

**Exercício 2.11\*** Considere o modelo (2.1)–(2.7) do Exemplo 2.4 na Seção 2.2. (a) Determine uma solução factível inicial, usando o seguinte procedimento: Faça  $x_{11} = 433$  toneladas (isto é, transporte 433 toneladas da pedreira 1 ao depósito 1). A demanda residual do depósito 1 é  $697 - 433 = 264$ . Faça  $x_{21} = 215$ . A demanda residual do depósito 1 é  $264 - 215 = 49$ . Faça  $x_{31} = 49$  (a demanda para o depósito 1 é satisfeita). A oferta residual da pedreira 3 é  $782 - 49 = 733$ . Continue o procedimento até que as demandas tenham sido satisfeitas. (b) Considere a matriz formada pelas colunas das variáveis positivas na solução de (a), por exemplo, coluna de  $x_{11}$ , de  $x_{21}$  etc. junto com a primeira coluna da matriz identidade  $7 \times 7$ . Mostre que essa matriz é básica. (c) Aplique o método simplex a partir da solução básica obtida nos itens (a) e (b) e determine a solução ótima. Observe a simplicidade na resolução dos sistemas básicos, que sempre podem ser resolvidos por substituição (no Capítulo 4 – Otimização em redes, esta característica é estudada em detalhes).



**Figura 2.53** Lançamento de esgoto em um rio.

**Exercício 2.12** Considere o modelo no Exemplo 2.1 na Seção 2.2 para determinar as frações ideais de farinha de osso, soja e peixe em uma ração animal. (a) Resolva o modelo pelo método das duas fases (Seção 2.8). (b) Há alguma característica do modelo que favoreça o uso do método dual simplex? (veja Exercício 2.33).

**Exercício 2.13** Os esgotos de três cidades, A, B, e C, depois de passarem por uma estação de tratamento, são jogados em um rio, conforme a Figura 2.53. Os esgotos das três cidades produzem quantidades diárias de poluentes, respectivamente,  $P_A$ ,  $P_B$  e  $P_C$  toneladas. O tratamento do esgoto pode reduzir a quantidade de poluentes em até 90%. Essa redução é denominada eficiência da estação de tratamento e o custo da estação  $i$  é diretamente proporcional à sua eficiência.

Por outro lado, devido à ação bioquímica (aeração etc.), no final de cada trecho AB e BC do rio, a quantidade de poluentes é reduzida em 10% e 20%, respectivamente. Quais são as eficiências das estações de tratamento, de modo que, em qualquer ponto do rio a quantidade de poluentes não ultrapasse  $P$  toneladas e o custo das estações de tratamento seja mínimo? Escreva um modelo de otimização linear para este problema.

**Exercício 2.14** Considere o problema de corte de bobinas na Seção 2.2 e o modelo de otimização linear em (2.19).

- Suponha que haja uma tolerância  $\tau_i$  na demanda do item  $i$ , isto é, os clientes aceitam qualquer quantidade do item  $i$  no intervalo  $[(1 - \tau_i)b_i, (1 + \tau_i)b_i]$ . Reformule o modelo (2.19) para considerar essa tolerância.
- Escreva um modelo para o Exemplo 2.9, considerando uma tolerância de  $\tau_i = 0,05$  (isto é,  $\pm 5\%$ ), de modo que a demanda do item de tamanho  $l_1 = 40$  é qualquer valor entre 4,75 e 5,25 toneladas).

**Exercício 2.15\*** Considere, por simplicidade, o modelo de corte de bobinas (2.16) com a tolerância no Exercício 2.14. Como a função objetivo busca minimizar a quantidade de bobinas cortadas com a tolerância introduzida, a solução do problema deve atender a demanda minimamente (menor a demanda, menor a quantidade cortada), que não necessariamente é de interesse da empresa. Reformule a função objetivo de modo que a contribuição ao lucro da empresa seja maximizada, considerando que o custo de cada bobina é  $\$ c$ , o preço de venda do item  $i$  é  $\$ p_i$ . Além disso, cada cm de sobra é vendido por  $\$ s$ . (Sugestão: cortar  $x_1$  bobinas conforme o padrão

de corte 1 produz  $a_{il}x_1$  itens  $i$  vendidos por  $\$ p_i a_{il} x_1$ , de modo que  $\$ \sum_{i=1}^m p_i a_{il} x_1$  é o valor das vendas dos itens decorrente do uso do padrão de corte 1. O comprimento, em cm, da sobra no padrão de corte 1 é  $L - \sum_{i=1}^m \ell_i a_{il}$ , de modo que  $\$ s(L - \sum_{i=1}^m \ell_i a_{il})$  é o valor da venda da sobra devido ao uso do padrão de corte 1. O custo das bobinas cortadas pelo padrão de corte 1 é  $\$ cx_1$ . O preço total de vendas menos o custo define a contribuição ao lucro da empresa a ser maximizada.)

**Exercício 2.16** Uma nova máquina deve ser instalada em uma fábrica cujo piso tem formato retangular e os cantos opostos têm as seguintes coordenadas (unidade em metros):  $(-40 \ -40)^T$  e  $(40 \ 20)^T$ . Existem quatro máquinas já instaladas nas posições:  $(0 \ 0)^T$ ,  $(-40 \ -40)^T$ ,  $(-30 \ 10)^T$  e  $(-35 \ 0)^T$ . Formule um modelo matemático para determinar a localização ótima da nova máquina, considerando que:

- A distância entre a nova máquina e as demais seja mínima. Use a distância “reticulada” (como se o deslocamento fosse pelas ruas de uma cidade), isto é, se  $x_1$  e  $x_2$  são as coordenadas da nova máquina, então a distância entre a nova máquina e outra instalada na posição  $(y_1 \ y_2)^T$  é dada por:  $|x_1 - y_1| + |x_2 - y_2|$ . Transforme o problema na forma linear (veja os exemplos de ajuste de curvas e controle ótimo na Seção 2.2).
- As movimentações de itens por hora, entre a nova máquina e as já instaladas, são dadas por: 10, 8, 15, 0. Reformule o problema para que a distância total percorrida pelos itens seja minimizada.
- A instalação da nova máquina deve ser tal que:  $x_1 \geq 10$ . Reescreva os modelos em (i) e (ii) com esta nova restrição.

**Exercício 2.17** Uma cooperativa de laticínios fabrica três produtos: leite pasteurizado, queijo fresco e iogurte. A cooperativa recebe diariamente 100 mil litros de leite por dia, os quais devem ser processados no mesmo dia. Há um compromisso de abastecer o mercado diário de no mínimo 50 mil litros de leite pasteurizado, cuja contribuição ao lucro é de \$0,07/litro. Um quilo de queijo fresco consome 10 litros de leite, enquanto um litro de iogurte consome 2,5 litros de leite. As contribuições ao lucro são: \$1,04 por quilo de queijo e \$0,20 por litro de iogurte. O máximo que o mercado assimila diariamente de cada produto é de 60 mil litros de leite pasteurizado, 3.500 quilos de queijo e 20 mil litros de iogurte. (a) Formule um modelo matemático para determinar as quantidades ideais de cada produto. (b) Fixe a venda de leite pasteurizado em 50.000 litros (produto menos lucrativo) e resolva o problema graficamente. (c) Resolva o problema usando o método simplex. (d) O departamento de vendas entende que as vendas de queijo e iogurte não são independentes e impõe que, para cada quilo de queijo produzido, pelo menos um litro de iogurte deve ser produzido. Escreva o modelo com esta restrição e resolva o problema graficamente, fixando-se a produção de leite pasteurizado em 50 mil litros. (e)\* Use o método dual simplex para reotimizar a solução obtida no item (c), depois da inclusão da nova restrição no item (d).

**Exercício 2.18** Considere o seguinte problema:

$$\begin{aligned} \text{Minimizar } f(x_1, x_2) &= -x_1 - x_2 \\ -x_1 + x_2 &\leq 2 \\ 2x_1 - x_2 &\leq 6 \\ x_1 \geq 0, \quad x_2 &\geq 0 \end{aligned}$$

- Resolva o problema graficamente (isto é, desenhe a região factível e identifique a(s) solução(ões) ótima(s)).

- b.** A solução  $x_1 = x_2 = 0$  é um vértice da região factível? Identifique todos os vértices da região factível.
- c.** Desenhe as soluções  $\mathbf{x}' = (x'_1, x'_2)^T = (1, 1)^T$  e  $\mathbf{x}'' = (x''_1, x''_2)^T = (5, 1)^T$ . Essas soluções são factíveis? Responda verificando que: i) pertence à região factível no gráfico do item (a) e ii) satisfaz as restrições.
- d.** Qual é a solução se o problema for: Maximizar  $f(x_1, x_2) = x_1 + x_2$ ?
- e.** Considere agora uma outra função objetivo: Minimizar  $f(x_1, x_2) = x_1 - x_2$ . Verifique se a solução ótima obtida no item (a) também é ótima considerando esta nova função objetivo. Há múltiplas soluções ótimas? Identifique no gráfico.
- f.** Suponha que, no problema original, a função objetivo seja alterada para:  
Minimizar  $f(x_1, x_2) = (-1 + \delta)x_1 - x_2$  (ou seja, o custo da variável  $x_1$  é alterado para:  $c_1 = -1 + \delta$ ). Desenhe no gráfico do item (a) os coeficientes da função objetivo para diferentes valores de  $\delta$ . Para que valores de  $\delta$  a solução ótima encontrada no item (a) permanece ótima?

**Exercício 2.19** Considere o seguinte problema:

$$\begin{aligned} \text{Minimizar } f(x_1, x_2) &= x_1 + x_2 \\ -x_1 + x_2 &\geq 2 \\ 2x_1 - x_2 &\leq 6 \\ x_1 \geq 0, \quad x_2 &\geq 0 \end{aligned}$$

- a.** Resolva o problema graficamente.
- b.** Considere agora: Maximizar  $f(x_1, x_2) = x_1 + x_2$  sujeito às mesmas restrições. O que mudou?
- c.** Construa uma nova função objetivo de modo que o problema tenha: i) um segmento de soluções ótimas; ii) uma semireta de soluções ótimas.
- d.** Considere o problema no item (b) e inclua a terceira restrição:  $x_1 + x_2 \leq 1$ . Resolva o problema resultante graficamente.
- e.** Considere perturbações no coeficiente da terceira restrição do tipo:  $b_3 = 1 + \delta$ ,  $\delta \geq 0$ . Represente graficamente essas perturbações. Observe que, para valores altos de  $\delta$ , o problema torna-se factível.
- f.** Formule um problema de otimização linear para determinar o menor valor de  $\delta$  que faz com que a região factível seja não-vazia. (Sugestão: encare  $\delta$  como uma variável do problema.)

**Exercício 2.20** Considere o sistema de equações lineares  $\mathbf{Ax} = \mathbf{b}$  dado por (2.25), fixe:  $x_3 = 0$ ,  $x_4 = 1$  e determine os valores das variáveis restantes. Identifique a solução obtida na Figura 2.35. Além disso, fixe:  $x_3 = 0$ ,  $x_4 = 2$ . Identifique esta nova solução na Figura 2.35. Repita com outros valores para  $x_4$ . Desenhe a reta obtida com a variação de  $x_4$ , mantendo  $x_3 = 0$ . Certifique-se que esta reta, no plano  $(x_1, x_2)$ , é dada por:  $x_1 + x_2 = 6$  (veja a primeira equação de (2.25)).

**Exercício 2.21** Fixe em (2.29) as variáveis em  $\mathbf{x}_N$ :  $x_3 = 0$  e  $x_4 = 1$ . Faça as operações usando matrizes e compare o resultado com o Exercício 2.20.

**Exercício 2.22** Reescreva o sistema de equações lineares  $\mathbf{Ax} = \mathbf{b}$  dado em (2.25), considerando  $x_2$  e  $x_3$  como as variáveis independentes, na forma equivalente  $\mathbf{Bx}_B + \mathbf{Nx}_N = \mathbf{b}$ . Explicite os índices básicos:  $B_1$ ,  $B_2$  e  $B_3$  e os índices não-básicos:  $N_1$  e  $N_2$ ; as matrizes  $\mathbf{B}$  e  $\mathbf{N}$ ; e os vetores de variáveis  $\mathbf{x}_B$  e  $\mathbf{x}_N$ . Fixe  $\mathbf{x}_N = \mathbf{0}$  (isto é,  $x_2 = x_3 = 0$ ) e compare com o sistema de equações lineares em (2.27).

**Exercício 2.23** Repita o Exercício 2.22, considerando  $x_1$  e  $x_2$  as variáveis independentes. Qual é o ponto na Figura 2.35 que corresponde à fixação das variáveis independentes em zero? É uma solução factível?

**Exercício 2.24** (*solução ótima única e custo relativo nulo*) Considere o seguinte problema de optimização linear:

$$\begin{aligned} \text{Minimizar } f(x_1, x_2) &= -x_1 - x_2 \\ x_1 + x_2 &\leq 6 \\ x_1 - x_2 &\leq 4 \\ x_2 &\leq 1 \\ x_1, x_2 &\geq 0 \end{aligned}$$

- Resolva graficamente e identifique a solução ótima única  $\mathbf{x}^* = (5 \ 1)^T$ .
- Considere a partição básica ótima em que  $x_3$  e  $x_5$  são não-básicas, isto é,  $\mathbf{B} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \mathbf{a}_4]$  (note que há outras partições básicas ótimas). Calcule a solução básica (é degenerada?).
- Verifique as condições de otimalidade (determine  $\boldsymbol{\lambda} = (-1 \ 0 \ 0)^T$  e  $\hat{c}_3 = 1$  e  $\hat{c}_5 = 0$ ).
- Como  $f(\mathbf{x}) = f(\mathbf{x}^*) + \hat{c}_3 x_3 + \hat{c}_5 x_5$  (veja (2.36)), a única maneira de obter soluções alternativas ótimas é aumentar  $x_5$  mantendo  $x_3 = 0$ . Mostre que a variável não-básica  $x_5$  não pode assumir valores positivos e conclua que um custo relativo nulo na solução ótima não necessariamente garante múltiplas soluções ótimas. (Sugestão: se fizermos  $x_5 = \varepsilon \geq 0$ , mantendo  $x_3 = 0$ , por (2.38),  $\mathbf{x}_B = \hat{\mathbf{x}}_B - \mathbf{y}\varepsilon$  (encontre  $\mathbf{y} = (-1 \ 1 \ 2)^T$ ), segue-se que  $x_1 = 5 + \varepsilon$ ,  $x_2 = 1 - \varepsilon$ ,  $x_4 = 0 - 2\varepsilon$  e, portanto,  $\varepsilon$  não pode ser positivo, ou seja, não é possível outra solução ótima. Note o papel de  $y_3 > 0$ ).
- Se a solução ótima é não degenerada e tem um custo relativo nulo, pode-se afirmar que existem múltiplas soluções ótimas? (Raciocine com a solução do item (d), imaginando  $\hat{x}_4 > 0$ .)
- Considere agora uma nova função objetivo: Minimizar  $f(x_1, x_2) = -x_1$ . Verifique graficamente que a solução ótima é a mesma do item (a). A partição básica no item (b) fornece essa solução ótima. Verifique, entretanto, que a condição de otimalidade não é verificada. Conclua que podemos ter em mãos uma solução ótima sem que a condição de otimalidade seja verificada. Isso poderia ocorrer para uma solução não degenerada?

**Exercício 2.25\*** Considere uma matriz básica  $m \times m$   $\mathbf{B} = [\mathbf{a}_{B_1} \cdots \mathbf{a}_{B_{\ell-1}} \mathbf{a}_{B_\ell} \mathbf{a}_{B_{\ell+1}} \cdots \mathbf{a}_{B_m}]$  em uma iteração do método simplex (ou dual simplex). Mostre que a nova matriz  $\mathbf{B}' = [\mathbf{a}_{B_1} \cdots \mathbf{a}_{B_{\ell-1}} \mathbf{a}_{N_k} \mathbf{a}_{B_{\ell+1}} \cdots \mathbf{a}_{B_m}]$ , obtida pelo método simplex (ou dual simplex), também é uma base, isto é, existe  $(\mathbf{B}')^{-1}$ . Apresente uma forma simples de calcular  $(\mathbf{B}')^{-1}$  a partir de  $\mathbf{B}^{-1}$ . Sugestão: Siga os seguintes passos:

- Multiplique  $\mathbf{B}^{-1}$  por  $\mathbf{B}'$  e obtenha a matriz  $\mathbf{E}$ , isto é,  $\mathbf{B}^{-1} \mathbf{B}' = \mathbf{E}$ . Como  $\mathbf{B}^{-1} \mathbf{B} = \mathbf{I}$ , segue-se que  $\mathbf{B}^{-1} \mathbf{a}_{B_i}$  é a  $i$ -ésima coluna da matriz identidade. Assim, a matriz  $\mathbf{E}$  difere da matriz identidade apenas pela  $\ell$ -ésima coluna:

$$\mathbf{B}^{-1} \mathbf{B}' = \left[ \mathbf{B}^{-1} \mathbf{a}_{B_1} \cdots \underbrace{\mathbf{B}^{-1} \mathbf{a}_{N_k}}_{=\mathbf{y}} \cdots \mathbf{B}^{-1} \mathbf{a}_{B_m} \right] = \begin{bmatrix} 1 & \cdots & y_1 & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & & y_\ell & & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & y_m & \cdots & 1 \end{bmatrix} \underbrace{\mathbf{E}}_{\mathbf{E}}$$

com  $y_\ell$  na diagonal da matriz  $\mathbf{E}$  e  $y_\ell > 0$  por (2.39) no caso do método simplex (ou  $y_\ell = -\eta_\ell^T \mathbf{a}_{N_k} < 0$  por (2.47) no caso do método dual simplex).

- ii) Mostre que a matriz  $\mathbf{E}$  tem inversa dada por:  $\mathbf{E}^{-1} = \begin{bmatrix} 1 & \dots & -\frac{y_1}{y_\ell} & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & & \frac{1}{y_\ell} & & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & -\frac{y_m}{y_\ell} & \dots & 1 \end{bmatrix}$

Basta fazer o produto  $\mathbf{EE}^{-1}$  e obter a matriz identidade. Esta matriz é chamada matriz elementar.

- iii) Dessa forma,  $\mathbf{B}^{-1}\mathbf{B}' = \mathbf{E} \Leftrightarrow \underbrace{\left(\mathbf{E}^{-1}\mathbf{B}^{-1}\right)}_{(\mathbf{B}')^{-1}} \mathbf{B}' = \mathbf{I}$ , do que se segue:  $(\mathbf{B}')^{-1} = \mathbf{E}^{-1}\mathbf{B}^{-1}$ , ou seja, a inversa de  $\mathbf{B}'$  existe por construção, portanto,  $\mathbf{B}'$  é uma base. O próximo passo mostra um esquema prático de como obter  $(\mathbf{B}')^{-1}$  sem a necessidade de se trabalhar explicitamente com  $\mathbf{E}^{-1}$ .

- iv) Seja  $\mathbf{v}_i$  a  $i$ -ésima linha de  $\mathbf{B}^{-1}$ , então

$$\mathbf{E}^{-1}\mathbf{B}^{-1} = \begin{bmatrix} \mathbf{v}_1 - \frac{y_1}{y_\ell} \mathbf{v}_\ell \\ \vdots \\ \frac{1}{y_\ell} \mathbf{v}_\ell \\ \vdots \\ \mathbf{v}_m - \frac{y_m}{y_\ell} \mathbf{v}_\ell \end{bmatrix},$$

ou seja, a  $i$ -ésima linha de  $(\mathbf{B}')^{-1}$ ,  $i \neq \ell$ , é obtida pela adição da  $\ell$ -ésima linha de  $\mathbf{B}^{-1}$  multiplicada por  $-\frac{y_i}{y_\ell}$  à  $i$ -ésima linha de  $\mathbf{B}^{-1}$ . Observe o esquema prático decorrente destas operações: coloque o vetor  $\mathbf{y}$  (direção simplex) ao lado da matriz  $\mathbf{B}^{-1}$  e faça as operações elementares necessárias para transformar o vetor  $\mathbf{y}$  na  $\ell$ -ésima coluna da matriz identidade, estendendo as mesmas operações sobre a matriz  $\mathbf{B}'$ .

**Exercício 2.26\*** Considere que a matriz  $\mathbf{B}'$  foi obtida de  $\mathbf{B}$ , substituindo-se a  $\ell$ -ésima coluna  $\mathbf{a}_{B_\ell}$  por  $\mathbf{a}_{N_k}$ , seja pelo método simplex ou dual simplex. Do Exercício 2.25, segue-se que  $(\mathbf{B}')^{-1} = \mathbf{E}^{-1}\mathbf{B}^{-1}$ , do que decorre um esquema prático para o produto  $\mathbf{E}^{-1}\mathbf{B}^{-1}$ , sem a necessidade de se explicitar  $\mathbf{E}^{-1}$ . Dê um esquema prático para o cálculo da solução básica e do vetor multiplicador simplex associados a  $\mathbf{B}'$ , isto é,  $\hat{\mathbf{x}}_{\mathbf{B}'} = (\mathbf{B}')^{-1}\mathbf{b}$  e  $(\lambda')^T = \mathbf{c}_{\mathbf{B}'}^T(\mathbf{B}')^{-1}$ .

Sugestões:

- i) Para o cálculo da solução básica, faça:  $\mathbf{x}_{\mathbf{B}_B} = (\mathbf{B}')^{-1}\mathbf{b} = \mathbf{E}^{-1}\mathbf{B}^{-1}\mathbf{b} = \mathbf{E}^{-1}\hat{\mathbf{x}}_{\mathbf{B}_B}$ . O produto  $\mathbf{E}^{-1}\hat{\mathbf{x}}_{\mathbf{B}_B}$  é obtido a partir da matriz elementar  $\mathbf{E}^{-1}$  dada no Exercício 2.25:

$$\begin{bmatrix} \hat{x}_{B'_1} \\ \vdots \\ \hat{x}_{B'_\ell} \\ \vdots \\ \hat{x}_{B'_m} \end{bmatrix} = \begin{bmatrix} 1 & \dots & -\frac{y_1}{y_\ell} & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & & \frac{1}{y_\ell} & & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & -\frac{y_m}{y_\ell} & \dots & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{B_1} \\ \vdots \\ \hat{x}_{B_\ell} \\ \vdots \\ \hat{x}_{B_m} \end{bmatrix} = \begin{bmatrix} \hat{x}_{B_1} - \frac{y_1}{y_\ell} \hat{x}_{B_\ell} \\ \vdots \\ \frac{1}{y_\ell} \hat{x}_{B_\ell} \\ \vdots \\ \hat{x}_{B_m} - \frac{y_m}{y_\ell} \hat{x}_{B_\ell} \end{bmatrix}$$

que consiste em colocar o vetor  $\mathbf{y}$  (direção simplex) ao lado da solução básica  $\hat{\mathbf{x}}_{\mathbf{B}}$  e fazer as operações elementares necessárias para transformar o vetor  $\mathbf{y}$  na  $\ell$ -ésima coluna da matriz identidade, estendendo as mesmas operações sobre  $\hat{\mathbf{x}}_{\mathbf{B}}$ , isto é, o mesmo esquema prático para atualizar a matriz inversa.

ii) Para o cálculo do vetor multiplicador  $(\lambda')^T = \mathbf{c}_{B'}^T (\mathbf{B}')^{-1}$ , os passos são mais elaborados. Considere a expressão para a matriz  $(\mathbf{B}')^{-1}$  no passo (iv) do Exercício 2.25, e  $\mathbf{c}_{B'}^T = (c_{B_1} \cdots c_{N_k} \cdots c_{B_m})$ . Assim,

$$(\lambda')^T = \mathbf{c}_{B'}^T (\mathbf{B}')^{-1} = (c_{B_1} \cdots c_{N_k} \cdots c_{B_m}) \begin{bmatrix} \mathbf{v}_1 - \frac{y_1}{y_\ell} \mathbf{v}_\ell \\ \vdots \\ \frac{1}{y_\ell} \mathbf{v}_\ell \\ \vdots \\ \mathbf{v}_m - \frac{y_m}{y_\ell} \mathbf{v}_\ell \end{bmatrix} = \sum_{i \neq \ell} c_{B_i} (\mathbf{v}_i - \frac{y_i}{y_\ell} \mathbf{v}_\ell) + \frac{1}{y_\ell} c_{N_k} \mathbf{v}_\ell \pm c_{B_\ell} \mathbf{v}_\ell$$

$$= \sum_{i=1}^m c_{B_i} \mathbf{v}_i + (-\sum_{i=1}^m c_{B_i} y_i + c_{N_k}) \frac{\mathbf{v}_\ell}{y_\ell}$$

Como  $\lambda^T = \mathbf{c}_B^T \mathbf{B}^{-1} = \sum_{i=1}^m c_{B_i} \mathbf{v}_i$  e  $\hat{c}_{N_k} = c_{N_k} - \underbrace{\mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{a}_{N_k}}_{\lambda} = c_{N_k} - \sum_{i=1}^m c_{B_i} y_i$ , segue-se do desenvolvimento anterior que:  $(\lambda')^T = \lambda^T + \frac{\hat{c}_{N_k}}{y_\ell} \mathbf{v}_\ell$ , que está de acordo com (2.43) e (2.47), já que  $\mathbf{v}_\ell = -\mathbf{v}_\ell$ .

Em resumo, do Exercício 2.25 e dos desenvolvimentos acima, segue-se um esquema prático para atualizar a inversa da base, a solução básica e o vetor multiplicador em uma iteração do método simplex ou dual simplex:

$$\left| \begin{array}{c|cc|c} \mathbf{v}_1 & \hat{x}_{B_1} & y_1 & \\ \vdots & \vdots & \vdots & \\ \mathbf{v}_\ell & \hat{x}_{B_\ell} & y_\ell & \\ \vdots & \vdots & \vdots & \\ \mathbf{v}_m & \hat{x}_{B_m} & y_m & \\ \hline \lambda^T & \hat{f} & -\hat{c}_{N_k} & \end{array} \right| \rightarrow \left| \begin{array}{c|cc|c} \mathbf{v}'_1 & \hat{x}_{B'_1} & 0 & \\ \vdots & \vdots & \vdots & \\ \mathbf{v}'_\ell & \hat{x}_{B'_\ell} & 1 & \\ \vdots & \vdots & \vdots & \\ \mathbf{v}'_m & \hat{x}_{B'_m} & 0 & \\ \hline \lambda'^T & f' & 0 & \end{array} \right|$$

O valor da função objetivo para a nova solução básica é calculado por:

$$\underbrace{f(x')}_{f'} = \underbrace{f(\hat{x})}_{\hat{f}} + c_{N_k} \underbrace{\hat{e}}_{\hat{x}_{B_\ell} / y_\ell} \quad \text{ou} \quad f' = \hat{f} + \frac{\hat{c}_{N_k}}{y_\ell} \hat{x}_{B_\ell}, \quad \text{que é o valor obtido na última posição da coluna}$$

que contém as variáveis básicas. Esta forma de atualizar a inversa da matriz básica, solução básica e vetor multiplicador é conhecida como *método simplex revisado*.

Considere a primeira iteração do Exemplo 2.26, em que a matriz básica é a identidade. A inversa da base, a solução básica e o vetor multiplicador simplex para a nova iteração podem ser calculados pelo procedimento anterior:

$$\left| \begin{array}{ccc|cc} 1 & 0 & 0 & 6 & 1 \\ 0 & 1 & 0 & 4 & -1 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 2 \end{array} \right| \xleftarrow{\ell=3} \quad \rightarrow \quad \left| \begin{array}{ccc|cc} 1 & 0 & -1 & 2 & 0 \\ 0 & 1 & 1 & 8 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & -2 & -8 & 0 \end{array} \right|$$

Da tabela à direita, obtemos a inversa da base na iteração 2, a solução básica, o vetor multiplicador e o valor da função objetivo:

$$\mathbf{B}^{-1} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad \hat{\mathbf{x}}_B = \begin{pmatrix} 2 \\ 8 \\ 4 \end{pmatrix} \quad \lambda = \begin{pmatrix} 0 \\ 0 \\ -2 \end{pmatrix} \quad f(\hat{\mathbf{x}}) = -8$$

Determine as iterações seguintes do método simplex para o Exemplo 2.26, fazendo uso das atualizações vistas. Reescreva o algoritmo dual simplex (Seção 2.10.5), considerando as atualizações de  $B^{-1}$ ,  $\hat{\mathbf{x}}_B$  e  $\boldsymbol{\lambda}$ .

**Exercício 2.27** Em um problema de minimização com  $x_i \geq 0$ , para que  $g(\boldsymbol{\lambda})$  seja finito, é necessário que  $c_i - \boldsymbol{\lambda}^T \mathbf{a}_i \geq 0$ . Verifique as exigências sobre  $\boldsymbol{\lambda}$  para as outras condições sobre o sinal das variáveis. Em resumo, para um problema de otimização linear (minimização ou maximização),  $g(\boldsymbol{\lambda})$  é finito se e somente se:

**problema de minimização**

- |                 |  |
|-----------------|--|
| se $x_i \geq 0$ | então $c_i - \boldsymbol{\lambda}^T \mathbf{a}_i \geq 0$ |
| se $x_i \leq 0$ | então $c_i - \boldsymbol{\lambda}^T \mathbf{a}_i \leq 0$ |
| se $x_i$ livre  | então $c_i - \boldsymbol{\lambda}^T \mathbf{a}_i = 0$    |

**problema de maximização**

- |                 |  |
|-----------------|--|
| se $x_i \geq 0$ | então $c_i - \boldsymbol{\lambda}^T \mathbf{a}_i \leq 0$ |
| se $x_i \leq 0$ | então $c_i - \boldsymbol{\lambda}^T \mathbf{a}_i \geq 0$ |
| se $x_i$ livre  | então $c_i - \boldsymbol{\lambda}^T \mathbf{a}_i = 0$    |

**Exercício 2.28** Considere o problema primal:

$$\begin{aligned} \text{Minimizar } f(\mathbf{x}) &= -x_1 - x_2 \\ x_1 - x_2 &\geq 1 \\ -x_1 + x_2 &\geq 1 \\ x_1 &\geq 0, x_2 \geq 0 \end{aligned}$$

Escreva o problema dual e mostre que ambos os problemas, primal e dual, são infactíveis.

**Exercício 2.29** Considere o seguinte problema primal. Calcule seu dual e resolva-o graficamente. O que podemos concluir do primal, sem resolvê-lo?

Primal:

$$\begin{aligned} \text{Minimizar } f(\mathbf{x}) &= x_1 + x_2 \\ -x_1 + x_2 &\geq 1 \\ x_1 - 2x_2 &\geq 1 \\ x_1 &\geq 0, x_2 \geq 0 \end{aligned}$$

**Exercício 2.30** Considere o par de problemas primal-dual no Exemplo 2.43. Suponha que a solução ótima do problema dual seja conhecida:  $\lambda_1 = \frac{5}{4}$ ,  $\lambda_2 = \frac{5}{6}$ . Utilize as folgas complementares (propriedade 2.12) para determinar uma solução ótima primal. Observe que a solução foi unicamente determinada (como também no Exemplo 2.43). Conclua que, se um dos problemas tem solução ótima não degenerada, então o outro tem solução ótima única.

**Exercício 2.31** Considere o problema de corte de peças no Exemplo 2.36. Suponha que a demanda seja  $\mathbf{b} = (108 \ 162)^T$ . Escreva o problema dual, resolva-o graficamente e identifique múltiplas soluções ótimas. Determine a solução ótima primal usando as folgas complementares (propriedade 2.12). O problema dual pode ter múltiplas soluções ótimas e o primal ter uma solução ótima não degenerada? Conclua que, se um dos problemas tem múltiplas soluções ótimas, então o outro tem solução ótima degenerada.

**Exercício 2.32** Represente em um gráfico a região factível do problema primal do exemplo 2.47 e as soluções básicas primais percorridas pelo método dual simplex até a solução ótima. Observe que as soluções básicas primais são infactíveis até que a solução ótima seja obtida.

**Exercício 2.33** Considere o problema de otimização linear em que  $\mathbf{c} \geq \mathbf{0}$ :

$$\text{Minimizar } f(\mathbf{x}) = \mathbf{c}^T \mathbf{x}$$

$$\mathbf{A}\mathbf{x} \geq \mathbf{b}$$

$$\mathbf{x} \geq \mathbf{0}.$$

Escreva o problema dual e verifique que  $\boldsymbol{\lambda} = \mathbf{0}$  é uma solução básica dual factível. (Sugestão: considere a base formada pelas colunas das variáveis de folga.)

# 3

## Otimização discreta

### 3.1 INTRODUÇÃO

A característica que distingue otimização discreta ou combinatória é que algumas das variáveis pertencem a um conjunto discreto, tipicamente, um subconjunto dos números inteiros. Otimização discreta também é conhecida como programação inteira e combinatória, e os termos otimização e programação são usados indistintamente neste livro.

Problemas de otimização discreta aparecem em profusão na vida real e aplicações ocorrem nas mais diversas áreas, tais como energia, transportes, telecomunicações, circuitos eletrônicos, biologia molecular, medicina e saúde, criptografia, aviação e finanças. Uma grande variedade de aplicações é encontrada em engenharia de produção, compreendendo planejamento e programação da produção, projeto de *layout* de sistemas de produção, localização de instalações e distribuição de produtos. Variáveis discretas importantes nessas aplicações envolvem, por exemplo, decidir se um produto é fabricado ou não em um período, escolher a melhor seqüência de itens a serem processados em uma máquina, escolher o melhor local, dentre vários candidatos, a instalar uma nova fábrica ou um centro de distribuição, e determinar as melhores rotas de distribuição de produtos.

Um problema com variáveis inteiras e reais é denominado problema de *programação (linear) inteira mista (PIM)* quando tem a seguinte forma:

$$\begin{aligned} z &= \max \mathbf{c}\mathbf{x} + \mathbf{d}\mathbf{y} \\ (\text{PIM}) \quad \mathbf{A}\mathbf{x} + \mathbf{D}\mathbf{y} &\leq \mathbf{b} \\ \mathbf{x} \in R_+^n, \mathbf{y} \in Z_+^p \end{aligned}$$

em que **A**, uma matriz  $(m \times n)$ , **D**, uma matriz  $(m \times p)$ , **c**, um vetor  $(1 \times n)$ , **d**, um vetor  $(1 \times p)$ , e **b**, um vetor  $(m \times 1)$ , representam os parâmetros do problema. Os vetores de variáveis são **x** e **y** com dimensões  $(n \times 1)$  e  $(p \times 1)$ .  $R_+^n$  representa o espaço dos vetores com  $n$  componentes reais e  $Z_+^p$  representa o espaço dos vetores com  $p$  componentes inteiras não-negativas.

Quando todas as variáveis são inteiras, tem-se um problema de *programação (linear) inteira (PI)*,

$$\begin{aligned} z &= \max \mathbf{c}\mathbf{x} \\ (\text{PI}) \quad \mathbf{A}\mathbf{x} &\leq \mathbf{b} \\ \mathbf{x} \in Z_+^n \end{aligned}$$

e se todas as variáveis assumem valores 0 ou 1, tem-se um problema de *programação 0-1 ou binária (PB)*, escrito como

$$(PB) \quad \begin{aligned} z &= \max \mathbf{c}\mathbf{x} \\ \mathbf{Ax} &\leq \mathbf{b} \\ \mathbf{x} &\in B^n \end{aligned}$$

em que  $B^n$  representa o espaço dos vetores com  $n$  componentes binárias.

Um outro tipo de problema é o problema de *otimização combinatória* (*OC*), que pode ser definido da seguinte forma. É dado um conjunto finito  $N = \{1, \dots, n\}$ , um conjunto de pesos  $c_j$  para cada  $j \in N$ , e uma família  $F$  de subconjuntos factíveis de  $N$ . O problema de otimização combinatória consiste em achar o subconjunto de peso mínimo de  $F$ , isto é

$$\min \left\{ \sum_{j \in S} c_j : S \in F \right\}$$

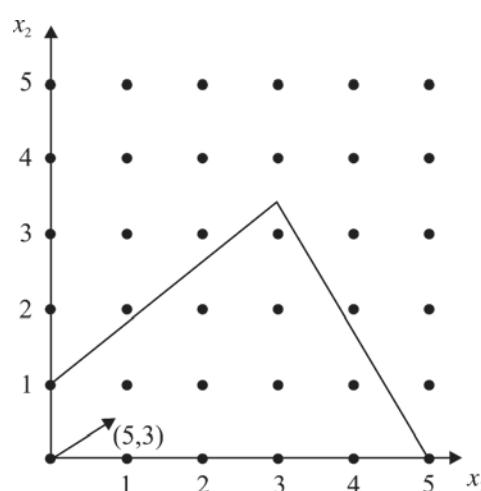
Nas Seções 3.4-3.6, apresentam-se diversos exemplos de problemas de programação inteira e otimização combinatória. De modo geral, um problema de *OC* pode ser formulado como um problema de *PI* ou *PB*. Problemas de programação inteira e combinatória são resolvidos por *métodos ótimos (exatos)*, que fornecem uma solução ótima; por *algoritmos aproximados*, que garantem a distância máxima entre o valor da solução subótima encontrada e o valor ótimo; ou por *métodos heurísticos*, que, em geral, fornecem uma solução subótima, sem conhecimento da qualidade desta em relação a uma solução ótima. Os métodos de resolução são apresentados nas Seções 3.9 e 3.11.

O exemplo a seguir ilustra um problema de programação inteira.

### Exemplo 3.1

$$(PI) \quad \begin{aligned} z &= \max 10x_1 + 6x_2 \\ 9x_1 + 5x_2 &\leq 45 \\ -4x_1 + 5x_2 &\leq 5 \\ \mathbf{x} &\in Z_+^2 \end{aligned}$$

A Figura 3.1 mostra o conjunto de soluções factíveis que satisfazem as restrições do Exemplo 3.1, que é dado por  $X = \{(0, 0), (0, 1), (1, 0), (1, 1), (2, 0), (2, 1), (2, 2), (3, 0), (3, 1), (3, 2), (3, 3)\}$ .



**Figura 3.1** Conjunto de soluções factíveis do Exemplo 3.1.

$(4, 0), (4, 1), (5, 0)\}$ . Ao se mover ao longo da direção do gradiente da função objetivo  $(5, 3)$ , observa-se que a solução ótima é  $\mathbf{x} = (5, 0)$ , com valor  $z = 50$ . ■

O seguinte exemplo ilustra um problema de programação binária.

### Exemplo 3.2

$$(PB) \quad \begin{aligned} z &= \max 2x_1 + 3x_2 \\ 6x_1 + 8x_2 &\leq 10 \\ \mathbf{x} &\in B^2 \end{aligned}$$

O conjunto de soluções factíveis do Exemplo 3.2 é dado por  $X = \{(0, 0), (1, 0), (0, 1)\}$ . Calculando o valor de cada solução, tem-se que a solução ótima é  $\mathbf{x} = (0, 1)$ , com valor  $z = 3$ . ■

A seguinte modificação do Exemplo 3.1 ilustra um problema de programação inteira-mista:

### Exemplo 3.3

$$(PIM) \quad \begin{aligned} z &= \max 10x_1 + 6x_2 \\ 9x_1 + 5x_2 &\leq 45 \\ -4x_1 + 5x_2 &\leq 5 \\ x_1 &\in R_+^1, x_2 \in Z_+^1 \end{aligned}$$

A Figura 3.2 mostra que o conjunto de soluções factíveis consiste em quatro segmentos de reta descritos por:  $X_1 = \{(x_1, 0), 0 \leq x_1 \leq 5\}$ ,  $X_2 = \{(x_1, 1), 0 \leq x_1 \leq 4\frac{4}{9}\}$ ,  $X_3 = \{(x_1, 2), 1\frac{1}{4} \leq x_1 \leq 3\frac{8}{9}\}$  e  $X_4 = \{(x_1, 3), 2\frac{1}{2} \leq x_1 \leq 3\frac{1}{3}\}$ . A solução ótima é  $\mathbf{x} = (3\frac{1}{3}, 3)$ , com valor  $z = 51\frac{1}{3}$ . ■

## 3.2 RELAXAÇÃO LINEAR

Considere agora o problema de *programação linear (PL)* quando a condição de integralidade do Exemplo 3.1 é relaxada, isto é,  $\mathbf{x} \in R_+^2$ .

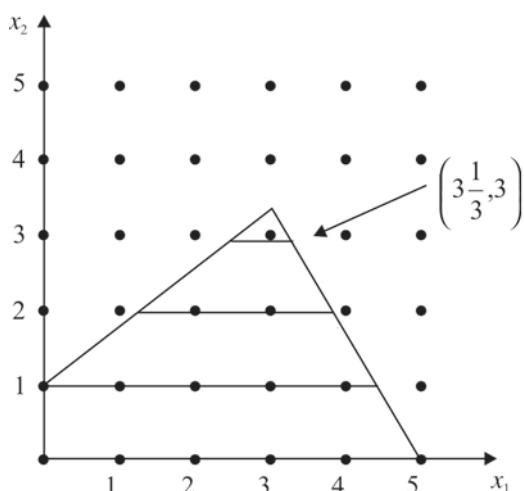


Figura 3.2 Conjunto de soluções factíveis do Exemplo 3.3.

### Exemplo 3.4

$$(PL) \quad \begin{aligned} \bar{z} &= \max 10x_1 + 6x_2 \\ 9x_1 + 5x_2 &\leq 45 \\ -4x_1 + 5x_2 &\leq 5 \\ \mathbf{x} &\in R_+^2 \end{aligned}$$

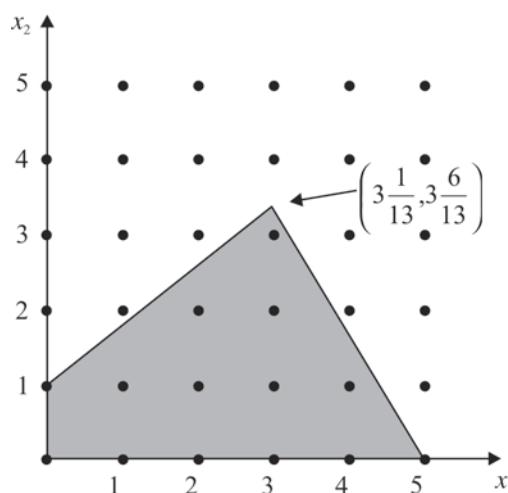
em que  $R_+^2$  denota o vetor com duas variáveis reais não-negativas.

O problema de programação linear  $PL$  é chamado de *relaxação linear* do problema de programação inteira  $PI$ . A Figura 3.3 mostra que a solução ótima de  $PL$  é  $\bar{\mathbf{x}} = \left(3\frac{1}{13}, 3\frac{6}{13}\right)$ , com valor  $\bar{z} = 51\frac{7}{13}$ . Note que  $\bar{\mathbf{x}}$  está bem distante da solução ótima inteira  $\mathbf{x} = (5, 0)$ . Como decorrência, a solução arredondada  $(3, 3)$  também está distante da solução ótima inteira, mostrando que a estratégia de arredondamento da solução ótima do problema relaxado de programação linear pode não ser satisfatória. Em problemas grandes, pode ser difícil achar uma solução arredondada que satisfaça as restrições do problema de programação inteira e, em caso extremo, o conjunto de soluções factíveis de um problema de programação inteira é vazio. ■

A relaxação linear é um conceito-chave para diversos métodos de resolução de problemas  $PI$  e  $PIM$ , pelo seguinte motivo. Sejam  $X_{PI}$ ,  $X_{PIM}$ ,  $X_{PL}$  os conjuntos de soluções factíveis de  $PI$ ,  $PIM$  e  $PL$ , respectivamente. Como  $Z_+^n \subset R_+^n$ , segue-se que  $X_{PI} \subset X_{PL}$  e  $X_{PIM} \subset X_{PL}$ , portanto, o valor  $\bar{z}$  da solução ótima de  $PL$  é um *limite superior* para o valor  $z$  da solução ótima de  $PI$  e  $PIM$ . No Exemplo 3.4,  $\bar{z} = 51\frac{7}{13}$ , e os valores ótimos de  $PI$  e  $PIM$  nos Exemplos 3.1 e 3.3 são, respectivamente, 50 e  $51\frac{1}{3}$ .

A relaxação linear do problema de programação binária do Exemplo 3.2 é dada por:

$$(PL_B) \quad \begin{aligned} \bar{z} &= \max 2x_1 + 3x_2 \\ 6x_1 + 8x_2 &\leq 10 \\ 0 \leq x_i &\leq 1, \quad i = 1, 2 \end{aligned}$$



**Figura 3.3** Conjunto de soluções factíveis do Exemplo 3.4.

Na Seção 3.9.1, mostra-se que a solução ótima de  $PL_B$  é dada por  $\bar{x} = (\frac{1}{3}, 1)$ , com valor  $\bar{z} = 3\frac{2}{3}$ , um limitante superior do valor ótimo de  $PB$  igual a 3.

### 3.3 MODELAGEM COM VARIÁVEIS BINÁRIAS

Um grande número de problemas de otimização discreta envolve a ocorrência ou não de um evento, e a decisão entre duas alternativas. Essa dicotomia é modelada por uma variável binária  $x$  tal que

$$x = \begin{cases} 1 & \text{se o evento ocorre} \\ 0 & \text{se o evento não ocorre} \end{cases}$$

A seguir são apresentadas diversas situações práticas com esse tipo de decisão e a maneira de traduzi-las matematicamente.

#### 3.3.1 Implicações “se – então”

Existem várias situações em que a ocorrência de um evento implica a ocorrência de outro evento. A seguir, são ilustrados exemplos dessas situações.

- *Custo fixo*

Suponha que o custo de produção  $K$  de um item consiste em um custo fixo  $s$  adicionado de um custo linear variável com taxa  $c$ , similarmente ao modelo (1.1)–(1.5) do Capítulo 1. Defina a variável

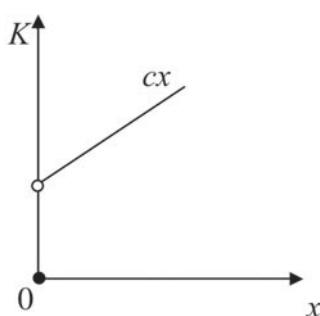
$x$  = quantidade produzida do item

Então, o custo de produção  $K(x)$  é dado por

$$K(x) = \begin{cases} s + cx & \text{se } x > 0 \\ 0 & \text{se } x = 0 \end{cases}$$

A Figura 3.4 ilustra a função  $K$ , que é não linear devido à descontinuidade no ponto  $x = 0$ . Para representar essa função de forma linear com uma variável inteira, seja  $M$  um limitante superior da produção do item, e considere a variável binária  $y$  tal que  $y = 0$  implica  $x = 0$  ou, equivalentemente,  $x > 0$  implica  $y = 1$ . Então, o custo  $K$  pode ser reescrito como

$$\begin{aligned} K &= sy + cx \\ x &\leq My \end{aligned}$$



**Figura 3.4** Função com custo fixo.

Note que, nesta forma,  $x = 0$  não implica  $y = 0$ , e que o ponto  $w_1 = (x = 0, y = 1)$  é factível com custo  $K(w_1) = s$ . Em geral, deseja-se minimizar o custo  $K$ , portanto, o ponto  $w_1$  não é um mínimo, pois a solução  $w_2 = (x = 0, y = 0)$  é factível com custo  $K(w_2) = 0$ .

- *Produção de itens*

Considere a situação em que, se o produto 1 é fabricado, então o produto 2 também deve ser fabricado. Seja

$$x_1 = \text{quantidade produzida do item 1}$$

$$x_2 = \text{quantidade produzida do item 2}$$

Seja  $y$  uma variável binária tal que

$$y = \begin{cases} 1 & \text{se } x_1 > 0 \\ 0 & \text{caso contrário} \end{cases}$$

Essa condição é expressa pela desigualdade

$$x_1 \leq My$$

em que  $M$  é um limitante superior da produção do item 1.

Deseja-se expressar a condição de que  $y = 1$  implica  $x_2 > 0$ . Isso é obtido por meio da desigualdade

$$x_2 \geq my$$

tal que  $m$  é um limitante inferior da quantidade a ser produzida do item 2.

- *Desigualdades de funções*

Em diversas situações é necessário impor que a desigualdade  $f(x_1, x_2, \dots, x_n) > 0$  implica que a desigualdade  $g(x_1, x_2, \dots, x_n) \geq 0$  é verdadeira. Considere uma variável binária  $y$ , e seja  $M$  um número grande, tal que  $f \leq M$  e  $-g \leq M$  para todo valor de  $x_1, x_2, \dots, x_n$ . A implicação é expressa pelas seguintes restrições:

$$-g(x_1, x_2, \dots, x_n) \leq My \quad (3.1)$$

$$f(x_1, x_2, \dots, x_n) \leq M(1-y) \quad (3.2)$$

Note que, se  $f > 0$ , então (3.2) implica  $y = 0$ , e de (3.1) segue-se que  $g \geq 0$ .

**Exemplo 3.5** Suponha que quatro itens podem ser produzidos em uma máquina denotada por  $k$ , e se o item 1 é produzido em  $k$ , então os outros itens, 2, 3 e 4, não podem ser processados em  $k$  e são processados em outras máquinas.

Defina as variáveis

$$x_{ik} = \begin{cases} 1 & \text{se o item } i \text{ é processado na máquina } k \\ 0 & \text{caso contrário} \end{cases}$$

Então  $x_{1k} = 1$  implica  $x_{2k} = x_{3k} = x_{4k} = 0$ . Como as variáveis são binárias, pode-se expressar essa condição da seguinte forma: se  $x_{1k} > 0$ , então  $x_{2k} + x_{3k} + x_{4k} \leq 0$ , ou  $-x_{2k} - x_{3k} - x_{4k} \geq 0$ . Utilizando-se as expressões (3.1) e (3.2), tem-se que

$$x_{2k} + x_{3k} + x_{4k} \leq 3y$$

$$x_{1k} \leq 3(1-y)$$

Note que  $M = 3$ , pois este é o valor máximo assumido pela soma  $x_{2k} + x_{3k} + x_{4k}$ . ■

### 3.3.2 Restrição ativada ou desativada

Seja a desigualdade

$$f(x_1, x_2, \dots, x_n) \leq 0$$

Defina uma variável binária  $y$  tal que  $y = 1$  implica que a desigualdade anterior é satisfeita ou está ativada. Isso pode ser expresso como

$$f(x_1, x_2, \dots, x_n) \leq M(1-y)$$

em que  $M$  é um número grande. Note que, se  $y = 0$ , então a restrição é desativada, isto é,  $f(x_1, x_2, \dots, x_n)$  pode assumir qualquer valor até seu limite superior  $M$ .

### 3.3.3 Restrições disjuntivas

Considere as desigualdades

$$f(x_1, x_2, \dots, x_n) \leq 0 \quad (3.3)$$

e

$$g(x_1, x_2, \dots, x_n) \leq 0 \quad (3.4)$$

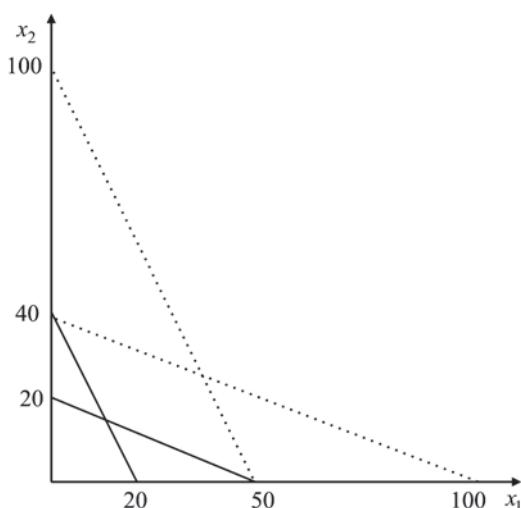
Deseja-se que somente uma das desigualdades esteja ativada. Defina uma variável binária  $y$  tal que, se  $y = 1$ , então somente (3.3) é ativada, e se  $y = 0$ , somente (3.4) é ativada. Isso é representado matematicamente por

$$f(x_1, x_2, \dots, x_n) \leq M(1-y)$$

e

$$g(x_1, x_2, \dots, x_n) \leq My$$

**Exemplo 3.6** Considere as restrições  $4x_1 + 2x_2 \leq 80$  e  $2x_1 + 5x_2 \leq 100$ ,  $x_1, x_2 \geq 0$ , ilustradas na Figura 3.5 em linha cheia. O acréscimo de um número grande  $M$  no lado direito das restrições faz com que a restrição desativada seja transladada paralelamente no quadrante superior direito do plano. As linhas pontilhadas indicam a translação mínima das duas retas de forma que somente uma das restrições seja ativada. Assim, a reta  $4x_1 + 2x_2 = 80$  é transladada até  $4x_1 + 2x_2 = 200$ , e



**Figura 3.5** Exemplo de restrições disjuntivas.

a reta  $2x_1 + 5x_2 = 100$  é transladada até  $2x_1 + 5x_2 = 200$ , portanto, o menor valor de  $M$  é dado por  $\max\{120, 100\} = 120$ . Um valor pequeno de  $M$  é desejável para acelerar os métodos de resolução de problemas de programação inteira, como mostrado na Seção 3.8. A representação matemática da disjunção é, então, dada por

$$\begin{aligned} 4x_1 + 2x_2 &\leq 80 + 120(1 - y) \\ 2x_1 + 5x_2 &\leq 100 + 120y \end{aligned}$$

■

Exemplos de restrições ativadas ou desativadas e restrições disjuntivas ocorrem em problemas de programação de tarefas em máquinas, que são apresentados na Seção 3.6.3.

### 3.3.4 Representação de função linear por partes

Funções lineares por partes, discutidas brevemente na Seção 2.2.10, são importantes porque podem representar, por exemplo, o custo de expansão da capacidade ou aproximar uma função não-linear. Considere a função  $f(x) = f_1(x) + f_2(x) + f_3(x)$ , ilustrada na Figura 3.6. As funções  $f_1(x), f_2(x), f_3(x)$  são lineares e especificadas pelos pontos  $(0, 0), (2, 10), (6, 14)$  e  $(10, 22)$ , respectivamente. Qualquer  $x$ , tal que  $0 \leq x \leq 10$ , pode ser escrito como

$$x = 0\lambda_1 + 2\lambda_2 + 6\lambda_3 + 10\lambda_4, \quad \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1, \quad \lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0$$

Então,

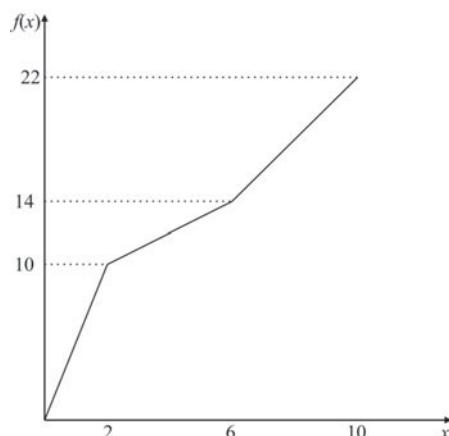
$$f(x) = 0\lambda_1 + 10\lambda_2 + 14\lambda_3 + 22\lambda_4, \quad \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1, \quad \lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0$$

se no máximo dois valores  $\lambda_i$  e  $\lambda_{i+1}$ ,  $i = 1, 2, 3$  forem positivos. Essa condição pode ser modelada por meio de variáveis binárias  $y_i$ , tal que  $y_i = 1$ , se  $a_i \leq x \leq a_{i+1}$ ,  $y_i = 0$ , caso contrário, em que  $a_1 = 0, a_2 = 2, a_3 = 6, a_4 = 10$ , e as restrições

$$\begin{aligned} \lambda_i \leq y_i, \quad \lambda_2 \leq y_1 + y_2, \quad \lambda_3 \leq y_2 + y_3, \quad \lambda_4 \leq y_3 \\ y_1 + y_2 + y_3 = 1 \end{aligned}$$

### 3.3.5 Relações lógicas

Variáveis binárias são usadas para relações lógicas. Algumas dessas relações são ilustradas a seguir.



**Figura 3.6** Função linear por partes.

Suponha que existam cinco tipos de investimento financeiro, e seja  $x_j$  a variável binária de decisão tal que

$$x_j = \begin{cases} 1 & \text{se o investimento } j \text{ é selecionado} \\ 0 & \text{caso contrário} \end{cases}$$

Considere agora as seguintes situações e as restrições representativas.

- No máximo três investimentos são selecionados.

$$x_1 + x_2 + x_3 + x_4 + x_5 \leq 3$$

- Exatamente um investimento é selecionado.

$$x_1 + x_2 + x_3 + x_4 + x_5 = 1$$

- O investimento 1 ou o investimento 2 é selecionado.

$$x_1 + x_2 \geq 1$$

- Se o investimento 2 é selecionado, então o investimento 1 também é selecionado.

$$x_2 \leq x_1$$

- Se os investimentos 2, 3 ou 4 são selecionados, então o investimento 1 é selecionado.

A primeira forma de representar essa condição é

$$x_2 + x_3 + x_4 \leq 3x_1 \quad (3.5)$$

e a segunda forma é

$$\begin{aligned} x_2 &\leq x_1 \\ x_3 &\leq x_1 \\ x_4 &\leq x_1 \end{aligned} \quad (3.6)$$

Ambas as representações são equivalentes do ponto de vista de programação inteira, mas não são equivalentes sob o ponto de vista de programação linear, em que as variáveis são reais. Considere os conjuntos:

$$X_B = \text{soluções binárias que satisfazem (3.5) ou (3.6)}$$

$$X_1 = \{x_2 + x_3 + x_4 \leq 3x_1, \quad 0 \leq x_i \leq 1, i = 1, 2, 3, 4\}$$

$$X_2 = \{x_2 \leq x_1, \quad x_3 \leq x_1, \quad x_4 \leq x_1, \quad 0 \leq x_i \leq 1, i = 1, 2, 3, 4\}$$

Note que  $X_B \subset X_2 \subset X_1$ . Por exemplo, os valores  $x_2 = 1/2, x_3 = x_4 = 1/4, x_1 = 1/3$  satisfazem (3.5), mas não satisfazem (3.6). O método mais usado para resolver problemas de programação inteira, denominado *branch-and-bound* e discutido na Seção 3.9.1, faz uso da programação linear para achar a solução ótima inteira. Quanto menor o conjunto de soluções factíveis da programação linear, menor é o esforço computacional para achar a solução ótima inteira. Portanto, o conjunto  $X_2$  é preferido em relação a  $X_1$ .

### 3.3.6 Representação de valores discretos

Considere um problema em que uma variável  $x$  só pode assumir valores do conjunto discreto  $\{4, 6, 8, 12, 20, 24\}$ . Para representar essa condição, defina as variáveis binárias  $y_i, i = 1, \dots, 6$  e as restrições

$$x = 4y_1 + 6y_2 + 8y_3 + 12y_4 + 20y_5 + 24y_6$$

$$y_1 + y_2 + y_3 + y_4 + y_5 + y_6 = 1$$

### 3.4 FORMULAÇÕES DE PROBLEMAS CLÁSSICOS

Linguagens algébricas, criadas a partir da década de 1980, tiveram um grande impacto prático para a resolução de problemas de otimização. Essas linguagens de alto nível permitem que o usuário escreva modelos genéricos de programação linear e não-linear em um formato parecido com o da notação algébrica. Além disso, o modelo é separado dos dados (armazenados em arquivos separados), e independente do tamanho do problema. Desta forma, essas linguagens concentram a atenção do usuário na modelagem, reduzem as fontes de erros, aceleram o desenvolvimento do modelo e facilitam a compreensão e possíveis modificações no futuro.

Existem diversas linguagens populares, tais como GAMS, AMPL, MPL, AIMMS e, ainda, as linguagens OPL, MOSEL e LINGO, desenvolvidas pelas empresas que produzem os pacotes de otimização CPLEX, XPRESS e LINDO. De modo geral, existem versões estudantis gratuitas que o leitor pode usar para resolver os modelos apresentados neste livro. Outra ferramenta computacional muito usada é a planilha Excel, que contém um *solver* para resolução de problemas de dimensão relativamente pequena.<sup>1</sup> O Excel, no entanto, não gera um modelo genérico como as linguagens algébricas.

A seguir, apresentamos formulações de diversos problemas clássicos e importantes de programação inteira.

#### 3.4.1 Problemas da mochila

Problemas da mochila envolvem a escolha de itens a serem colocados em uma ou mais mochilas de forma a maximizar uma função de utilidade.

##### *Mochila 0-1*

Um exemplo de problema da mochila 0-1 é a seleção de projetos. Considere  $n$  projetos e um capital  $b$  para investimento. O projeto  $j$  tem um custo  $a_j$  e um retorno esperado  $p_j$ . O problema consiste em selecionar os projetos que maximizam o retorno total esperado sem ultrapassar o limite de capital.

Defina as variáveis

$$x_j = \begin{cases} 1 & \text{se o projeto } j \text{ é selecionado} \\ 0 & \text{caso contrário} \end{cases}$$

O problema é, então, formulado como:

$$\max \sum_{j=1}^n p_j x_j \quad (M1)$$

$$\sum_{j=1}^n a_j x_j \leq b \quad (M2)$$

$$\mathbf{x} \in B^n \quad (M3)$$

A função objetivo (M1) expressa a maximização do retorno esperado, e a restrição (M2) indica que o capital  $b$  não pode ser excedido. A restrição (M3) representa o tipo das variáveis.

Este problema é denominado problema da mochila devido à analogia do problema que envolve a decisão de quais itens carregar em uma mochila sem exceder um dado limite de peso, e apare-

<sup>1</sup> Para exemplos de aplicação de planilha eletrônica na solução de problemas de pesquisa operacional, veja, por exemplo, Lachtermacher (2004).

ce como subproblema em diversos problemas de otimização combinatória. Quando existem outras restrições, como, por exemplo, o volume da mochila, tem-se um problema da mochila multidimensional.

**Exemplo 3.7** Considere um capital para investimento  $b = 100$ ,  $n = 8$  projetos, e os seguintes vetores de parâmetros:

$$\mathbf{p} = [p_j] = [41 \ 33 \ 14 \ 25 \ 32 \ 32 \ 9 \ 19]$$

$$\mathbf{a} = [a_j] = [47 \ 40 \ 17 \ 27 \ 34 \ 23 \ 5 \ 44]$$

A solução ótima é dada por  $x_2 = x_4 = x_6 = x_7 = 1$ , com valor 99. Esta solução utiliza  $40 + 27 + 23 + 5 = 95$  unidades do capital. ■

### Mochila inteira

Neste problema,<sup>2</sup> a variável de decisão  $x_j \geq 0$ ,  $j = 1, \dots, n$ , pode assumir um valor inteiro não-negativo, isto é, pode-se levar diversas unidades do mesmo item na mochila. A formulação é idêntica à da mochila 0-1, ao substituir-se  $\mathbf{x} \in B^n$  por  $\mathbf{x} \in Z_+^n$ .

### Múltiplas mochilas

Considere  $n$  itens que devem ser colocados em  $m$  mochilas de capacidades distintas  $b_i$ ,  $i = 1, \dots, m$ . Cada item  $j$  tem uma lucratividade  $p_j$  e um peso  $w_j$ , e o problema consiste em selecionar  $m$  subconjuntos distintos de itens, tal que cada subconjunto ocupe uma capacidade de no máximo  $b_i$  e o lucro total seja maximizado.<sup>3</sup> Este problema ocorre em situações de carregamento de contêineres<sup>4</sup> e em situações de corte nas indústrias de papel e aço.<sup>5</sup>

Defina as variáveis

$$x_{ij} = \begin{cases} 1 & \text{se o item } j \text{ é colocado na mochila } i \\ 0 & \text{caso contrário} \end{cases}$$

O problema é formulado como:

$$\max \sum_{i=1}^m \sum_{j=1}^n p_j x_{ij} \quad (\text{M4})$$

$$\sum_{j=1}^n w_j x_{ij} \leq b_i, \quad i = 1, \dots, m \quad (\text{M5})$$

$$\sum_{i=1}^m x_{ij} \leq 1, \quad j = 1, \dots, n \quad (\text{M6})$$

$$\mathbf{x} \in B^{mn} \quad (\text{M7})$$

A função objetivo (M4) representa a maximização do lucro total. As restrições (M5) garantem que a capacidade  $b_i$  da mochila  $i$  não pode ser excedida. As restrições (M6) indicam que se o item  $j$  é escolhido, então ele é colocado em uma única mochila. A restrição (M7) representa o tipo das variáveis.

<sup>2</sup> Um método de solução para este problema é proposto em Yanasse e Soma (1987) e Yanasse *et al.* (2000).

<sup>3</sup> Ver, por exemplo, Ferreira *et al.* (1996).

<sup>4</sup> Eilon e Christofides (1971).

<sup>5</sup> Sinuany-Stern e Winer (1994).

Uma generalização deste problema envolve a existência de  $K$  tipos de peso  $w_{jk}, k = 1, \dots, K$ , e cada mochila  $i$  com  $K$  tipos de capacidade  $b_{ik}, k = 1, \dots, K$ , de modo que a restrição (M5) assume a forma  $\sum_{j=1}^n w_{ij}x_{ij} \leq b_{ik}, \quad i = 1, \dots, m, \quad k = 1, \dots, K$ . Essa generalização modela a situação de carregar  $m$  contêineres (veículos) com um subconjunto de  $n$  itens, tal que existem limitantes não somente no peso total, mas também em outras características de um item, tais como volume e valor.<sup>6</sup>

### *Empacotamento em mochilas*

No problema anterior, o número de mochilas é dado e alguns itens podem não ser colocados nas mochilas. No problema de empacotamento em mochilas, conhecido como *bin packing*, deseja-se determinar o número mínimo de mochilas de mesma capacidade  $b$  que empacotem  $n$  itens de peso  $w_j, j = 1, \dots, n$ . Este é um dos diversos problemas de empacotamento, que envolvem também o arranjo de objetos em dispositivos bidimensionais e tridimensionais,<sup>7</sup> tais como nos problemas de carregamentos de produtos sobre paletes ou dentro de contêineres ou caminhões.

Defina as variáveis

$$y_i = \begin{cases} 1 & \text{se a mochila } i \text{ é usada} \\ 0 & \text{caso contrário} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{se o item } j \text{ é colocado na mochila } i \\ 0 & \text{caso contrário} \end{cases}$$

Tem-se, então, a seguinte formulação:

$$\min \sum_{i=1}^n y_i \tag{M8}$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n \tag{M9}$$

$$\sum_{j=1}^n w_j x_{ij} \leq b y_i, \quad i = 1, \dots, n \tag{M10}$$

$$\mathbf{x} \in B^{nn}, \mathbf{y} \in B^n \tag{M11}$$

A função objetivo (M8) minimiza o número de mochilas. Note que o limite superior do número de mochilas é igual ao número de itens  $n$ . As restrições (M9) asseguram que cada item  $j$  é colocado em uma única mochila, e as restrições (M10) impõem que se a mochila  $i$  é usada, então sua capacidade  $b$  não pode ser excedida. A restrição (M11) indica o tipo das variáveis.

### **3.4.2 Problemas de corte**

Em diversos processos industriais, itens são produzidos a partir do corte de peças maiores que estão disponíveis em estoque ou são fabricadas na própria indústria ou, ainda, compradas de terceiros. Essas peças podem ter apenas uma dimensão relevante para o problema de corte (unidimensional), por exemplo, o comprimento, tais como barras de aço, bobinas de papel, rolos de filme; ou duas dimensões, como o comprimento e a largura, por exemplo, placas de madeira, tecido, chapas de aço; ou, ainda, três dimensões, tais como blocos de matéria-prima para colchões e

<sup>6</sup> Uma heurística para este problema é proposta por Ahuja e Cunha (2005).

<sup>7</sup> Problemas de carregamento tridimensionais são complexos e usualmente resolvidos por heurísticas; ver Silva *et al.* (2003), Lins *et al.* (2002) e Kohayakawa *et al.* (2004).

travesseiros. Apresenta-se a seguir o problema de corte unidimensional, discutido brevemente na Seção 2.2.7, cuja modelagem pode ser estendida para problemas bidimensionais e tridimensionais.

### Corte unidimensional

Este problema consiste em cortar barras disponíveis de tamanho único  $L$ , para a produção de  $m$  tipos de itens (barras de tamanhos menores) com tamanhos  $l_1, l_2, \dots, l_m$ , e demandas,  $b_1, b_2, \dots, b_m$ , respectivamente. O objetivo é minimizar o número de barras usadas, dado um limite superior de barras disponíveis  $n$ . A primeira formulação deste problema é atribuída a Kantorovich (1939), e reproduzida a seguir.

Defina as variáveis

$$y_i = \begin{cases} 1 & \text{se a barra } i \text{ é usada} \\ 0 & \text{caso contrário} \end{cases}$$

$x_{ij}$  = número de vezes que o item  $j$  é cortado na barra  $i$ .

Tem-se, então, o seguinte modelo:

$$\min \sum_{i=1}^n y_i \quad (C1)$$

$$\sum_{i=1}^n x_{ij} \geq b_j, \quad j = 1, \dots, m \quad (C2)$$

$$\sum_{j=1}^m l_j x_{ij} \leq L y_i, \quad i = 1, \dots, n \quad (C3)$$

$$y \in B^n, x \in Z_+^{mn} \quad (C4)$$

A função objetivo (C1) representa a minimização do número de barras. As restrições (C2) asseguram que cada demanda  $b_j$  é atendida e as restrições (C3) garantem que se a barra  $i$  é usada, então o comprimento total dos itens é limitado pelo comprimento  $L$  da barra. A restrição (C4) indica o tipo das variáveis.

O problema de empacotamento em mochilas e o problema de corte unidimensional são similares. Nos dois problemas, deseja-se minimizar o número de objetos com um dado tamanho e, se o objeto  $i$  é usado, os itens colocados no objeto ou cortados do objeto não podem exceder esse tamanho. O corte de itens a partir de objetos pode ser visto como empacotar itens dentro de objetos, por isso, problemas de corte e empacotamento têm sido estudados em conjunto na literatura.<sup>8</sup>

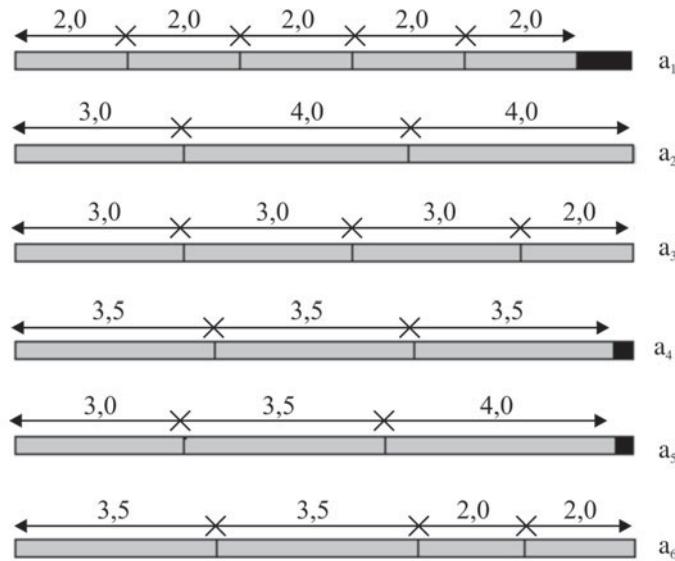
O modelo anterior tem várias restrições da mochila, uma para cada barra  $i$ . Como o objetivo é minimizar o número de barras a serem cortadas, basta considerar o conjunto de soluções inteiras que satisfaz a restrição da mochila de forma máxima, isto é, a capacidade remanescente da mochila é zero ou uma fração de  $\min_{j=1}^m l_j$ . Essas soluções inteiras são chamadas padrões de corte ou de empacotamento.

Para  $L = 11$ ,  $m = 4$ ,  $l_1 = 2$ ,  $l_2 = 3$ ,  $l_3 = 3,5$ ,  $l_4 = 4$ , a Figura 3.7 ilustra alguns padrões de corte. As bordas em negrito correspondem a sobras da barra.

Seja  $n$  o número total de possíveis padrões de corte, e para cada padrão de corte  $j$ ,  $j = 1, 2, \dots, n$ , associe um vetor  $m$ -dimensional  $\mathbf{a}_j = (a_{1j}, \dots, a_{mj})$ , tal que

$a_{ij}$  = número de barras do tipo  $i$  no padrão  $j$

<sup>8</sup> Dyckhoff *et al.* (1997) e Waescher *et al.* (2006).



**Figura 3.7** Exemplos de padrões de corte.

Para o exemplo da Figura 3.7, tem-se:

$$\mathbf{a}_1 = \begin{bmatrix} 5 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{a}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 2 \end{bmatrix} \quad \mathbf{a}_3 = \begin{bmatrix} 1 \\ 3 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{a}_4 = \begin{bmatrix} 0 \\ 0 \\ 3 \\ 0 \end{bmatrix} \quad \mathbf{a}_5 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{a}_6 = \begin{bmatrix} 2 \\ 0 \\ 2 \\ 0 \end{bmatrix}$$

Seja  $\mathbf{b}$  = o vetor de demandas, e defina a variável

$x_j$  = número de barras cortadas conforme o padrão de corte  $j$

O modelo matemático a seguir foi proposto por Gilmore e Gomory (1961, 1963) e é uma alternativa ao modelo de Kantorovich (ver Seção 2.2.7):

$$\min \sum_{j=1}^n x_j \quad (C5)$$

$$\sum_{j=1}^n \mathbf{a}_j x_j = \mathbf{b} \quad (C6)$$

$$\mathbf{x} \in \mathbb{Z}_+^n \quad (C7)$$

A função objetivo (C5) minimiza o número de barras cortadas de acordo com os  $n$  padrões. As restrições (C6) garantem o atendimento das demandas, enquanto (C7) indica o tipo das variáveis. Este modelo é muito utilizado na prática, com resolução por meio de programação linear e técnicas de arredondamento de variáveis.

### Corte bidimensional

A seguir apresentamos um exemplo de um problema de geração de padrão de corte bidimensional. Este problema envolve apenas uma placa que deve ser cortada em um número de peças retangulares de modo a otimizar algum objetivo, por exemplo, maximizar a utilização da placa. Considere uma placa retangular ( $L, W$ ) de comprimento  $L$  e largura  $W$  a ser cortada em  $m$  tipos de

peças retangulares menores; a peça do tipo  $i$  tem tamanho  $(l_i, w_i)$  e valor  $v_i$ . Sejam  $P_i$  e  $Q_i$  o número mínimo e máximo de peças do tipo  $i$  que podem ser cortadas da placa,  $0 \leq P_i \leq Q_i$ ,  $i = 1, \dots, m$ . Admita que

- $L, W, l_i, w_i, i = 1, \dots, m$  são inteiros.
- Os cortes são ortogonais a um lado da placa.
- A orientação das peças é fixa, isto é, uma peça de comprimento  $l_i$  e largura  $w_i$  é distinta de uma peça de comprimento  $l_i$  e largura  $w_i$ ,  $l_i \neq w_i$ .

Defina

$$a_{ilws} = \begin{cases} 1 & \text{se a peça do tipo } i, \text{ quando cortada com vértice inferior à esquerda com} \\ & \text{coordenadas } (l, w) \text{ exclui o ponto } (r, s) \text{ (veja Figura 3.8)} \\ 0 & \text{caso contrário} \end{cases}$$

Para impedir contagem dupla quando duas peças são adjacentes, defina

$$a_{ilws} = \begin{cases} 1 & \text{se } 0 \leq l \leq r \leq l + l_i - 1 \leq L - 1 \text{ e } 0 \leq w \leq s \leq w + w_i - 1 \leq W - 1; \\ 0 & \text{caso contrário.} \end{cases}$$

Seja  $X = \{0, 1, 2, \dots, L - 1\}$  o conjunto de comprimentos para localização do vértice inferior à esquerda do corte das peças e  $Y = \{0, 1, 2, \dots, W - 1\}$  o conjunto correspondente para as larguras.

Defina as variáveis

$$x_{ilw} = \begin{cases} 1 & \text{se a peça do tipo } i \text{ é cortada com seu vértice inferior à esquerda no ponto } (l, w), \\ & \text{tal que } 0 \leq l \leq L - l_i \text{ e } 0 \leq w \leq W - w_i; \\ 0 & \text{caso contrário.} \end{cases}$$

O problema de corte bidimensional não guilhotinado pode ser formulado da seguinte maneira:<sup>9</sup>

$$\max \sum_{i=1}^m \sum_{l \in X} \sum_{w \in Y} v_i x_{ilw} \quad (C8)$$

$$\sum_{i=1}^m \sum_{l \in X} \sum_{w \in Y} a_{ilws} x_{ilw} \leq 1, \quad r \in X, \quad s \in Y \quad (C9)$$

$$P_i \leq \sum_{l \in X} \sum_{w \in Y} x_{ipq} \leq Q_i, \quad i = 1, \dots, m \quad (C10)$$

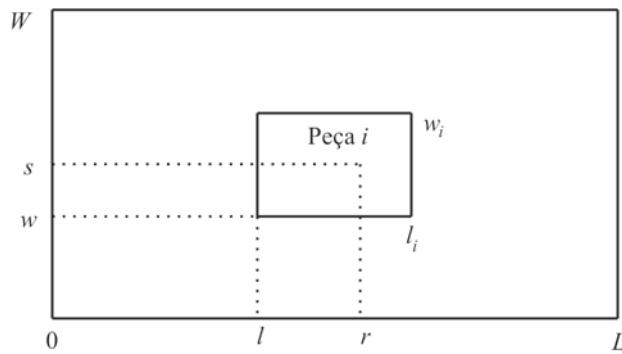
$$x \in B^{mlX \times lY} \quad (C11)$$

A função objetivo (C8) representa a maximização da utilidade total das peças cortadas. As restrições (C9) garantem que qualquer ponto  $(r, s)$  é excluído no máximo por uma peça (restrições de sobreposição de peças), e as restrições (C10) forçam que o número de peças de cada tipo  $i$  esteja situado na faixa requerida. A restrição (C11) indica o tipo das variáveis.

O problema de carregamento de paletes com caixas iguais pode ser visto como um caso particular do modelo (C8)–(C11) em que temos apenas dois tipos de peças, com tamanhos  $(l, w)$  e  $(w, l)$ , correspondendo às duas maneiras de arranjar a face inferior das caixas sobre o palete.<sup>10</sup> Outro exemplo de problema de corte bidimensional, que aparece em diversos processos industriais, envolve os chamados padrões guilhotinados, em que os cortes são realizados conforme uma

<sup>9</sup> Os conjuntos  $X$  e  $Y$  podem ser reduzidos, sem perda de generalidade, redefinindo-os como as combinações lineares inteiras dos comprimentos  $l_1, l_2, \dots, l_m$  e larguras  $w_1, w_2, \dots, w_m$ , respectivamente; veja Beasley (1985).

<sup>10</sup> Ver, por exemplo, Alvarez-Valdes *et al.* (2005) e Birgin *et al.* (2005).



**Figura 3.8** Localização de um corte de peça.

guilhotina de papel.<sup>11</sup> Devido à alta complexidade dos problemas de corte e empacotamento, estes são, em geral, resolvidos por métodos heurísticos ou algoritmos aproximados.<sup>12</sup>

### 3.4.3 Problemas de designação

Este tipo de problema envolve a designação (ou atribuição) de tarefas a agentes, que são termos genéricos com significado específico em cada problema prático.

#### Designação

Este é o problema mais simples da categoria e envolve  $n$  tarefas e  $n$  agentes, tal que cada tarefa é executada por um único agente e cada agente executa uma única tarefa. A execução da tarefa  $j$  pelo agente  $i$  tem um custo  $c_{ij}$ . O problema, então, consiste em designar tarefas a agentes de modo a minimizar o custo total.

Este problema ocorre em diversas aplicações de forma isolada ou como subproblema de problemas mais complexos. Como exemplos, pode-se citar: designação de frações de um lote de um item para processamento em máquinas,<sup>13</sup> designação de centros de distribuição de jornais aos centros de impressão de jornais,<sup>14</sup> e designação de clientes em roteamento de veículos.<sup>15</sup>

Defina as variáveis

$$x_{ij} = \begin{cases} 1 & \text{se a tarefa } j \text{ é designada ao agente } i \\ 0 & \text{caso contrário} \end{cases}$$

O modelo que representa o problema é dado por (ver Seção 2.2.2):

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (D1)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n \quad (D2)$$

11 Ver, por exemplo, Oliveira e Ferreira (1990), Yanasse *et al.* (1991), Carnieri *et al.* (1994), Christofides e Hadjiconstantinou (1995), Morabito e Arenales (2000) e Parada *et al.* (2000).

12 Dyckhoff e Finke (1992) e Waescher *et al.* (2006); veja também algoritmos aproximados em Miyazawa e Wakabayashi (2003).

13 LeBlanc *et al.* (1999).

14 Ree e Yoon (1996).

15 Fisher e Jaikumar (1981).

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \quad (D3)$$

$$\mathbf{x} \in B^{mn} \quad (D4)$$

A função objetivo (D1) minimiza o custo total de designação de tarefas a agentes. As restrições (D2) e (D3) asseguram que cada tarefa  $j$  é designada a um único agente, e cada agente  $i$  executa exatamente uma tarefa. A restrição (D4) indica o tipo das variáveis.

Como mencionado na Seção 2.2.2, pode-se demonstrar que a solução ótima do problema de programação linear em que as variáveis  $\mathbf{x} \in R_+^{mn}$  também é a solução ótima do problema anterior, isto é, a solução ótima do problema relaxado de programação linear é inteira. Este problema é discutido na Seção 4.3.1.

#### *Designação generalizada*

Neste problema, tem-se  $m$  agentes e  $n$  tarefas, com  $m < n$ ; cada tarefa deve ser executada por um único agente, e um agente pode executar mais de uma tarefa. A execução da tarefa  $j$  pelo agente  $i$  requer uma quantidade  $a_{ij}$  de recurso do agente  $i$ , com custo  $c_{ij}$ . O agente  $i$  tem capacidade de recurso  $b_i$ . Ao se utilizarem as variáveis binárias do problema anterior, obtém-se o seguinte modelo:

$$\min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (D5)$$

$$\sum_{i=1}^m x_{ij} = 1, \quad j = 1, \dots, n \quad (D6)$$

$$\sum_{j=1}^n a_{ij} x_{ij} \leq b_i, \quad i = 1, \dots, m \quad (D7)$$

$$\mathbf{x} \in B^{mn} \quad (D8)$$

A função objetivo (D5) minimiza o custo total de designação de tarefas a agentes. As restrições (D6) garantem que cada tarefa  $j$  é executada por um único agente, e as restrições (D7) impõem que a capacidade  $b_i$  de cada agente  $i$  não é excedida. A restrição (D8) indica o tipo das variáveis. Ao se associarem agentes com mochilas e itens com tarefas, nota-se a semelhança do problema de designação generalizada com o problema de empacotamento em mochilas. No primeiro problema, o número de mochilas é dado, enquanto no segundo este número deve ser minimizado.

**Exemplo 3.8** Considere  $m = 3$  agentes,  $n = 8$  tarefas e os seguintes parâmetros:

$$\mathbf{C} = [c_{ij}] = \begin{bmatrix} 15 & 61 & 3 & 94 & 86 & 68 & 69 & 51 \\ 21 & 28 & 76 & 48 & 54 & 85 & 39 & 72 \\ 21 & 21 & 46 & 43 & 21 & 3 & 84 & 44 \end{bmatrix}$$

$$\mathbf{A} = [a_{ij}] = \begin{bmatrix} 31 & 69 & 14 & 87 & 51 & 65 & 35 & 54 \\ 23 & 20 & 71 & 86 & 91 & 57 & 30 & 74 \\ 20 & 55 & 39 & 60 & 83 & 67 & 35 & 32 \end{bmatrix}$$

$$\mathbf{b} = [b_i] = [100 \ 100 \ 100]$$

A solução ótima é dada por  $x_{13} = x_{15} = x_{17} = 1$ ;  $x_{21} = x_{22} = x_{26} = 1$ ;  $x_{34} = x_{38} = 1$ , isto é, as tarefas 3, 5 e 7 são designadas ao agente 1, as tarefas 1, 2 e 6 são designadas ao agente 2, e as ta-

refas 4 e 8 são designadas ao agente 3. O valor da solução ótima é 379. Note que somente o agente 3 tem folga de recurso de 8 unidades. Se a capacidade dos agentes 1 ou 2 é reduzida para 99, então o exemplo não tem solução factível. ■

#### *Designação generalizada com múltiplos recursos*

Este problema envolve múltiplos recursos  $k = 1, \dots, r$ , associados a cada agente  $i$ . Seja  $a_{ijk}$  a quantidade do recurso  $k$  usada pelo agente  $i$  para executar a tarefa  $j$ , e  $b_{ik}$  a quantidade do recurso  $k$  disponível para o agente  $i$ . A única alteração em relação ao modelo (D5–8) ocorre nas restrições (D7), que são substituídas pelas restrições

$$\sum_{j=1}^n a_{ijk} x_{ij} \leq b_{ik}, \quad i = 1, \dots, m, \quad k = 1, \dots, r \quad (\text{D9})$$

que restringem a utilização do recurso  $k$  do agente  $i$ . Este problema aparece em diversas aplicações, tais como alocação de bases de dados entre nós de um sistema de computadores distribuídos<sup>16</sup> e programação da produção em *job shop*.<sup>17</sup>

#### *Designação generalizada com múltiplos níveis*

Neste problema, cada tarefa  $j$  é executada por um único agente em um nível de eficiência  $k = 1, \dots, l$ . Seja  $a_{ijk}$  a quantidade do recurso usada pelo agente  $i$  no nível  $k$  para executar a tarefa  $j$ , e  $b_i$  a quantidade do recurso disponível para o agente  $i$ . O custo de designar a tarefa  $j$  ao agente  $i$  no nível  $k$  é  $c_{ijk}$ . Em problemas reais, a relação entre custo e utilização de recursos é tal que, se  $a_{ijk} < a_{ijk'}$ , então  $c_{ijk} > c_{ijk'}$ .

Defina as variáveis

$$x_{ijk} = \begin{cases} 1 & \text{se a tarefa } j \text{ é designada ao agente } i \text{ no nível } k \\ 0 & \text{caso contrário} \end{cases}$$

O problema é, então, formulado como

$$\min \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^l c_{ijk} x_{ijk} \quad (\text{D10})$$

$$\sum_{i=1}^m \sum_{k=1}^l x_{ijk} = 1, \quad j = 1, \dots, n \quad (\text{D11})$$

$$\sum_{j=1}^n \sum_{k=1}^l a_{ijk} x_{ijk} \leq b_i, \quad i = 1, \dots, m \quad (\text{D12})$$

$$\mathbf{x} \in B^{mn l} \quad (\text{D13})$$

A função objetivo (D10) expressa a minimização do custo total de designação de tarefas a agentes nos níveis. As restrições (D11) garantem que cada tarefa  $j$  é executada por um único agente  $i$  em um nível  $k$ , e as restrições (D12) impõem que a capacidade  $b_i$  de cada agente  $i$  não é excedida. A restrição (D13) indica o tipo das variáveis.

Este problema aparece na designação de produção em máquinas,<sup>18</sup> em que  $c_{ijk}$  representa o custo (preparação, produção e estoque por unidade de tempo) de designar o item  $j$  à máquina  $i$

16 Pirkul (1986).

17 Shtub e Kogan (1998).

18 Glover *et al.* (1979).

com tamanho de lote  $k$ . Portanto, para um par específico item-máquina, um lote pequeno resulta em alto custo e vice-versa.

### *Designação quadrática*

Este problema<sup>19</sup> envolve duas decisões de designação simultâneas e é melhor ilustrado por meio do *layout* de facilidades.<sup>20</sup> Existem  $n$  facilidades a serem designadas a  $n$  locais, tal que cada instalação é designada a um único local, e cada local contém exatamente uma facilidade. A distância do local  $k$  para o local  $l$  é  $c_{kl}$  e o fluxo de volume da facilidade  $i$  para a facilidade  $j$  é  $f_{ij}$ . O problema consiste em designar facilidade a locais de forma que o produto fluxo-distância total seja minimizado.

Defina as variáveis

$$x_{ik} = \begin{cases} 1 & \text{se a facilidade } i \text{ é designada ao local } k \\ 0 & \text{caso contrário.} \end{cases}$$

O modelo para este problema é dado por

$$\min \sum_{\substack{i=1 \\ j \neq i}}^n \sum_{k=1}^n \sum_{\substack{l=1 \\ l \neq k}}^n f_{ij} c_{kl} x_{ik} x_{jl} \quad (\text{D14})$$

sujeito a (D2–D4).

A função objetivo (D14) representa a minimização do produto total fluxo-distância. Note que o termo  $f_{ij} c_{kl}$  ocorre somente se  $x_{ik} = 1$  e  $x_{jl} = 1$ , portanto, a função objetivo é quadrática.

### **3.4.4 Problemas de cobertura, partição e empacotamento de conjuntos**

Estes problemas têm uma estrutura semelhante e envolvem a seleção de uma coleção de subconjuntos  $S_j$ ,  $j = 1, \dots, n$  de um conjunto  $S$ , de forma que a coleção constitua uma cobertura, uma partição ou um empacotamento com relação a  $S$ . Para ilustrar estes problemas, considere o seguinte exemplo. Seja o conjunto  $S = \{1, 2, 3, 4, 5\}$ , e os seguintes subconjuntos de  $S$ ,

$$S_1 = \{1, 2\}, S_2 = \{1, 3, 5\}, S_3 = \{2, 4, 5\}, S_4 = \{3\}, S_5 = \{1\}, S_6 = \{4, 5\}$$

Uma cobertura de  $S$  requer que a união dos subconjuntos seja igual a  $S$ , por exemplo,  $S_1 \cup S_3 \cup S_4 = S$ . Um empacotamento envolve uma união de subconjuntos disjuntos, por exemplo, a união  $S_4 \cup S_5 \cup S_6$  é um empacotamento, pois  $S_4 \cap S_5 \cap S_6 = \emptyset$ . Uma partição de  $S$  é uma cobertura e um empacotamento com relação a  $S$ . Os subconjuntos  $S_1, S_4, S_6$  constituem uma partição de  $S$ , pois  $S_1 \cup S_4 \cup S_6 = S$  e  $S_1 \cap S_4 \cap S_6 = \emptyset$ . A Figura 3.9 ilustra graficamente os exemplos dados para cobertura, empacotamento e partição.

De modo geral, um custo ou valor  $c_j$  está associado à  $S_j$ , e os problemas de otimização combinatória associados podem ser assim enunciados. Selecione uma coleção de subconjuntos de custo mínimo para os problemas de cobertura e partição, e de valor máximo para o problema de empacotamento, respectivamente.

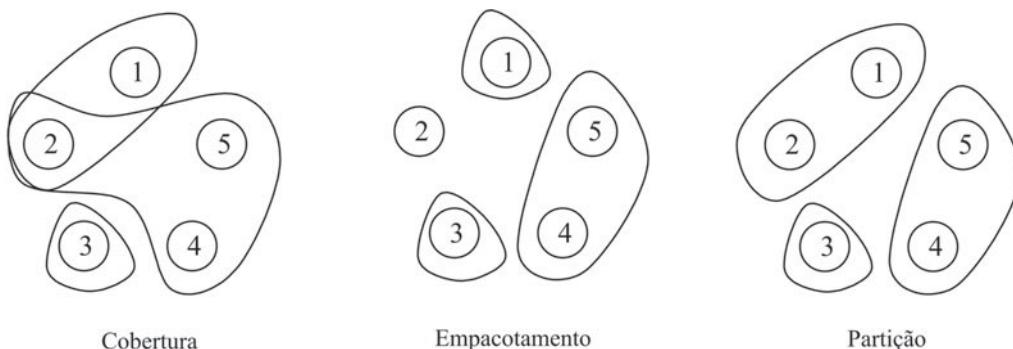
A seguir, são apresentados problemas práticos que podem ser modelados como problemas de cobertura, partição e empacotamento.

A localização de facilidades de emergência, tais como ambulâncias e estações de bombeiros, foi um dos primeiros tipos de problemas modelado como um problema de cobertura.<sup>20</sup> Suponha que cada subconjunto  $S_j, j = 1, \dots, 6$ , definido anteriormente, é tal que:

$S_j$  = subconjunto de bairros de uma cidade que podem ser atendidos dentro de 10 minutos (ou distância especificada) a partir de uma facilidade de atendimento  $j$  com custo de instalação  $c_j$ .

<sup>19</sup> Uma revisão de abordagens sobre estes problemas pode ser visto em Loiola *et al.* (2004). Para um exemplo de aplicação no transporte de carga em redes do tipo *hub-and-spoke*, ver Silva e Cunha (2005).

<sup>20</sup> Toregas *et al.* (1971).



**Figura 3.9** Ilustração dos problemas de empacotamento, cobertura e participação.

Determine os locais das facilidades de modo que todos os bairros sejam atendidos por pelo menos uma facilidade, e o custo total de instalação seja minimizado.

Defina as variáveis:

$$x_j = \begin{cases} 1 & \text{se a facilidade } j \text{ é selecionada} \\ 0 & \text{caso contrário.} \end{cases}$$

A formulação do problema é a seguinte:

$$\min \sum_{j=1}^6 c_j x_j \quad (\text{CO1})$$

$$\begin{aligned}
 x_1 + x_2 &+ x_5 \geq 1 && \text{(bairro 1)} \\
 x_1 + x_3 &\geq 1 && \text{(bairro 2)} \\
 x_2 + x_4 &\geq 1 && \text{(bairro 3)} \\
 x_3 + x_6 &\geq 1 && \text{(bairro 4)} \\
 x_2 + x_5 &+ x_6 \geq 1 && \text{(bairro 5)}
 \end{aligned} \tag{CO2}$$

$$\mathbf{x} \in B^6 \quad (\text{CO3})$$

A função objetivo (CO1) expressa a minimização do custo total de instalação de facilidades. As restrições (CO2) indicam que cada bairro do conjunto  $S = \{1, 2, 3, 4, 5\}$ , nesta ordem, tem de ser atendido por pelo menos uma facilidade. A restrição (CO3) indica o tipo das variáveis.

De maneira genérica, o problema de cobertura de conjuntos pode ser descrito da seguinte maneira:

$$\min \mathbf{c}^T \mathbf{x}$$

$$Ax \geq 1$$

$$\mathbf{x} \in B^n$$

em que  $\mathbf{1}$  é um vetor  $(m \times 1)$  com componentes iguais a 1, e os elementos da matriz de incidência  $\mathbf{A}$  são dados por

$$a_{ij} = \begin{cases} 1 & \text{se } i \in S_j \\ 0 & \text{caso contrário} \end{cases}$$

No exemplo anterior, esta matriz é dada por

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

**Exemplo 3.9** Para ilustrar este problema, considere  $\mathbf{c} = [20 \ 76 \ 16 \ 23 \ 23 \ 18]$ , tal que  $c_1 = 20$  representa o custo de  $S_1$ , e assim por diante até  $c_6 = 18$ , que representa o custo de  $S_6$ . A solução ótima é  $x_1 = x_3 = x_4 = 1$ , com valor 59, e é ilustrada na Figura 3.9. ■

Problemas de programação de tripulação de aeronaves podem ser modelados como problemas de partição ou cobertura.<sup>21</sup> Para uma dada programação diária ou semanal de vôos, a empresa deve alojar uma tripulação para cada vôo, de forma a atender a regras de segurança, requisitos de sindicato, política da empresa (restrições) e minimizar os custos. A unidade básica de planejamento é uma rota ou *pairing*, que consiste em uma seqüência de vôos atendida por uma tripulação, iniciada e terminada na cidade base em que esta reside. A formação de rotas que atendem as restrições acima descritas e o cálculo de seus custos é uma tarefa complexa, realizada por meio de programa computacional.

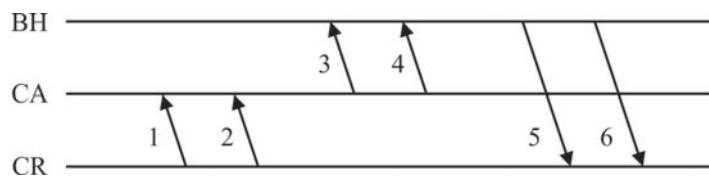
**Exemplo 3.10** A Figura 3.10 ilustra as rotas entre os aeroportos de Curitiba (CR), Campinas (CA) e Belo Horizonte (BH), com os respectivos números dos vôos (6 vôos). A Tabela 3.1 mostra todas as 24 possíveis rotas com três vôos. Cada rota  $j$  tem um custo  $c_j$ , e o objetivo é minimizar o custo total de forma que um vôo seja atendido exatamente por uma tripulação. Trata-se, portanto, de um problema de partição do conjunto  $S = \{1, \dots, 6\}$ .

Defina as variáveis:

$$x_j = \begin{cases} 1 & \text{se a rota } j \text{ é selecionada} \\ 0 & \text{caso contrário} \end{cases}$$

O custo  $c_j$  de cada rota  $j = 1, \dots, 24$  é mostrado na função objetivo da seguinte formulação do problema:

$$\min 20x_1 + 16x_2 + 17x_3 + 18x_4 + 20x_5 + 13x_6 + 14x_7 + 15x_8 + 20x_9 + 21x_{10} + 31x_{11} + 15x_{12} + 11x_{13} + 43x_{14} + 54x_{15} + 45x_{16} + 34x_{17} + 23x_{18} + 34x_{19} + 12x_{20} + 34x_{21} + 11x_{22} + 43x_{23} + 12x_{24}$$



**Figura 3.10** Vôos entre cidades.

<sup>21</sup> Arabeyre *et al.* (1969). Muitos outros problemas também podem ser modelados como problemas de partição ou cobertura, por exemplo, ver Xavier e Oliveira (2004), Gomes e Palhanob (2006).

**Tabela 3.1**  
**Rotas com três vôos.**

	<i>Rotas</i>		<i>Rotas</i>		<i>Rotas</i>
1	1-3-5	9	3-5-1	17	5-1-3
2	1-3-6	10	3-5-2	18	5-1-4
3	1-4-5	11	3-6-1	19	5-2-3
4	1-4-6	12	3-6-2	20	5-2-4
5	2-3-5	13	4-5-1	21	6-1-3
6	2-3-6	14	4-5-2	22	6-1-4
7	2-4-5	15	4-6-1	23	6-2-3
8	2-4-6	16	4-6-2	24	6-2-4

$$x_1 + x_2 + x_3 + x_4 + x_9 + x_{11} + x_{13} + x_{15} + x_{17} + x_{18} + x_{21} + x_{22} = 1 \quad (\text{vôo } 1)$$

$$x_5 + x_6 + x_7 + x_8 + x_{10} + x_{12} + x_{14} + x_{16} + x_{19} + x_{20} + x_{23} + x_{24} = 1 \quad (\text{vôo } 2)$$

$$x_1 + x_2 + x_5 + x_6 + x_9 + x_{10} + x_{11} + x_{12} + x_{17} + x_{19} + x_{21} + x_{23} = 1 \quad (\text{vôo } 3)$$

$$x_3 + x_4 + x_7 + x_8 + x_{13} + x_{14} + x_{15} + x_{16} + x_{18} + x_{20} + x_{22} + x_{24} = 1 \quad (\text{vôo } 4) \quad (P1)$$

$$x_1 + x_3 + x_5 + x_7 + x_9 + x_{10} + x_{13} + x_{14} + x_{17} + x_{18} + x_{19} + x_{20} = 1 \quad (\text{vôo } 5)$$

$$x_2 + x_4 + x_6 + x_8 + x_{11} + x_{12} + x_{15} + x_{16} + x_{21} + x_{22} + x_{23} + x_{24} = 1 \quad (\text{vôo } 6)$$

$$\mathbf{x} \in B^{24} \quad (P2)$$

A função objetivo expressa a minimização do custo total das rotas. As restrições (P1) indicam que cada vôo tem de ser atendido por uma única tripulação, e a restrição (P2) estabelece o tipo das variáveis. A solução ótima deste exemplo é  $x_6 = x_{13} = 1$ , com valor 24.

De forma geral, o problema de partição de conjuntos pode ser descrito da seguinte maneira:

$$\min \mathbf{c}^T \mathbf{x}$$

$$\mathbf{A}\mathbf{x} = \mathbf{1}$$

$$\mathbf{x} \in B^n$$

em que o vetor **1** e a matriz de incidência **A** são idênticos ao problema de cobertura.

Para ilustrar o problema de empacotamento de conjuntos, considere uma grande empresa com lojas de departamentos distribuídas no país, em que seu setor de logística controla a cadeia de suprimentos que conecta o vendedor (produtor), os centros de distribuição e as lojas de varejo. Grande parte das operações envolve a contratação de transportadoras para enviar mercadorias entre cidades. Para tal, a empresa promove um leilão entre transportadoras para vender trechos, em que cada trecho consiste no transporte entre duas cidades. Em um leilão convencional, só são permitidos lances em trechos individuais, enquanto em um leilão combinatório,<sup>22</sup> permitem-se

<sup>22</sup> Ledyard *et al.* (2000), Vries e Vohra (2003).

lances em combinações de trechos. Leilões combinatórios começaram a ser praticados a partir da década de 1980 para permitir economias. Por exemplo, uma transportadora faz um lance de R\$80 mil para o trecho A-B entre as cidades A e B, e outra transportadora faz um lance de R\$30 mil para o trecho C-A entre as cidades C e A. Uma terceira transportadora faz um lance de R\$100 mil para o conjunto de trechos A-B e C-A. Obviamente, o último lance é o vencedor, e a empresa economiza R\$10 mil em relação à contratação dos trechos individuais propostos pelas duas primeiras transportadoras.

Usando a terminologia de conjuntos, o conjunto  $S$  consiste de todos os trechos e  $S_j$ ,  $j = 1, \dots, J$ , representa os subconjuntos distintos de trechos propostos por todas as transportadoras  $k = 1, \dots, K$ . Considere os seguintes parâmetros:

$$a_{ijk} = \begin{cases} 1 & \text{se a transportadora } k \text{ quer o subconjunto } S_j \text{ que contém o trecho } i \\ 0 & \text{caso contrário} \end{cases}$$

$R_j$  = soma dos valores individuais dos trechos do subconjunto  $S_j$ , isto é, o valor pago pela empresa se contratasse uma transportadora para cada trecho de  $S_j$ . Por exemplo, se  $S_j$  contém os trechos A-B e C-A mencionados acima, com valores individuais de R\$80 mil e R\$30 mil, respectivamente, então  $R_j$  resulta em R\$110 mil

$b_{jk}$  = valor mínimo para a transportadora  $k$  conseguir o subconjunto  $S_j$ . Por exemplo, se houver um lance de R\$100 mil da transportadora  $k$  para os trechos A-B e C-A em  $S_j$ , então  $b_{jk}$  corresponde a esse valor.

Defina as variáveis

$$x_{jk} = \begin{cases} 1 & \text{se o subconjunto } S_j \text{ é designado à transportadora } k \\ 0 & \text{caso contrário} \end{cases}$$

Este problema é, então, modelado da seguinte maneira:

$$\max \sum_{j=1}^J \sum_{k=1}^K (R_j - b_{jk}) x_{jk} \quad (\text{E1})$$

$$\sum_{j=1}^J \sum_{k=1}^K a_{ijk} x_{jk} \leq 1, \quad \forall \text{ trecho } i \in S \quad (\text{E2})$$

$$\mathbf{x} \in B^{JK} \quad (\text{E3})$$

A função objetivo (E1) representa a economia total obtida pela empresa de lojas de departamento. As restrições (E2) impõem que cada trecho  $i$  pode ser alocado a no máximo um subconjunto  $S_j$  e transportadora  $k$ , e a restrição (E3) indica o tipo das variáveis.

De forma geral, o problema de empacotamento consiste em selecionar subconjuntos disjuntos do conjunto  $S$  de forma a maximizar o valor do empacotamento. Sua formulação genérica é dada por

$$\max \mathbf{c}^T \mathbf{x}$$

$$\mathbf{Ax} \leq \mathbf{1}$$

$$\mathbf{x} \in B^n$$

em que  $\mathbf{A}$  é a matriz de incidência bidimensional, como definida no problema de cobertura. Note, no exemplo acima, que  $\mathbf{A}$  é uma matriz tridimensional e a variável binária é bidimensional. Observe, também, que, se a restrição (C10) do problema de corte bidimensional (Seção 3.4.2) é eliminada, então este é um problema de empacotamento com matriz de incidência quadrimensional e variável binária tridimensional.

### 3.4.5 Problemas de caixeiro-viajante

Problemas de caixeiro-viajante (CV) envolvem um conjunto de cidades, em que o caixeiro sai de uma cidade base ou um depósito, visita todas as cidades ou um subconjunto delas somente uma vez, e retorna à cidade base de modo a otimizar um ou mais objetivos. Problemas de caixeiro-viajante pertencem à classe de problemas de *roteamento em nós*, e são definidos em grafos orientados ou não orientados (ver Seção 4.1).

#### *Caixeiro-viajante – distância*

Iniciamos pelo problema clássico e mais importante, objeto de grande número de trabalhos na literatura, que trata da minimização da distância da rota percorrida. Considere um grafo não orientado  $G = (N, E)$ , em que o conjunto  $N$  consiste de  $n$  cidades e  $E$  representa o conjunto de arestas entre cidades. Suponha que  $G$  é um grafo completo, isto é, para qualquer par de cidades  $i, j \in N, i \neq j$ , existe uma aresta  $(i, j)$ . A distância entre as cidades  $i$  e  $j$  é  $c_{ij}$ , e quando  $c_{ij} = c_{ji}$ , o problema é dito simétrico; caso contrário, é chamado de assimétrico. Um caixeiro deve visitar  $n$  cidades, passando por cada cidade somente uma vez, e retornar à cidade de partida. Este percurso é denominado ciclo hamiltoniano<sup>23</sup> do grafo  $G$ , e o problema consiste em determinar o ciclo hamiltoniano, ou rota, de distância mínima. Este é um dos problemas combinatórios mais conhecidos e pesquisados devido à sua aplicação em diversas áreas, tais como manufatura de circuitos, programação da produção, telecomunicações e seqüenciamento de DNA.

Defina as variáveis

$$x_{ij} = \begin{cases} 1 & \text{se o caixeiro vai diretamente da cidade } i \text{ à cidade } j, i \neq j \\ 0 & \text{caso contrário} \end{cases}$$

e considere o seguinte modelo:

$$\min \sum_{i=1}^n \sum_{j>i} c_{ij} x_{ij} \quad (\text{CV1})$$

$$\sum_{j< i} x_{ji} + \sum_{j> i} x_{ij} = 2, \quad i = 1, \dots, n \quad (\text{CV2})$$

$$\mathbf{x} \in B^{n(n-1)/2} \quad (\text{CV3})$$

A função objetivo (CV1) expressa a minimização da distância total da rota, e as restrições (CV2) impõem que cada cidade tenha somente uma cidade sucessora imediata e uma cidade predecessora imediata. A restrição (CV3) indica o tipo das variáveis. Uma solução para o modelo anterior pode gerar sub-rotas desconexas, como mostra a Figura 3.11, para um exemplo com  $n = 7$  cidades.

Seja  $S$  uma sub-rota, por exemplo,  $S = \{1, 2, 3, 4\}$  na Figura 3.11. A eliminação de sub-rotas pode ser obtida através do conjunto de restrições

$$\sum_{i \in S} \sum_{\substack{j \in S \\ j > i}} x_{ij} + \sum_{i \notin S} \sum_{\substack{j \in S \\ j > i}} x_{ij} \geq 2, \quad S \subset N, \quad 3 \leq |S| \leq \left\lfloor \frac{n}{2} \right\rfloor, \quad (\text{CV4})$$

que garantem que, para cada conjunto  $S$ , existem no mínimo duas arestas entre as cidades de  $S$  e as cidades não pertencentes a  $S$ . A cardinalidade de  $S$  é no mínimo 3, pois um ciclo em um grafo não orientado tem pelo menos 3 nós, e no máximo  $\left\lfloor \frac{n}{2} \right\rfloor$ , pois ao se eliminar ciclos com  $k$  nós, eliminam-se ciclos com  $n-k$  nós. Por exemplo, a restrição que elimina a sub-rota  $S = \{1, 2, 3, 4\}$  é dada por:

$$x_{15} + x_{16} + x_{17} + x_{25} + x_{26} + x_{27} + x_{35} + x_{36} + x_{37} + x_{45} + x_{46} + x_{47} \geq 2$$

23. Em homenagem a R. Hamilton que propôs um jogo que consistia em determinar uma rota iniciando e terminando no mesmo vértice de um dodecaedro e passando por todos os vértices sem repetir uma visita (ver, por exemplo, Goldberg e Luna, 2000).

Uma forma alternativa de eliminação de sub-rotas é dada por

$$\sum_{i \in S} \sum_{\substack{j \in S \\ j > i}} x_{ij} \leq |S| - 1, \quad S \subset N, \quad 3 \leq |S| \leq \left\lfloor \frac{n}{2} \right\rfloor \quad (\text{CV5})$$

que corresponde a eliminar uma aresta da sub-rota. Por exemplo, a sub-rota  $S = \{1, 2, 3, 4\}$  é eliminada ao se impor

$$x_{12} + x_{23} + x_{34} + x_{14} \leq 3$$

Como o número de subconjuntos distintos de um conjunto de cardinalidade  $k$  é  $2^k$ , segue-se que os conjuntos de restrições (CV4) e (CV5) têm cardinalidade da ordem de  $2^k$ ,  $k \geq 6$ , mostrando o crescimento exponencial dessas restrições com o número de cidades. Para  $k \leq 5$ , somente as restrições (CV2) impedem a ocorrência de sub-rotas.

Em problemas definidos em um grafo orientado completo  $G = (N, E)$ , como, por exemplo em problemas assimétricos, isto é,  $c_{ij} \neq c_{ji}$ , em que  $E$  agora representa o conjunto de arcos, a formulação do caixeiro-viajante é alterada para

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n c_{ij} x_{ij} \\ & \sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n, \quad j \neq i \end{aligned} \quad (\text{CV6})$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n, \quad i \neq j \quad (\text{CV7})$$

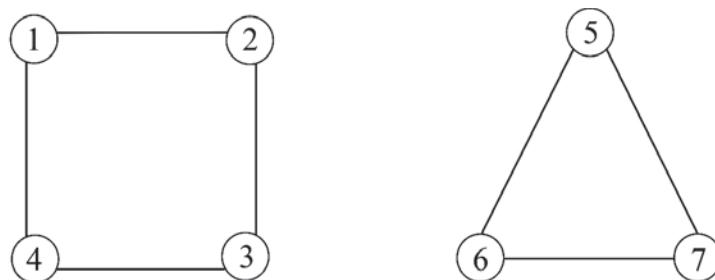
$$\mathbf{x} \in B^{n(n-1)} \quad (\text{CV8})$$

e as restrições (CV4) e (CV5) tornam-se

$$\sum_{i \in S, j \notin S} x_{ij} \geq 1, \quad S \subset N, \quad 2 \leq |S| \leq N \left\lfloor \frac{n}{2} \right\rfloor \quad (\text{CV9})$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad S \subset N, \quad 2 \leq |S| \leq \left\lfloor \frac{n}{2} \right\rfloor \quad (\text{CV10})$$

Note que as restrições (CV6) e (CV7) correspondem às restrições do problema de designação. A restrição (CV8) mostra que o número de variáveis é o dobro em relação ao caso simétrico. Além



**Figura 3.11** Ilustração de sub-rotas.

disso, cada rota deve entrar e sair de cada subconjunto  $S$  de cidades. Assim, a eliminação de uma sub-rota requer que a rota deixe cada subconjunto  $S$  pelo menos uma vez, e isso é expresso pelas restrições (CV9). As restrições (CV10) têm a mesma interpretação de (CV5).

**Exemplo 3.11** Considere o seguinte exemplo de um problema simétrico de caixeiro-viajante com 12 cidades, e com distância euclidiana  $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$  entre as cidades  $i$  e  $j$ . As coordenadas  $(x_i, y_i)$  das 12 cidades são indicadas a seguir e ilustradas na Figura 3.12(a). Por exemplo, as coordenadas da cidade 1 são  $(x_1, y_1) = (43, 23)$ .

$$\begin{aligned}\mathbf{x} &= [43 \ 58 \ 53 \ 21 \ 78 \ 46 \ 79 \ 60 \ 42 \ 87 \ 77 \ 99] \\ \mathbf{y} &= [23 \ 76 \ 64 \ 38 \ 68 \ 57 \ 6 \ 5 \ 30 \ 2 \ 97 \ 79]\end{aligned}$$

A Figura 3.12(b) mostra a solução do exemplo sem incluir restrições de sub-rotas. A solução resultante tem distância total 237,828, mas é infactível, pois apresenta quatro sub-rotas, tais que

Sub-rotas	Solução
1) 1 – 4 – 9 – 1	1) $x_{1,4} = x_{1,9} = x_{4,9} = 1$
2) 6 – 2 – 3 – 6	2) $x_{2,6} = x_{2,3} = x_{3,6} = 1$
3) 5 – 11 – 12 – 5	3) $x_{5,11} = x_{5,12} = x_{11,12} = 1$
4) 10 – 7 – 8 – 10	4) $x_{7,8} = x_{7,10} = x_{8,10} = 1$

Para eliminar essas sub-rotas, incluem-se as seguintes restrições:

$$\begin{aligned}x_{1,4} + x_{4,9} + x_{1,9} &\leq 2 \\ x_{2,6} + x_{2,3} + x_{3,6} &\leq 2 \\ x_{5,11} + x_{11,12} + x_{5,12} &\leq 2 \\ x_{7,10} + x_{7,8} + x_{8,10} &\leq 2\end{aligned}$$

A Figura 3.12(c) mostra a rota ótima com distância total 283,635, e solução

$$x_{1,8} = x_{1,9} = x_{2,3} = x_{2,11} = x_{3,6} = x_{4,6} = x_{4,9} = x_{5,10} = x_{5,12} = x_{7,8} = x_{7,10} = x_{11,12} = 1$$

■

### *m-caixeiros-viajantes – distância*

Este problema difere do problema anterior pela existência de  $m$  caixeiros-viajantes que saem de um depósito, denotado por 0, para visitar um conjunto de  $n$  cidades. Seja  $N = \{0, 1, \dots, n\}$  e admita que o grafo orientado  $G = (N, E)$  é completo. O objetivo é determinar  $m$  rotas, todas partindo do depósito e retornando ao depósito, de forma a minimizar a distância total percorrida.

Este problema pode ser formulado como:

$$\min \sum_{i=0}^n \sum_{\substack{j=0 \\ j \neq i}}^n c_{ij} x_{ij} \quad (\text{CV11})$$

$$\sum_{j=1}^n x_{0j} = m \quad (\text{CV12})$$

$$\sum_{i=1}^n x_{i0} = m \quad (\text{CV13})$$

$$(\text{CV6}) - (\text{CV7})$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad S \setminus \{0\} \subset N, \quad 2 \leq |S| \leq \left\lfloor \frac{n}{2} \right\rfloor \quad (\text{CV14})$$

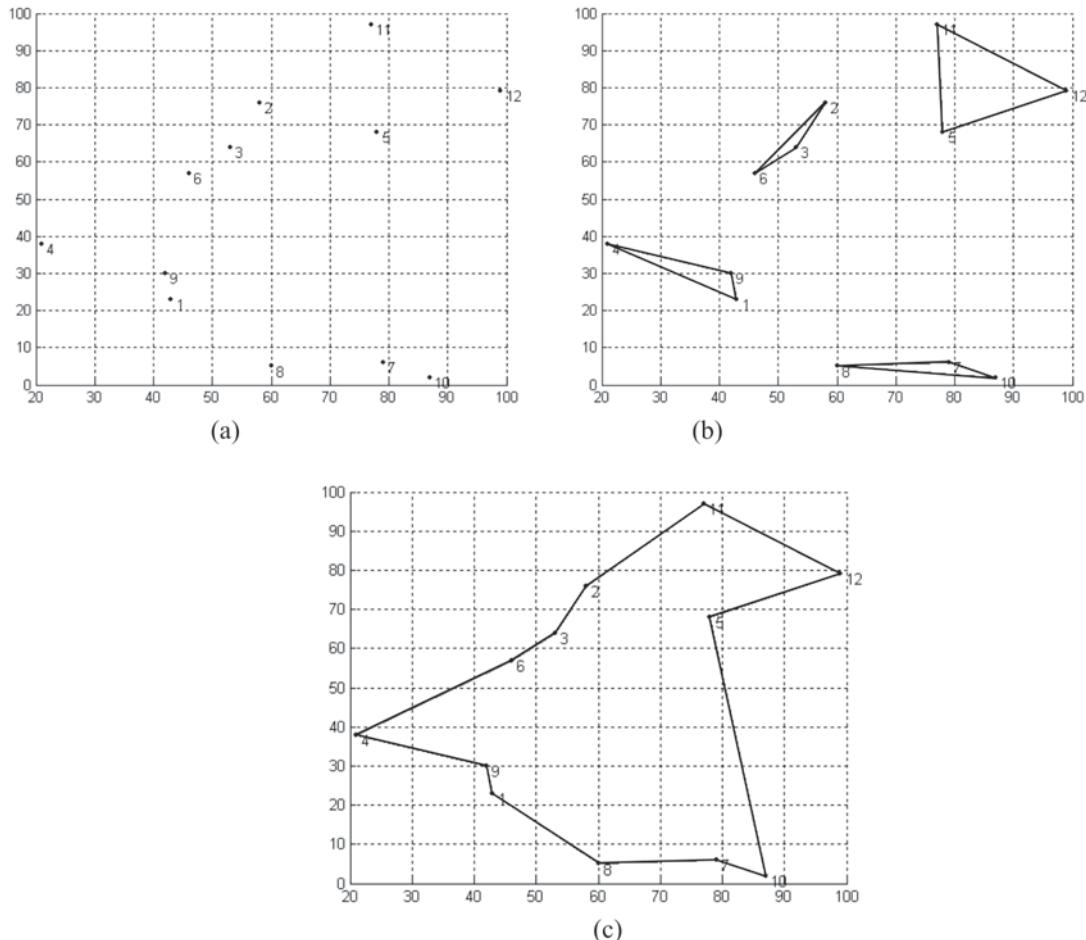
$$\mathbf{x} \in B^{n(n-1)+2n} \quad (\text{CV15})$$

A diferença do  $m$ -caixeiro-viajante em relação a um único caixeiro-viajante é expressa pelas restrições (CV12) e (CV13), que garantem que  $m$  caixeiros saem do depósito e que eles retornam ao depósito. Note que os subconjuntos  $S \setminus \{0\}$  não contêm o depósito nas restrições de eliminação de sub-rotas (CV14).

#### Caixeiro-viajante – aquisição

Este problema consiste em determinar uma rota de custo mínimo por meio de um subconjunto de mercados, tal que o custo total é a soma dos custos de viagem e da aquisição de produtos.

Considere um depósito denotado por 0, um conjunto de mercados  $M = \{1, \dots, n\}$ , e um conjunto de produtos  $K = \{1, \dots, m\}$ . Seja  $G = (N, E)$  um grafo não orientado em que  $N = \{0\} \cup M$  é o conjunto de nós, e  $E = \{(i, j) : i, j \in N\}$  é o conjunto de arestas. Cada produto  $K$  pode ser adquirido em um subconjunto  $M_k$  de mercados. O número de unidades do produto  $k$  a ser adquirido é igual a  $d_k$  e somente  $q_{ki}$  unidades de  $k$  estão disponíveis no mercado  $i$ . Admita que



**Figura 3.12** Exemplo de problema de caixeiro-viajante.

$0 < q_{ki} \leq d_k$  e  $\sum_{j \in M_k \setminus \{i\}} q_{kj} \geq d_k$  para todo  $k \in K$  e  $i \in M_k$ . O preço de produto  $k$  no mercado  $i$  é  $b_{ki}$ , e o custo de viagem na aresta  $e = (i, j)$  é representado por  $c_e$ .

O problema do caixeiro-viajante de aquisição (*traveling purchaser problem*) consiste em determinar um ciclo em  $G$  que passe pelo depósito e um subconjunto de mercados tal que todos os produtos sejam adquiridos a um custo mínimo total, que é a soma do custo de roteamento e do custo de aquisição.

Para  $S \subset M$ , defina  $\delta(S) = \{(i, j) \in E : i \in S, j \in N \setminus S\}$ . Defina também

$$M^* = \{0\} \cup \left\{ i \in M : \text{existe } k \in K \text{ tal que } \sum_{j \in M_k \setminus \{i\}} q_{kj} < d_k \right\}$$

o conjunto de nós que necessariamente faz parte de uma solução factível, e

$$K^* = \left\{ k \in K : \sum_{i \in M_k} q_{ki} = d_k \right\}$$

o conjunto de produtos sem opções de decisão de mercado. Considere as seguintes variáveis:

$$z_e = \begin{cases} 1 & \text{se a aresta } e \text{ pertence à solução} \\ 0 & \text{caso contrário} \end{cases} \quad (\text{CV16})$$

$$y_i = \begin{cases} 1 & \text{se o nó } i \text{ pertence à solução} \\ 0 & \text{caso contrário} \end{cases} \quad (\text{CV17})$$

$x_{ki}$  = quantidade do produto  $p_k$  adquirida no mercado  $i$ .

O problema pode ser formulado como:

$$\min \sum_{e \in E} c_e z_e + \sum_{k \in K} \sum_{i \in M_k} b_{ki} x_{ki} \quad (\text{CV18})$$

$$\sum_{e \in \delta(\{i\})} z_e = 2y_i, \quad i \in N \quad (\text{CV19})$$

$$\sum_{e \in \delta(S)} z_e \geq 2y_i, \quad S \subset M, \quad i \in S \quad (\text{CV20})$$

$$\sum_{i \in M_k} x_{ki} = d_k, \quad k \in K \quad (\text{CV21})$$

$$x_{ki} \leq q_{ki} y_i, \quad k \in K, \quad i \in M_k \quad (\text{CV22})$$

$$\mathbf{z}_e \in B^{|E|}, \quad \mathbf{y} \in B^{|M \setminus M^*|} \quad (\text{CV23})$$

$$y_i = 1 \text{ se } i \in M^*, \quad \mathbf{x} \in R_+^{mn} \quad (\text{CV24})$$

A função objetivo (CV18) minimiza a soma dos custos de viagem e da aquisição de produtos. As restrições (CV19) garantem que, se o nó  $i$  faz parte da solução, então o número de arestas que incidem em  $i$  (grau de  $i$ ) é igual a 2. As restrições (CV20) asseguram a conectividade entre um nó  $i \in S$  e um nó  $\in N \setminus S$ . As restrições (CV21) garantem que a demanda do produto  $k$  é adquirida, e as restrições (CV22) impõem que um produto  $k$  não pode ser adquirido em um mercado  $i$ .

não visitado. As restrições (CV23) e (CV24) indicam o tipo das variáveis. Note que, se  $k \in K^*$ , nas restrições (CV21), tem-se  $x_{ki} = q_{ki}$ .

#### *Caixeiro-viajante – lucro*

Nestes problemas, um lucro está associado a cada nó e o objetivo consiste em determinar uma rota de custo mínimo por meio de um subconjunto dos nós, de forma a otimizar a soma dos custos de viagem e dos lucros coletados. Estes problemas são semelhantes ao problema anterior, no sentido de que apenas um subconjunto de nós é visitado e, por esse motivo, usa-se a mesma notação.

Considere um depósito denotado por 0 e um conjunto de nós  $M = \{1, \dots, n\}$ . Seja  $G = (N, E)$  um grafo não orientado em que  $N = \{0\} \cup M$  é o conjunto de nós, e  $E = \{(i, j) : i, j \in N\}$  é o conjunto de arestas. Um lucro  $p_i$  é associado a cada nó  $i \in M$ , e o custo de viagem na aresta  $e = (i, j)$  é representado por  $c_e$ . Considere também as variáveis binárias  $z_e$  e  $y_i$  definidas em (CV16) e (CV17), respectivamente.

Existem três problemas genéricos de caixeiro-viajante com lucro, que dependem da forma como os objetivos são considerados:

1. Os objetivos são combinados em uma função objetivo; a meta é encontrar uma rota que minimize o custo de viagem, menos o lucro coletado. Este problema é conhecido na literatura como o problema da rota lucrativa (*profitable tour problem*), e é formulado da seguinte maneira:

$$\min \sum_{e \in E} c_e z_e - \sum_{i \in N} p_i y_i$$

sujeito a (CV19) e

$$\sum_{e \subset S} z_e \leq |S| - 1, \quad S \setminus \{0\} \subset N, \quad 3 \leq |S| \leq \left\lfloor \frac{n}{2} \right\rfloor \quad (\text{CV25})$$

$$\mathbf{z}_e \in B^{|E|}, \quad \mathbf{y} \in B^n \quad (\text{CV26})$$

2. O custo de viagem é uma restrição; a meta é encontrar uma rota que maximize o lucro coletado, de modo que o custo de viagem não exceda um valor especificado  $c_{\max}$ . Este problema é conhecido como problema de orientação (*orienteering problem*), um esporte em que participantes têm um mapa de uma área e devem visitar uma sequência de pontos de controle da forma mais rápida possível. Este problema é formulado como:

$$\max \sum_{i \in N} p_i y_i$$

sujeito a (CV19), (CV25), (CV26) e

$$\sum_{e \in E} c_e z_e \leq c_{\max}$$

3. O lucro é uma restrição; o alvo é encontrar uma rota que minimize o custo de viagem de modo que o lucro coletado não seja menor que um valor especificado  $p_{\min}$ . Este problema é conhecido como problema de coleta de prêmio (*prize-collecting problem*), e é formulado como:

$$\min \sum_{e \in E} c_e z_e$$

sujeito a (CV19), (CV25), (CV26) e

$$\sum_{i \in N} p_i y_i \geq p_{\min}$$

Outro tipo de problema é o caixeiro-viajante dependente do tempo, em que o custo da viagem entre duas cidades  $i, j$ , representado por  $c_{ijt}$ , depende da distância entre as cidades, e se  $i$  é visitada no período de tempo  $t - 1$ , e  $j$  é visitada no período de tempo  $t$ . Este problema encontra diversas aplicações, tais como programação da produção com custos ou tempos de preparação dependentes da seqüência, e minimização do tempo de espera do início de serviço de clientes.<sup>24,25</sup> Outras variantes incluem o caixeiro-viajante em que o custo entre cidades é uma função contínua do tempo, e problemas dinâmicos de caixeiro-viajante, em que novos pedidos de serviço surgem durante o percurso, pedidos estes que podem ser atendidos ou não.<sup>26</sup>

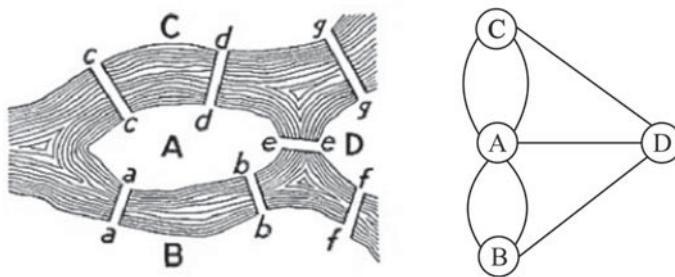
### 3.4.6 Problemas de carteiro chinês

Problemas de carteiro chinês pertencem à classe de problemas de *rroteamento em arcos*, e são definidos em grafos orientados ou não orientados. Em problemas de rroteamento em arcos, a meta é determinar a travessia de custo mínimo de um conjunto especificado de arcos (ou arestas) de um grafo, com ou sem restrições. Estes problemas aparecem em diversos contextos práticos, tais como entrega e coleta de correspondência, coleta de lixo, rroteamento de ônibus escolar e patrulhamento de ruas pela polícia.

O problema das pontes de Königsberg é, talvez, o primeiro problema de rroteamento em arcos. Königsberg é uma cidade que pertencia à Alemanha e hoje pertence à Rússia, com o nome de Kaliningrad. A questão é determinar se existe um caminho fechado passando em cada ponte somente uma vez. A Figura 3.13 ilustra as pontes e o grafo associado.

O famoso matemático suíço Euler, em 1736, encontrou condições para a existência de um caminho fechado e mostrou que não existe tal caminho neste caso específico. Essa descoberta de Euler foi um dos marcos iniciais da teoria de grafos.

Considere um grafo não orientado  $G = (N, E)$  conexo, isto é, existe um caminho formado por arestas entre quaisquer dois nós do grafo. Euler demonstrou o seguinte resultado: existe um ciclo euleriano que atravessa cada aresta do grafo exatamente uma vez se e somente se todo nó tem grau par. A necessidade da condição do grau é óbvia: um nó que aparece  $k$  vezes no ciclo euleriano deve ter grau  $2k$ , isto é,  $k$  entradas e  $k$  saídas. A suficiência da condição é demonstrada em livros de grafos.<sup>27</sup> Um exemplo de um grafo com nós pares e um ciclo euleriano é mostrado na Figura 3.14.



**Figura 3.13** Pontes de Königsberg e grafo associado.

24 Ver Gouveia e Voß (1995) para uma classificação de formulações.

25 Ver Lucena (1990) para um método exato de minimização do tempo médio de chegada nas cidades.

26 Ver Hagnani e Jung (2005) para uma heurística para o problema dinâmico do caixeiro viajante, em que o custo é uma função contínua do tempo.

27 Ver, por exemplo, Diestel (2005).

Quando não existe um ciclo euleriano, uma questão natural é determinar um caminho fechado de comprimento mínimo que atravesse cada aresta pelo menos uma vez. Este problema, conhecido como problema do carteiro chinês, foi proposto pelo matemático Guan em 1962, quando trabalhava em um correio durante a revolução cultural chinesa. Guan enunciou este problema da seguinte maneira: “Um carteiro tem de cobrir seu segmento designado antes de retornar ao posto de correio. O problema é encontrar o caminho mais curto para o carteiro.”

Considere agora o problema do carteiro chinês em um grafo não orientado  $G = (N, E)$ , tal que  $c_e$  é o custo da aresta  $e \in E$ . Se todos os nós têm grau par, então existe um ciclo euleriano do grafo, e qualquer ciclo tem o mesmo custo, isto é, a soma dos custos de todas as arestas.

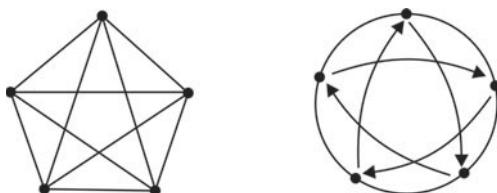
No caso em que existam vértices com grau ímpar, Guan observou que a adição de arestas aos vértices de grau ímpar, isto é, replicando arestas com o mesmo custo, gera um grafo em que todos os vértices têm grau par. O problema do carteiro chinês consiste em determinar quais arestas devem ser duplicadas de forma a obter um ciclo euleriano de custo mínimo. A Figura 3.15(a) mostra um grafo não euleriano, e as Figuras 3.15(b) e 3.15(c) mostram dois grafos eulerianos obtidos por duplicação de arestas. Os custos dos ciclos eulerianos nas Figuras 3.15(a), 3.15(b) e 3.15(c) são 19, 24 e 27, respectivamente.

Dado um grafo  $G = (N, E)$ , sejam  $\hat{E}$  e  $\check{E}$  conjuntos de arestas orientadas, em que um conjunto contém uma cópia orientada de cada aresta de  $E$  em uma direção (arbitrária), e o outro conjunto contém uma cópia de cada aresta de  $E$  na direção oposta. Para uma dada aresta  $e \in E$ , sejam  $\hat{e} \in \hat{E}$  e  $\check{e} \in \check{E}$  e as arestas com direções opostas associadas a  $e$ . Seja  $\delta_v^+$  o número de arestas orientadas que entram no nó  $v$ , e  $\delta_v^-$  o número de arestas orientadas que saem do nó  $v$ .

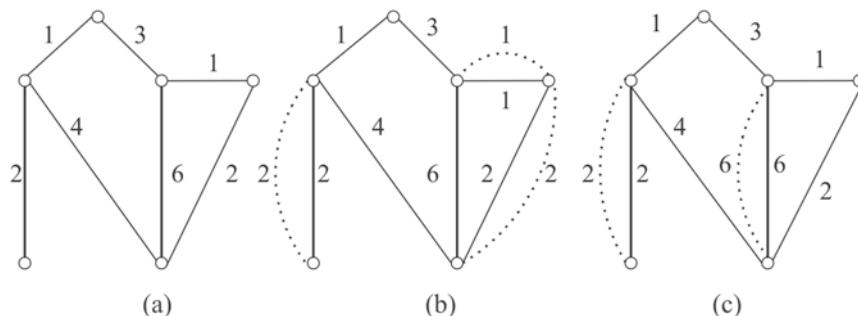
Defina as variáveis

$$x_{\hat{e}} = \begin{cases} 1 & \text{se a aresta } \hat{e} \text{ associada à aresta } e \text{ é escolhida} \\ 0 & \text{caso contrário,} \end{cases}$$

$$x_{\check{e}} = \begin{cases} 1 & \text{se a aresta } \check{e} \text{ associada à aresta } e \text{ é escolhida} \\ 0 & \text{caso contrário,} \end{cases}$$



**Figura 3.14** Exemplo de ciclo euleriano.



**Figura 3.15** Soluções para um exemplo do carteiro chinês.

Considere o modelo

$$\min \sum_{\bar{e} \in \bar{E}} c_{\bar{e}} x_{\bar{e}} + \sum_{\check{e} \in \check{E}} c_{\check{e}} x_{\check{e}} \quad (\text{CC1})$$

$$x_{\bar{e}} + x_{\check{e}} \geq 1, \quad \forall \text{ par } (\bar{e}, \check{e}) \text{ associado a } e \in E \quad (\text{CC2})$$

$$\sum_{\bar{e} \in S^+(v)} x_{\bar{e}} + \sum_{\check{e} \in S^+(v)} x_{\check{e}} - \left( \sum_{\hat{e} \in S^-(v)} x_{\hat{e}} + \sum_{\check{e} \in S^-(v)} x_{\check{e}} \right) = 0, \quad \forall v \in N \quad (\text{CC3})$$

$$x_{\bar{e}}, x_{\check{e}} \in \{0,1\}, \quad \forall \bar{e} \in \bar{E}, \quad \forall \check{e} \in \check{E} \quad (\text{CC4})$$

A função objetivo (CC1) minimiza o custo do ciclo euleriano. As restrições (CC2) garantem que um ciclo passa no mínimo uma vez em cada aresta de  $E$ . As restrições (CC3) impõem que o grau de  $v$ , ao se retirar a orientação das arestas, é par. A restrição (CC4) indica o tipo da variável.

Considere agora o problema do carteiro chinês em um grafo orientado  $G=(N,E)$ . Neste caso, a condição necessária e suficiente para a existência de um ciclo euleriano é que o número de arcos que entram e saem de um nó deve ser igual.<sup>28</sup> A Figura 3.16 ilustra um grafo sem solução para este problema.

Defina as variáveis

$x_{ij}$  = número de vezes que o arco  $(i, j)$  é atravessado

O problema é, então, modelado como

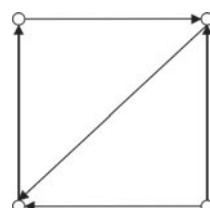
$$\min \sum_{(i,j) \in E} c_{ij} x_{ij} \quad (\text{CC5})$$

$$\sum_{\{j:(i,j) \in E\}} x_{ij} - \sum_{\{j:(j,i) \in E\}} x_{ji} = 0, \quad i \in N \quad (\text{CC6})$$

$$x_{ij} \geq 1, \quad \forall (i, j) \in E \quad (\text{CC7})$$

$$\mathbf{x} \in Z_+^{|E|} \quad (\text{CC8})$$

A função objetivo (CC5) minimiza o custo do ciclo euleriano. As restrições (CC6) garantem que o ciclo entre e saia de um nó um mesmo número de vezes, e as restrições (CC7) impõem que cada arco seja visitado pelo menos uma vez. A restrição (CC8) indica o tipo das variáveis. A formulação do problema do carteiro chinês em um grafo misto, com arestas e arcos, é uma combinação dos dois modelos anteriores.<sup>29</sup>



**Figura 3.16** Grafo sem solução para o problema do carteiro chinês.

<sup>28</sup> Ford e Fulkerson (1962).

<sup>29</sup> Ralphs (1993).

No problema do caixeiro chinês, busca-se atravessar todas as arestas ou arcos do grafo. No entanto, em várias aplicações de roteamento em arcos, busca-se atravessar um subconjunto das arestas ou arcos do grafo. Neste caso, o problema é chamado de carteiro rural, e as formulações de problemas deste tipo são mais complexas.

### 3.5 PROBLEMAS DE LOGÍSTICA

Esta seção contém a descrição e formulação de problemas importantes nas áreas de distribuição e localização de facilidades.

#### 3.5.1 Roteamento de veículos

Problemas de roteamento<sup>30</sup> de veículos envolvem o projeto de rotas de entrega e/ou coleta de custo mínimo, partindo de um ou mais depósitos para um número de clientes, sujeito a restrições adicionais. Este tipo de problema tem um papel fundamental na área de gerenciamento da distribuição e logística.<sup>31</sup> A seguir, apresentam-se alguns problemas de roteamento de veículos que procuram cobrir nós de um grafo. Inicia-se pela apresentação do clássico problema de roteamento de veículos (PRV), que é representado por um grafo orientado completo  $G = (N, E)$ , em que  $N = C \cup \{0, n+1\}$ ,  $C = \{1, \dots, n\}$  é o conjunto de nós que representam os clientes, e  $0, n+1$  são os nós que representam o depósito. O conjunto  $E = \{(i, j) : i, j \in N, i \neq j, i \neq n+1, j \neq 0\}$  corresponde aos arcos associados às conexões entre nós. Nenhum arco termina no nó 0 e nenhum arco começa no nó  $n+1$ . Todas as rotas começam em 0 e terminam em  $n+1$ . Um custo  $c_{ij}$  e um tempo de viagem  $t_{ij}$  estão associados a cada arco  $(i, j) \in E$ . O tempo de viagem  $t_{ij}$  inclui o tempo de serviço do cliente  $i$ . Cada cliente  $i$  tem uma demanda  $d_i$ . Um conjunto  $K$  de veículos idênticos, cada veículo  $k \in K$  com capacidade  $Q$ , está situado no depósito. O objetivo é minimizar o custo total de viagens, sujeito às seguintes restrições:

- Cada rota inicia e termina no depósito.
- Cada cliente pertence somente a uma rota.
- A demanda total de uma rota não pode exceder a capacidade  $Q$  do veículo.
- O tempo de viagem de uma rota não pode exceder o limite  $D$ .

Defina as variáveis

$$x_{ijk} = \begin{cases} 1 & \text{se o veículo } k \text{ percorre o arco } (i, j), \forall k \in K, \forall (i, j) \in E \\ 0 & \text{caso contrário} \end{cases}$$

e considere a seguinte formulação:

$$\min \sum_{k \in K} \sum_{(i, j) \in E} c_{ij} x_{ijk} \quad (\text{RV1})$$

$$\sum_{k \in K} \sum_{j \in N} x_{ijk} = 1, \quad \forall i \in C \quad (\text{RV2})$$

$$\sum_{i \in C} d_i \sum_{j \in N} x_{ijk} \leq Q, \quad \forall k \in K \quad (\text{RV3})$$

<sup>30</sup> O termo roteirização de veículos, embora não encontrado nos dicionários da língua portuguesa, tem sido usado como equivalente à palavra inglesa *routing* (ou *routeing*). O termo roteamento de veículos também é utilizado alternativamente por alguns autores brasileiros.

<sup>31</sup> Toth e Vigo (2002), Ghiani *et al.* (2004) e Novaes (1989).

$$\sum_{i \in N} \sum_{j \in N} t_{ij} x_{ijk} \leq D, \quad \forall k \in K \quad (\text{RV4})$$

$$\sum_{j \in N} x_{0jk} = 1, \quad \forall k \in K \quad (\text{RV5})$$

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0, \quad \forall h \in C, \quad \forall k \in K \quad (\text{RV6})$$

$$\sum_{i \in N} x_{i,n+1,k} = 1, \quad \forall k \in K \quad (\text{RV7})$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1, \quad S \subset C, 2 \leq |S| \leq \left\lfloor \frac{n}{2} \right\rfloor, \quad \forall k \in K \quad (\text{RV8})$$

$$\mathbf{x} \in B^{K|E|} \quad (\text{RV9})$$

A função objetivo (RV1) representa a minimização do custo total das rotas. As restrições (RV2) asseguram que cada cliente  $i$  é designado a um único veículo. As restrições (RV3) impõem que a demanda total de cada rota do veículo  $k$  não excede a capacidade  $Q$  do veículo. As restrições (RV4) garantem que a duração de cada rota do veículo  $k$  não excede o limite  $D$ . As restrições (RV5), (RV6) e (RV7) representam restrições de fluxo em redes, que exigem que cada veículo  $k$  parte do depósito (nó 0) somente uma vez, deixe o nó  $h$  se e somente se entrar neste nó, e retorne ao depósito (nó  $n + 1$ ) somente uma vez. As restrições (RV7) são redundantes, mas são mantidas no modelo para enfatizar a estrutura de redes. As restrições (RV8) garantem a não-existência de sub-rotas, e a restrição (RV9) indica o tipo de variável.<sup>32</sup>

Se o número de veículos no modelo (RV1)-(RV9) for um parâmetro fixo ou um limitante superior igual a  $K$ , remova o arco  $(0, n+1)$ . Caso contrário, se o número de veículos é uma variável, atribua um custo  $c_v$  a cada veículo usado. Isso é feito impondo-se  $c_{0,n+1} = -c_v$ . Se  $c_v$  é grande, o modelo primeiramente minimiza o número de veículos e, em segundo lugar, minimiza o custo de viagem.

#### Roteamento de veículos com janelas de tempo

Considere uma extensão do PRV tal que, para cada cliente, o início do serviço deve estar dentro de uma janela de tempo  $[a_i, b_i]$ ,  $i \in C$ . Os veículos deixam o depósito no instante 0 e devem retornar durante o intervalo  $[a_{n+1}, b_{n+1}]$ . Um veículo pode chegar em um cliente antes de sua janela de tempo e esperar sem custo. Este é um problema de roteamento de veículos com janelas de tempo (PRVJT) e, para formulá-lo, considere a variável adicional

$$S_{ik} = \text{instante em que o veículo } k \text{ começa a servir o cliente } i, \quad \forall k \in K, \forall i \in C.$$

#### Restrições adicionais

- Se o veículo  $k$  deixa o nó  $i$  e viaja para o nó  $j$ , então não pode chegar em  $j$  antes de  $s_{ik} + t_{ij}$ :

$$x_{ijk} (s_{ik} + t_{ij} - s_{jk}) \leq 0, \quad \forall (i, j) \in E, \quad \forall k \in K \quad (\text{RV10})$$

- Todas as janelas de tempo devem ser respeitadas:

$$a_i \leq s_{ik} \leq b_i, \quad \forall i \in N - \{0\}, \quad \forall k \in K \quad (\text{RV11})$$

A restrição (RV10) pode ser linearizada da seguinte maneira. Seja  $M_{ij} = \max\{b_i + s_{ik} + t_{ij}, 0\}$ . Então (RV10) é equivalente a

$$s_{ik} + t_{ij} \leq s_{jk} + (1 - x_{ijk}) M_{ij}, \quad \forall (i, j) \in E, \quad \forall k \in K \quad (\text{RV12})$$

<sup>32</sup> Fukasawa et al. (2005) apresentam uma nova formulação para o PRV e um método exato de resolução.

O modelo matemático associado ao PRVJT consiste de (RV1) sujeito a (RV2), (RV3), (RV5), (RV6), (RV7) (RV9), (RV11), e (RV10) ou (RV12). A restrição (RV11), para  $i = n+1$ , limita o tempo total de viagem, e as restrições de eliminação de sub-rotas (RV8) não são necessárias aqui, pois as restrições (RV12) impedem sub-rotas. Considere, por exemplo, a sub-rota  $S = \{1, 2, 3, 4\}$  da Figura 3.11. De (RV12) tem-se:  $s_{1k} + t_{12} \leq s_{2k}$ ,  $s_{2k} + t_{23} \leq s_{3k}$ ,  $s_{3k} + t_{34} \leq s_{4k}$ , e  $s_{4k} + t_{41} \leq s_{1k}$ , que é impossível.

**Exemplo 3.12** Este exemplo ilustra os problemas de roteamento de veículos com e sem janelas de tempo. As restrições de tempo total de viagem são desconsideradas, e o custo  $c_{ij}$  é igual à distância euclidiana entre os nós  $i$  e  $j$ . Considere os seguintes parâmetros:

- Número de clientes  $n = 9$
- Coordenadas  $(x_i, y_i)$ ,  $i = 1, \dots, 9$  dos clientes

$$\begin{aligned}\mathbf{x} &= [50 \quad 16 \quad 23 \quad 40 \quad 9 \quad 97 \quad 78 \quad 20 \quad 71 \quad 64 \quad 50] \\ \mathbf{y} &= [50 \quad 32 \quad 1 \quad 65 \quad 77 \quad 71 \quad 24 \quad 26 \quad 98 \quad 55 \quad 50]\end{aligned}$$

- Demanda dos clientes

$$\mathbf{d} = [d_i] = [11 \quad 35 \quad 2 \quad 9 \quad 3 \quad 18 \quad 8 \quad 10 \quad 11]$$

- Janelas de tempo

$$\begin{aligned}\mathbf{a} = [a_i] &= [0 \quad 45 \quad 11 \quad 25 \quad 20 \quad 15 \quad 50 \quad 10 \quad 40 \quad 10 \quad 0] \\ \mathbf{b} = [b_i] &= [0 \quad 70 \quad 145 \quad 40 \quad 100 \quad 80 \quad 190 \quad 110 \quad 190 \quad 45 \quad 400]\end{aligned}$$

- Capacidade dos veículos:  $Q = 60$

A solução ótima para o PRVJT é dada por:

Rota do veículo 1: 0 – 1 – 7 – 2 – 10

$$s_{11} = 45,00 \quad s_{71} = 52,21 \quad s_{21} = 77,39 \quad s_{10,1} = 200,95$$

Rota do veículo 2: 0 – 9 – 5 – 8 – 6 – 10

$$s_{92} = 14,87 \quad s_{52} = 51,54 \quad s_{82} = 89,02 \quad s_{62} = 163,35 \quad s_{10,2} = 228,21$$

Rota do veículo 3: 0 – 3 – 4 – 10

$$s_{33} = 25,00 \quad s_{43} = 58,24 \quad s_{10,3} = 149,09$$

O custo ótimo é 428,733, e a capacidade usada dos veículos 1, 2 e 3 é 54, 42 e 11, respectivamente. A Figura 3.17 mostra a solução do PRVJT. Note o cruzamento na rota do veículo 2, causado pelas janelas de tempo. ■

A solução ótima para o PRV é dada por:

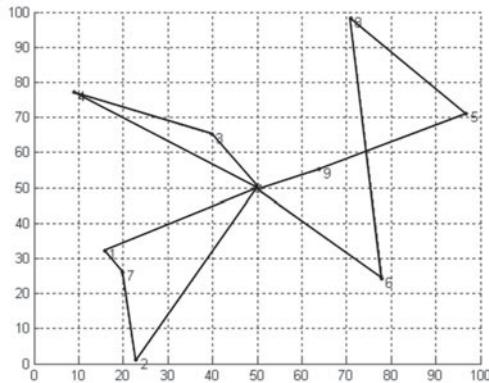
Rota do veículo 1: 0 – 2 – 7 – 1 – 10

Rota do veículo 2: 0 – 3 – 4 – 8 – 5 – 6 – 9 – 10

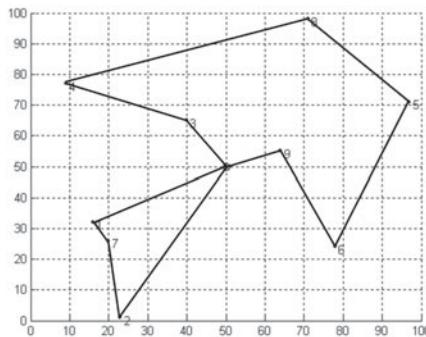
O custo ótimo é igual a 380,596, e a capacidade usada dos veículos 1 e 2 é 53 e 54, respectivamente. A Figura 3.18 mostra a solução.

#### Roteamento periódico de veículos

Considere agora o problema de roteamento periódico de veículos (PRPV), outra extensão do PRV. O PRPV opera com um horizonte de planejamento, e cada cliente  $i$  especifica uma freqüência de visitas  $f_i$  e um conjunto  $R_i$  de combinações possíveis de visitas. Por exemplo, se  $f_i = 2$ , e  $R_i = \{\{1, 3\}, \{2, 4\}, \{3, 5\}\}$ , então o cliente  $i$  deve ser visitado duas vezes e as visitas devem ocorrer nos dias 1 e 3, ou nos dias 2 e 4 ou nos dias 3 e 5.



**Figura 3.17** Representação gráfica da solução para o PRVJT.



**Figura 3.18** Representação gráfica da solução para o PRV.

Considere um multigrafo orientado  $G = (N, E)$ , uma extensão do conceito de grafo que permite múltiplos arcos ligando dois nós. O PRPV é definido em um multigrafo em que  $N = C \cup \{0\}$ ,  $C = \{1, \dots, n\}$  é o conjunto de nós que representam os clientes, e 0 é o nó que representa o depósito. O conjunto  $E = \{(i, j)^{k,l} : i, j \in N, i \neq j\}$  é o conjunto de arcos, em que  $k$  e  $l$  referem-se ao número do veículo e ao dia de visita, respectivamente. Todas as rotas começam em 0 e terminam em 0. A cada arco  $(i, j)^{k,l}$  está associado um custo  $c_{ijkl}$  e um tempo de viagem  $t_{ijkl}$ , que inclui o tempo de serviço no cliente  $i$ . Um conjunto  $K$  de veículos, cada veículo  $k \in K$  com capacidade  $Q$ , está situado no depósito. Para um horizonte  $T$  finito de dias e um cliente  $i$ , é dada a demanda  $d_i$  de  $i$ , a freqüência  $f_i$  de visitas a  $i$  e a combinação  $R_i$  de dias permitidos para visitas. O PRPV consiste em programar as visitas aos clientes e determinar as rotas dos veículos em cada dia do horizonte de tempo de forma a minimizar o custo total de viagens, sujeito às seguintes restrições:

- Cada rota inicia e termina no depósito.
- Cada cliente no dia  $l$  pertence somente a uma rota.
- A demanda total de uma rota não deve exceder a capacidade  $Q$  do veículo.
- O tempo total de uma rota não deve exceder o limite  $D$ .

Considere o parâmetro binário

$$a_{rl} = \begin{cases} 1 & \text{se o dia } l \text{ pertence à combinação de dias de visita } r \\ 0 & \text{caso contrário} \end{cases}$$

Defina as variáveis

$$x_{ijkl} = \begin{cases} 1 & \text{se o veículo } k \text{ percorre o arco } (i, j) \text{ no dia } l, \forall k \in K, \forall (i, j) \in E, l \in T \\ 0 & \text{caso contrário} \end{cases}$$

$$y_{ir} = \begin{cases} 1 & \text{se a combinação de dias de visita } r \text{ é designada ao cliente } i, r \in R_i, i \in C \\ 0 & \text{caso contrário} \end{cases}$$

e considere a seguinte formulação:

$$\min \sum_{k \in K} \sum_{(i, j) \in E} \sum_{l \in T} c_{ijkl} x_{ijkl} \quad (\text{RV13})$$

$$\sum_{r \in C_i} y_{ir} = 1, \quad \forall i \in C \quad (\text{RV14})$$

$$\sum_{j \in N} \sum_{k \in K} x_{ijkl} - \sum_{r \in C_i} a_{rl} y_{ir} = 0, \quad \forall i \in C, \quad \forall l \in T \quad (\text{RV15})$$

$$\sum_{i \in C} d_i \sum_{j \in N} x_{ijkl} \leq Q, \quad \forall k \in K, \quad \forall l \in T \quad (\text{RV16})$$

$$\sum_{i \in N} \sum_{j \in N} t_{ijkl} x_{ijkl} \leq D, \quad \forall k \in K, \quad \forall l \in T \quad (\text{RV17})$$

$$\sum_{j \in C} x_{0,jkl} \leq 1, \quad \forall k \in K, \quad \forall l \in T \quad (\text{RV18})$$

$$\sum_{i \in N} x_{ihkl} - \sum_{j \in N} x_{hkjl} = 0, \quad \forall h \in C, \quad \forall k \in K, \quad \forall l \in T \quad (\text{RV19})$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijkl} \leq |S| - 1, \quad S \subset C, 2 \leq |S| \leq \left\lfloor \frac{n}{2} \right\rfloor, \quad \forall k \in K, \quad \forall l \in T \quad (\text{RV20})$$

$$x \in B^{TK|E|} \quad (\text{RV21})$$

A função objetivo (RV13) expressa a minimização do custo total das rotas. As restrições (RV14) asseguram que cada cliente  $i$  é designado a uma única combinação de dias, enquanto as restrições (RV15) garantem que cada cliente  $i$  é visitado somente nos dias correspondentes à combinação de dias a ele designada. As restrições (RV16) impõem que a demanda total de cada rota do veículo  $k$ , em cada dia  $l$ , não excede a capacidade  $Q$  de um veículo. As restrições (RV17) garantem que a duração de cada rota do veículo  $k$ , em cada dia  $l$ , não excede o limite  $D$ . As restrições (RV18) e (RV19) representam restrições de fluxo em redes, que exigem que cada veículo  $k$  seja usado no máximo uma vez em cada dia  $l$  e, se usado, deixe o nó  $h$ , se e somente se entrar neste nó, e retorne ao nó 0. As restrições (RV20) garantem a não-existência de sub-rotas, e a restrição (RV21) indica o tipo de variável.

#### Roteamento de veículos com múltiplos depósitos

Para a formulação do problema de roteamento de veículos com múltiplos depósitos (PRVMD), considere, no PRPV (RV13)-(RV21), que o horizonte  $T$  tem  $L$  dias, que a freqüência de visitas é  $f_i = 1$  e  $R_i = \{\{1\}, \dots, \{L\}\}, i \in C$ . Associe depósitos com dias, e defina  $c_{0ikl}$  e  $c_{i0kl}$  como os custos de viagem entre o depósito  $l$  e o cliente  $i$ , usando o veículo  $k$ . A inclusão de nós de depósito

artificiais para a minimização do número de veículos em PRPV e PRVMD é feita de modo análogo ao do PRV.

### 3.5.2 Localização de facilidades

A localização de facilidades é um aspecto crítico do planejamento estratégico de empresas privadas e públicas. Exemplos típicos no setor público envolvem decisões de localização de centros de saúde, escolas e estações de bombeiros, enquanto no setor privado tem-se a localização de fábricas, armazéns e centros de distribuição. Em diversas situações, tais como em sistemas de distribuição, as decisões da localização de facilidades e de designação de clientes a facilidades são feitas simultaneamente. A seguir, apresentam-se modelos matemáticos de alguns problemas importantes de localização. Para tal, considere os seguintes parâmetros:

- $J$  = conjunto de nós  $j$  que representam os clientes,  $j = 1, \dots, n$
- $I$  = conjunto de locais  $i$  candidatos à localização de facilidades,  $i = 1, \dots, m$
- $q_j$  = demanda do cliente  $j$
- $d_{ij}$  = distância do cliente  $j$  à facilidade localizada em  $i$
- $c_{ij}$  = custo de atender a demanda  $q_j$  a partir de uma facilidade localizada em  $i$
- $f_i$  = custo fixo de instalação de uma facilidade no local  $i$
- $Q_i$  = capacidade da facilidade instalada no local  $i$

Uma variável comum a todos os problemas descritos a seguir é:

$$y_i = \begin{cases} 1, & \text{se a facilidade é aberta no local } i \\ 0, & \text{caso contrário.} \end{cases}$$

#### P-medianas

Este problema envolve a localização de  $p$  facilidades e a designação de clientes a facilidades, de modo a minimizar a soma das distâncias de clientes a facilidades.<sup>33</sup> Permite-se que os nós das instalações estejam em qualquer lugar dos arcos que ligam nós de clientes, incluindo os nós dos clientes. Demonstra-se<sup>34</sup> que existe uma solução ótima em que as facilidades situam-se no conjunto  $J$  dos clientes, portanto,  $I \subset J$ . Considere as variáveis

$$y_i = \begin{cases} 1, & \text{se a facilidade é aberta no local } i \\ 0, & \text{caso contrário} \end{cases}$$

e a seguinte formulação:

$$\min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \quad (\text{LF1})$$

$$\sum_{i \in I} x_{ij} = 1, \quad \forall j \in J \quad (\text{LF2})$$

$$x_{ij} \leq y_i, \quad \forall i \in I, \quad \forall j \in J \quad (\text{LF3})$$

$$\sum_{i \in I} y_i = p \quad (\text{LF4})$$

$$\mathbf{x} \in B^{|I||J|}, \mathbf{y} \in B^{|I|} \quad (\text{LF5})$$

<sup>33</sup> Ver Galvão (2004) para uma revisão de contribuições ao estudo deste problema e outros problemas de localização sem restrição de capacidade. Para um método ótimo e uma heurística para o problema das p-medianas, ver Senne *et al.* (2005) e Resende e Werneck (2004), respectivamente.

<sup>34</sup> Hakimi (1964, 1965).

A função objetivo (LF1) minimiza o custo total de designação de clientes a facilidades. As restrições (LF2) garantem que cada cliente  $j$  é atendido por uma única facilidade. As restrições (LF3) asseguram que cada cliente  $j$  só pode ser designado a uma facilidade aberta no local  $i$ . A restrição (LF4) indica que exatamente  $p$  facilidades são abertas, e a restrição (LF5) representa o tipo das variáveis.

### P-centros

Este problema envolve a localização de  $p$  facilidades e a designação de clientes a facilidades, de modo a minimizar a distância máxima de clientes a facilidades. Este problema admite variações do modelo básico. O problema de p-centros-nós restringe os nós de facilidades aos nós de clientes, enquanto o problema de p-centros-absolutos permite que os nós de facilidades estejam em qualquer lugar dos arcos que ligam nós de clientes.

Para formular este problema, considere as variáveis do problema de  $p$ -medianas e a seguinte variável adicional:

$r$  = distância máxima de um cliente quando designado a uma facilidade

$$\min r \quad (\text{LF6})$$

$$r \geq \sum_{i \in I} d_{ij} x_{ij}, \quad \forall j \in J \quad (\text{LF7})$$

(LF2)–(LF5)

A função objetivo (LF6) minimiza a distância máxima de um cliente a uma facilidade. A restrição (LF7) expressa  $r$  como um limitante superior da distância de cada cliente  $j$  a uma facilidade. As demais restrições são idênticas às do problema das  $p$ -medianas.

### P-medianas e p-centros com capacidade limitada

Nestes problemas, associa-se uma capacidade  $Q_i$  à facilidade no local<sup>35</sup>  $i$ . Neste caso, as restrições (LF3) são, alteradas para

$$\sum_{j \in J} q_j x_{ij} \leq Q_i y_i, \quad \forall i \in I \quad (\text{LF8})$$

Note que, ao se abrir um conjunto  $O = \{i \in I : y_i = 1\}$  de facilidades, o problema resultante é um problema de designação generalizada (ver Seção 3.4.3), que pode ser factível ou não.

### Localização de facilidades com capacidade ilimitada

Este problema envolve a localização de facilidades e a designação de clientes a facilidades, de modo a minimizar o custo fixo de implantação de facilidades e o custo variável de atendimento das demandas dos clientes. Considere as seguintes variáveis:

$x_{ij}$  = fração da demanda  $q_j$  atendida pela facilidade localizada em  $i$ ,  $\forall i \in I, \forall j \in J$

<sup>35</sup> Para aplicações em localização de escolas e seleção de locais para realização de provas de vestibular, ver Correa *et al.* (2004) e Pizzolato *et al.* (2004).

O modelo que representa este problema é dado por:

$$\min \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \quad (\text{LF9})$$

$$\sum_{i \in I} x_{ij} = 1, \quad \forall j \in J \quad (\text{LF10})$$

$$x_{ij} \leq y_i, \quad \forall i \in I, \forall j \in J \quad (\text{LF11})$$

$$\mathbf{y} \in B^{|I|}, 0 \leq x_{ij} \leq 1, \forall i \in I, \forall j \in J \quad (\text{LF12})$$

A função objetivo (LF9) expressa a minimização da soma dos custos fixos e variáveis. As restrições (LF10) asseguram que a demanda de cada cliente  $j$  é atendida. As restrições (LF11) garantem que uma fração não-nula de demanda só pode ser enviada a partir de uma facilidade aberta no local  $i$ , e a restrição (LF12) indica o tipo das variáveis.

Seja  $O = \{i \in I : y_i = 1\}$  um conjunto de instalações abertas, e defina o índice  $i$  de uma variável ou parâmetro  $w_i$  como o seu argumento, isto é,  $i = \arg w_i$ . Tem-se, então, um problema de programação linear com variáveis canalizadas, cuja solução ótima é analítica: para todo cliente  $j$ ,  $x_{i^*j} = 1$ , de modo que  $i^* = \arg \min_{i \in O} c_{ij}$ ,  $x_{ij} = 0$ , caso contrário, isto é, cada cliente  $j$  tem sua demanda atendida pela facilidades  $i$  de menor custo  $c_{ij}$ .

#### *Localização de instalações com capacidade limitada*

Este problema difere do anterior pela existência da capacidade  $Q_j$  associada à facilidade no local  $j$ . Neste caso, a restrição (LF11) é substituída pelas restrições (LF8). Ao se fixar um conjunto  $O = \{i \in I : y_i = 1\}$  de facilidades abertas, o problema resultante é um problema de transporte (veja Seção 2.2.2), que tem solução factível se e somente se  $\sum_{j \in J} q_j \leq \sum_{i \in O} Q_i$ . A resolução do problema de transporte é abordada na Seção 4.3.1.

#### *Localização de facilidades com capacidade limitada e fonte única*

Este problema difere do problema anterior pelo fato de que um cliente deve ser atendido a partir de uma única instalação. Considere as variáveis:

$$x_{ij} = \begin{cases} 1 & \text{se o cliente } j \text{ é designado à facilidade localizada em } i \\ 0 & \text{caso contrário} \end{cases}$$

Este problema pode ser modelado da seguinte maneira:

$$\min \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \quad (\text{LF13})$$

$$\sum_{i \in I} x_{ij} = 1, \quad \forall j \in J$$

$$\sum_{j \in J} q_j x_{ij} \leq Q_i y_i, \quad \forall i \in I \quad (\text{LF14})$$

$$\mathbf{x} \in B^{|I||J|}, \quad \mathbf{y} \in B^{|I|} \quad (\text{LF15})$$

As restrições (LF13) garantem que cada cliente  $j$  é designado a uma única facilidades. As restrições (LF14) garantem que, se uma instalação é aberta no local  $i$ , então a quantidade enviada

de  $i$  para os clientes  $j$  é limitada pela capacidade  $Q_i$ . A restrição (LF15) indica o tipo das variáveis. Novamente, ao se abrir um conjunto  $O = \{i \in I : y_i = 1\}$  de facilidades, o problema resultante é um problema de designação generalizada, que pode ser factível ou não.

**Exemplo 3.13** Este exemplo ilustra os dois últimos problemas. Considere os seguintes parâmetros:

Para  $i = 1, \dots, 5$  e  $j = 1, \dots, 10$ , seja

	995	756	092	951	964	1171	1194	1113	1104	1276
	1056	993	1038	1049	1044	1230	1239	1143	954	1222
$c_{ij}$ =	1192	938	894	1285	1149	1227	893	1179	916	1234
	951	1049	2085	932	1387	1478	985	1248	1102	753
	864	1044	1249	1387	981	1123	953	1253	953	862

Para  $i = 1, \dots, 5$  e  $j = 11, \dots, 20$ , seja

	1089	382	517	28	1025	1240	969	841	981	934
	942	1252	955	1042	944	664	1046	715	924	1044
$c_{ij}$ =	953	975	1093	1070	783	987	1247	1054	1323	1069
	1053	935	936	963	826	1185	1691	640	1164	970
	1285	1296	974	935	1070	1041	185	847	1056	530

$$\mathbf{Q} = [Q_i] = [1324, 1975, 1506, 1938, 2010]$$

$$\mathbf{q} = [q_j] = [716, 181, 182, 567, 125, 330, 153, 234, 117, 510, 928, 541, 148, 291, 360, 434, \\= 148, 560, 321, 583]$$

$$\mathbf{f} = [f_j] = [50201, 44208, 58800, 51130, 4940]$$

### Localização de instalações com capacidade limitada

Solução ótima:

$$x_{2,2} = x_{2,8} = x_{2,11} = x_{2,13} = x_{11,6} = x_{11,9} = 1 \\x_{2,13} = 0,39$$

$$x_{3,2} = x_{3,3} = x_{3,7} = x_{3,9} = x_{3,15} = 1 \\x_{3,12} = 0,95$$

$$x_{4,4} = x_{4,10} = x_{4,18} = 1 \\x_{4,12} = 0,05 \\x_{4,13} = 0,61 \\x_{4,14} = 0,63$$

$$x_{5,1} = x_{5,5} = x_{5,6} = x_{5,17} = 1 \\x_{5,14} = 0,37$$

$$y_2 = y_3 = y_4 = y_5 = 1$$

Valor da solução ótima: 176.052

Capacidade utilizada nas facilidades abertas:

Facilidade	2	3	4	5
Capacidade	1975	1506	1938	2010

### Localização de facilidades com capacidade limitada e fonte única

Solução ótima:

$$\begin{aligned}
 x_{2,2} &= x_{2,3} = x_{2,4} = x_{2,9} = x_{2,11} = 1 \\
 x_{3,5} &= x_{3,7} = x_{3,8} = x_{3,16} = x_{3,18} = 1 \\
 x_{4,1} &= x_{4,12} = x_{4,15} = x_{4,19} = 1 \\
 x_{5,6} &= x_{5,10} = x_{5,13} = x_{5,14} = x_{5,17} = x_{5,20} = 1 \\
 x_{5,5} &= x_{5,6} = x_{5,17} = x_{5,20} = 1 \\
 y_2 &= y_3 = y_4 = y_5 = 1
 \end{aligned}$$

O valor da solução ótima é 177.801, e a capacidade utilizada nas facilidades abertas é a mesma do caso anterior.

Como esperado, o custo do problema de facilidade com fonte única é maior. Além disso, as soluções dos dois problemas utilizam toda a capacidade disponível das facilidades abertas. ■

### Custo fixo em fluxos em redes

Considere um grafo orientado  $G = (N, E)$  com capacidade  $u_{ij}$  e custo unitário de fluxo  $c_{ij}$  para todo arco  $(i, j) \in E$ . A cada nó  $i$  é associada uma demanda  $q_i$ , que corresponde a uma fonte, um sorvedouro ou um ponto de passagem, se  $q_i$  for positivo, negativo ou zero, respectivamente. Se existe um fluxo positivo no arco  $(i, j)$ , incorre-se em um custo fixo  $f_{ij}$  associado à abertura desse arco.

O problema de custo fixo em fluxos em redes consiste em determinar quais arcos abrir e qual o fluxo em cada arco aberto, de forma a minimizar o custo total. Note que os problemas de localização anteriores estavam relacionados ao custo de abertura de nós, enquanto o problema presente consiste em localizar ou abrir arcos. Este problema ocorre em uma grande variedade de projetos de sistemas, tais como redes de suprimento de água, aquecimento, eletricidade e telecomunicações.<sup>36</sup>

Considere as variáveis:

$$y_{ij} = \begin{cases} 1 & \text{se o arco } (i, j) \text{ é aberto} \\ 0 & \text{caso contrário} \end{cases}$$

$$x_{ij} = \text{fluxo no arco } (i, j)$$

Este problema é formulado da seguinte maneira:

$$\min \sum_{(i,j) \in E} f_{ij} y_{ij} + c_{ij} x_{ij} \quad (\text{LF16})$$

$$\sum_{j \in N} x_{ji} - \sum_{j \in N} x_{ij} = q_i, \quad \forall i \in N \quad (\text{LF17})$$

<sup>36</sup> Ver Mateus *et al.* (2000) para heurísticas em projetos de redes de telecomunicações.

$$x_{ij} \leq u_{ij} y_{ij}, \quad \forall (i, j) \in E \quad (\text{LF18})$$

$$\mathbf{x} \in R_+^{|E|}, \quad \mathbf{y} \in B^{|E|} \quad (\text{LF19})$$

A função objetivo (LF16) minimiza a soma dos custos fixos e variáveis. As restrições (LF17) correspondem à conservação do fluxo em cada nó  $i$ , e as restrições (LF18) impõem que o fluxo em um arco  $(i, j)$  existe somente se este arco for aberto. A restrição (LF19) indica o tipo das variáveis.

A Seção 4.2.4 discute o caso particular em que não há custos fixos no problema (fluxo de custo mínimo).

### 3.6 PROBLEMAS DE PRODUÇÃO

Esta seção contém a descrição e a formulação de problemas importantes nas áreas de planejamento e programação da produção. Problemas na área de produção em geral são decompostos hierarquicamente em três níveis: estratégico, tático e operacional. O nível mais alto é o estratégico, em que as decisões são de longo prazo e envolvem altos investimentos. Esse nível trata da escolha e do projeto do processo, relacionados ao arranjo de máquinas e outros equipamentos e com a determinação da capacidade destes, em função da demanda futura. Um dos parâmetros determinados nesse nível é a quantidade de produção unitária, isto é, a menor quantidade em que um produto é fabricado. Tamanhos de lotes determinados no nível tático são múltiplos inteiros da quantidade de produção unitária, por exemplo, uma unidade de produção de refrigerante é uma caixa com seis vasilhames e um lote corresponde a 100 caixas.

O nível tático trata do planejamento das atividades, que consiste de dois subníveis: planejamento agregado da produção e planejamento de quantidades de produção. O planejamento agregado envolve decisões como níveis de mão-de-obra, hora extra e subcontratação, e duram tipicamente de vários meses a um ano. As necessidades são determinadas por uma medida agregada de demanda, por exemplo, horas de produção. O planejamento das quantidades de produção envolve a determinação, para cada produto, de quanto e quando produzir em um horizonte tipicamente de algumas semanas a seis meses. Nesse horizonte, os níveis de mão-de-obra e tempo disponível de máquina são, em geral, considerados fixos. O programa mestre de produção (*master production schedule – MPS*) e o planejamento de recursos de produção (*material resource planning – MRP II*) são exemplos clássicos de sistemas de planejamento de quantidades de produção. A partir das demandas externas ou independentes, o MPS determina quanto e quando produzir de cada produto final. Partindo das quantidades determinadas pelo MPS, o MRP II planeja de forma sincronizada as necessidades de componentes e matéria-prima para os produtos finais.

De modo geral, um “bom” plano de produção é aquele que satisfaz as demandas sem atraso, respeita a capacidade dos recursos disponíveis e minimiza os custos de produção. Os custos de estoque e de preparação (*setup*) de máquinas são, em geral, os componentes mais importantes do custo total de produção, e têm caráter conflitante, isto é, o custo de preparação total ao longo do horizonte de planejamento diminui com a quantidade produzida, enquanto o custo total de estoque aumenta com a quantidade produzida. Quando diversos itens distintos compartilham o processamento em uma máquina, incorre-se em um tempo e/ou custo de preparação para as operações necessárias de um dado item. Se os estados de preparação são distintos para cada item, é necessário determinar a seqüência de produção desses itens. Uma tendência recente na literatura consiste no tratamento simultâneo do planejamento e da programação da produção em cada período (tipicamente, uma semana) do horizonte de planejamento.

O nível operacional controla as atividades diárias das ordens de produção provenientes do nível tático. As principais decisões envolvidas nesse nível são: designação de tarefas (*jobs*) a máquinas e programação (*scheduling*) das tarefas em cada máquina, isto é, a seqüência de processamento das tarefas e o instante de início e término do processamento de cada tarefa.

### 3.6.1 Planejamento da produção

A seguir, são apresentados alguns modelos importantes de planejamento da produção conhecidos na literatura como modelos de dimensionamento de lotes (*lot sizing*), brevemente discutidos na Seção 2.2.3. Os modelos apresentados possuem as seguintes características comuns. O horizonte de planejamento é finito e dividido em períodos, e a demanda de cada item em cada período é dinâmica, isto é, varia ao longo do horizonte. A demanda e outros parâmetros dos modelos são supostos conhecidos, isto é, trata-se de modelos determinísticos. Existem outros modelos em que o horizonte pode ser infinito, a variável tempo é contínua, a demanda é expressa como uma taxa em relação ao tempo, e ainda modelos em que a demanda é estocástica.<sup>37</sup> Para enunciar os modelos matemáticos, considere os seguintes parâmetros:

- $d_{it}$  = demanda do item  $i$  no período  $t$
- $b_i$  = tempo para produzir uma unidade do item  $i$
- $C_t$  = capacidade de produção em horas de uma máquina ou instalação no período  $t$
- $sp_i$  = tempo de preparação de máquina para processar o item  $i$
- $s_i$  = custo de preparação do item  $i$
- $h_i$  = custo unitário de estoque do item  $i$
- $I_{i0}$  = estoque inicial do item  $i$
- $n$  = número de itens finais
- $T$  = número de períodos do horizonte de planejamento

Considere também as seguintes variáveis:

- $x_{it}$  = quantidade do item  $i$  produzida no período  $t$  (tamanho do lote)
- $I_{it}$  = estoque do item  $i$  no fim do período  $t$
- $y_{it} = \begin{cases} 1 & \text{se o item } i \text{ é produzido no período } t \\ 0 & \text{caso contrário} \end{cases}$

#### Um item sem restrição de capacidade

O problema mais simples de dimensionamento de lotes envolve um único item, sem restrição de capacidade, e seu modelo é o seguinte:

$$\min \sum_{t=1}^T (sy_t + hI_t) \quad (\text{DL1})$$

$$I_t = I_{t-1} + x_t - d_t, \quad t = 1, \dots, T, \quad I_0 = I_T = 0 \quad (\text{DL2})$$

$$x_t \leq \left( \sum_{\tau=t}^T d_{\tau} \right) y_t, \quad t = 1, \dots, T \quad (\text{DL3})$$

$$\mathbf{x} \in R_+^T, \quad \mathbf{I} \in R_+^T, \quad \mathbf{y} \in B^T \quad (\text{DL4})$$

A função objetivo (DL1) minimiza o custo total de preparação e estoque. As restrições (DL2) representam equações de balanceamento de estoque em cada período  $t$ . Se  $I_0 > 0$ , use este estoque inicial para abater demandas no horizonte, portanto, pode-se assumir, sem perda de generalidade, que  $I_0 = 0$ . Além disso, na solução ótima,  $I_T = 0$  é uma decorrência da minimização do custo de estoque. As restrições (DL3) garantem que a produção no período  $t$  é limitada superior-

<sup>37</sup> Para detalhes destes modelos, ver Johnson e Montgomery (1974), Hax e Candea (1984) e Nahmias (1995).

mente pela demanda acumulada do período  $t$  ao último período  $T$ , e que o tamanho do lote é positivo, isto é,  $x_t > 0$ , somente se há produção no período  $t$  ( $y_t = 1$ ). A restrição (DL4) indica o tipo das variáveis.

Este foi o primeiro problema com demanda dinâmica abordado na literatura por Wagner e Within em 1958, que propuseram um algoritmo de programação dinâmica apresentado na Seção 5.4.

Seja  $c_t$  o custo unitário de produção. Neste caso, a função objetivo (DL1) é modificada para

$$\min \sum_{t=1}^T (c_t x_t + s y_t + h I_t)$$

de modo que  $c_t x_t$  representa o custo variável no período  $t$ .

Suponha que o custo unitário de produção é constante ao longo do tempo, isto é,  $c_t = c$ ,  $t = 1, \dots, T$ . A equação (DL2) pode ser reescrita na forma

$$I_t = \sum_{\tau=1}^t x_{\tau} - \sum_{\tau=1}^t d_{\tau}, \quad t = 1, \dots, T$$

e como  $I_T = 0$ , tem-se, para  $t = T$ ,

$$\sum_{t=1}^T x_t = \sum_{t=1}^T d_t$$

Portanto,

$$\sum_{t=1}^T c_t x_t = c \sum_{t=1}^T x_t = c \sum_{t=1}^T d_t$$

isto é, o custo variável total é uma constante e pode ser retirado da função objetivo, resultando, assim, na função objetivo (DL1).

No caso em que a demanda pode ser atendida com atraso, atribui-se uma penalidade  $\delta$  por unidade de demanda não atendida no período  $t$ . Considere as variáveis:

$I_t^+$  = estoque no fim do período  $t$

$I_t^-$  = falta (demanda não atendida) no período  $t$

A formulação anterior é modificada da seguinte maneira:

$$\min \sum_{t=1}^T (s y_t + h I_t^+ + \delta I_t^-) \quad (\text{DL5})$$

$$I_t^+ - I_t^- = I_{t-1}^+ - I_{t-1}^- + x_t - d_t, \quad t = 1, \dots, T, \quad I_0^+ = I_0^- = 0 \quad (\text{DL6})$$

$$x_t \leq \left( \sum_{\tau=t}^T d_{\tau} \right) y_t, \quad t = 1, \dots, T \quad (\text{DL7})$$

$$\mathbf{x} \in R_+^T, \mathbf{I}^+ \in R_+^T, \mathbf{I}^- \in R_+^T, \mathbf{y} \in B^T \quad (\text{DL8})$$

A função objetivo (DL5) minimiza o custo total de preparação, estoque e demanda não atendida. As restrições de balanceamento (DL6) levam em consideração o estoque e a demanda não atendida em cada período  $t$ . As restrições (DL7) são idênticas às restrições (DL3), e a restrição (DL8) indica o tipo das variáveis. A consideração de demanda não atendida pode ser incluída nos modelos a seguir, de acordo com o exposto acima.

### Múltiplos itens e restrição de capacidade

Este problema, conhecido na literatura como *capacitated lot sizing problem*,<sup>38</sup> trata de  $n$  itens que devem ser processados em uma única máquina ou facilidade, com restrições de capacidade, e pode ser modelado da seguinte maneira:

$$\min \sum_{i=1}^n \sum_{t=1}^T (s_i y_{it} + h_i I_{it}) \quad (\text{DL9})$$

$$I_{it} = I_{i,t-1} + x_{it} - d_{it}, \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (\text{DL10})$$

$$\sum_{i=1}^n (sp_i y_{it} + b_i x_{it}) \leq C_t, \quad t = 1, \dots, T \quad (\text{DL11})$$

$$x_{it} \leq M_{it} y_{it}, \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (\text{DL12a})$$

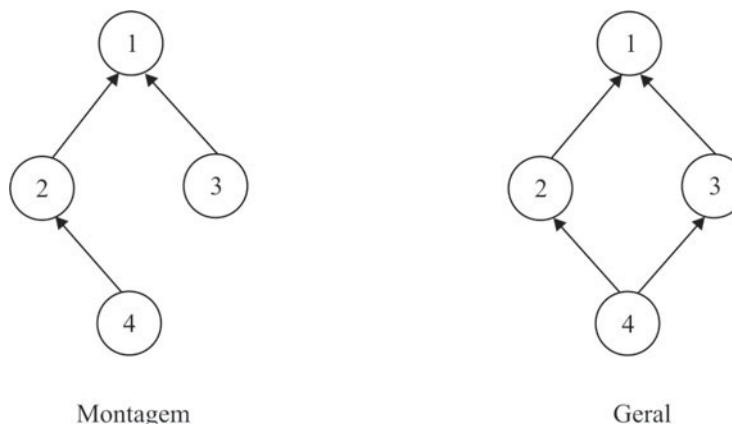
$$M_{it} = \min\left\{\frac{C_t - sp_i}{b_i}, \sum_{\tau=t}^T d_{i\tau}\right\}, \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (\text{DL12b})$$

$$\mathbf{x} \in R_+^{nT}, \quad \mathbf{I} \in R_+^{nT}, \quad \mathbf{y} \in B^{nT} \quad (\text{DL13})$$

A função objetivo (DL9) minimiza o custo total de preparação e estoque. As restrições (DL10) correspondem ao balanceamento de estoque de cada item  $i$  em cada período  $t$ . As restrições (DL11) expressam que, em cada período  $t$ , o tempo total de preparação e produção é limitado pela capacidade disponível. As restrições (DL12a) implicam que  $x_{it} > 0$  somente se  $y_{it} = 1$ . O limitante  $M_{it}$  em (DL12b) é o mínimo entre a capacidade restante no período  $t$  se  $i$  é produzido neste período, e a demanda acumulada do período  $t$  ao período  $T$ . A restrição (DL13) indica o tipo das variáveis.

### Múltiplos níveis

Um item final freqüentemente utiliza itens componentes, e a estrutura de interdependência desses itens pode ser representada por um grafo orientado. Quando um item tem um único item sucessor, diz-se que a estrutura é de montagem, e quando tem mais de um item sucessor, a estrutura é chamada geral. A Figura 3.19 ilustra exemplos dessas estruturas.



**Figura 3.19** Estruturas de produto.

<sup>38</sup> Para um modelo de fluxo em redes com resolução pelo método *branch-and-bound*, veja, por exemplo, Armentano *et al.* (1999). Para resultados de complexidade (ver Seção 3.7) deste problema, ver Bitran e Yanasse (1982).

A produção em múltiplos níveis ou estágios requer vários recursos, tais como processamento em máquinas distintas e mão-de-obra. Considere os seguintes parâmetros adicionais:

- $d_{it}$  = demanda independente do item  $i$  (final ou componente) no período  $t$
- $r_{ij}$  = número de itens  $i$  necessários para produzir uma unidade do item  $j$
- $S_i$  = conjunto de sucessores imediatos do item  $i$
- $K$  = número de recursos
- $R_k$  = conjunto de índices de itens que usam o recurso  $k$
- $b_{ki}$  = capacidade necessária do recurso  $k$  para produzir uma unidade do item  $i$
- $C_{kt}$  = capacidade do recurso  $k$  disponível no período  $t$

O modelo matemático associado a múltiplos níveis é mostrado a seguir.

$$\min \sum_{i=1}^n \sum_{t=1}^T (s_i y_{it} + h_i I_{it}) \quad (\text{DL14})$$

$$I_{it} = I_{i,t-1} + x_{it} - \sum_{j \in S_i} r_{ij} x_{jt} - d_{it}, \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (\text{DL15})$$

$$\sum_{i \in R_k} (sp_i y_{it} + b_{ki} x_{it}) \leq C_{kt}, \quad k = 1, \dots, K, \quad t = 1, \dots, T \quad (\text{DL16})$$

$$x_{it} \leq M_{it} y_{it}, \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (\text{DL17a})$$

$$M_{it} = \min \left\{ \left( \min_{k:i \in R_k} \frac{C_{kt} - sp_i}{b_{ki}} \right), \sum_{\tau=j}^T d_{i\tau} \right\}, \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (\text{DL17b})$$

$$\mathbf{x} \in R_+^{nT}, \mathbf{I} \in R_+^{nT}, \mathbf{y} \in B^{nT} \quad (\text{DL18})$$

A função objetivo (DL14) minimiza o custo total de preparação e estoque. As restrições (DL15) correspondem ao balanceamento de estoque de cada item  $i$  em cada período  $t$ , de modo que a somatória do lado direito de (DL15) corresponde à demanda dependente (produção  $x_{jt}$ ). As restrições (DL16) expressam que, em cada período  $t$  e para cada recurso  $k$ , o tempo total de preparação e produção é limitado pela capacidade disponível  $C_{kt}$ . As restrições (DL17a) garantem que  $x_{it} > 0$  somente se  $y_{it} = 1$ , e  $M_{it}$  em (DL17b) corresponde à máxima quantidade de produção do item  $i$  no período  $t$ . A restrição (DL18) indica o tipo das variáveis.

#### *Múltiplos itens, restrição de capacidade e preservação de preparação*

Os modelos anteriores (DL9)–(DL13) e (DL14)–(DL18) não consideram a utilização do tempo de preparação ( $TP$ ) de máquina para o item que é produzido em último lugar no período  $t-1$ , e em primeiro lugar no período  $t$ . Esses modelos são denominados modelos sem preservação do estado de preparação (MSPP). Neste caso, o tempo restante no período  $t-1$  para produção do lote desse item pode ser insuficiente, então a produção desse mesmo item no início do período  $t$  exige um novo tempo de preparação, o que aumenta o custo de preparação, desperdiça capacidade produtiva e, em caso de alta utilização da capacidade, pode resultar em plano de produção infactível em relação à capacidade. No modelo com preservação do estado de preparação (MCPP), o estado de preparação do último item produzido no período  $t-1$  é mantido no início do período  $t$  para continuar a produção desse item. Essas duas situações estão ilustradas na Figura 3.20. No modelo MSPP, após o tempo de preparação  $TP$  no período  $t-1$  para um novo item, uma quantidade deste é produzida e a máquina é desativada no fim desse período. No início do período  $t$ , outro tempo de preparação  $TP$  é gasto para continuar a produção do mesmo item. No

modelo MCPP, a máquina não é desativada no fim do período  $t-1$  e seu estado de preparação para a produção do mesmo item no período  $t$  é mantido.

Quando os tempos de preparação são pequenos, o modelo MSPP pode ser aceitável, mas, quando os tempos de preparação são elevados, é necessário utilizar o modelo MCPP. A preservação do tempo de preparação adiciona uma nova decisão ao problema de dimensionamento de lotes, que envolve a determinação do primeiro e do último item produzido em cada período.

A seguir, é apresentado o modelo MCPP em uma situação mais geral, em que a produção requer vários recursos.<sup>39</sup>

Defina as variáveis:

$$w_{it} = \begin{cases} 1 & \text{se o estado de preparação para o produto } i \text{ continua do período } t-1 \text{ ao período } t \\ 0 & \text{caso contrário} \end{cases}$$

$$q_{tk} = \begin{cases} 1 & \text{se não existe preparação de itens em } t, \text{ que utilizam o recurso } k, \text{ dado que a} \\ & \text{preparação de um item específico ocorreu em } t-1 \text{ e é mantida até } t+1 \\ 0 & \text{caso contrário} \end{cases}$$

$$\min \sum_{i=1}^n \sum_{t=1}^T (s_i y_{it} + h_i I_{it}) \quad (\text{DL19})$$

$$I_{it} = I_{i,t-1} + x_{it} - d_{it}, \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (\text{DL20})$$

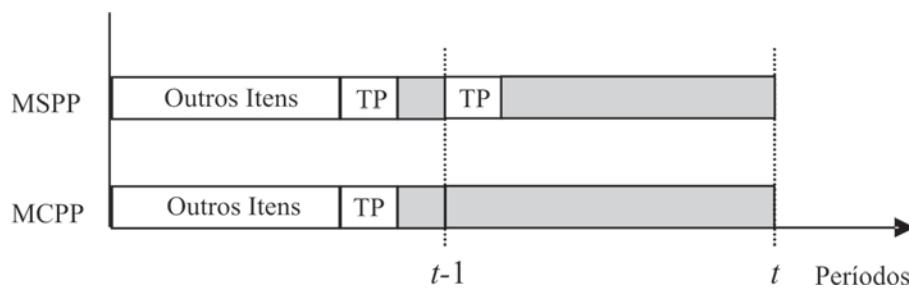
$$\sum_{i \in R_k} (sp_i y_{it} + b_{ki} x_{it}) \leq C_{kt}, \quad k = 1, \dots, K, \quad t = 1, \dots, T \quad (\text{DL21})$$

$$\sum_{i \in R_k} w_{it} \leq 1, \quad k = 1, \dots, K, \quad t = 2, \dots, T \quad (\text{DL22})$$

$$w_{it} \leq y_{i,t-1} + w_{i,t-1}, \quad i = 1, \dots, n, \quad t = 2, \dots, T \quad (\text{DL23})$$

$$w_{i,t+1} + w_{it} \leq 1 + q_{tk}, \quad k = 1, \dots, K, \quad i \in R_k, \quad t = 1, \dots, T-1 \quad (\text{DL24})$$

$$y_{it} + q_{tk} \leq 1, \quad k = 1, \dots, K, \quad i \in R_k, \quad t = 1, \dots, T \quad (\text{DL25})$$



**Figura 3.20** Produção com e sem preservação de preparação.

<sup>39</sup> Sueri e Stadtler (2003).

$$x_{it} \leq M_{it}(y_{it} + w_{it}), \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (\text{DL26})$$

$$q_{tk} \geq 0, \quad k = 1, \dots, K, \quad t = 1, \dots, T-1 \quad (q_{1k} = q_{Tk} = 0) \quad (\text{DL27})$$

$$w_{it} \in \{0, 1\}, \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (w_{il} = 0) \quad (\text{DL28})$$

$$\mathbf{x} \in R_+^{nT}, \mathbf{I} \in R_+^{nT}, \mathbf{y} \in B^{nT} \quad (\text{DL29})$$

A função objetivo (DL19) minimiza o custo total de preparação e estoque. As restrições (DL20) correspondem ao balanceamento de estoque de cada item  $i$  em cada período  $t$ . As restrições (DL21) expressam que, em cada período  $t$  e para cada recurso  $k$ , o tempo total de preparação e produção é limitado pela capacidade disponível  $C_{kt}$ . As restrições (DL22) garantem, para cada recurso  $k$ , que no máximo uma preparação é preservada de um período para o próximo. As restrições (DL23) asseguram que uma preparação é preservada no período  $t$  ( $w_{it} = 1$ ) somente se o item  $i$  foi preparado em  $t - 1$  ( $y_{i,t-1} = 1$ ) ou a preparação foi preservada do período  $t - 2$  ao período  $t - 1$  ( $w_{i,t-1} = 1$ ). As restrições (DL24) impõem que uma preparação pode ser mantida por dois períodos consecutivos  $t$  e  $t + 1$ , isto é,  $q_{tk} = 1$  se  $w_{it} = 1$  e  $w_{i,t+1} = 1$ , e as restrições (DL25) garantem que isso é possível se não existir preparação nesse período. As restrições (DL26) asseguram que a produção do item  $i$  no período  $t$  pode ocorrer se o item estiver preparado no período  $t$  ( $y_{it} = 1$ ) ou se a preparação em  $t - 1$  foi preservada ( $w_{it} = 1$ ). A produção máxima  $M_{it}$  do item  $i$  no período  $t$  é dada por (DL17b). As restrições (DL27)–(DL29) indicam o tipo das variáveis. Note que, embora as variáveis  $q_{tk}$  assumam valores 0 ou 1, elas não precisam ser definidas explicitamente como variáveis binárias.

O modelo acima pode ser facilmente transformado em um modelo para produtos com estrutura de múltiplos níveis. Basta substituir as restrições (DL20) pelas restrições (DL15).

**Exemplo 3.14** Os modelos (DL14)–(DL18) e (DL19)–(DL29) são ilustrados através do exemplo abaixo. Considere os seguintes parâmetros:

$$T = 4; n = 3; I_{i0} = 0, \quad i = 1, 2, 3; \text{ um único recurso e } R_l = \{1, 2, 3\};$$

$$\mathbf{sp} = [sp_i] = [40 \quad 40 \quad 40]$$

$$\mathbf{h} = [h_i] = [150 \quad 100 \quad 70]$$

$$\mathbf{s} = [s_i] = [350 \quad 100 \quad 90]$$

$$\mathbf{C} = [C_t] = [280 \quad 320 \quad 280 \quad 400]$$

$$\mathbf{b} = [b_i] = [20 \quad 10 \quad 20]$$

$$\mathbf{D} = [d_{it}] = \begin{bmatrix} 1 & 10 & 3 & 10 \\ 2 & 4 & 0 & 5 \\ 2 & 4 & 0 & 5 \end{bmatrix}$$

Solução ótima para o modelo com preservação da preparação:

$$y_{11} = y_{12} = 1$$

$$y_{21} = y_{24} = 1$$

$$y_{31} = y_{34} = 1$$

$$x_{11} = 1,00$$

$$x_{12} = 10,00$$

$$x_{13} = 4,50$$

$$x_{14} = 8,50$$

$$x_{21} = 2,00$$

$$x_{22} = 4,00$$

$$x_{24} = 5,00$$

$$x_{31} = 6,00$$

$$x_{34} = 5,00$$

$$\begin{aligned}w_{13} &= w_{14} = 1 \\w_{22} &= 1\end{aligned}$$

$$\begin{aligned}q_3 &= 1 \\I_{11} &= 1,50 \\I_{31} &= 4,00\end{aligned}$$

Custo ótimo: 1.585,00

A Figura 3.21 ilustra a solução ótima. O espaço em branco com o número do item corresponde a seu tempo de preparação, o espaço em cinza é associado ao processamento do item, e o espaço pontilhado corresponde à ociosidade da máquina. No período 2, por exemplo, a máquina é preparada para o item 1, que é então processado de acordo com o tamanho do lote desse item nesse período. A máquina fica ociosa, mas em estado de preparação do item 1, que é novamente processado no início do período 3. Isso mostra que ociosidade pode ser desejável para não incorrer em custo adicional de estoque.

Solução ótima para o modelo sem preservação da preparação:

$$y_{11} = y_{12} = y_{13} = y_{14} = 1$$

$$y_{21} = y_{22} = y_{24} = 1$$

$$y_{31} = y_{33} = y_{34} = 1$$

$$x_{11} = 1,00$$

$$x_{12} = 10,00$$

$$x_{13} = 3,00$$

$$x_{14} = 10,00$$

$$x_{21} = 2,00$$

$$x_{22} = 4,00$$

$$x_{24} = 5,00$$

$$x_{31} = 6,00$$

$$x_{33} = 3,50$$

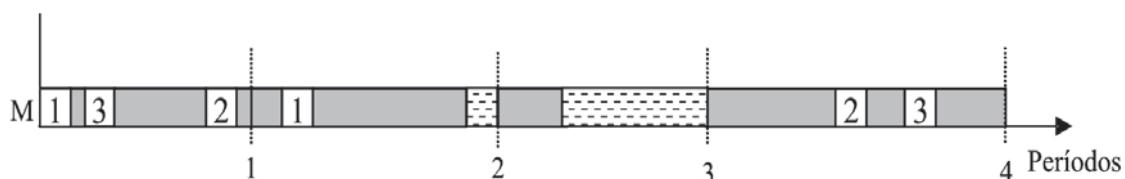
$$x_{34} = 1,5$$

$$I_{31} = 4,00$$

$$I_{33} = 3,50$$

Custo ótimo: 2.495,00

O custo ótimo sem preservação da preparação é aproximadamente 57% maior que o custo ótimo com preservação. A solução ótima é ilustrada na Figura 3.22, e mostra dez preparações de máquina contra cinco preparações na Figura 3.21. Note que no período 1, a máquina é preparada para o item 2, e novamente preparada no início do período 2. ■



**Figura 3.21** Representação da solução ótima do modelo com preservação da preparação.

### 3.6.2 Dimensionamento e programação de lotes

O tamanho do período nos problemas de dimensionamento de lotes discutidos anteriormente é grande, tipicamente uma semana, e é chamado de macroperíodo. No macroperíodo toma-se a decisão de quais itens produzir, sem a preocupação de estabelecer uma seqüência de produção destes itens. Uma tendência recente na literatura consiste em integrar as decisões de dimensionamento e programação de lotes (*lot sizing and scheduling*) em cada período. Para ilustrar esta integração, apresentamos a seguir dois problemas.

#### *Problema discreto de dimensionamento e programação de lotes*

A divisão do macroperíodo em vários microperíodos, tipicamente uma hora ou um turno, leva ao problema discreto de dimensionamento e programação de lotes (*discrete lot sizing and scheduling problem*). A hipótese fundamental deste problema é conhecida como produção “tudo-ou-nada”, isto é, somente um item é produzido por período e, se produzido, utiliza a capacidade total. A produção de um lote pode levar vários microperíodos, portanto, incorre-se um custo de preparação somente quando um novo lote é processado. Para modelar esta situação, considere uma nova variável definida como

$$z_{it} = \begin{cases} 1 & \text{se a máquina está preparada para o item } i \text{ no microperíodo } t \\ 0 & \text{caso contrário} \end{cases}$$

e o seguinte parâmetro adicional

$$z_{i0} = \begin{cases} 1 & \text{se a máquina está preparada para o item } i \text{ no início do microperíodo 1} \\ 0 & \text{caso contrário} \end{cases}$$

O problema pode, então, ser formulado como:

$$\min \sum_{i=1}^n \sum_{t=1}^T s_i y_{it} + h_i I_{it} \quad (\text{DLS1})$$

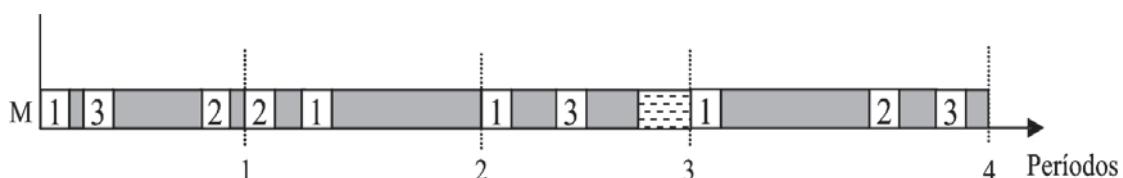
$$I_{it} = I_{i,t-1} + x_{it} - d_{it}, \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (\text{DLS2})$$

$$b_i x_{it} = C_i z_{it}, \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (\text{DLS3})$$

$$\sum_{i=1}^n z_{it} \leq 1, \quad t = 1, \dots, T \quad (\text{DLS4})$$

$$y_{it} \geq z_{it} - z_{i,t-1}, \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (\text{DLS5})$$

$$\mathbf{z} \in B^{nT}, \mathbf{I} \in R_+^{nT}, \mathbf{x} \in R_+^{nT}, \mathbf{y} \in R_+^{nT} \quad (\text{DLS6})$$



**Figura 3.22** Representação da solução ótima do modelo sem preservação da preparação.

A função objetivo (DLS1) minimiza a soma dos custos fixos de produção e custos de estoque. As restrições (DLS2) correspondem ao balanceamento de estoque para cada item  $i$  e período  $t$ . As restrições (DLS3) impõem o consumo total de capacidade do item  $i$  no microperíodo  $t$ . As restrições (DLS4) garantem que no máximo um item é produzido por microperíodo. As restrições (DLS5) indicam o início de um novo lote, isto é, se  $z_{it} = 1$  e  $z_{i,t-1} = 0$ , então  $y_{ih} = 1$ . A restrição (DLS6) indica o tipo das variáveis. Note que, embora as variáveis  $y_{ih}$  assumam valores 0 ou 1, elas não precisam ser definidas explicitamente como variáveis binárias, devido a (DLS5) e (DLS1).

#### *Problema geral de dimensionamento e programação de lotes*

O problema geral de dimensionamento e programação de lotes (*general lot sizing and scheduling problem*) considera macroperíodos  $t$ ,  $t = 1, \dots, T$ , de modo que cada período  $t$  é dividido em microperíodos de tamanho variável, sem superposição. Todos os microperíodos são seqüenciados na ordem  $h = 1, \dots, H$ . O número de microperíodos  $|H_t|$  dentro do macroperíodo  $t$  é definido *a priori* para permitir a modelagem matemática.

O tamanho do microperíodo é uma variável de decisão, expressa pela quantidade produzida no microperíodo. Uma seqüência de microperíodos em que um mesmo item é produzido define um lote, e a quantidade produzida durante esses microperíodos corresponde ao tamanho do lote. Note que os microperíodos constituem duas decisões em cada macroperíodo: a seqüência de produção dos itens e o tamanho do lote desses itens.

Considere as variáveis:

$x_{ih}$  = quantidade do item  $i$  produzida no microperíodo  $h$ ;

$$y_{ih} = \begin{cases} 1 & \text{se o item } i \text{ é produzido no microperíodo } h \\ 0 & \text{caso contrário} \end{cases}$$

$$z_{ih} = \begin{cases} 1 & \text{se a máquina está preparada para o item } i \text{ no microperíodo } h \\ 0 & \text{caso contrário} \end{cases}$$

Este problema é formulado da seguinte maneira:

$$\min \sum_{i=1}^n \sum_{h=1}^{|H_t|} s_i y_{ih} + \sum_{i=1}^n \sum_{t=1}^T h_i I_{it} \quad (\text{DLS7})$$

$$I_{it} = I_{i,t-1} + \sum_{h=1}^{|H_t|} x_{ih} - d_{it}, \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (\text{DLS8})$$

$$b_i x_{ih} \leq C_t z_{ih}, \quad i = 1, \dots, n, \quad t = 1, \dots, T, \quad h = 1, \dots, |H_t| \quad (\text{DLS9})$$

$$\sum_{i=1}^n \sum_{h=1}^{|H_t|} b_i x_{ih} \leq C_t, \quad t = 1, \dots, T \quad (\text{DLS10})$$

$$\sum_{i=1}^n z_{ih} \leq 1, \quad h = 1, \dots, H \quad (\text{DLS11})$$

$$y_{ih} \geq z_{ih} - z_{i,h-1}, \quad i = 1, \dots, n, \quad h = 1, \dots, H \quad (\text{DLS12})$$

$$\mathbf{z} \in B^{nH}, \quad \mathbf{I} \in R_+^{nT}, \quad \mathbf{x} \in R_+^{nH}, \quad \mathbf{y} \in R_+^{nH} \quad (\text{DLS13})$$

A função objetivo (DLS7) minimiza a soma dos custos fixos de produção e dos custos de estoque. As restrições (DLS8) correspondem ao balanceamento de estoque para cada item  $i$  e pe-

ríodo  $t$ . As restrições (DLS9) asseguram que, se a máquina está preparada para o item  $i$  no microperíodo  $h$  do período  $t$ , então esse item pode ser produzido. As restrições de capacidade em cada período  $t$  estão representadas em (DLS10). As restrições (DLS11) forçam um único estado de preparação de máquina, e as restrições (DLS12) indicam o início de um novo lote. A restrição (DLS13) indica o tipo das variáveis.

Podemos, ainda, ter modelos que considerem que os tempos e custos de preparação são dependentes da seqüência de produção. Por exemplo, em uma linha de envasamento de bebidas, o tempo de preparação para a produção de um lote de refrigerante *diet* depois da produção de um lote de refrigerante com açúcar é bem diferente do tempo de preparação para a produção da seqüência inversa.<sup>40</sup>

### 3.6.3 Programação da produção

Como já destacado, as principais decisões envolvidas no nível operacional são: designação de tarefas (*jobs*) a máquinas e a programação (*scheduling*) das tarefas em cada máquina, isto é, a seqüência de processamento das tarefas e o instante de início e término do processamento de cada tarefa. A literatura nesta área é extensa devido à enorme variedade dos ambientes de produção, das características dos produtos e das medidas de desempenho de programas de produção. Novas características no ambiente de produção foram introduzidas com sistemas flexíveis de manufatura. Esse tipo de sistema é um conjunto integrado, controlado por computadores, de dispositivos automatizados de manuseio de materiais e máquinas de controle numérico que podem processar volumes médios de uma variedade de tipos de partes. A meta desse sistema é atingir a eficiência da produção automatizada de grande escala, utilizando a flexibilidade de um *job shop* manual para processar simultaneamente diversos tipos de partes.<sup>41</sup>

Com relação à avaliação da qualidade de um programa de produção, as principais medidas de desempenho são: *makespan*, tempo de fluxo total, atraso máximo, atraso total, *lateness* e número de tarefas atrasadas. O *makespan* é o instante de término de processamento de todas as tarefas a serem processadas e é uma medida da utilização do sistema de produção. O tempo de fluxo total é a soma dos instantes de término de processamento das tarefas, que mede o estoque em processamento. Todas as demais medidas citadas estão relacionadas às datas de entrega das tarefas. Tarefas que terminam atrasadas são penalizadas, ao passo que tarefas que terminam antes da data de entrega podem incorrer em custo de estoque. De forma geral, essas medidas são conflitantes e a otimização de mais de uma medida deve levar em consideração o *trade-off* caracterizado pela otimização com múltiplos objetivos, em que a melhoria segundo um critério causa uma deterioração em outro critério.

A seguir são apresentados alguns modelos matemáticos para ilustrar a vasta área de programação da produção.

#### *Uma máquina*

Considere  $n$  tarefas a serem processadas em uma máquina. Todas as tarefas estão disponíveis para processamento no instante zero e admite-se que a interrupção do processamento (*preemption*) de qualquer tarefa não é permitida. Considere os seguintes parâmetros inteiros e não-negativos:

$p_i$  = tempo de processamento da tarefa  $i$

$d_i$  = data de entrega da tarefa  $i$

$M$  = número grande

<sup>40</sup> Para detalhes de modelos de dimensionamento e programação de lotes com tempo e custo de preparação dependente da seqüência de produção, ver Drexl e Kimms (1997) e Karimi *et al.* (2003). Para exemplos de aplicação destes modelos na indústria de bebidas e nutrição animal, ver, por exemplo, Rangel e Ferreira (2003) e Toso e Morabito (2005).

<sup>41</sup> Stecke (1983).

Defina as seguintes variáveis:

$C_i$  = instante de término do processamento da tarefa  $i$

$$x_{ij} = \begin{cases} 1 & \text{se a tarefa } i \text{ precede imediatamente a tarefa } j \\ 0 & \text{caso contrário} \end{cases}$$

$$T_i = \max \{C_i - d_i, 0\} = \text{atraso da tarefa } i$$

$$E_i = \max \{d_i - C_i, 0\} = \text{avanço ou adiantamento da tarefa } i$$

$$L_i = C_i - d_i = \text{lateness da tarefa } i$$

Seja zero (0) uma tarefa fictícia que precede imediatamente a primeira tarefa e sucede imediatamente a última tarefa de uma seqüência de tarefas. A partir desses parâmetros e variáveis, é possível formular problemas com critérios distintos de otimização. As seguintes restrições são comuns a todos problemas:

$$\sum_{i=0, i \neq j}^n x_{ij} = 1, \quad j = 0, 1, \dots, n \quad (\text{S1})$$

$$\sum_{j=0, j \neq i}^n x_{ij} = 1, \quad i = 0, 1, \dots, n \quad (\text{S2})$$

$$C_j \geq C_i - M + (p_j + M)x_{ij}, \quad i = 0, \dots, n, \quad j = 1, \dots, n \quad (\text{S3})$$

$$C_i \geq 0, \quad i = 1, \dots, n, \quad C_0 = 0 \quad (\text{S4})$$

$$\mathbf{x} \in B^{(n+1)(n+1)} \quad (\text{S5})$$

As restrições (S1) e (S2) garantem que cada tarefa tem apenas uma tarefa imediatamente predecessora e uma tarefa imediatamente sucessora, respectivamente. Se  $x_{ij} = 1$ , a restrição (S3) implica que

$$C_j \geq C_i + p_j$$

e se  $x_{ij} = 0$ , tem-se que

$$C_j - C_i \geq -M$$

isto é, a restrição (S3) é desativada. As restrições (S4) e (S5) indicam o tipo das variáveis.

#### Minimização do tempo de fluxo total

O tempo de fluxo total corresponde à soma dos tempos de término das tarefas, isto é,  $\sum_{i=1}^n C_i$ . O problema é, então, expresso como

$$\min \sum_{i=1}^n C_i$$

(S1)–(S5)

Pode-se demonstrar que a solução ótima deste problema é dada pela regra SPT (*shortest processing time*), em que as tarefas são seqüenciadas em ordem não decrescente dos tempos de processamento, isto é, as tarefas são processadas na seqüência: [1], [2], ..., [n], tal que  $p_{[1]} \leq p_{[2]} \leq \dots \leq p_{[n]}$  e [i] corresponde à tarefa na  $i$ -ésima posição. Por exemplo,  $p_{[2]}$  é o tempo de processamento da tarefa que ocupa a segunda posição na seqüência de produção.<sup>42</sup>

#### Minimização do atraso máximo

O atraso máximo, representado por  $T_{\max}$ , está associado à tarefa com maior diferença entre seu instante de término e data de entrega, isto é,  $T_{\max} = \max_n T_i$ . Note que  $T_i = \max \{C_i - d_i, 0\}$  implica  $T_i \geq C_i - d_i$  e  $T_i \geq 0$ . O problema da minimização de  $T_{\max}$  pode, então, ser formulado como:

<sup>42</sup> Este resultado ainda é válido mesmo que os tempos de processamento das tarefas sejam variáveis aleatórias; ver Seção 7.8.

$$\begin{aligned}
 & \min T_{\max} \\
 & T_{\max} \geq T_i, \quad i = 1, \dots, n \\
 & T_i \geq C_i - d_i, \quad i = 1, \dots, n \\
 & \mathbf{T} \in R_+^n \\
 & \text{(S1)}-\text{(S5)}
 \end{aligned}$$

Demonstra-se que a solução ótima deste problema é dada pela regra EDD (*earliest due date*), que consiste em seqüenciar as tarefas em ordem não decrescente das datas de entrega, isto é, as tarefas são processadas na seqüência: [1], [2], ..., [n], tal que  $d_{[1]} \leq d_{[2]} \leq \dots \leq d_{[n]}$ .

#### Minimização da soma dos atrasos

Este problema é modelado como

$$\begin{aligned}
 & \min \sum_{i=1}^n T_i \\
 & T_i \geq C_i - d_i, \quad i = 1, \dots, n \\
 & \mathbf{T} \in R_+^n \\
 & \text{(S1)}-\text{(S5)}
 \end{aligned}$$

#### Minimização da soma dos atrasos e avanços

A formulação deste problema é dada por

$$\begin{aligned}
 & \min \sum_{i=1}^n (T_i + E_i) \\
 & T_i \geq C_i - d_i, \quad i = 1, \dots, n \\
 & E_i \geq d_i - C_i, \quad i = 1, \dots, n \\
 & \mathbf{T} \in R_+^n, \mathbf{E} \in R_+^n \\
 & \text{(S1)}-\text{(S5)}
 \end{aligned}$$

#### Minimização do número de tarefas atrasadas

Defina as seguintes variáveis:

$$y_i = \begin{cases} 1 & \text{se a tarefa } i \text{ está atrasada} \\ 0 & \text{caso contrário} \end{cases}$$

A formulação do problema é dada por

$$\begin{aligned}
 & \min \sum_{i=1}^n y_i \\
 & T_i \geq C_i - d_i \quad i = 1, \dots, n \\
 & T_i \leq M y_i, \quad i = 1, \dots, n \\
 & \mathbf{T} \in R_+^n, \mathbf{y} \in B^n \\
 & \text{(S1)}-\text{(S5)}
 \end{aligned}$$

Note que, se  $T_i > 0$ , então  $y_i = 1$ .

## Minimização do lateness máximo

Seja  $L_{\max} = \max_{i=1}^n L_i$  o lateness máximo e, como a variável  $L_i$  é livre, defina  $L_i = L_i^+ - L_i^-$ . Desta maneira, o problema é representado pelo seguinte modelo

$$\begin{aligned} \min L_{\max} \\ L_{\max} &\geq L_i^+ - L_i^- \quad i = 1, \dots, n \\ L_i^+ - L_i^- &= C_i - d_i \quad i = 1, \dots, n \\ \mathbf{L}^+ &\in R_+^n, \mathbf{L}^- \in R_+^n \\ (\text{S1})-(\text{S5}) \end{aligned}$$

Demonstra-se que a solução ótima deste problema também é dada pela regra EDD.

**Exemplo 3.15** O exemplo a seguir ilustra a minimização das medidas de desempenho citadas anteriormente. Considere os parâmetros:

$$\mathbf{p} = [p_i] = [64 \quad 53 \quad 63 \quad 99 \quad 189 \quad 44 \quad 50 \quad 22]$$

$$\mathbf{d} = [d_i] = [100 \quad 70 \quad 150 \quad 601 \quad 118 \quad 590 \quad 107 \quad 180]$$

Apresentam-se, a seguir, a solução ótima e seu valor para cada critério.

Fluxo total	Valor 1880
Seqüência	8    6    7    2    3    1    4    5
Instante de término	22    66    116    169    232    296    395    584

Atraso máximo	Valor 269
Seqüência	5    2    7    1    3    8    6    4
Instante de término	189    242    292    354    419    441    485    584
Atraso	71    172    185    254    269    261    0    0

Soma dos atrasos	Valor 499
Seqüência	2    7    3    8    1    5    6    4
Instante de término	53    103    166    188    252    441    485    584
Atraso	0    0    16    8    152    323    0    0

Soma dos atrasos e avanços	Valor 575
Seqüência	2    7    3    8    1    5    4    6
Instante de término	53    103    166    188    252    441    546    590
Atraso	0    0    16    8    152    323    0    0
Avanço	17    4    0    0    0    0    55    0

Número de tarefas atrasadas	Valor 3
Seqüência	8    7    3    6    4    5    1    2
Instante de término	22    72    135    179    278    467    531    584
Atraso	0    0    0    0    0    349    431    514

<i>Lateness</i> máximo	Valor 269
Seqüência	1    7    2    5    3    8    4    6
Instante de término	64    114    167    356    419    441    540    584
<i>Lateness</i>	-36    7    97    238    269    261    -61    -6

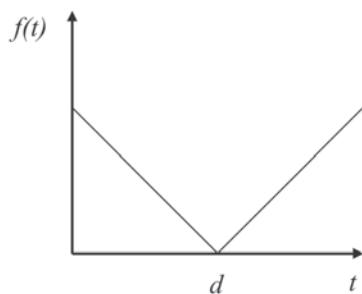
Note a existência de uma seqüência ótima alternativa que minimiza o atraso máximo  $T_{\max}$  e o *lateness* máximo  $L_{\max}$  em relação à seqüência EDD dada por  $2 - 1 - 7 - 5 - 3 - 8 - 6 - 4$ . A seqüência encontrada para a minimização do tempo de fluxo total é a seqüência SPT.

No critério de soma de atrasos e avanços, a máquina fica ociosa entre o término do processamento da tarefa 5 (instante 441) e o início do processamento da tarefa 4 (instante 447). Isso ocorre porque a data de entrega da tarefa 4 é 601, e o início do processamento desta tarefa no instante 441 aumentaria seu custo de avanço em seis unidades. Note, também, que a tarefa 6 termina na sua data de entrega (590), portanto, sua antecipação também acarretaria em aumento do custo de avanço. A medida da soma de atrasos e avanços  $f(t)$  como função do instante de término do processamento  $t$  para uma dada tarefa com data de entrega  $d$  é mostrada na Figura 3.23. Essa função decresce linearmente até o instante  $d$  (o custo de avanço diminui) e a partir daí cresce linearmente (o custo de atraso aumenta).

Pode-se demonstrar que as demais medidas são não decrescentes com o acréscimo do instante de término de processamento de todas as tarefas.<sup>43</sup> Intuitivamente, isso implica que, para as demais medidas, a máquina não fica ociosa, pois uma antecipação no início do processamento de uma tarefa não aumenta a medida. Um aspecto importante a ser destacado é o conflito entre objetivos, isto é, uma solução ótima para um objetivo, em geral, não é a solução ótima para outro objetivo.<sup>44</sup>

Considere agora que um tempo de preparação de máquina é necessário para processar a tarefa  $i$ . Se o tempo de preparação é independente da seqüência, ele é denotado por  $s_i$ , e basta definir um tempo para a execução da tarefa  $i$  como  $s_i + p_i$ . Quando o tempo de preparação é dependente da seqüência, considere a seguinte notação:

$s_{ij} = \text{tempo de preparação da máquina para processar a tarefa } j \text{ imediatamente após a tarefa } i$



**Figura 3.23** Medida de soma de avanços e atrasos para uma tarefa.

43 Ver, por exemplo, French (1982).

44 Ver Armentano e Arroyo (2004) para uma heurística para problemas com múltiplos objetivos; ver também Friedlander *et al.* (1990).

No caso de tempo de preparação independente da seqüência, a restrição (S3) é alterada para

$$C_j \geq C_i - M + (s_j + p_j + M)x_{ij}, \quad i = 0, \dots, n \quad j = 1, \dots, n \quad (S6)$$

e no caso de tempo de preparação dependente da seqüência, a restrição (S3) torna-se

$$C_j \geq C_i - M + (s_{ij} + p_j + M)x_{ij}, \quad i = 0, \dots, n \quad j = 1, \dots, n \quad (S7)$$

Todas as formulações anteriores continuam válidas ao substituir-se (S3) por (S6) ou (S7). No caso de tempo de preparação dependente da seqüência, o problema de minimização do *makespan*, representado por  $\underset{n}{\max} C_{\max}$ , consiste no tempo total para processar todas as tarefas, isto é,

$C_{\max} = \max_{i=1}^n C_i$  e pode ser formulado como:

$$\begin{aligned} & \min C_{\max} \\ & C_{\max} \geq C_i, \quad i = 1, \dots, n \end{aligned} \quad (S1)-(S2)-(S7)-(S4)-(S5)$$

Considere a notação:

$s_{[k],[k+1]}$  = tempo de preparação da tarefa na posição  $k + 1$  quando sucede a tarefa na posição  $k$

Por exemplo,  $s_{[3],[4]}$  é o tempo de preparação para produzir a tarefa que ocupa a quarta posição na seqüência de produção, logo após a produção da tarefa que ocupa a terceira posição. Então, o *makespan* pode ser expresso como

$$C_{\max} = \sum_{k=1}^n s_{[k-1],[k]} + s_{[n],[0]} + \sum_{i=1}^n p_i$$

Como a segunda somatória é constante, então o problema do *makespan* corresponde ao problema do caixeiro-viajante que começa na cidade 0 com distância  $s_{ij}$  entre as cidades  $i$  e  $j$ . Quando o tempo de preparação é zero, ou quando este não depende da seqüência, este problema não existe, pois todas as seqüências geram o mesmo valor de *makespan*.

### Máquinas paralelas

Existem três categorias de ambientes de máquinas paralelas: idênticas, uniformes e não relacionadas. Em máquinas idênticas, cada tarefa tem o mesmo tempo de processamento e o mesmo tempo de preparação. Em máquinas uniformes, o tempo de processamento de uma tarefa e o tempo de preparação de máquinas mais modernas são proporcionais aos tempos correspondentes à máquina mais antiga. Em máquinas não relacionadas, não existe uma relação entre os tempos de processamento e preparação de máquinas distintas. Os modelos a seguir referem-se a  $m$  máquinas paralelas e  $n$  tarefas disponíveis para processamento no instante zero, sem interrupção de processamento de qualquer tarefa.

Para o problema de minimização do *makespan* em máquinas idênticas, seja

$p_i$  = tempo de processamento da tarefa  $i$ ,  $i = 1, \dots, n$

Considere as seguintes variáveis

$$\begin{aligned} C_{\max} &= \text{makespan} \\ x_{ij} &= \begin{cases} 1 & \text{se a tarefa } i \text{ é processada na máquina } j \\ 0 & \text{caso contrário} \end{cases} \end{aligned}$$

e o seguinte modelo:

$$\min C_{\max} \quad (S8)$$

$$\sum_{j=1}^m x_{ij} = 1, \quad i = 1, \dots, n \quad (S9)$$

$$C_{\max} \geq \sum_{i=1}^n p_i x_{ij}, \quad j = 1, \dots, m \quad (S10)$$

$$C_{\max} \geq 0, \mathbf{x} \in B^{mn} \quad (S11)$$

A função objetivo (S8) representa a minimização do *makespan*. As restrições (S9) asseguram que uma tarefa  $i$  é designada a exatamente uma máquina, enquanto as restrições (S10) impõem que o *makespan* é o maior tempo de processamento entre todas as máquinas. A restrição (S11) indica o tipo das variáveis.<sup>45</sup>

A seguir, apresenta-se um modelo para minimização da soma dos avanços e atrasos que pode ser aplicado a qualquer dos três tipos de máquinas anteriormente mencionados. Sejam os parâmetros inteiros não-negativos:

$p_{ik}$  = tempo de processamento da tarefa  $i$  na máquina  $k$

$s_{ijk}$  = tempo de preparação da máquina  $k$  para processar a tarefa  $j$  imediatamente após a tarefa  $i$

$d_i$  = data de entrega da tarefa  $i$

$M$  = número grande

Defina as seguintes variáveis

$C_{ik}$  = instante de término do processamento da tarefa  $i$  na máquina  $k$

$$x_{ijk} = \begin{cases} 1 & \text{se a tarefa } i \text{ precede imediatamente a tarefa } j \text{ na máquina } k \\ 0 & \text{caso contrário} \end{cases}$$

$$T_i = \max \{C_i - d_i, 0\} = \text{atraso da tarefa } i$$

$$E_i = \max \{d_i - C_i, 0\} = \text{avanço da tarefa } i$$

e considere a seguinte formulação:

$$\min \sum_{i=1}^n (T_i + E_i) \quad (S12)$$

$$\sum_{k=1}^m \sum_{i=0}^n x_{ijk} = 1, \quad j = 1, \dots, n \quad (S13)$$

$$\sum_{j=1}^n x_{0jk} \leq 1, \quad k = 1, \dots, m \quad (S14)$$

$$\sum_{\substack{i=0 \\ i \neq h}}^n x_{ihk} - \sum_{\substack{j=0 \\ j \neq h}}^n x_{hjk} = 0, \quad h = 1, \dots, n, \quad k = 1, \dots, m \quad (S15)$$

$$C_{jk} \geq C_{ik} - M + (s_{ijk} + p_{jk} + M)x_{ijk}, \quad i = 0, \dots, n, \quad j = 1, \dots, n \quad (S16)$$

---

<sup>45</sup> Ver França *et al.* (1996) para uma heurística em máquinas paralelas idênticas.

$$T_i \geq C_i - d_i \quad i = 1, \dots, n \quad (\text{S17})$$

$$E_i \geq d_i - C_i \quad i = 1, \dots, n \quad (\text{S18})$$

$$\mathbf{T} \in R_+^n, \mathbf{E} \in R_+^n, \mathbf{x} \in B^{m(n+1)(n+1)} \quad (\text{S19})$$

A função objetivo (S12) expressa a minimização da soma total dos atrasos e avanços das tarefas. As restrições (S13) impõem que cada tarefa  $j$  tem uma única tarefa predecessora imediata em uma única máquina. As restrições (S14) garantem que cada máquina  $k$ , se usada, tem uma única seqüência de processamento. As restrições (S15) asseguram que cada tarefa  $j$  tem uma única tarefa sucessora imediata, com exceção da tarefa 0, que estabelece o início e o final da seqüência de processamento em uma máquina  $k$ . Se  $x_{ijk} = 1$ , a restrição (S16) implica que, na máquina  $k$ ,

$$C_{jk} \geq C_{ik} + s_{ijk} + p_{jk}$$

e, se  $x_{ijk} = 0$ , tem-se que

$$C_{jk} - C_{ik} \geq -M$$

isto é, a restrição (S16) fica desativada. A restrição (S19) indica o tipo das variáveis.

Modelos para outros critérios de otimização, como vistos para uma máquina, podem ser facilmente obtidos por modificações na formulação (S12)–(S19).

### *Job shop*

Um *job shop* clássico é um ambiente de produção com  $n$  tarefas e  $m$  máquinas, em que cada tarefa é processada nas  $m$  máquinas, de acordo com um roteiro preestabelecido. Considere, por exemplo, 5 tarefas e 3 máquinas, denotadas por 1, 2 e 3. A Figura 3.24 mostra um exemplo com a matriz de operações de cada tarefa  $O$  e a matriz de tempos de processamento nas máquinas  $P$ . Por exemplo, a primeira linha da matriz  $O$  indica que a tarefa 1 é processada nas máquinas 2-1-3, nesta ordem, com tempos de processamento de 5, 7, e 10, respectivamente, correspondentes aos elementos da primeira linha da matriz  $P$ .<sup>46</sup>

Admita que as  $n$  tarefas estão disponíveis para processamento no instante zero e que a interrupção do processamento de qualquer tarefa não é permitida. Considere os parâmetros:

$p_{ik}$  = tempo de processamento da tarefa  $i$  na máquina  $k$

$i(1), \dots, i(m)$  = roteiro de processamento da tarefa  $i$  nas  $m$  máquinas, isto é, a seqüência de máquinas em que as operações dessa tarefa são processadas

$M$  = número grande

$$\mathbf{O} = \begin{bmatrix} 2 & 1 & 3 \\ 1 & 2 & 3 \\ 3 & 2 & 1 \\ 2 & 1 & 3 \\ 3 & 1 & 2 \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} 5 & 7 & 10 \\ 9 & 5 & 3 \\ 5 & 8 & 2 \\ 2 & 7 & 4 \\ 8 & 8 & 8 \end{bmatrix}$$

**Figura 3.24** Matrizes de operações e tempos de processamento em *job shop*.

<sup>46</sup> O caso particular de tarefas com tempo unitário é considerado em Szwarcfiter (1987); ver também discussão em Santoro e Pacheco (2000).

Defina as variáveis

$C_{ik}$  = instante de término do processamento da tarefa  $i$  na máquina  $k$

$$x_{ijk} = \begin{cases} 1 & \text{se a tarefa } i \text{ precede a tarefa } j \text{ na máquina } k \\ 0 & \text{caso contrário} \end{cases}$$

e considere o seguinte modelo:

$$\min \sum_{i=1}^n C_{i(m)} \quad (S20)$$

$$C_{i,i(1)} \geq p_{i,i(1)}, \quad i = 1, \dots, n \quad (S21)$$

$$C_{i,i(k+1)} \geq C_{i(k)} + p_{i,i(k+1)}, \quad i = 1, \dots, n, \quad k = 1, \dots, m-1 \quad (S22)$$

$$C_{jk} \geq C_{ik} + p_{jk} - M(1 - x_{ijk}), \quad i = 1, \dots, n, \quad j = 1, \dots, n, \quad k = 1, \dots, m \quad (S23)$$

$$C_{ik} \geq C_{jk} + p_{ik} - Mx_{ijk}, \quad i = 1, \dots, n, \quad j = 1, \dots, n, \quad k = 1, \dots, m \quad (S24)$$

$$\mathbf{C} \in R_+^{mn}, \mathbf{x} \in B^{mn} \quad (S25)$$

A função objetivo (S20) expressa a minimização do tempo de fluxo total das tarefas. As restrições (S21) garantem que a primeira operação de cada tarefa  $i$  é completada após o respectivo tempo de processamento. As restrições (S22) impõem que a operação  $k + 1$  da tarefa  $i$  é concluída depois do término da operação  $k$  e do tempo de processamento da operação  $k + 1$ . As restrições (S23) e (S24) são restrições disjuntivas que indicam se, na máquina  $k$ , a tarefa  $i$  precede a tarefa  $j$ , ou a tarefa  $j$  precede a tarefa  $i$ . Se  $x_{ijk} = 1$ , então, de (S23) e (S24), tem-se que

$$C_{jk} \geq C_{ik} + p_{jk}$$

$$C_{ik} - C_{jk} \geq p_{ik} - M$$

isto é, a restrição (S24) é desativada. De modo análogo, se  $x_{ijk} = 0$ , então

$$C_{jk} - C_{ik} \geq p_{jk} - M$$

$$C_{ik} \geq C_{jk} + p_{ik}$$

isto é, a restrição (S23) é desativada. A restrição (S25) estabelece o tipo das variáveis.

Variantes do modelo acima para outros critérios de otimização<sup>47</sup> podem ser obtidas por modificações adequadas na formulação (S20)–(S25).

**Exemplo 3.16** Para as matrizes de roteiro e de processamento da Figura 3.24, a solução ótima é dada por:

$$\begin{aligned} x_{131} &= x_{151} = x_{211} = x_{231} = x_{241} = x_{251} = x_{411} = x_{431} = x_{451} = x_{531} = 1 \\ x_{122} &= x_{132} = x_{152} = x_{232} = x_{252} = x_{352} = x_{412} = x_{422} = x_{432} = x_{452} = 1 \\ x_{213} &= x_{313} = x_{323} = x_{343} = x_{413} = x_{423} = x_{513} = x_{523} = x_{533} = x_{543} = 1 \end{aligned}$$

$$\begin{array}{lll} C_{12} = 14 & C_{11} = 19 & C_{13} = 32 \\ C_{21} = 9 & C_{22} = 19 & C_{23} = 22 \\ C_{33} = 11 & C_{32} = 27 & C_{31} = 32 \\ C_{42} = 7 & C_{41} = 11 & C_{43} = 15 \\ C_{53} = 8 & C_{51} = 27 & C_{52} = 35 \end{array}$$

<sup>47</sup> Ver Armentano e Scrich (2000) para uma heurística para minimização do atraso total em *job shop*.

Valor do fluxo ótimo:  $C_{13} + C_{23} + C_{31} + C_{43} + C_{52} = 136$

A representação gráfica das operações em máquinas ao longo do tempo, conhecida como diagrama de Gantt, é mostrada na Figura 3.25. ■

### Flow shop

O ambiente de produção *flow shop* é um caso particular do ambiente *job shop*, em que as  $n$  tarefas têm o mesmo roteiro nas  $m$  máquinas. Quando a seqüência das tarefas é a mesma em todas as máquinas, tem-se um *flow shop* permutacional. O modelo do *job shop* pode ser adaptado de forma a gerar um modelo para o *flow shop*. No entanto, esta formulação é fraca<sup>48</sup> para obtenção da solução ótima por pacotes de otimização e, por esse motivo, apresenta-se uma formulação mais adequada.

Seja  $j$  o índice que indica a posição na seqüência de tarefas, e considere o parâmetro:

$p_{ik}$  = tempo de processamento da tarefa  $i$  na máquina  $k$

Defina as variáveis

$s_{kj}$  = instante de início de processamento da tarefa na posição  $j$  na máquina  $k$

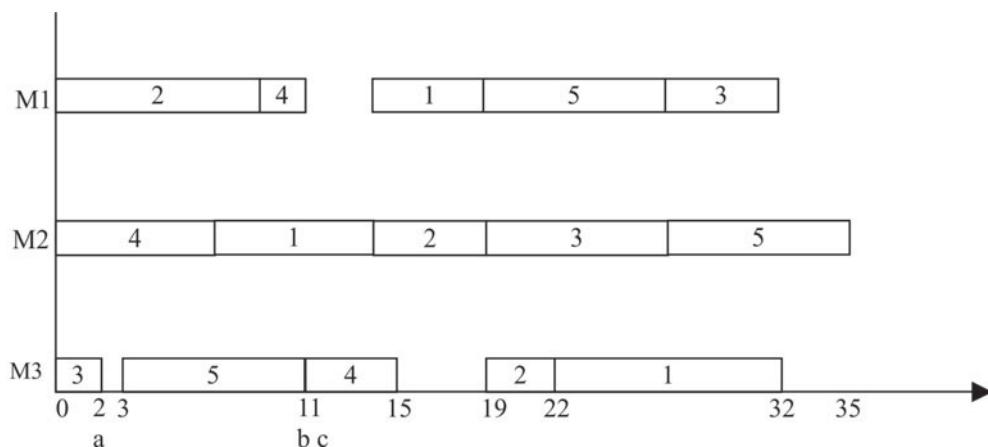
$$z_{ij} = \begin{cases} 1 & \text{se a tarefa } i \text{ é designada à posição } j \\ 0 & \text{caso contrário} \end{cases}$$

e considere o seguinte modelo

$$\min C_{\max} = s_{mn} + \sum_{i=1}^n p_{im} z_{in} \quad (\text{S26})$$

$$\sum_{j=1}^n z_{ij} = 1, \quad i = 1, \dots, n \quad (\text{S27})$$

$$\sum_{i=1}^n z_{ij} = 1, \quad j = 1, \dots, n \quad (\text{S28})$$



**Figura 3.25** Diagrama de Gantt para exemplo de *job shop*.

<sup>48</sup> Stafford Jr. et al. (2005).

$$s_{1j} + \sum_{i=1}^n p_{i1} z_{ij} = s_{1,j+1}, \quad j = 1, \dots, n-1 \quad (\text{S29})$$

$$s_{11} = 0 \quad (\text{S30})$$

$$s_{k1} + \sum_{i=1}^n p_{ik} z_{il} = s_{k+1,1}, \quad k = 1, \dots, m-1 \quad (\text{S31})$$

$$s_{kj} + \sum_{i=1}^n p_{ik} z_{ij} \leq s_{k+1,j}, \quad j = 2, \dots, n, \quad k = 1, \dots, m-1 \quad (\text{S32})$$

$$s_{kj} + \sum_{i=1}^n p_{ik} z_{ij} \leq s_{k,j+1}, \quad j = 1, \dots, n-1, \quad k = 2, \dots, m \quad (\text{S33})$$

$$\mathbf{s} \in R_+^{nm}, \mathbf{z} \in B^{nn} \quad (\text{S34})$$

A função objetivo (S26) corresponde à minimização do *makespan*. As restrições (S27) garantem que cada tarefa  $i$  está associada a uma única posição, e as restrições (S28) asseguram que cada posição  $j$  está associada a uma única tarefa. As restrições (S29) forçam a tarefa na posição  $j$  a iniciar seu processamento na máquina 1 depois que sua tarefa predecessora tenha sido processada nesta máquina. A restrição (S30) estabelece que a primeira tarefa da seqüência comece seu processamento na máquina 1 no instante 0. As restrições (S31) garantem que a primeira tarefa na seqüência seja processada imediatamente na próxima máquina  $k+1$ , desde que seu processamento na máquina corrente  $k$  tenha sido completado. As restrições (S32) asseguram que uma tarefa na posição  $j$  não pode ser iniciada na próxima máquina  $k+1$  antes do término do seu processamento na máquina corrente  $k$ . As restrições (S33) garantem que uma tarefa na posição  $j+1$  não pode iniciar em uma máquina  $k$  antes que o processamento da tarefa na posição  $j$  na mesma máquina  $k$  tenha sido completado. A restrição (S34) indica o tipo das variáveis.<sup>49</sup>

Seja

$$C_l = s_{lm} + \sum_{i=1}^n p_{im} z_{il}, \quad l = 1, \dots, n$$

o instante de término do processamento na posição  $l$  na máquina  $m$ , e  $d_l$ , a correspondente data de entrega. Então

$$T_l = \max\{C_l - d_l, 0\}, \quad l = 1, \dots, n$$

é o atraso da tarefa na posição  $l$ . Assim, o fluxo total e o atraso total podem ser expressos como

$$\sum_{l=1}^n C_l \text{ e } \sum_{l=1}^n T_l.$$

Existem diversas variantes dos problemas básicos de *job shop* (*flow shop*) modelados anteriormente que são descritas a seguir.

- Job shop com número distinto de operações por tarefa

Este é o caso em que cada tarefa é processada por um subconjunto de  $m$  máquinas disponíveis.

- Job shop com interrupção (preemption)

Em todos os modelos anteriores, assumimos que o processamento de uma tarefa não pode ser interrompido. No entanto, existem situações, principalmente em programação de tarefas em computadores, em que uma tarefa com maior prioridade torna-se disponível para processamento. Nesse caso, a tarefa menos importante tem seu processamento interrompido.

<sup>49</sup> Uma heurística para minimização do *makespan* é proposta por Moccellin e Santos (2000); ver também Nagano e Moccellin (2002). O caso particular de tarefas com tempos unitários é considerado em Blazewicz *et al.* (1988).

- Job shop com instantes de disponibilidade distintos

Os modelos anteriores também pressupõem que todas as tarefas estão disponíveis para processamento no mesmo instante. No entanto, há situações em que esses instantes de disponibilidade (*ready times*) são distintos.

- Job shop flexível

Uma operação pode ser executada por qualquer máquina de um dado conjunto. Um *flow shop* flexível é caracterizado por estágios, cada estágio com máquinas paralelas. Esses ambientes flexíveis de produção permitem vários roteiros de processamento de uma tarefa.

- Job shop com tempos de preparação dependentes da seqüência

A extensão das formulações dos problemas de *job shop* e *flow shop* apresentadas anteriormente para a consideração de tempos de preparação dependentes da seqüência não é imediata.<sup>50</sup>

- Job shop e flow shop com buffers

Tarefas que esperam para serem processadas em uma máquina situam-se, em geral, em um *buffer* de capacidade limitada.<sup>51</sup> Nos modelos anteriores de *job shop* e *flow shop*, assume-se que a capacidade é ilimitada. Duas variantes adicionais são importantes em indústrias de processamento: bloqueio (*blocking*) e sem espera (*no-wait*).<sup>52</sup> No *flow shop* com bloqueio, não existem *buffers* entre máquinas. O término da operação de uma tarefa em uma máquina bloqueia esta máquina até que a máquina seguinte esteja disponível para processamento.<sup>53</sup> O *flow shop* sem espera não permite *buffers* entre máquinas. Quando uma tarefa começa seu processamento na primeira máquina, deve ser processada continuamente até a última máquina, sem interrupção. Neste caso, o bloqueio não existe. Uma das razões para esses ambientes de produção é a tecnologia de produção. Em alguns processos, a temperatura ou outras características (como a viscosidade) do material requer que cada operação siga imediatamente a anterior. Este tipo de situação ocorre em processos químicos, farmacêuticos e de aço.

- Job shop e flow shop com reentrada

Uma reentrada ocorre quando uma tarefa completa uma passada pelo ambiente de produção e entra novamente neste ambiente para um novo conjunto de operações. Esta situação é típica da indústria de semicondutores.<sup>54</sup>

- Restrições de precedência

Este tipo de restrição impõe que uma tarefa só pode iniciar seu processamento após o término do processamento de outras tarefas, denominadas tarefas predecessoras. Isso ocorre em sistemas de produção<sup>55</sup> e de computação.<sup>56</sup>

De forma geral, as decisões de programação da produção devem ser tomadas em tempo pequeno (alguns segundos ou minutos), o que inviabiliza a utilização de métodos ótimos, que, em geral, exigem grande tempo computacional para a geração de uma solução ótima. Neste caso, uma prática comum é a utilização de regras de despacho que ordenam as tarefas de acordo com algum critério, como, por exemplo, as regras SPT e EDD.<sup>57</sup>

<sup>50</sup> Para formulações em *flow shop* e *job shop*, ver Stafford Jr. e Tseng (2002), e Cheung e Zhou (2001), respectivamente.

<sup>51</sup> Nowicki (1999).

<sup>52</sup> Hall e Sriskandarajah (1996).

<sup>53</sup> Para um algoritmo *branch-and-bound* para este problema, ver Ronconi e Armentano (2001).

<sup>54</sup> Pearn *et al.* (2004).

<sup>55</sup> Strusevich (1997).

<sup>56</sup> Kwok e Ahmad (2005).

<sup>57</sup> Baker (1974). Ver, por exemplo, Regras de despacho são utilizadas por Kiyuzato *et al.* (2002) para minimizar a soma dos atrasos em aplicação em produção de autopartes.

### 3.6.4 Outros problemas de programação

Existem outros problemas de programação (*scheduling*) que não podem ser descritos como o processamento de tarefas em máquinas. A seguir, apresentamos tipos distintos de problemas comuns e importantes de programação.

#### *Balanceamento de linha de montagem*

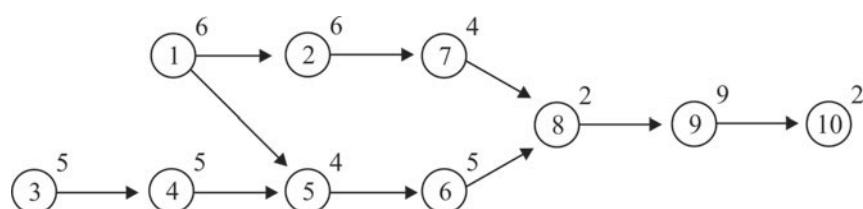
Uma linha de montagem é um sistema de fluxo de produção de alto volume de produtos padronizados, criada por Henry Ford em 1913 para a produção de automóveis. Atualmente, também tem sido usada para a produção de baixo volume de produtos personalizados.<sup>58</sup> Uma linha de montagem consiste em estações de trabalho dispostas ao longo de uma esteira, e as tarefas são lançadas na linha, movendo-se de uma estação para outra. Em cada estação, um conjunto de operações é executado em um intervalo de tempo, denominado *tempo de ciclo*. O problema de partitionar (balancear) o trabalho de montagem entre estações com relação a algum objetivo é denominado problema de balanceamento de linha de montagem. A instalação de uma linha de montagem é uma decisão de longo prazo e, em geral, requer um grande investimento.

Cada operação de manufatura em um produto é chamada de tarefa que tem uma dada duração, e devido a condições tecnológicas e organizacionais, existem restrições de precedência entre tarefas. Esses elementos podem ser representados em um grafo de precedência, em que cada nó representa uma tarefa, o peso do nó corresponde à duração da tarefa e arcos representam as relações de precedência. A Figura 3.26<sup>59</sup> mostra um grafo de precedência com dez nós. A duração da tarefa correspondente ao nó 5 é de 4 unidades de tempo, e essa tarefa só pode ser executada quando o processamento das tarefas 1 e 4 (nós predecessores imediatos do nó 5) é completado.

Seja  $S_k$  o conjunto de tarefas designadas à estação  $k$ ,  $k = 1, \dots, m$ , e  $d_j$  a duração da tarefa  $j$ ,  $j = 1, \dots, n$ . Então, a carga da estação  $k$  é dada por  $d(S_k) = \sum_{j \in S_k} d_j$ . Quando um tempo de ciclo  $c$  é dado, então uma linha balanceada é factível somente se a carga da estação não excede  $c$ . Quando  $d(S_k) < c$ , a estação  $k$  tem um tempo ocioso de  $c - d(S_k)$  unidades de tempo.

Existem linhas com montagem de um produto ou de vários produtos. No caso de um único produto, o objetivo pode ser minimizar  $m$  para um dado  $c$  ou, ainda, minimizar  $c$  (maximizar a taxa de produção) para um dado  $m$ . O objetivo mais geral consiste em maximizar a eficiência  $E$  da linha, definida como  $E = d_{\text{total}} / mc$ , em que  $d_{\text{total}} = \sum_{j=1}^n d_j$ .

Para o exemplo da Figura 3.26, uma solução ótima, dado  $c = 11$ , corresponde à carga de  $m = 5$  estações,  $S_1 = \{1, 3\}$ ,  $S_2 = \{2, 4\}$ ,  $S_3 = \{5, 6\}$ ,  $S_4 = \{7, 8\}$ ,  $S_5 = \{9, 10\}$ . As estações 2 e 5 não têm



**Figura 3.26** Grafo de precedência em linha de montagem.

<sup>58</sup> Para uma revisão de problemas generalizados de linhas de montagem, ver Becker e Scholl (2006).

<sup>59</sup> Figura extraída de Scholl e Becker (2006).

tempo ocioso, e as estações 1, 3 e 4 têm tempo ocioso de 1, 2 e 5 unidades de tempo, respectivamente. Uma solução ótima, dado  $m = 6$ , corresponde às cargas das estações  $S_1 = \{3, 4\}$ ,  $S_2 = \{1, 5\}$ ,  $S_3 = \{2, 7\}$ ,  $S_4 = \{6, 8\}$ ,  $S_5 = \{9\}$ ,  $S_6 = \{10\}$ , com tempo de ciclo mínimo  $c = 11$ . A eficiência ótima é dada por  $E = 48/11 \times 5 = 0,87$ , que é atingida pela primeira solução ótima.

### Programação de projetos

Um projeto consiste de atividades que têm uma duração, relações de precedência e competem por recursos. Quando os recursos são ilimitados, uma prática comum é utilizar os métodos CPM e PERT,<sup>60</sup> descritos na Seção 4.2.2, para analisar o efeito do possível atraso de atividades. Projetos são encontrados em diversas áreas, tais como: construção de edifícios, pontes, estradas; planejamento da produção de produtos sob encomenda, por exemplo, aviões e navios; desenvolvimento e pesquisa.<sup>61</sup>

Problemas de programação de projetos, brevemente discutidos na Seção 2.2.4, consistem em encontrar um instante de início para todas as atividades de forma a minimizar um ou mais objetivos, como, por exemplo, o instante de término do projeto (*makespan*) quando os recursos são limitados. Existe uma grande variedade de problemas de acordo com os elementos do projeto. Atividades podem ser executadas de diversos modos, como, por exemplo, um trabalhador em oito períodos de tempo ou quatro trabalhadores em dois períodos de tempo. Existem categorias de recursos, tais como recursos renováveis, que estão disponíveis em cada período (máquinas, equipamento, mão-de-obra), e recursos não-renováveis, que são limitados ao longo do horizonte de planejamento (capital disponível). Além disso, há uma diversidade de objetivos, tais como minimização do *makespan* ou do custo,<sup>62</sup> e maximização da qualidade ou do valor presente, quando existe um fluxo de caixa ao longo do projeto, com despesas com atividades e receitas geradas pelo término de partes do projeto.

Para ilustrar a minimização do *makespan* em um projeto com recursos limitados, vamos considerar o Exemplo 4.19 da Seção 4.2.2. Um projeto pode ser representado por um grafo orientado em que nós representam atividades e arcos indicam relações de precedência. Por exemplo, a atividade 9 pode ser iniciada somente quando as atividades 4, 6 e 3 estiverem completadas. As durações das atividades correspondem a pesos dos nós. A Figura 3.27 mostra o grafo de precedência do projeto, com dez atividades e dois nós fictícios *O* e *F* que representam atividades de início e fim do projeto. Cada atividade  $i$ , com duração  $d_i$ , requer  $r_{ik}$  unidades do recurso  $k$ , com disponibilidade  $R_k$ . Assuma que cada atividade requer dois tipos de recursos, isto é,  $k = 1, 2$ , e considere os seguintes parâmetros:

$$\mathbf{d} = [d_i] = [0 \quad 4 \quad 10 \quad 5 \quad 1 \quad 5 \quad 7 \quad 4 \quad 3 \quad 2 \quad 2 \quad 0]$$

$$\mathbf{r} = [r_{ik}] = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 & 1 & 1 & 3 & 0 & 2 & 1 & 1 & 0 \end{bmatrix}$$

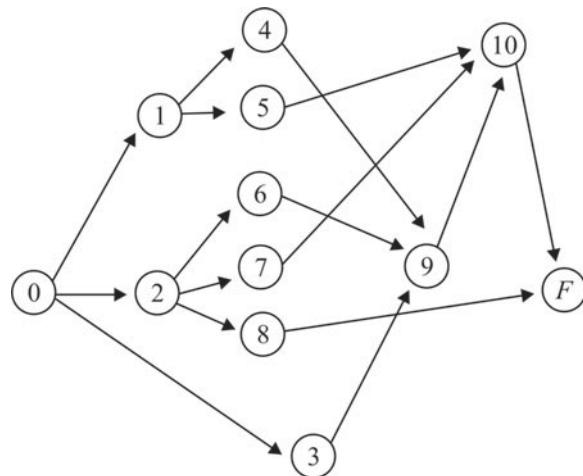
$$\mathbf{R} = [R_k] = [2 \quad 3]$$

As Figuras 3.28 e 3.29 mostram o diagrama de Gantt para o projeto com recursos ilimitados e limitados, respectivamente. A limitação de recursos, em geral, provoca um aumento do *makespan*, no caso, de 21 para 24. Note, por exemplo, que a atividade 3 é atrasada em relação ao projeto ilimitado, pois as atividades 1 e 2 consomem a disponibilidade de 3 unidades do recurso 2.

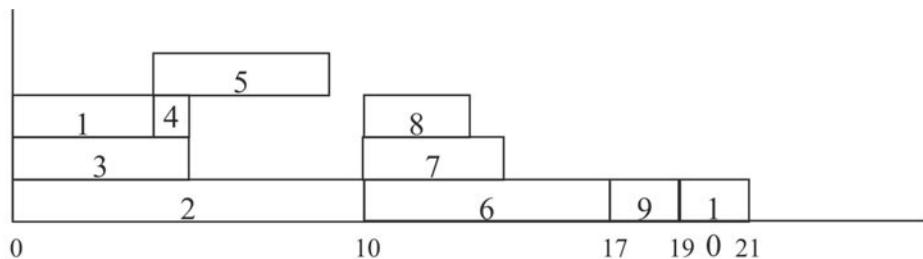
60 Outros métodos para programar atividades com duração aleatória são propostos por Mohring e Stork (2000).

61 Um bom livro introdutório é Moder *et al.* (1983) e, para uma revisão abrangente, ver Kolisch e Padman (2001).

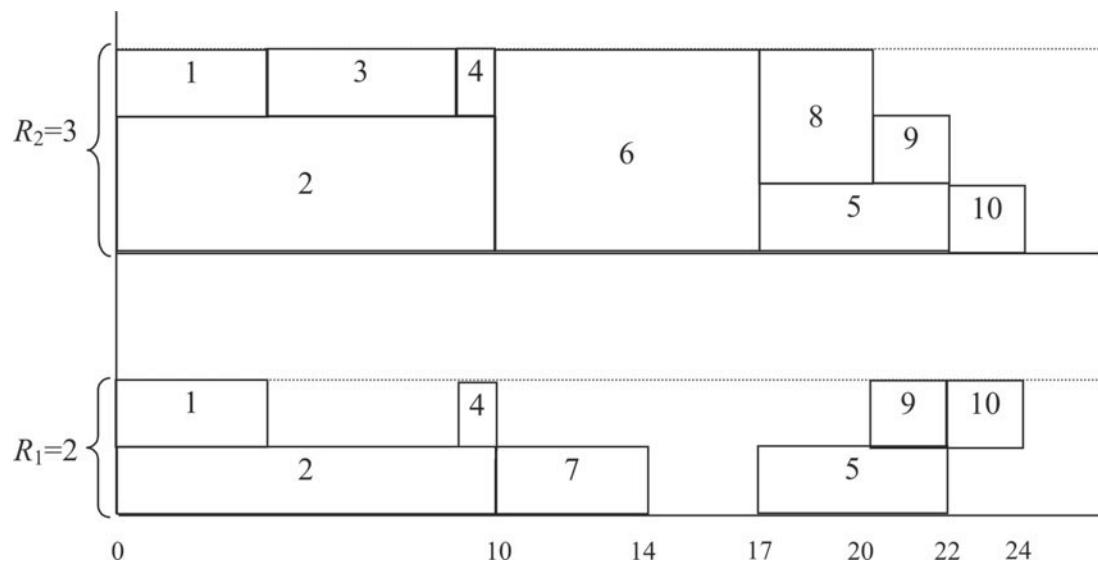
62 Ver, por exemplo, Yamashita *et al.* (2006).



**Figura 3.27** Grafo de precedência em projeto.



**Figura 3.28** Diagrama de Gantt para projeto com recursos ilimitados.



**Figura 3.29** Diagrama de Gantt para projeto com recursos limitados.

### *Programação de serviços*

A programação de atividades em serviços difere da programação de tarefas em máquinas em vários aspectos. A primeira diferença é que em serviços não há bens a serem estocados. Uma tarefa pode esperar pelo processamento em uma máquina, mas em serviço uma atividade pode estar associada a um cliente que não tolera esperar. A segunda distinção é que o número de recursos (por exemplo, máquinas) em manufatura é, em geral, fixo, enquanto em serviço o número de recursos (por exemplo, pessoas, quartos, caminhões) pode variar com o tempo. Outra distinção deve-se ao fato de que negar um serviço a um cliente é mais comum do que não entregar um produto no contexto de manufatura. A seguir são apresentados exemplos de tipos de problemas de programação em serviços.

### *Sistemas de reserva*

Uma agência de aluguel de carros mantém uma frota com vários tipos de carros. Clientes podem ser flexíveis ou não com relação ao tipo de carro que desejam alugar. Um cliente deseja fazer uma reserva para um período de dias determinado, e a agência tem de decidir se provê ou não o carro ao cliente. Pode ser vantajoso negar a reserva ao cliente quando existe uma chance de alugar o carro para outro cliente por um período de tempo maior. O objetivo da empresa é maximizar o número de dias de aluguel de seus carros.

### *Programação de horários em instituições de ensino*

O problema de programação de horários (*timetabling*) faz parte do cotidiano de instituições educacionais. A dificuldade de encontrar soluções satisfatórias manualmente nesses problemas tem motivado a pesquisa e o desenvolvimento de ferramentas de confecção automatizada de horários. Em geral, busca-se a confecção de um quadro de horários que satisfaça alguns requisitos essenciais (restrições *hard*), como, por exemplo, não alocar uma sala para duas aulas distintas no mesmo horário, e atenda, quando possível, outros requisitos não essenciais (restrições *soft*), como, por exemplo, a preferência de um professor por uma aula em um dado horário. Embora o problema de programação de horários seja bastante dependente das particularidades da instituição, em geral, pode ser classificado em três categorias, descritas a seguir.

- Escolas de ensino médio (e também algumas universidades): as turmas são conjuntos disjuntos de estudantes que compartilham o mesmo currículo e, em geral, devem ter horário completamente preenchido, enquanto os professores especificam restrições e/ou preferências de horários.
- Universidades: neste caso, há liberdade na seleção de disciplinas por parte dos alunos, e busca-se a montagem de um horário em que os alunos possam assistir a todas as aulas das disciplinas que pretendem cursar.
- Exames no final de período letivo: cada aluno apresenta um conjunto de exames que deve prestar e estes devem ser agendados de modo que nenhum aluno tenha dois exames simultâneos e, se possível, com intervalos de tempo razoáveis entre os exames de cada aluno.

Nos primeiros dois problemas, as aulas devem ser agendadas em um período, usualmente, uma semana dividida em dias. Pré-alocações de aulas podem ser requisitadas. Outra consideração bastante comum na formulação do problema é a alocação de recursos limitados, tais como salas de aula e projetores multimídia.

A definição das restrições *hard* e *soft* é bastante dependente do problema tratado. Em escolas, por exemplo, uma possível restrição *hard* é que uma turma não pode ter mais de duas aulas do mesmo assunto por dia. Uma possível restrição *soft* envolve a satisfação dos professores por meio

da geração de horários compactos, aulas concentradas em alguns dias, e sem períodos de inatividade entre aulas.<sup>63</sup>

#### Programação e Grade de Horários em Torneios de Esporte

Ligas de esportes profissionais constituem uma grande atividade econômica no mundo. As redes de televisão norte-americanas pagam mais de 400 milhões de dólares por ano para transmitir somente jogos de beisebol,<sup>64</sup> enquanto no Brasil uma rede de televisão comprometeu-se a pagar R\$707,7 milhões, no mínimo, para a transmissão dos jogos do campeonato brasileiro no triênio 2006-2008.<sup>65</sup>

Uma questão-chave em torneios de esportes é o programa de jogos das equipes. Os detentores dos direitos de transmissão não querem jogos sem atração em um horário nobre, e as equipes não querem que o investimento em jogadores e infra-estrutura seja prejudicado por um programa pobre de jogos. Portanto, programas eficientes são de grande interesse para as equipes, ligas, patrocinadores, mídia e fãs. Isso tem motivado a pesquisa em programação de jogos em diversos esportes, tais como futebol, basquetebol, beisebol, hóquei e críquete.

A distância entre cidades em países grandes é um fator complicador. No caso do campeonato brasileiro de futebol, uma viagem de São Paulo a Belém leva de 8 a 10 horas para cobrir uma distância de aproximadamente três mil quilômetros. Para reduzir a distância percorrida, o time de São Paulo pode viajar, por exemplo, de Belém a Fortaleza para outro jogo, antes de retornar para São Paulo.

A seguir, apresentamos o problema do torneio viajante (*traveling tournament problem*), que é um problema de grade de horários de um campeonato que abstrai certas características de problemas de programação de jogos.

Dados  $n$  times, em que  $n$  é par, um torneio simples é tal que cada time joga uma vez em  $n - 1$  rodadas. Cada rodada contém  $n/2$  jogos, de modo que o número total de jogos é  $n(n-1)/2$ , que corresponde à soma dos números inteiros  $1, 2, \dots, n-1$ . Por exemplo, em um torneio com 4 times, A, B, C e D, o time D joga com os times B, C e D, o time B joga com os times C e D, e o time C joga com o time D, totalizando  $(4 \times 3)/2 = 6$  jogos. Um torneio duplo é aquele que tem  $2(n-1)$  rodadas e cada par de times joga duas vezes, um em casa e outro na casa do oponente. Admite-se que cada time tenha um estádio em casa e que as distâncias entre as cidades são conhecidas. Cada time está em casa no início do torneio e retorna para casa depois da última partida fora de casa. Quando um time joga duas vezes consecutivas fora de casa, ele viaja da cidade do primeiro oponente para a cidade do segundo oponente sem retornar para casa.

Para um torneio duplo, o problema do torneio viajante consiste em minimizar a soma das distâncias viajadas por todos os times, sujeito à restrição de que o número de jogos consecutivos fora de casa e em casa é limitado (tipicamente, três). Existem duas variantes deste problema que incluem uma restrição adicional:

- Variante sem repetição: não existem times  $i, j$  tal que, se  $i$  joga com  $j$ , então  $j$  joga com  $i$  na próxima rodada.
- Variante espelho:<sup>66</sup> os jogos da rodada  $k$  são os mesmos da rodada  $k + (n-1)$ ,  $k = 1, \dots, n-1$ .

#### Programação em transporte

Existe um grande número de problemas de programação no setor de transportes, devido aos diversos meios de transporte, tais como navios, aviões, trens e ônibus. A programação de veículos

<sup>63</sup> Ver Santos *et al.* (2004) para uma heurística aplicada à programação de horário em escolas de ensino médio no Brasil.

<sup>64</sup> Easton (2001).

<sup>65</sup> Folha de S. Paulo, 31 de outubro de 2005.

<sup>66</sup> Ver Ribeiro (2005) para heurísticas para a variante espelho e aplicação no campeonato brasileiro de futebol de 2004.

e da tripulação são dois dos mais importantes problemas no planejamento operacional de uma empresa de transporte público, que pode ser ônibus, metrô, trem ou empresa aérea.

As entradas para o processo de planejamento operacional são as linhas e a freqüência destas. A partir dessa informação, constrói-se a grade de horários com as viagens e os correspondentes instantes e locais de partida e chegada. A programação de veículos consiste em designar veículos a viagens, resultando em um programa para cada veículo. Os veículos que não estão em uso ficam estacionados em um depósito. Um programa para um veículo consiste em blocos de viagens consecutivas, cada bloco começando no depósito. Em cada bloco existem tarefas que devem ser designadas ao período de trabalho de uma tripulação. A programação da tripulação corresponde ao planejamento de curto prazo, isto é, designar tarefas à tripulação a cada dia.

Restrições importantes para veículos envolvem atender a demanda de passageiros e respeitar o primeiro e o último horário de partida em cada dia. Restrições importantes associadas à tripulação compreendem tempo máximo de trabalho diário, limite de tempo de hora extra e tempo de descanso dentro de uma janela de tempo do período de trabalho. Os objetivos estão relacionados a aspectos operacionais, tais como minimizar o número de veículos e tripulações e duração de hora extra.<sup>67</sup>

#### Programação de força de trabalho

A alocação de força de trabalho e a programação de pessoas lidam com programas de trabalho e designação de pessoas a turnos de modo a atender à demanda por recursos que variam ao longo do tempo. Exemplos destes problemas são operadores de telefonia, enfermeiras em hospitais, policiais, pessoal de transporte (tripulação de avião, motorista de ônibus), dentre outros. As operações nesses ambientes são prolongadas e irregulares, e a necessidade de pessoas varia ao longo do tempo.

Considere, por exemplo, a programação de enfermeiras em hospitais. Cada hospital tem uma necessidade de enfermeiras que varia a cada dia. Por exemplo, o número de enfermeiras necessário em fins de semana é menor, e pode ou não ser menor no período noturno. Existem diversas restrições impostas por leis trabalhistas e pelo hospital, gerando, assim, diversas possibilidades de programas por turno, com custos distintos. O horizonte de planejamento varia de uma semana a um mês, tipicamente. Enfermeiras são, em geral, classificadas em níveis de acordo com seu conhecimento e experiência. Além disso, elas explicitam, no início do horizonte de planejamento, suas preferências por dias de folga. A combinação desses dois fatores resulta em programas com diversos níveis de qualidade. A meta é designar pessoas a turnos, de modo que as necessidades diárias sejam atendidas e as restrições sejam satisfeitas a um custo mínimo, custo este que pode envolver hora extra e baixa qualidade de atendimento.<sup>68</sup>

### 3.7\* COMPLEXIDADE COMPUTACIONAL

Uma instância  $I$  de um problema  $P$  é definida pelos valores dos parâmetros. Por exemplo,

$$\max \{2x_1 + 4x_2 + 3x_3 : 3x_1 + 2x_2 + 4x_3 \leq 5, x \in B^3\}$$

corresponde a uma instância do problema da mochila

$$\max \left\{ \sum_{j=1}^n c_j x_j : \sum_{j=1}^n a_j x_j \leq b, x \in B^n \right\}$$

<sup>67</sup> Ver Rodrigues *et al.* (2006) para a otimização da programação integrada de veículos e tripulação, com aplicação em empresas que operam em São Paulo e São Bernardo do Campo.

<sup>68</sup> Métodos heurísticos são propostos por Dias *et al.* (2003) para a programação de enfermeiras do Hospital das Clínicas da Universidade Estadual de Campinas.

Para uma instância  $I$  de um problema  $P$ , o comprimento da entrada  $L = L(I)$  é o comprimento da representação binária de  $I$ . No caso do problema da mochila, tem-se

$$L(I) = \sum_{j=1}^n \lceil \log_2 c_j \rceil + \sum_{j=1}^n \lceil \log_2 a_j \rceil$$

Dado um problema  $P$ , um algoritmo  $A$  para este problema, e uma instância  $I$ , seja  $f_A(I)$  o número de operações elementares para aplicar o algoritmo  $A$  à instância  $I$ . A função complexidade  $f_A^*(k) = \max_I \{f_A(I) : L(I) = k\}$  corresponde ao tempo de execução do algoritmo  $A$ .

A teoria da complexidade computacional distingue dois tipos de função, que representam um algoritmo “eficiente” e um algoritmo “não eficiente”. Um algoritmo eficiente tem função complexidade polinomial, isto é,  $f_A^*(k) \leq ck^p$ , para algum inteiro positivo  $p$  e uma constante positiva  $c$ . Um algoritmo não eficiente é caracterizado por uma função exponencial de  $k$ , isto é, para constantes  $c_1, c_2 > 0$ ,  $d_1, d_2 > 1$  e um inteiro positivo  $k$ , tem-se

$$c_1 d_1^k \leq f_A^*(k) \leq c_2 d_2^k \quad \text{para todo inteiro } k \geq k'.$$

Exemplos típicos de função exponencial correspondem à enumeração de  $2^k$  vetores binários de dimensão  $k$  e à enumeração de  $n!$  permutações de seqüências com  $n$  elementos.

Os fundamentos da teoria da NP-completude associada a problemas de decisão foram iniciados com o problema de satisfabilidade (*satisfiability*).<sup>69</sup> Este é um problema clássico na lógica booleana, em que uma variável  $x$  assume o valor 0 ou 1. Os valores 0 e 1 são associados aos termos *falso* e *verdadeiro* associados a proposições em lógica. O complemento de uma variável booleana  $x$  é definido como  $\bar{x} = 1 - x$ . Seja  $X = \{x_1, \dots, x_n\}$  um conjunto de variáveis booleanas. Se  $f$  é uma função das variáveis booleanas  $x_1, \dots, x_n$ , diz-se que cada termo  $x_i$  ou seu complemento  $\bar{x}_i$ ,  $i = 1, \dots, n$ , é um *literal*. Uma cláusula  $C$  definida no conjunto  $X$  é uma disjunção de literais. Por exemplo,  $C = x_1 \vee \bar{x}_4 \vee x_7$  é uma cláusula, em que o símbolo “ $\vee$ ” representa “ou”. Diz-se que uma cláusula é *satisfiuta* se pelo menos um de seus membros é verdadeiro. Neste exemplo, a cláusula é satisfiuta a menos que  $x_1 = 0, x_4 = 1, x_7 = 0$ .

Uma coleção de cláusulas  $C_1, C_2, \dots, C_m$  sobre  $X$  é *satisfiuta* se existe uma atribuição de valores aos literais de cada cláusula que satisfaça simultaneamente todas as cláusulas, isto é, que satisfaça a conjunção das cláusulas  $C_1 \wedge C_2 \wedge \dots \wedge C_m$ , em que o símbolo “ $\wedge$ ” representa “e”.

Como ilustração, considere as cláusulas  $C_1 = x_1 \vee \bar{x}_2 \vee x_3$ ,  $C_2 = x_2 \vee x_3$ ,  $C_3 = x_1 \vee \bar{x}_3$ . A fórmula  $C_1 \wedge C_2 \wedge C_3$  é satisfiuta ao se fazer  $x_1 = 1, x_2 = 1, x_3 = 0$ . Este problema pode ser traduzido para a existência de uma solução factível para o seguinte problema binário:

$$\begin{aligned} x_1 + (1 - x_2) + x_3 &\geq 1 \\ x_2 + x_3 &\geq 1 \\ x_1 + (1 - x_3) &\geq 1 \end{aligned}$$

De modo geral, dadas  $m$  cláusulas  $C_i = (C_i^+, C_i^-)$ ,  $i = 1, \dots, m$ , existe

$$x \in B^n : \sum_{j \in C_i^+} x_j + \sum_{j \in C_i^-} (1 - x_j) \geq 1 \quad \text{para } i = 1, \dots, m?$$

Em seguida, demonstrou-se<sup>70</sup> que uma coleção de problemas de decisão associados a problemas de otimização combinatória, incluindo o problema do caixeiro-viajante, são tão “difíceis” quanto o problema de satisfabilidade. Por exemplo, a versão decisão do problema de otimização da mochila corresponde a: para um dado  $k$ , existe

$$x \in B^n : \sum_{j=1}^n a_j x_j \leq b \quad \text{e} \quad \sum_{j=1}^n c_j x_j \geq k ?$$

<sup>69</sup> Cook (1971).

<sup>70</sup> Karp (1972).

Em 1979, Garey e Johnson publicaram um livro considerado um marco na teoria da complexidade computacional. Como ressaltado pelos autores, a distinção entre os algoritmos polinomiais e exponenciais é bastante significativa ao se considerarem instâncias grandes. A Tabela 3.2 ilustra taxas de crescimento de tempo de execução para algumas funções de cada tipo, admitindo um computador que execute  $10^6$  operações por segundo. Note o crescimento explosivo das três funções de complexidade exponencial.

Para a maioria dos problemas de programação inteira mista, incluindo a maioria dos problemas formulados anteriormente, não existem algoritmos polinomiais de resolução.

### 3.8\* FORMULAÇÕES ALTERNATIVAS

Podem existir diversas formulações para um dado problema e algumas delas são melhores. Inicialmente, define-se o conceito de formulação.

**Definição 3.1** Um subconjunto de  $R^n$  descrito por restrições lineares  $P = \{\mathbf{x} \in R^n : \mathbf{Ax} \leq \mathbf{b}\}$  é um poliedro.

**Definição 3.2** Um poliedro  $P \subset R^{n+p}$  é uma formulação para um conjunto  $X \subset Z^n \times R^p$  se e somente se  $X = P \cap (Z^n \times R^p)$ .

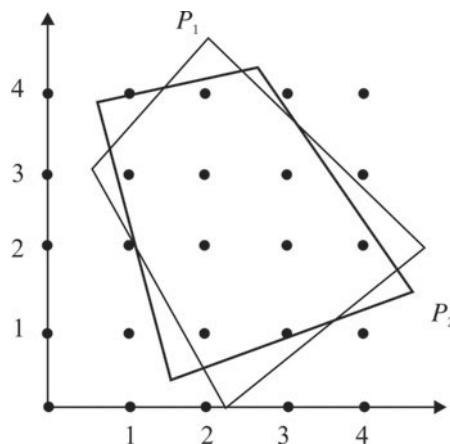
**Exemplo 3.17** A Figura 3.30 apresenta duas formulações distintas  $P_1$  e  $P_2$  para o conjunto

$$X = \{(1,3), (2,1), (2,2), (2,3), (2,4), (3,1), (3,2), (3,3), (4,2)\}$$

■

**Tabela 3.2**  
**Tempo computacional para funções polinomiais e exponenciais.**

Função Complexidade	Tamanho n					
	10	20	30	40	50	60
$n$	0,00001 segundo	0,00002 segundo	0,00003 segundo	0,00004 segundo	0,00005 segundo	0,00006 segundo
$n^2$	0,0001 segundo	0,0004 segundo	0,0009 segundo	0,0016 segundo	0,0025 segundo	0,0036 segundo
$n^3$	0,001 segundo	0,008 segundo	0,027 segundo	0,064 segundo	0,125 segundo	0,216 segundo
$n^5$	0,1 segundo	3,2 segundos	24,3 segundos	1,7 minuto	5,2 minutos	13,0 minutos
$2^n$	0,001 segundo	1,0 segundo	17,9 minutos	12,7 dias	35,7 anos	366 séculos
$3^n$	0,059 segundo	58 minutos	6,5 anos	3855 séculos	$2 \times 10^8$ séculos	$1,3 \times 10^{13}$ séculos
$n!$	3,628 segundos	771,4 séculos	$8,4 \times 10^{16}$ séculos	$2,5 \times 10^{32}$ séculos	$9,6 \times 10^{48}$ séculos	$2,6 \times 10^{66}$ séculos



**Figura 3.30** Duas formulações distintas para um problema de programação inteira.

**Definição 3.3** Um conjunto  $X \subset R^n$  é chamado de conjunto convexo se, para quaisquer  $\mathbf{x}_1, \mathbf{x}_2 \in X$  e todo número real  $\alpha$ ,  $0 < \alpha < 1$ , o ponto  $\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2 \in X$ .

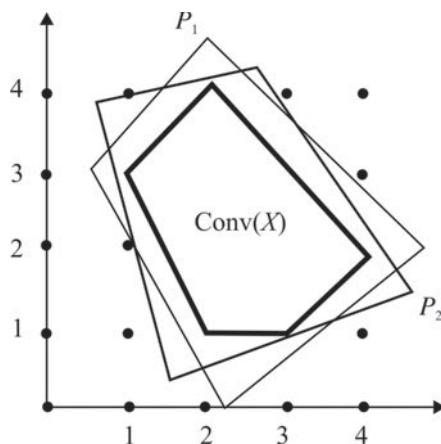
A interpretação geométrica desta definição é que, se um conjunto é convexo, então, dados dois pontos no conjunto, todo ponto no segmento de reta que liga esses dois pontos também é um elemento do conjunto.

**Definição 3.4** Dado um conjunto  $X \subset R^n$ , a envoltória convexa (*convex hull*) de  $X$ , representada por  $\text{conv}(X)$ , é definida como a interseção de todos os conjuntos convexos que contêm  $X$ , isto é, o menor conjunto convexo que contém  $X$ .

A Figura 3.31 ilustra a envoltória convexa do conjunto  $X$  citado, que corresponde à combinação convexa dos pontos  $\mathbf{x}_1 = (1, 3)$ ,  $\mathbf{x}_2 = (2, 1)$ ,  $\mathbf{x}_3 = (2, 4)$ ,  $\mathbf{x}_4 = (3, 1)$ ,  $\mathbf{x}_5 = (4, 2)$ , isto é,

$$\text{conv}(X) = \left\{ \sum_{j=1}^5 \alpha_j \mathbf{x}_j, \quad \sum_{j=1}^5 \alpha_j = 1, \quad \alpha \geq 0, \forall j \right\}$$

As Figuras 3.32 e 3.33 mostram a envoltória convexa de conjuntos mais complexos.



**Figura 3.31** Envoltória convexa do conjunto  $X$ .

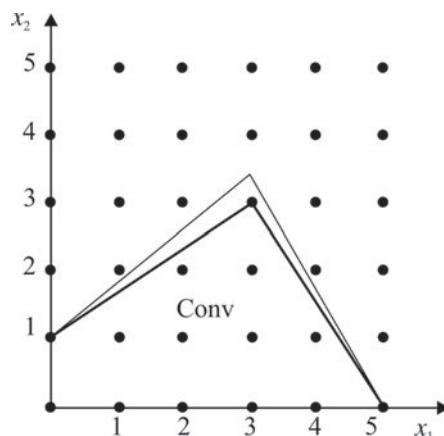
xa dos conjuntos de soluções factíveis dos exemplos de *PI* e *PIM* nas Figuras 3.1 e 3.2, respectivamente.

Note que as soluções ótimas dos exemplos de *PI* e *PIM* são pontos extremos das envoltórias convexas dos conjuntos de soluções factíveis de *PI* e *PIM*, respectivamente. Demonstra-se que isso é verdadeiro para problemas de programação inteira e programação inteira mista. Portanto, podemos expressar o problema de programação inteira mista *PIM* como o seguinte problema de programação linear.

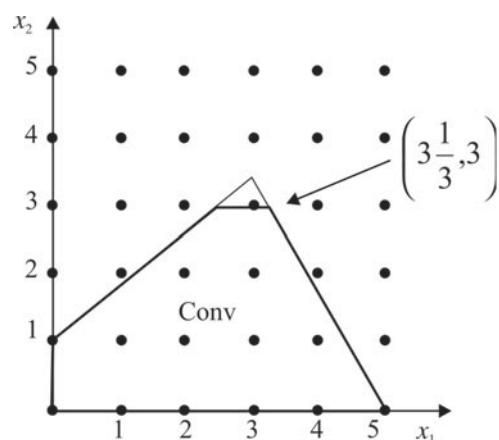
$$\begin{aligned} z &= \max \mathbf{c}^T \mathbf{x} + \mathbf{d}^T \mathbf{y} \\ \tilde{\mathbf{A}}\mathbf{x} + \tilde{\mathbf{D}}\mathbf{y} &\leq \tilde{\mathbf{b}} \\ \mathbf{x} &\in R_+^n, \mathbf{y} \in R_+^p \end{aligned}$$

em que  $\{(\mathbf{x}, \mathbf{y}): \tilde{\mathbf{A}}\mathbf{x} + \tilde{\mathbf{D}}\mathbf{y} \leq \tilde{\mathbf{b}}, \mathbf{x} \in R_+^n, \mathbf{y} \in R_+^p\} = \text{conv}\{(\mathbf{x}, \mathbf{y}): \mathbf{A}\mathbf{x} + \mathbf{D}\mathbf{y} \leq \mathbf{b}, \mathbf{x} \in R_+^n, \mathbf{y} \in Z_+^p\}$ .

A partir dessa caracterização de *PIM*, é possível definir a melhor de duas formulações  $P_1$  e  $P_2$  para um dado conjunto  $X \subset R^n$ . Como  $X \subseteq \text{conv}(X) \subseteq P$ , para qualquer formulação  $P$ ,  $P_1$  é uma formulação melhor que  $P_2$  se  $P_1 \subset P_2$ .



**Figura 3.32** Envoltória convexa do Exemplo 3.1 de *PI*.



**Figura 3.33** Envoltória convexa do Exemplo 3.3 de *PIM*.

A determinação de  $\text{conv}(X)$  é tão difícil quanto resolver  $PIM$ , e os métodos de resolução *branch-and-bound* e *branch-and-cut*, discutidos na Seção 3.7, operam de forma a fazer com que a solução ótima de  $PIM$  seja a solução ótima de um problema de programação linear. Esses métodos eliminam regiões da formulação do problema  $P$  sem penetrar em  $\text{conv}(X)$ , para evitar a perda de uma possível solução ótima. Por esse motivo, quanto mais próxima é a formulação  $P$  em relação a  $\text{conv}(X)$ , menor o número de regiões a serem eliminadas, e os métodos convergem mais rapidamente para a solução ótima. Esta é a motivação para encontrar formulações melhores.

#### *Formulações equivalentes para o problema da mochila 0-1*

Considere o conjunto de vetores binários para o problema da mochila 0-1.

$X = \{(0,0,0,0), (1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1), (1,1,0,0), (1,0,0,1), (0,1,0,1) (1,1,0,1)\}$   
e as seguintes formulações:

$$\begin{aligned} P_1 &= \{\mathbf{x} \in R^4 : 0 \leq x_j \leq 1, j = 1, 2, 3, 4, x_1 + 9x_2 + 29x_3 + 4x_4 \leq 30\} \\ P_2 &= \{\mathbf{x} \in R^4 : 0 \leq x_j \leq 1, j = 1, 2, 3, 4, 0,3x_1 + 2x_2 + 6x_3 + x_4 \leq 6\} \\ P_3 &= \{\mathbf{x} \in R^4 : \begin{array}{lll} x_1 + & x_3 & \leq 1 \\ x_2 + x_3 & \leq 1 \\ x_3 + x_4 & \leq 1 \\ 0 \leq x_j \leq 1, & j = 1, 2, 3, 4 \end{array}\} \end{aligned}$$

Suponha que  $\mathbf{x} \in P_3$ . Então, multiplicando por 2 ambos os lados das desigualdades em  $P_3$  e depois somando as desigualdades tem-se,

$$2x_1 + 2x_3 + 2x_2 + 2x_3 + 2x_3 + 2x_4 = 2x_1 + 2x_2 + 6x_3 + 2x_4 \leq 6$$

Isto implica que  $0,3x_1 + 2x_2 + 6x_3 + x_4 \leq 6$ , portanto,  $\mathbf{x} \in P_2$ .

Se  $\mathbf{x} \in P_2$ , então multiplicando por 5 ambos os lados da desigualdade anterior resulta,

$$1,5x_1 + 10x_2 + 30x_3 + 5x_4 \leq 30$$

Isso implica que  $x_1 + 9x_2 + 29x_3 + 4x_4 \leq 30$ , portanto,  $\mathbf{x} \in P_1$ . Note que o ponto  $\mathbf{x}_2 = (1;1;0,45;1) \in P_2$ , mas  $\notin P_3$ , e o ponto  $\mathbf{x}_1 = (0;0;1;0,25) \in P_1$ , mas  $\notin P_2$ . Portanto,  $P_3 \subset P_2 \subset P_1$ . Pode-se mostrar que  $P_3 = \text{conv}(X)$ .

#### *Formulações para localização de facilidades com capacidade ilimitada*

Seja  $P_1$  a formulação relaxada para o problema de localização de facilidades com capacidade ilimitada (LF9)-(LF12), e considere a formulação alternativa  $P_2$  apresentada a seguir.

$$\min \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \quad (\text{LF9})$$

$$\sum_{i \in I} x_{ij} = 1, \quad \forall j \in J \quad (\text{LF10})$$

$$\sum_{j \in J} x_{ij} \leq n y_i, \quad \forall i \in I \quad (\text{LF11a})$$

$$0 \leq x_{ij} \leq 1, \quad \forall i \in I, \quad \forall j \in J, \quad \mathbf{y} \in B^{|I|} \quad (\text{LF12})$$

Note que a formulação alternativa difere pelo conjunto de restrições (LF11a) que corresponde à soma em  $j \in J$  das restrições

$$x_{ij} \leq y_i, \quad \forall i \in I, \forall j \in J \quad (\text{LF11})$$

Mostra-se a seguir que  $P_1 \subset P_2$ . Seja um ponto  $(x, y)$  que satisfaz (LF11), então somando sobre  $j \in J$ , segue-se que este ponto também satisfaz (LF11a). Portanto  $P_1 \subseteq P_2$ . Para mostrar que  $P_1 \subset P_2$ , basta exibir um ponto de  $P_2$  que não está em  $P_1$ . Suponha, por simplicidade, que  $n = km$ , com  $k \geq 2$ . Então, um ponto no qual cada facilidade atende  $k$  clientes, dado por  $x_{ij} = 1$ ,  $j = k(i-1)+1, \dots, k(i-1)+k$ ,  $i = 1, \dots, m$ , e  $x_{ij} = 0$  em caso contrário, e  $y_i = k/n$  para  $i = 1, \dots, m$  situa-se em  $P_2 \setminus P_1$ , que representa o conjunto de pontos que pertencem a  $P_2$  mas não pertencem a  $P_1$ . Por exemplo, se  $n = 12$  e  $m = 3$ , então  $k = 4$ ,  $x_{1j} = 1$ ,  $j = 1, 2, 3, 4$ ,  $x_{1j} = 0$  em caso contrário,  $x_{2j} = 1$ ,  $j = 5, 6, 7, 8$ ,  $x_{2j} = 0$  em caso contrário,  $x_{3j} = 1$ ,  $j = 9, 10, 11, 12$ ,  $x_{3j} = 0$  em caso contrário, e  $y_1 = y_2 = y_3 = 1/3$ . Portanto, para  $i = 1, 2, 3$ ,  $\sum_{j=1}^{12} x_{ij} = 4 = 12 \times \frac{1}{3}$ , que satisfaz (LF11a), mas viola (LF11).

#### *Formulações para dimensionamento de lotes de um item*

Considere a formulação relaxada  $P_1$  do problema descrito por (DL1)–(DL4) na Seção 3.6.1, aqui repetida para facilidade de leitura.

$$\begin{aligned} \min & \sum_{t=1}^T (sy_t + h_t I_t) \\ I_t &= I_{t-1} + x_t - d_t, \quad t = 1, \dots, T, \quad I_0 = I_T = 0 \\ x_t &\leq \left( \sum_{\tau=t}^T d_\tau \right) y_t, \quad t = 1, \dots, T \\ x_t, I_t &\geq 0, \quad 0 \leq y_t \leq 1, \quad t = 1, \dots, T \end{aligned}$$

A seguir é apresentada uma formulação alternativa  $P_2$  para este problema. Para tal, considere o parâmetro  $h_{\tau,t-1} = h_\tau + \dots + h_{t-1}$  e a variável

$u_{\tau,t}$  = quantidade produzida no período  $\tau$  para atender à demanda no período  $t$

Tem-se, então, a seguinte formulação:

$$\min \sum_{\tau=1}^T \sum_{t=\tau}^T h_{\tau,t-1} u_{\tau,t} + \sum_{t=1}^T sy_t \tag{DL1a}$$

$$\sum_{\tau=1}^t u_{\tau,t} = d_t \quad t = 1, \dots, T \tag{DL2a}$$

$$u_{\tau,t} \leq d_t y_t \quad \tau, t = 1, \dots, T; \quad \tau \leq t \tag{DL3a}$$

$$u_{\tau,t} \geq 0, \quad 0 \leq y_t \leq 1, \quad \tau, t = 1, \dots, T, \quad \tau \leq t \tag{DL4a}$$

A função objetivo (DL1a) expressa a minimização do custo total de estoque e produção. As restrições (DL2a) asseguram que a demanda é atendida em cada período  $t$ , enquanto as restrições (DL3a) impõem um limite superior às variáveis quando o item é produzido no período  $t$ . As restrições (DL4a) indicam o tipo das variáveis.

As variáveis de produção das duas formulações estão relacionadas por

$$x_\tau = \sum_{t=\tau}^T u_{\tau,t} \quad \tau = 1, \dots, T$$

Qualquer solução de  $P_2$  pertence a  $P_1$ . Além disso, o ponto  $x_t = d_t$ ,  $y_t = d_t / \sum_{\tau=1}^t d_\tau$  pertence a  $P_1$ , mas não pertence a  $P_2$ , pois  $u_{tt} = d_t$  e, de (DL3a), segue-se que  $y_t = 1 > d_t / \sum_{\tau=1}^t d_\tau$ . Portanto,  $P_2 \subset P_1$ . Demonstra-se que  $P_2$  representa a envoltória convexa das soluções deste problema.

### 3.9 MÉTODOS DE ENUMERAÇÃO IMPLÍCITA E DE PLANOS DE CORTE

Considere o problema de programação inteira-mista

$$(PIM) \quad \begin{aligned} z &= \max \mathbf{c}^T \mathbf{x} + \mathbf{d}^T \mathbf{y} \\ \mathbf{Ax} + \mathbf{Dy} &\leq \mathbf{b} \\ \mathbf{x} &\in R_+^n, \mathbf{y} \in Z_+^p \end{aligned}$$

em que  $\mathbf{A}$ , uma matriz  $(m \times n)$ ,  $\mathbf{D}$ , uma matriz  $(m \times p)$ ,  $\mathbf{c}^T$ , um vetor  $(1 \times n)$ ,  $\mathbf{d}^T$ , um vetor  $(1 \times p)$ , e  $\mathbf{b}$ , um vetor  $(m \times 1)$ , representam os parâmetros do problema. Os vetores de variáveis são  $\mathbf{x}$  e  $\mathbf{y}$  com dimensões  $(n \times 1)$  e  $(p \times 1)$ .  $R_+^n$  representa o espaço dos vetores com  $n$  componentes reais não-negativos e  $Z_+^p$  representa o espaço dos vetores com  $p$  componentes inteiros não-negativos.

Os métodos mais bem-sucedidos para resolver problemas genéricos de programação inteira-mista são baseados nos enfoques de enumeração implícita, ou *branch-and-bound*, e de planos de corte. A combinação desses enfoques gerou, na década de 1980, o método *branch-and-cut*, hoje presente nos pacotes comerciais de otimização, tais como CPLEX, XPRESS e LINDO.

Um conceito fundamental usado nesses métodos é a relaxação linear, que consiste em substituir em *PIM*  $\mathbf{y} \in Z_+^p$  por  $\mathbf{y} \in R_+^p$ , que se torna, portanto, um problema de programação linear. Para facilitar a exposição a seguir, considere o problema de programação inteira

$$(P) \quad z = \max \left\{ \mathbf{c}^T \mathbf{x} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \in Z_+^n \right\}$$

e sua relaxação linear

$$(PL) \quad \bar{z} = \max \left\{ \mathbf{c}^T \mathbf{x} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \in R_+^n \right\}$$

Como observado na Seção 3.1, o valor ótimo do problema relaxado *PL* é um limitante superior do problema de programação inteira *P*, isto é,  $\bar{z} \geq z$ .

#### 3.9.1 Método *branch-and-bound*

Considere os problemas *P* e *PL*. Se o poliedro  $\bar{P} = \left\{ \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \in R_+^n \right\}$  é fechado, então o número de soluções inteiras factíveis é finito. Uma forma de obter uma solução ótima para o problema *P* é utilizar um processo exaustivo, denominado enumeração completa, em que o valor de todas as soluções factíveis é calculado e escolhe-se a de maior valor.

Uma forma de reduzir o espaço de busca consiste em usar informação do problema de programação linear *PL* para executar um procedimento denominado *enumeração implícita*, em que subconjuntos de soluções são implicitamente considerados e descartados, pois não contêm uma solução ótima, ou então contêm soluções ótimas distintas, mas de mesmo valor ótimo. Esses subconjuntos são obtidos pela estratégia *dividir para conquistar*, que separa o problema original *P* em problemas menores de mais fácil resolução. Esta estratégia é usada no método *branch-and-bound* (*B&B*), cujos conceitos são ilustrados no seguinte exemplo:

#### Exemplo 3.18

$$(P) \quad \begin{aligned} z &= \max 5x_1 - x_2 \\ 7x_1 - 5x_2 &\leq 13 \\ 3x_1 + 2x_2 &\leq 17 \\ x &\in Z_+^2 \end{aligned}$$

### Resolução gráfica do exemplo

A Figura 3.34 mostra a região factível do problema  $P$ , a direção do gradiente da função objetivo  $(5, -1)$  e a solução ótima real do problema relaxado  $PL$ ,  $\bar{x} = (\bar{x}_1, \bar{x}_2) = (3\frac{24}{29}, 2\frac{22}{29})$  com valor  $\bar{z} = 16\frac{11}{29}$ , que é um limitante superior para o valor ótimo  $z$ .

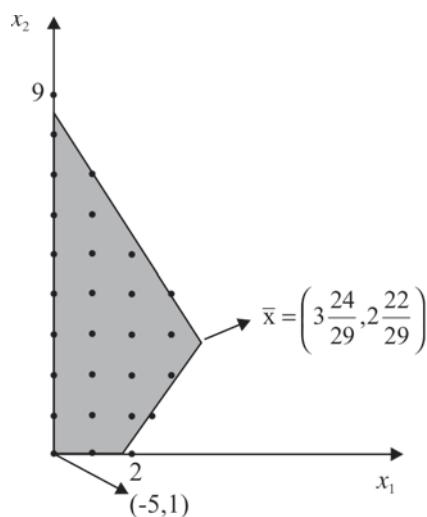
Considere a notação  $\lfloor x \rfloor = \max \{y \in Z : y \leq x\}$ , o maior inteiro contido em  $x$ , e  $\lceil x \rceil = \min \{y \in Z : y \geq x\}$ , o menor inteiro que contém  $x$ . Considere a divisão do problema  $P$  em dois problemas  $P^1$  e  $P^2$  da seguinte maneira. Adicione a restrição  $x_1 \leq \lfloor \bar{x}_1 \rfloor = 3$  para gerar o problema  $P^1$ , e a restrição  $x_1 \geq \lceil \bar{x}_1 \rceil = 4$  para gerar o problema  $P^2$ .

Essa partição pode ser representada por meio de uma árvore em que cada nó corresponde a um problema. Essa árvore, denominada árvore  $B\&B$ , é mostrada na Figura 3.35. Os nós descendentes do nó  $P$ ,  $P^1$  e  $P^2$  são chamados de nós *filhos*, e diz-se também que a variável  $x_1$  foi *ramificada*.

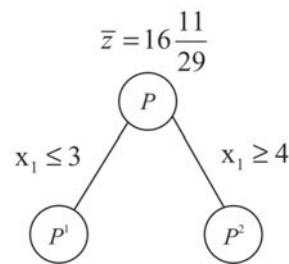
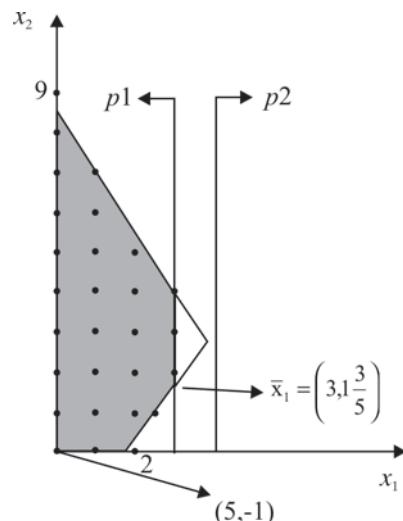
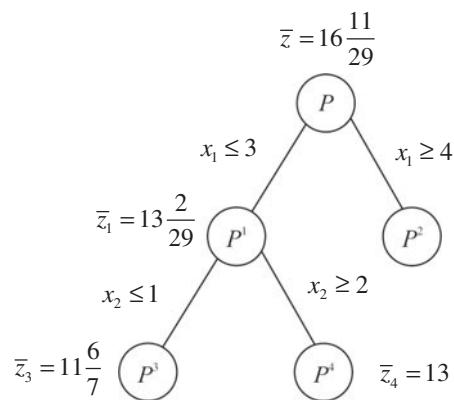
A Figura 3.36 mostra o espaço das soluções de  $P^1$  e  $P^2$ . Note que o espaço de soluções factíveis de  $P^2$ , relaxação linear de  $P^2$ , e consequentemente  $P^2$ , é vazio, o que implica que o problema  $P^2$  pode ser eliminado. A solução ótima da relaxação linear  $PL^1$  é dada por  $\bar{x}_1 = (3, 1\frac{3}{5})$ , com valor  $\bar{z}^1 = 13\frac{2}{5}$ , que é um limitante superior mais próximo ou apertado do valor ótimo de  $P$ .

Considere a partição do problema  $P^1$  em dois problemas  $P^3$  e  $P^4$  da seguinte maneira. Adicione a restrição  $x_2 \leq \lfloor 1\frac{3}{5} \rfloor = 1$  para gerar o problema  $P^3$ , e a restrição  $x_2 \geq \lceil 1\frac{3}{5} \rceil = 2$  para gerar o problema  $P^4$ . Essa partição é ilustrada na Figura 3.37, por meio da árvore  $B\&B$ .

Suponha agora que o nó correspondente ao problema  $P^4$  seja escolhido para exame. A relaxação linear  $PL^4$  resulta na solução ótima  $\bar{x}_4 = (3, 2)$ , que é uma solução inteira com valor  $\bar{z}^4 = 13$ . Considerando-se agora o nó correspondente ao problema  $P^3$ , tem-se que a relaxação linear  $PL^3$  fornece a solução ótima  $\bar{x}_3 = (2\frac{4}{7}, 1)$ , com valor  $\bar{z}^3 = 11\frac{6}{7}$ . Como  $\bar{z}^3 < \bar{z}^4$ , conclui-se



**Figura 3.34** Representação gráfica das soluções de  $P$  e  $PL$ .

**Figura 3.35** Partição do problema  $P$ .**Figura 3.36** Representação gráfica das soluções de  $P^1$ .**Figura 3.37** Partição do problema  $P^1$ .

que a solução  $\bar{x}_4 = (3, 2)$  é a solução ótima do problema  $P$ . A Figura 3.38 mostra o espaço das soluções de  $P^3$  e  $P^4$ , e a solução ótima  $\bar{x}_4$ .

### Resolução analítica do exemplo

Como visto anteriormente, o método *B&B* faz uso de programação linear e de adição de restrições, como mostram as Figuras 3.36 e 3.38, para cortar pontos extremos não inteiros, correspondentes a soluções ótimas dos problemas de programação linear. Isso também ocorre em métodos baseados em planos de corte, como o método *branch-and-cut* apresentado na Seção 3.9.3.

Ao se colocar uma nova restrição ao problema de programação linear, utiliza-se a informação da tabela ou quadro ótimo gerado pelo método simplex (veja Seção 2.7) para reotimizar o novo problema de programação linear. A reotimização é feita por meio do método dual simplex, apresentado na seção 2.10.4 para o problema primal de minimização. A seguir é apresentada uma forma alternativa de derivação da expressão da variável que entra na base no algoritmo dual simplex para problemas de maximização.

Considere o problema primal de programação linear

$$\bar{z} = \max \left\{ \mathbf{c}^T \mathbf{x} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \in R_+^n \right\}$$

e o problema dual associado

$$w = \min \left\{ \boldsymbol{\lambda}^T \mathbf{b} : \boldsymbol{\lambda}^T \mathbf{A} \geq \mathbf{c}^T \right\}$$

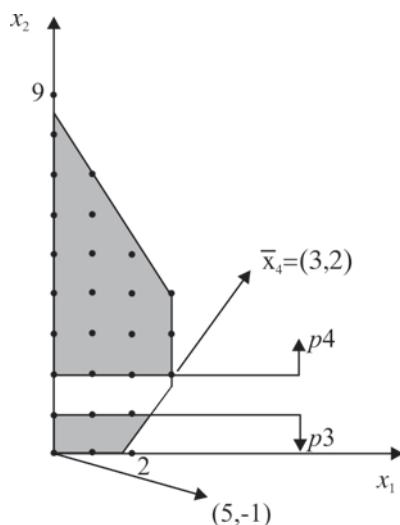
em que  $\boldsymbol{\lambda}^T$  corresponde ao vetor  $(1 \times m)$  de variáveis duais. Seja a partição

$$\mathbf{A} = [\mathbf{B} \ \mathbf{N}] \quad \mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} \quad \mathbf{c}^T = [\mathbf{c}_B^T \ \mathbf{c}_N^T]$$

tal que  $\mathbf{B}$  é uma matriz  $(m \times n)$  não-singular,  $\mathbf{x}_B$  e  $\mathbf{x}_N$  correspondem a vetores de variáveis básicas e não-básicas, respectivamente. Da partição, tem-se

$$\mathbf{Bx}_B + \mathbf{Nx}_N = \mathbf{b} \tag{3.7}$$

$$\bar{z} = \mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_N^T \mathbf{x}_N \tag{3.8}$$



**Figura 3.38** Representação gráfica das soluções de  $P^3$  e  $P^4$ .

De (3.7), tem-se que

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N \quad (3.9)$$

Substituindo (3.9) em (3.8), tem-se

$$\bar{z} = \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b} - (\mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N} - \mathbf{c}_N^T) \mathbf{x}_N \quad (3.10)$$

Seja

$$\mathbf{x}_{B_0} = \bar{z}, \quad \mathbf{x}_B^T = [x_{B_1}, \dots, x_{B_m}]$$

$NB$ , conjunto de índices das colunas de  $\mathbf{N}$ , e  $\hat{a}_j$ ,  $j \in NB$ , uma coluna de  $\mathbf{N}$ . Defina

$$\hat{\mathbf{a}}_0 = \begin{bmatrix} \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b} \\ \mathbf{B}^{-1} \mathbf{b} \end{bmatrix} = \begin{bmatrix} \hat{a}_{00} \\ \hat{a}_{10} \\ \vdots \\ \hat{a}_{m0} \end{bmatrix} \quad \hat{\mathbf{a}}_j = \begin{bmatrix} \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{a}_j - c_j \\ \mathbf{B}^{-1} \mathbf{a}_j \end{bmatrix} = \begin{bmatrix} \hat{a}_{0j} \\ \hat{a}_{1j} \\ \vdots \\ \hat{a}_{mj} \end{bmatrix}$$

De (3.9) e (3.10),

$$x_{B_0} = \hat{a}_{00} - \sum_{j \in NB} \hat{a}_{0j} x_j \quad (3.11)$$

$$x_{B_i} = \hat{a}_{i0} - \sum_{j \in NB} \hat{a}_{ij} x_j \quad i = 1, \dots, m \quad (3.12)$$

A solução básica  $x_{B_i} = y_{i0}$ ,  $i = 1, \dots, m$ , é ótima se:

- $\hat{a}_{i0} \geq 0$ ,  $i = 1, \dots, m$
- $\hat{a}_{0j} \geq 0$ ,  $\forall j \in NB$

Seja  $\mathbf{B}$  uma base do problema primal tal que  $\mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N} - \mathbf{c}_N^T \geq 0$ , isto é,  $\hat{a}_{0j} \geq 0$ ,  $\forall j \in NB$ . Se  $\mathbf{B}$  é factível, então  $\mathbf{x}_B = \mathbf{B}^{-1} \mathbf{b}$  é uma solução básica ótima. Defina  $\lambda^T = \mathbf{c}_B^T \mathbf{B}^{-1}$ . Então,

$$\lambda^T \mathbf{A} = \lambda^T [\mathbf{B} \ \mathbf{N}] = [\mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{B} \quad \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N}] = [\mathbf{c}_B^T \quad \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N}] \geq [\mathbf{c}_B^T \quad \mathbf{c}_N^T],$$

isto é,  $\lambda^T$  é dual factível e solução ótima do problema dual, pois  $w = \lambda^T \mathbf{b} = \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b} = z$ .

Admita agora que  $\hat{a}_{0j} \geq 0$ ,  $\forall j \in NB$ , e que  $x_{B_r} = \hat{a}_{r0} < 0$ , isto é, a solução básica atual é infactível. A variável  $x_{B_r}$  deve sair da base e uma variável deve entrar na base de modo a aumentar o valor da solução dual, atualmente dado por  $\hat{a}_{00} = \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b}$ .

Se  $\hat{a}_{rj} > 0$ , então, após a operação de pivotamento, tem-se

$$\hat{a}_{00} - \hat{a}_{0j} \frac{\hat{a}_{r0}}{\hat{a}_{rj}} \geq \hat{a}_{00}$$

Portanto, deve-se ater a  $\hat{a}_{rj} < 0$ . Defina

$$\frac{\hat{a}_{0k}}{\hat{a}_{rk}} = \max_{j \in NB} \left\{ \frac{\hat{a}_{0j}}{\hat{a}_{rj}}, \hat{a}_{rj} < 0 \right\} \quad (3.13)$$

Segue-se que

$$\frac{\hat{a}_{0k}}{\hat{a}_{rk}} \geq \frac{\hat{a}_{0j}}{\hat{a}_{rj}}, \quad \forall j : \hat{a}_{rj} < 0$$

portanto,

$$\hat{a}_{0j} - \frac{\hat{a}_{0k}\hat{a}_{rj}}{\hat{a}_{rk}} \geq 0, \forall j : \hat{a}_{rj} < 0$$

Como  $\hat{a}_{0j} \geq 0$  e  $\hat{a}_{rk} < 0$ , então

$$\hat{a}_{0j} - \frac{\hat{a}_{0k}\hat{a}_{rj}}{\hat{a}_{rk}} \geq 0, \forall j : \hat{a}_{rj} \geq 0$$

portanto, (3.13) define a variável a entrar na base.

Note que, se  $\hat{a}_{rj} \geq 0, \forall j \in NB$ , então o problema primal é infactível, pois

$$x_{B_r} = \hat{a}_{r0} - \sum_{j \in NB} \hat{a}_{rj} x_j < 0, \forall x_j \geq 0, j \in NB$$

Considere agora que a solução básica ótima de um problema de programação linear tenha sido obtida e deseja-se incluir a restrição

$$\sum_{j=1}^n a_{pj} x_j \leq b_p$$

ou

$$\sum_{j=1}^n a_{pj} x_j + s_p = b_p, \quad s_p \geq 0$$

Tomando  $s_p$  como variável básica e expressando-a em termos das variáveis não-básicas associadas à solução básica ótima, tem-se

$$s_p = \hat{a}_{p0} - \sum_{j=1}^n \hat{a}_{pj} x_j$$

Se  $\hat{a}_{p0} \geq 0$ , a nova solução básica com  $s_p = \hat{a}_{p0}$  é ótima. Se  $\hat{a}_{p0} < 0$ , então a nova solução com  $s_p = y_{p0}$  é primal infactível, mas dual factível, pois  $\hat{a}_{0j} \geq 0, \forall j \in NB$ . Aplica-se, então, o dual simplex para reotimizar o problema com a restrição adicional.

O método dual simplex é agora aplicado para a reotimização dos problemas de programação linear envolvidos no exemplo ilustrativo do método *B&B*.

### 1) Nô P

A relaxação linear do problema *P* do Exemplo 3.18 e a inclusão das variáveis de folga  $s_1$  e  $s_2$  resultam no problema *PL*:

$$(PL) \quad \begin{aligned} \bar{z} &= \max 5x_1 - x_2 \\ 7x_1 - 5x_2 + s_1 &= 13 \\ 3x_1 + 2x_2 + s_2 &= 17 \\ x_1, x_2, s_1, s_2 &\geq 0 \end{aligned}$$

cuja solução ótima é mostrada no Quadro 3.1.

### 2) Nô $P^1$

Insere-se a restrição  $x_1 \leq 3^{71}$  ou, equivalentemente,  $x_1 + s_3 = 3, s_3 \geq 0$ , ao problema *P* e resolve-se a relaxação linear *PL*<sup>1</sup>. Usando o Quadro 3.1, a variável  $s_3$  é expressa em termos das variáveis não-básicas  $s_1$  e  $s_2$ :

<sup>71</sup> Neste ponto, estamos escolhendo arbitrariamente a variável  $x_2$  para ramificação; ver Linderoth e Savelsbergh (1999) para um estudo computacional sobre métodos de seleção de variáveis a serem ramificadas e de escolha de nós.

$$s_3 = 3 - 3 \frac{24}{29} + \frac{2}{29} s_1 + \frac{5}{29} s_2 \quad (3.14)$$

A variável  $s_3$  sai da base e, de (3.13), tem-se de

$$\max \left\{ \frac{\frac{13}{29}}{-\frac{2}{29}}, \frac{\frac{18}{29}}{-\frac{5}{29}} \right\} = -\frac{\frac{18}{29}}{\frac{5}{29}}$$

portanto, a variável  $s_2$  entra na base.

Incluindo a restrição (3.14) no Quadro 3.1 e executando operações de pivotamento, obtém-se a solução ótima de  $PL^1$ , mostrada no Quadro 3.2.

### 3) Nô $P^2$

Insere-se a restrição  $x_1 \geq 4$  ou, equivalentemente,  $x_1 - s_4 = 4$ ,  $s_4 \geq 0$ , ao problema  $P$  e resolve-se a relaxação linear  $PL^2$ . Usando o Quadro 3.1, a variável  $s_4$  é expressa em termos das variáveis não-básicas  $s_1$  e  $s_2$ :

$$s_4 = -4 - 3 \frac{24}{29} - \frac{2}{29} s_1 - \frac{5}{29} s_2$$

Como os coeficientes de  $s_1$  e  $s_2$  são negativos, segue-se que  $PL^2$  e, portanto,  $P^2$  é infactível.

**Quadro 3.1**

	$x_1$	$x_2$	$s_1$	$s_2$	
$\bar{z}$	0	0	$\frac{13}{29}$	$\frac{18}{29}$	$16\frac{11}{29}$
$x_1$	1	0	$\frac{2}{29}$	$\frac{5}{29}$	$3\frac{24}{29}$
$x_2$	0	1	$-\frac{3}{29}$	$\frac{7}{29}$	$2\frac{22}{29}$

**Quadro 3.2**

	$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	
$\bar{z}^1$	0	0	$\frac{1}{5}$	0	$3\frac{3}{5}$	$13\frac{2}{5}$
$x_1$	1	0	0	0	1	3
$x_2$	0	1	$-\frac{1}{5}$	0	$1\frac{2}{5}$	$1\frac{3}{5}$
$s_2$	0	0	$\frac{2}{5}$	1	$-5\frac{4}{5}$	$4\frac{4}{5}$

**4) Nó  $P^3$** 

Insere-se a restrição  $x_2 \leq 1$  ou, equivalentemente,  $x_2 + s_5 = 1$ ,  $s_5 \geq 0$ , ao problema  $P^1$  e resolve-se a relaxação linear  $PL^3$ . Usando o Quadro 3.2, expressa-se a variável  $s_5$  em termos das variáveis não-básicas  $s_1$  e  $s_3$ :

$$s_5 = 1 - x_2 = 1 - 1 - \frac{3}{5} - \frac{1}{5}s_1 + \frac{7}{5}s_3 = -\frac{3}{5} - \frac{1}{5}s_1 + \frac{7}{5}s_3 \quad (3.15)$$

Incluindo a restrição (3.15) no Quadro 3.2, e utilizando a expressão (3.13), segue-se que a variável  $s_5$  sai da base, a variável  $s_3$  entra na base e, após operações de pivotamento, obtém-se o Quadro 3.3 com a solução ótima de  $PL^3$ .

**5) Nó  $P^4$** 

Insere-se a restrição  $x_2 \geq 2$  ou, equivalentemente,  $x_2 - s_6 = 2$ ,  $s_6 \geq 0$ , ao problema  $P^1$  e resolve-se a relaxação linear  $PL^4$ . Usando o Quadro 3.2, expressa-se a variável  $s_6$  em termos das variáveis não-básicas  $s_1$  e  $s_3$ :

$$s_6 = x_2 - 2 = \frac{8}{5} - 2 + \frac{1}{5}s_1 - \frac{7}{5}s_3 = -\frac{2}{5} + \frac{1}{5}s_1 - \frac{7}{8}s_3 \quad (3.16)$$

Incluindo a restrição (3.16) no Quadro 3.2 e utilizando a expressão (3.13), segue-se que a variável  $s_6$  sai da base, a variável  $s_1$  entra na base e, após operações de pivotamento, obtém-se o Quadro 3.4 com a solução ótima  $\bar{x} = (3, 2)$  de  $PL^4$  e de  $P$ .

*Testes para eliminação de nós na árvore branch-and-bound*

Como ilustrado no exemplo anterior, um problema  $P^i$  correspondente a um nó na árvore *B&B* é eliminado ou descartado por testes que utilizam sua relaxação linear  $PL^i$ . Para formalizar esses testes, considere a seguinte notação:

$F(P^i)$ : região factível do problema  $P^i$

$F(PL^i)$ : região factível do problema  $PL^i$

$z^i$ : valor ótimo do problema  $P^i$  (limitante inferior de  $z$ )

**Quadro 3.3**

	$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$s_5$	
$\bar{z}^3$	0	0	$\frac{5}{7}$	0	0	$2\frac{4}{7}$	$11\frac{6}{7}$
$x_1$	1	0	$\frac{1}{7}$	0	0	$\frac{5}{7}$	$2\frac{4}{7}$
$x_2$	0	1	0	0	0	1	1
$s_2$	0	0	$-\frac{3}{7}$	1	0	$-4\frac{1}{7}$	$7\frac{2}{7}$
$s_3$	0	0	$-\frac{1}{7}$	0	1	$-\frac{5}{7}$	$\frac{3}{7}$

$\bar{z}^i$ : valor ótimo do problema  $PL^i$  (limitante superior de  $z$ )

$\mathbf{x}^*$ : melhor solução encontrada até o momento (solução incumbente)

$z^*$ : valor de  $\mathbf{x}^*$

O problema  $P^i$  é eliminado se satisfizer um dos seguintes testes de eliminação:

1)  $F(PL^i) = \emptyset$ :  $P^i$  é eliminado por infactibilidade.

O resultado decorre da relação  $F(P^i) \subseteq F(PL^i)$ .

2)  $\bar{z}^i \leq z^* : P^i$  é eliminado por qualidade.

Como  $z^i \leq \bar{z}^i \leq z^*$ , segue-se o resultado.

3) Se a solução ótima de  $PL^i$  é inteira:  $P^i$  é eliminado por otimalidade.

Como a solução ótima de  $PL^i$  é inteira, segue-se que  $\bar{z}^i \leq z^i$ . Além disso,  $z^i \leq \bar{z}^i$ , portanto,  $z^i = \bar{z}^i$ .

#### Algoritmo branch-and-bound

A partir dos testes de eliminação de problemas, apresenta-se um algoritmo básico do método *B&B*. O nó 0 da árvore *B&B* corresponde ao problema original de programação inteira  $P$ . Um nó não eliminado pelos testes acima e que ainda não foi dividido é chamado nó *ativo*. Estes nós são armazenados em uma lista  $L$ .

*Passo 0* (Inicialização). Faça  $\bar{z} = \infty$ ,  $z^* = -\infty$ ,  $\mathbf{x}^* = \emptyset$ ,  $L = \{P\}$ .

*Passo 1* (Seleção de nó). Selecione o nó ativo  $i$ , associado ao problema  $P^i$ , da lista de nós ativos. Se a lista estiver vazia, vá para o *Passo 6*.

*Passo 2* (Teste de eliminação 1). Se a região factível de  $PL^i$  for vazia, vá para o *Passo 1*.

*Passo 3* (Teste de eliminação 2). Se o valor  $\bar{z}^i$  da solução ótima de  $PL^i$  é tal que  $\bar{z}^i \leq z^*$ , vá para o *Passo 1*.

*Passo 4* (Teste de eliminação 3). Se a solução ótima  $\bar{\mathbf{x}}_i$  de  $PL^i$  é inteira com valor  $\bar{z}^i$ , e se  $\bar{z}^i > z^*$ , atualize  $\mathbf{x}^*$  e  $z^*$ . Elimine nós ativos  $i$  da lista  $L$ , tais que  $\bar{z}^i \leq z^*$ , e volte para o *Passo 1*.

*Passo 5* (Ramificação). Selecione uma variável da solução ótima  $\bar{\mathbf{x}}_i$  de  $PL^i$  com valor não inteiro e divida  $P^i$  em dois problemas. Adicione estes problemas à lista  $L$  e vá para o *Passo 1*.

*Passo 6* (Fim). Se  $z^* = -\infty$ , não existe solução factível; caso contrário, a solução incumbente  $\mathbf{x}^*$  é uma solução ótima.

Nos pacotes comerciais, o usuário pode parar o algoritmo *B&B* com uma lista não vazia e uma solução subótima com tolerância especificada  $\varepsilon$ , como, por exemplo  $\varepsilon = 0,1$ . Seja  $\bar{z}$  o maior

**Quadro 3.4**

	$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$s_6$	
$\bar{z}^4$	0	0	0	0	5	1	13
$x_1$	1	0	0	0	1	0	3
$x_2$	0	1	0	0	0	-1	2
$s_2$	0	0	0	1	-3	2	4
$s_1$	0	0	1	0	-7	-5	2

limitante superior dos nós na lista e seja  $z^*$  tal que  $(\bar{z} - z^*)/\bar{z} \leq \varepsilon$ . Então, a solução  $\mathbf{x}^*$  é  $\varepsilon$ -subótima. Em outras palavras, o valor da solução  $\mathbf{x}^*$  está a, no máximo,  $(100 \times \varepsilon)\%$  do valor da solução ótima.

#### Seleção de nós na árvore *branch-and-bound*

Considere uma árvore *B&B* com a lista de nós ativos. A questão consiste em escolher o próximo nó a ser examinado. Existem duas alternativas: regras *a priori*, que determinam previamente a ordem de escolha dos nós, e regras adaptativas, que determinam o nó a partir de informação dos nós ativos.

A regra *a priori* mais utilizada é a busca em profundidade com *backtracking*, também conhecida como *last-in, first-out*, o que significa que o último nó incluído na lista é o primeiro a ser examinado. Na busca em profundidade, se o nó corrente não é eliminado, então o próximo nó a ser examinado é um de seus filhos. O termo *backtracking* implica que, quando um nó é eliminado, retorna-se ao longo do caminho em direção ao nó raiz até encontrar o primeiro nó que tem um filho ainda não examinado. Ao se especificar que o nó filho à esquerda é examinado em primeiro lugar, então esta regra *a priori* define completamente a ordem de desenvolvimento da árvore.

A busca em profundidade tem duas vantagens:

- A experiência mostra que soluções factíveis são mais prováveis de serem encontradas em níveis profundos em relação ao nó raiz.
- Dada a solução ótima do nó pai, resolve-se o problema de programação linear para os nós filhos através da reotimização pelo algoritmo dual simplex. Além disso, o tamanho máximo da lista de nós ativos, em geral, não é muito grande. Note que o armazenamento dos nós ativos corresponde a guardar os quadros ótimos da resolução dos problemas de programação linear de cada um desses nós.

A grande desvantagem desta regra é que, por não usar informação, tende a gerar uma árvore com muitos nós.

Um exemplo de regra adaptativa muito usada consiste em selecionar o nó que tem o maior limitante superior. Neste caso, tem-se uma expectativa otimista em relação à obtenção de uma nova solução factível com valor máximo em relação a soluções factíveis geradas pelos outros nós. Suponha que  $z^*$  é o valor da solução incumbente atual e que  $z^{**}$  é o valor de uma nova solução incumbente gerada a partir do nó com maior limitante superior, isto é,  $z^{**} > z^*$ . Se o limitante superior deste nó estiver próximo de seu valor ótimo inteiro, então um grande número de nós ativos com limitantes superiores no intervalo  $(z^*, z^{**}]$  é eliminado. Esta discussão mostra que um fator crucial para a eficiência do método *branch-and-bound* é a qualidade do limitante superior, isto é, sua proximidade em relação ao valor ótimo em cada nó.

A vantagem desta regra adaptativa é que, em geral, produz uma árvore *B&B* com um número bem menor de nós quando comparada à busca em profundidade. Por outro lado, sua grande desvantagem é o armazenamento de um número muito grande de nós ativos, o que pode inviabilizar a solução de um problema pelo limite de memória computacional.

Para ilustrar a aplicação das estratégias descritas, considere o problema da mochila

$$\begin{aligned} z &= \max \sum_{j=1}^n p_j x_j \\ &\quad \sum_{j=1}^n a_j x_j \leq b \\ x &\in B^n \end{aligned}$$

e assuma que  $p_j, a_j > 0, \forall j$ .

**Proposição 3.1** Se  $\frac{p_1}{a_1} \geq \dots \geq \frac{p_n}{a_n}$ ,  $\sum_{j=1}^{r-1} a_j \leq b$ ,  $\sum_{j=1}^r a_j > b$ , a solução do problema relaxado de programação linear, em que  $0 \leq x_j \leq 1$ ,  $\forall j$ , é  $x_j = 1$ ,  $j = 1, \dots, r-1$ ,  $x_r = (b - \sum_{j=1}^{r-1} a_j) / a_r$ ,  $x_j = 0$ ,  $j = r+1, \dots, n$ .

**Demonstração.** A restrição linear pode ser escrita como

$$\sum_{j=1}^n a_j x_j + y = b, \quad y \geq 0$$

O problema de programação linear tem variáveis canalizadas,  $0 \leq x_j \leq 1$ ,  $\forall j$  e uma restrição linear. Segue-se que uma variável é básica e  $n$  variáveis são não-básicas com valores em 0 ou 1.

Elegendo  $x_r$  como variável básica, tem-se

$$x_r = \frac{b}{a_r} - \frac{a_1}{a_r} x_1 - \dots - \frac{a_n}{a_r} x_n - \frac{1}{a_r} y$$

Substituindo a expressão anterior na função objetivo da relaxação linear, temos

$$\bar{z} = \max(p_1 - \frac{p_r a_1}{a_r} x_1 + \dots + (p_{r+1} - \frac{p_r a_{r+1}}{a_r}) x_{r+1} + \dots + (p_n - \frac{p_r a_n}{a_r}) x_n - \frac{p_r}{a_r} y + \frac{p_r b}{a_r})$$

Por hipótese,  $\frac{p_1}{a_1} \geq \dots \geq \frac{p_n}{a_n} > 0$ , o que implica que os coeficientes das variáveis  $x_1, \dots, x_{r-1}$  são positivos e os coeficientes das variáveis  $x_{r+1}, \dots, x_n, y$  são negativos. Atribuindo-se o valor 1 para  $x_1, \dots, x_{r-1}$ ,  $(b - \sum_{j=1}^{r-1} a_j) / a_r$  para  $x_r$ , e 0 para  $x_{r+1}, \dots, x_n, y$ , tem-se a solução ótima.

Essa proposição mostra que a solução ótima da relaxação linear é obtida de forma analítica, e isso é usado no exemplo abaixo.

### Exemplo 3.19

$$z = \max 31x_1 + 126x_2 + 131x_3 + 37x_4 + 180x_5 + 170x_6 + 182x_7 + 123x_8 + 160x_9 + 80x_{10}$$

$$13x_1 + 111x_2 + 101x_3 + 27x_4 + 174x_5 + 136x_6 + 146x_7 + 99x_8 + 145x_9 + 76x_{10} \leq 606$$

Note que

$$\begin{aligned} \frac{p_1}{a_1} &= 2,384 > \frac{p_4}{a_4} = 1,370 > \frac{p_3}{a_3} = 1,297 > \frac{p_6}{a_6} = 1,280 > \frac{p_7}{a_7} = 1,246 > \frac{p_8}{a_8} = 1,242 > \frac{p_2}{a_2} = 1,135 > \\ \frac{p_9}{a_9} &= 1,1034 > \frac{p_{10}}{a_{10}} = 1,052 = \frac{p_5}{a_5} = 1,034 \end{aligned}$$

portanto, pela Proposição 3.1, a solução ótima do problema relaxado de programação linear é:  $x_1 = 1$ ;  $x_2 = 0,76$ ;  $x_3 = 1$ ;  $x_4 = 1$ ;  $x_5 = 0$ ;  $x_6 = 1$ ;  $x_7 = 1$ ;  $x_8 = 1$ ;  $x_9 = 0$ ;  $x_{10} = 0$ , com valor de função objetivo  $\bar{z} = 769,35$ . ■

A seguir, o método *B&B* é aplicado a este problema com as estratégias de seleção de nós baseadas em busca em profundidade e busca pelo nó com maior limitante superior.

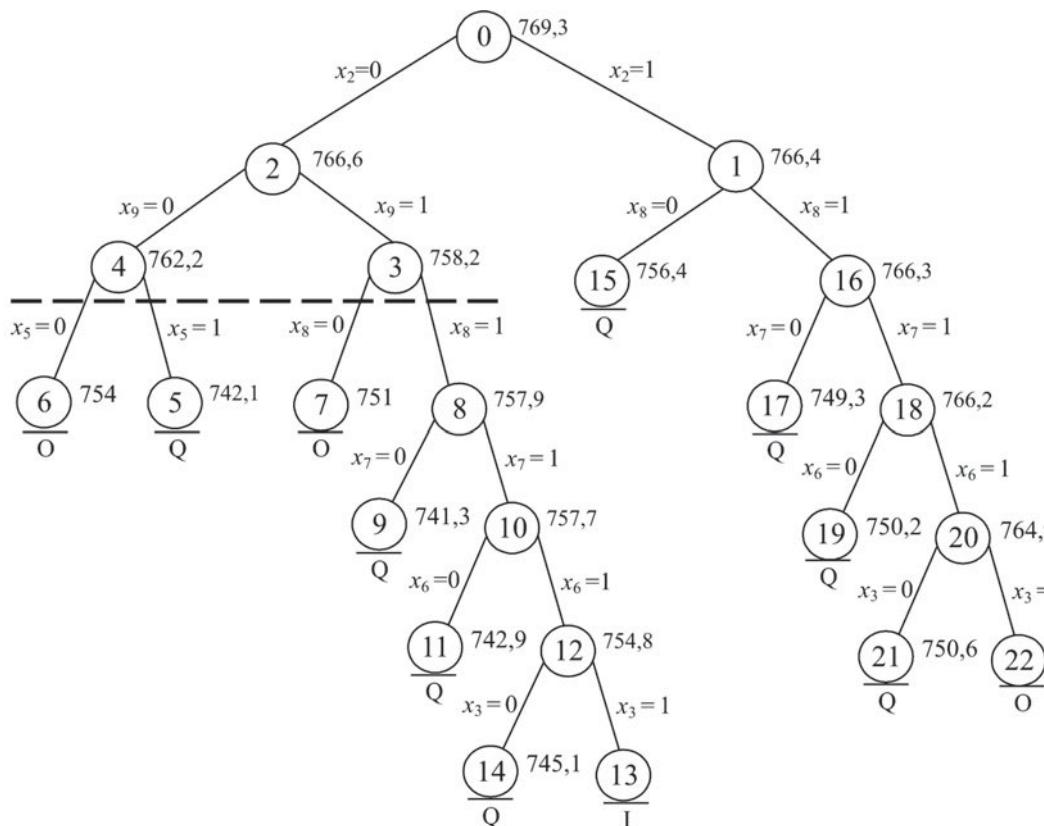
Inicialmente, note que o problema de programação inteira envolve variáveis binárias, portanto, um nó com uma variável com valor fracionário entre 0 e 1 é ramificado em dois nós filhos, com variáveis fixadas em 0 e 1. A Figura 3.39 mostra a árvore *B&B* resultante da

estratégia de busca em profundidade. Como regra, examina-se em primeiro lugar o nó filho com maior limitante superior. As letras I, Q e O nos nós indicam os critérios pelos quais estes são eliminados, a saber, infactibilidade, qualidade ou otimalidade. A solução ótima  $x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 0, x_5 = 0, x_6 = 1, x_7 = 1, x_8 = 1, x_9 = 0, x_{10} = 0$ , com valor 763, é encontrada no nó 22. Note que essa estratégia gera três soluções factíveis.

A árvore relativa à estratégia da busca pelo maior limitante superior é igual à árvore da Figura 3.39, exceto pelos nós que estão abaixo da linha tracejada. Essa estratégia gera apenas uma solução factível, que é a solução ótima.

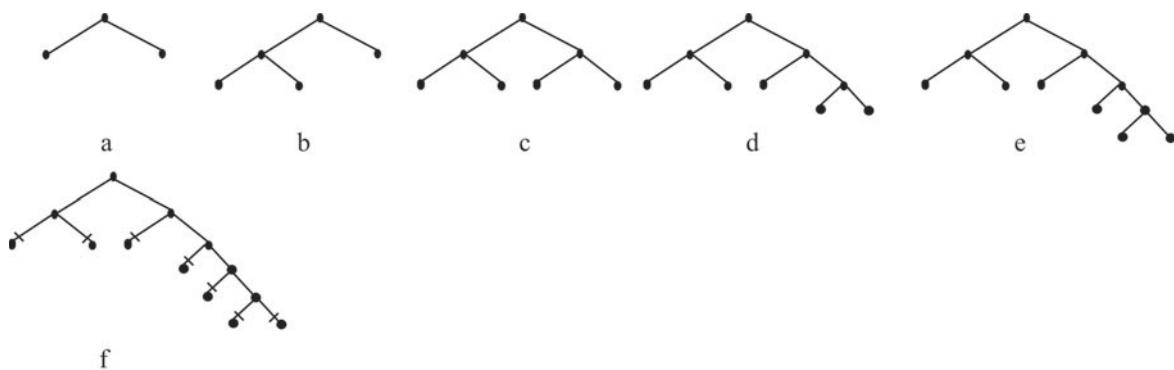
As Figuras 3.40 e 3.41 mostram a evolução de ambas as árvores. Na busca pelo nó de maior limitante superior, armazenam-se no máximo seis nós ativos, enquanto na busca em profundidade armazenam-se no máximo três nós ativos. Os ramos das árvores com traços indicam os nós eliminados. Na Figura 3.40, nenhum nó foi eliminado até a árvore em 3.40(e), que tem seis nós ativos. A Figura 3.41 mostra que a árvore 3.41(b) tem três nós ativos, e as restantes têm menos que três nós ativos.<sup>72</sup>

Este exemplo mostra a vantagem da busca pelo maior limitante superior em termos de número menor de nós criados pela árvore *B&B*. Por outro lado, a estratégia da busca em profundidade gera mais soluções factíveis e armazena menos nós ativos. Na prática, os pacotes comerciais de otimização citados na Seção 3.4 combinam essas duas estratégias e possibilitam também que o usuário selecione outras estratégias disponíveis.<sup>72</sup>

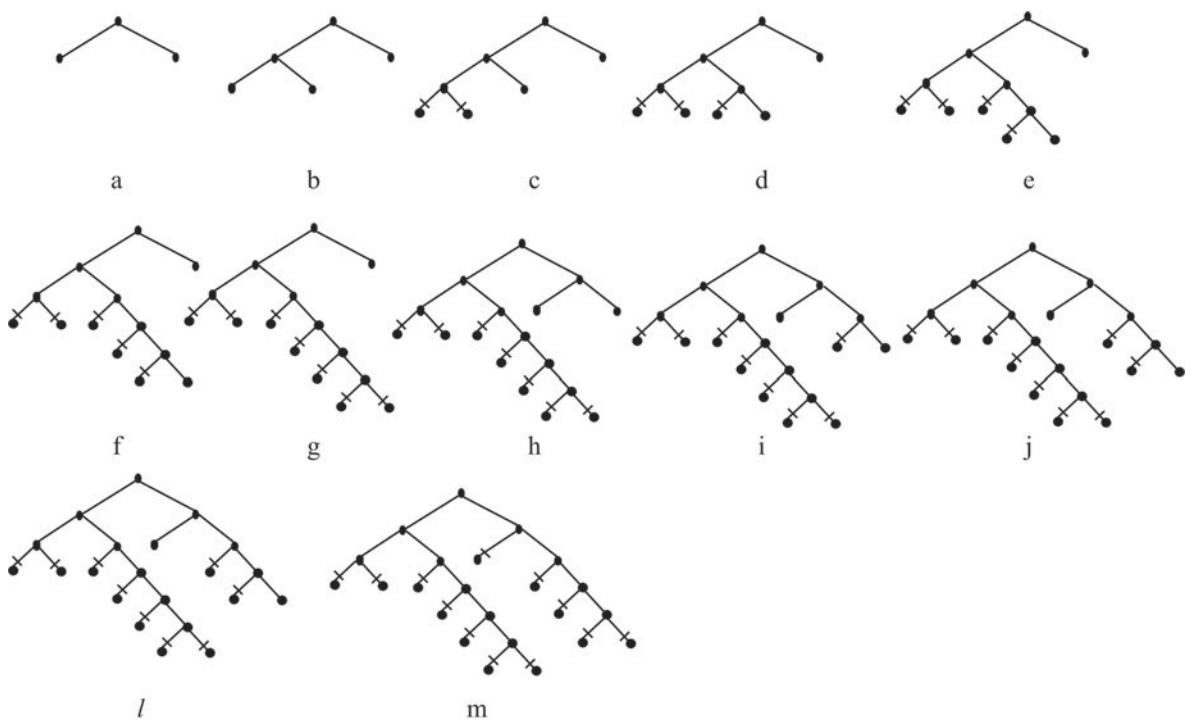


**Figura 3.39** Árvores para duas estratégias de busca.

<sup>72</sup> Algoritmos do tipo *branch-and-bound*, ou seja, que ramificam e podam a árvore de busca, podem ser desenvolvidos para resolver os mais diversos tipos de problemas. Por exemplo, em Yanasse (1994) é apresentado um algoritmo que seleciona o melhor tamanho dos objetos em um problema de corte bidimensional.



**Figura 3.40** Evolução da busca pelo maior limitante superior para o problema da mochila.



**Figura 3.41** Evolução da busca em profundidade para o problema da mochila.

### 3.9.2 Algoritmo de planos de corte de Gomory

Algoritmos de planos de corte buscam obter uma aproximação da envoltória convexa da região factível de um problema de programação inteira que contenha um ponto extremo correspondente a uma solução ótima. Essa aproximação é obtida por meio de cortes ou desigualdades válidas, definidas a seguir.

**Definição 3.5** Uma desigualdade  $\phi\mathbf{x} \leq \phi_0$  é uma desigualdade válida para  $X \subseteq R^n$  se  $\phi\mathbf{x} \leq \phi_0$  para todo  $\mathbf{x} \in X$ .

Em palavras, uma desigualdade é válida se o conjunto  $X$  situa-se em um dos semi-espacos definidos pelo hiperplano  $\phi\mathbf{x} = \phi_0$ . Desigualdades válidas em programação inteira são geradas a partir de uma simples observação.

**Proposição 3.2** A desigualdade  $x \leq \lfloor b \rfloor$  é válida para  $X = \{x \in Z^1 : x \leq b\}$ .

O procedimento geral para a construção de uma desigualdade válida para o conjunto  $X = \{\mathbf{x} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \in Z_+^n\}$ , tal que  $\mathbf{A}$  é uma matriz  $m \times n$  com colunas  $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$  e  $\mathbf{u} \in R_+^m$ , conhecido como procedimento de Chvátal-Gomory, é descrito a seguir.

- A desigualdade

$$\sum_{j=1}^n \mathbf{u}^T \mathbf{a}_j x_j \leq \mathbf{u}^T \mathbf{b}$$

é válida para  $X$ , pois  $\mathbf{u} \geq 0$  e  $\sum_{j=1}^n \mathbf{a}_j x_j \leq \mathbf{b}$ .

- A desigualdade

$$\sum_{j=1}^n \lfloor \mathbf{u}^T \mathbf{a}_j \rfloor x_j \leq \mathbf{u}^T \mathbf{b}$$

é válida para  $X$ , pois  $\mathbf{x} \geq 0$ .

- A desigualdade

$$\sum_{j=1}^n \lfloor \mathbf{u}^T \mathbf{a}_j \rfloor x_j \leq \lfloor \mathbf{u}^T \mathbf{b} \rfloor$$

é válida para  $X$ , pois  $\mathbf{x}$  é inteiro, portanto  $\sum_{j=1}^n \lfloor \mathbf{u}^T \mathbf{a}_j \rfloor x_j$  é inteiro.

Pode-se demonstrar que toda desigualdade válida para  $X$  pode ser obtida pela aplicação do procedimento de Chvátal-Gomory por um número finito de vezes.

Os exemplos a seguir ilustram a geração de desigualdades válidas em programação inteira.

**Exemplo 3.20** Identifique uma desigualdade para cortar o ponto  $(0, 0, 0, 35/6)$  do conjunto:

$$X = \{x \in Z_+^4 : 5x_1 + 7x_2 + 4x_3 + 6x_4 \leq 35\}$$

Multiplicando a desigualdade acima por  $u = 1/6$  e aplicando o procedimento de Chvátal-Gomory, tem-se

$$\lfloor 5/6 \rfloor x_1 + \lfloor 7/6 \rfloor x_2 + \lfloor 4/6 \rfloor x_3 + x_4 \leq \lfloor 35/6 \rfloor$$

portanto, a desigualdade válida  $x_2 + x_4 \leq 5$  corta o ponto. ■

**Exemplo 3.21** Considere o conjunto  $X = \{7x_1 - 5x_2 \leq 13, 3x_1 + 2x_2 \leq 17, x \in Z_+^2\}$  associado às restrições do Exemplo 3.17.

- Considere a soma ponderada das restrições com  $\mathbf{u} = (3/20, 0)$ , que resulta na desigualdade

$$\frac{21}{20}x_1 - \frac{3}{4}x_2 \leq \frac{39}{20}$$

$$\left\lfloor \frac{21}{20} \right\rfloor x_1 + \left\lfloor -\frac{3}{4} \right\rfloor x_2 \leq \left\lfloor \frac{39}{20} \right\rfloor$$

- A desigualdade válida é, então,  $x_2 - x_1 \leq 1$ , mostrada na Figura 3.42.

*Corte de Gomory*

Seja o problema de programação inteira

$$(P) \quad z = \max \left\{ \mathbf{c}^T \mathbf{x} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \in Z_+^n \right\}$$

e sua relaxação linear

$$(PL) \quad \bar{z} = \max \left\{ \mathbf{c}^T \mathbf{x} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \in R_+^n \right\}$$

Considere a solução ótima do problema relaxado de programação linear, em que  $x_{B_i}$ ,  $i = 1, \dots, m$  representam as variáveis básicas e  $NB$  denota o conjunto das variáveis não-básicas. Utilizando a notação (3.11) e (3.12), o problema  $PL$  pode ser reescrito como

$$\begin{aligned} z &= \max \hat{a}_{00} + \sum_{j \in NB} \hat{a}_{0j} x_j \\ x_{B_i} + \sum_{j \in NB} \hat{a}_{ij} x_j &= \hat{a}_{i0}, \quad i = 1, \dots, m \\ \mathbf{x} &\in R_+^n \end{aligned} \tag{3.17}$$

com  $\hat{a}_{0j} \leq 0$  para  $j \in NB$  e  $\hat{a}_{i0} \geq 0$ ,  $i = 1, \dots, m$ .

Se a solução ótima não é inteira, então existe uma linha  $i$  com  $y_{i0}$  não inteiro. O corte de Chvátal-Gomory para essa linha  $i$  é

$$x_{B_i} + \sum_{j \in NB} \lfloor \hat{a}_{ij} \rfloor x_j \leq \lfloor \hat{a}_{i0} \rfloor \tag{3.18}$$

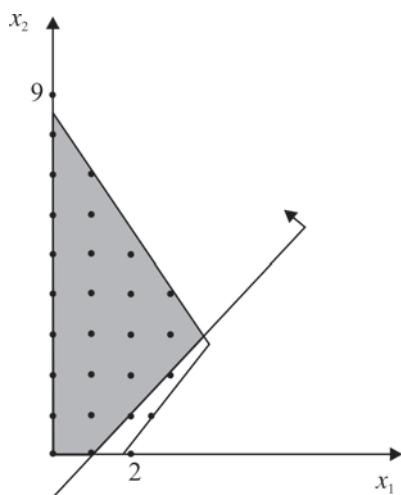
Substituindo  $x_{B_i}$  de (3.17) em (3.18), tem-se a desigualdade

$$\sum_{j \in NB} (\hat{a}_{ij} - \lfloor \hat{a}_{ij} \rfloor) x_j \geq \hat{a}_{i0} - \lfloor \hat{a}_{i0} \rfloor$$

ou

$$\sum_{j \in NB} f_{ij} x_j \geq f_{i0} \tag{3.19}$$

com  $f_{ij} = \hat{a}_{ij} - \lfloor \hat{a}_{ij} \rfloor$ ,  $0 \leq f_{ij} < 1$ , e  $f_{i0} = \hat{a}_{i0} - \lfloor \hat{a}_{i0} \rfloor$ ,  $0 < f_{i0} < 1$ . A desigualdade (3.19) corresponde ao corte de Gomory.



**Figura 3.42** Ilustração de desigualdade válida.

Como na solução básica ótima  $x_j = 0$  para todo  $j \in NB$ , a desigualdade (3.19) corta a solução básica ótima. Introduzindo a variável de sobra  $s_i \geq 0$  em (3.19), tem-se

$$s_i = \sum_{j \in NB} f_{ij}x_j - f_{i0}$$

De (3.17), segue-se que

$$x_{B_i} = f_{i0} - \sum_{j \in NB} f_{ij}x_j + \lfloor \hat{a}_{i0} \rfloor - \sum_{j \in NB} \lfloor \hat{a}_{ij} \rfloor x_j$$

Reescrevendo a equação anterior como

$$x_{B_i} = -s_i + \lfloor \hat{a}_{i0} \rfloor - \sum_{j \in NB} \lfloor \hat{a}_{ij} \rfloor x_j$$

segue-se que  $s_i$  é não-negativa e inteira, dado que os outros termos da equação são inteiros.

### *Algoritmo de Gomory*

*Passo 1* (Inicialização). Faça  $k = 0$  e  $PL^0 = PL$ , em que  $PL$  é a relaxação linear do problema  $P$ .

*Passo 2* (Reotimização). Resolva o problema linear

$$\bar{z}^k = \max \{ \mathbf{c}^T \mathbf{x} : \mathbf{x} \in PL^k \}$$

pelo método dual simplex.

*Passo 3* (Ottimalidade). Se a solução for inteira, então é uma solução ótima de  $P$ . Caso contrário, vá para o Passo 3.

*Passo 4* (Corte). Escolha uma linha  $i$  com  $f_{i0} > 0$ , construa o corte de Gomory e insira-o no fim do quadro ótimo de  $PL^k$ . Faça  $k = k + 1$  e vá para o Passo 2.

Note que este algoritmo gera uma seqüência de problemas de programação linear  $PL^i$ ,  $i = 0, 1, \dots, n$ , em que  $X = \bar{X}^0 \supseteq \bar{X}^1 \supseteq \dots \supseteq \bar{X}^n$ , e  $\bar{z} = \bar{z}^0 \geq \bar{z}^1 \geq \dots \geq \bar{z}^n \geq z$ , de modo que  $\bar{X}^i$  e  $\bar{z}^i$  representam, respectivamente, a região factível e a solução ótima de  $PL^i$ . Demonstra-se que este algoritmo, sob algumas hipóteses, converge em um número finito de passos. Uma desvantagem deste algoritmo é que a primeira solução factível gerada é a solução ótima.

Uma regra razoável no Passo 4 do algoritmo, visando reduzir o número de cortes necessários para resolver o problema  $(P)$ , é escolher a linha  $r$ , tal que  $f_{r0} = \max_i f_{i0}$ ,  $f_{i0} > 0$ , que corresponde à linha cuja variável apresenta o maior desvio entre seu valor fracionário e o maior número inteiro.

### **Exemplo 3.22**

Considere o problema do Exemplo 3.18:

$$(P) \quad \begin{aligned} z &= \max 5x_1 - x_2 \\ 7x_1 - 5x_2 + s_1 &= 13 \\ 3x_1 + 2x_2 + s_2 &= 17 \\ \mathbf{x}, \mathbf{s} &\in Z_+^2 \end{aligned}$$

A solução ótima do problema relaxado  $PL$  com  $\mathbf{x}, \mathbf{s} \in R_+^2$  é mostrada no Quadro 3.5.

A Figura 3.34 mostra a região factível do problema  $P$ , a direção do gradiente da função objetivo  $(5, -1)$  e a solução ótima real do problema relaxado  $PL$ ,  $\bar{\mathbf{x}} = (3\frac{24}{29}, 2\frac{22}{29})$ , com valor  $\bar{z} = 16\frac{11}{29}$ , que é um limitante superior para o valor ótimo  $z$ .

Como  $f_{10} = \frac{24}{29} > f_{20} = \frac{22}{29}$ , escolhe-se a linha da variável  $x_1$  e, utilizando a desigualdade de Gomory, obtém-se o primeiro corte

$$\frac{2}{29}s_1 + \frac{5}{29}s_2 \geq \frac{24}{29} \quad (3.20)$$

A partir da formulação  $P$ , pode-se escrever este corte em termos das variáveis originais do problema, resultando em  $x_1 \leq 3$ . Introduzindo uma variável de sobra em (3.20), tem-se

$$s_3 - 2s_1 - 5s_2 = -24, \quad s_3 \geq 0 \quad (3.21)$$

Incluindo a restrição (3.21) no fim do Quadro 3.5, segue-se que  $s_3 = -24$ , pois  $s_1$  e  $s_2$  são não-básicas. Reotimizando através do dual simplex, obtém-se o Quadro 3.6.

Escolhendo a linha  $s_2$ , obtém-se o segundo corte

$$2s_1 + s_3 \geq 4 \quad (3.22)$$

que corresponde à desigualdade  $3x_1 - 2x_2 \leq 5$ , em termos das variáveis originais. Inserindo uma variável de sobra em (3.22), tem-se

$$s_4 - 2s_1 - s_3 = -4 \quad (3.23)$$

Introduzindo a restrição (3.23) no Quadro 3.6 e reotimizando através do dual simplex, obtém-se o Quadro 3.7 com a solução ótima  $\mathbf{x} = (3, 2)$ . Os dois cortes de Gomory são mostrados na Figura 3.43.

**Quadro 3.5**

	$x_1$	$x_2$	$s_1$	$s_2$	
$\bar{z}$	0	0	$\frac{13}{29}$	$\frac{18}{29}$	$16\frac{11}{29}$
$x_1$	1	0	$\frac{2}{29}$	$\frac{5}{29}$	$3\frac{24}{29}$
$x_2$	0	1	$-\frac{3}{29}$	$\frac{7}{29}$	$2\frac{22}{29}$

**Quadro 3.6**

	$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	
$\bar{z}$	0	0	$\frac{1}{5}$	0	$3\frac{3}{5}$	$13\frac{2}{5}$
$x_1$	1	0	0	0	1	3
$x_2$	0	1	$-\frac{1}{5}$	0	$1\frac{2}{5}$	$1\frac{3}{5}$
$s_2$	0	0	$\frac{2}{5}$	1	$-5\frac{4}{5}$	$4\frac{4}{5}$

Quadro 3.7

	$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$s_4$	
$\bar{z}$	0	0	0	0	$3\frac{1}{2}$	$\frac{1}{2}$	13
$x_1$	1	0	0	0	1	0	3
$x_2$	0	1	0	0	$1\frac{1}{2}$	$-\frac{1}{2}$	2
$s_2$	0	0	0	1	-6	1	4
$s_1$	0	0	1	0	$\frac{1}{2}$	$-2\frac{1}{2}$	2

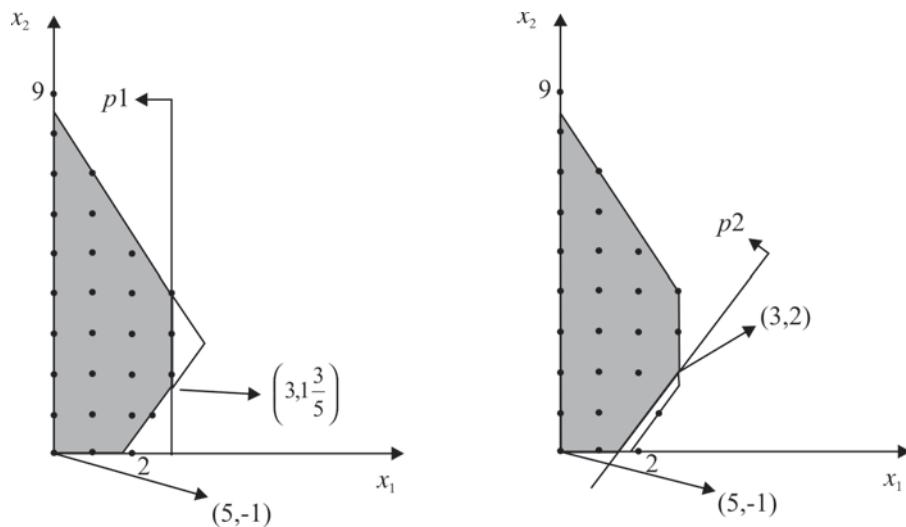


Figura 3.43 Cortes de Gomory.

### 3.9.3 Método *branch-and-cut*

Este método combina as estratégias dos métodos *branch-and-bound* e de planos de cortes, com o objetivo de reduzir o número de nós na árvore *B&B*. Em cada nó da árvore *B&B*, adicionam-se desigualdades válidas de modo a obter um limitante superior mais apertado no nó. As desigualdades de Gomory para problemas inteiros e inteiros mistos foram as primeiras propostas na literatura. Hoje, existem vários tipos de desigualdades válidas para problemas genéricos de programação inteira mista e desigualdades válidas para problemas específicos.

Para ilustrar o poder da adição de desigualdades na redução do número de nós na árvore, considere o seguinte exemplo, em que desigualdades de Gomory foram introduzidas somente no nó raiz 0.

**Exemplo 3.23**

$$z = \max \quad 25x_1 + 2x_2 + 31x_3 + 30x_4$$

$$14x_1 + 25x_2 + 14x_3 + 6x_4 \leq 146$$

$$19x_1 + 30x_2 + 24x_3 + 29x_4 \leq 239$$

$$x \in Z_+^4$$

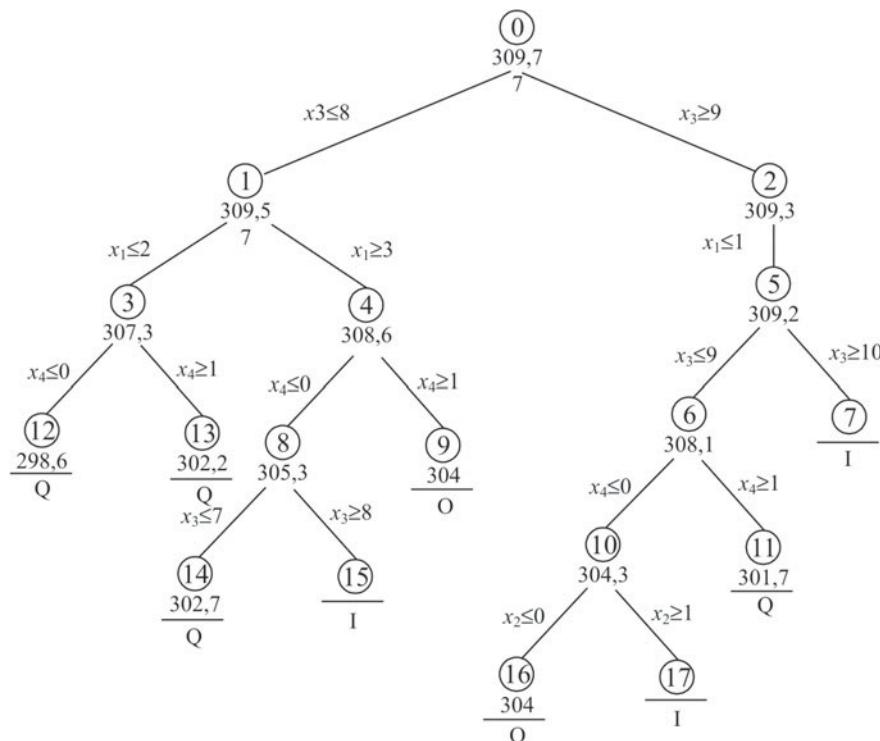
A Figura 3.44 mostra a árvore *B&B* com a estratégia de busca pelo nó de maior limitante superior. As letras I, Q e O indicam que o nó foi eliminado por infactibilidade, qualidade ou otimalidade. Duas soluções ótimas foram encontradas:

$$x_1 = 1, x_2 = 0, x_3 = 9, x_4 = 0, \text{ e } x_1 = 6, x_2 = 0, x_3 = 4, x_4 = 1, \text{ com valor } z = 304$$

O Quadro 3.8 correspondente à solução óptima do problema relaxado no nó 0:

**Quadro 3.8**

	$x_1$	$x_3$	$x_2$	$x_4$	$s_1$	$s_2$	
$\bar{z}$	0	0	$37\frac{13}{14}$	$5\frac{26}{35}$	$\frac{11}{70}$	$1\frac{1}{5}$	$309\frac{26}{35}$
$x_1$	1	0	$2\frac{4}{7}$	$-3\frac{26}{35}$	$\frac{12}{35}$	$-\frac{1}{5}$	$2\frac{9}{35}$
$x_3$	0	1	$-\frac{11}{14}$	$4\frac{6}{35}$	$-\frac{19}{70}$	$\frac{1}{5}$	$8\frac{6}{35}$

**Figura 3.44** Árvore *B&B* com busca pelo nó de maior limitante superior.

Escolhendo a linha  $x_1$ , obtém-se o corte

$$\frac{4}{7}x_2 + \frac{9}{35}x_4 + \frac{12}{35}s_1 + \frac{4}{5}s_2 - s_3 = \frac{9}{35} \quad (3.24)$$

ou, em termos das variáveis originais,

$$20x_1 + 32x_2 + 24x_3 + 25x_4 \leq 241$$

Incluindo o corte (3.24) no fim do Quadro 3.8 e reotimizando através do dual simplex, obtém-se o Quadro 3.9.

**Quadro 3.9**

	$x_1$	$x_3$	$x_5$	$x_2$	$x_4$	$s_2$	$s_3$	
$\bar{z}$	0	0	0	$37\frac{2}{3}$	$5\frac{5}{8}$	$\frac{5}{6}$	$\frac{11}{24}$	$309\frac{5}{8}$
$x_1$	1	0	0	2	-4	-1	1	2
$x_3$	0	1	0	$-\frac{1}{3}$	$4\frac{3}{8}$	$\frac{5}{6}$	$-\frac{19}{24}$	$8\frac{3}{8}$
$s_1$	0	0	1	$1\frac{2}{3}$	$\frac{3}{4}$	$2\frac{1}{3}$	$-2\frac{11}{12}$	$\frac{3}{4}$

Escolhendo a linha  $x_3$ , obtém-se o corte

$$\frac{2}{3}x_2 + \frac{3}{8}x_4 + \frac{5}{6}s_2 + \frac{5}{24}s_3 - s_4 = \frac{3}{8} \quad (3.25)$$

ou, em termos de variáveis originais,

$$20x_1 + 31x_2 + 25x_3 + 29x_4 \leq 249$$

Incluindo o corte (3.25) no Quadro 3.9 e reotimizando através do dual simplex, obtém-se o Quadro 3.10.

Escolhendo a linha  $x_7$ , obtém-se o corte

$$\frac{11}{14}x_2 + \frac{3}{7}x_4 + \frac{1}{14}x_5 - s_5 = \frac{3}{7} \quad (3.26)$$

ou, em termos de variáveis originais,

$$x_1 + x_2 + x_3 \leq 10$$

Incluindo o corte (3.26) no Quadro 3.10 e reotimizando através do dual simplex, obtém-se o Quadro 3.11.

O limitante superior no Quadro 3.8 é  $309\frac{26}{35}$ , ao passo que o limitante superior no Quadro 3.11 após a inclusão de três cortes, é reduzido para  $308\frac{4}{5}$ . A partir do Quadro 3.11, aplica-se o método *B&B* e a árvore resultante é mostrada na Figura 3.45. As duas soluções ótimas também foram encontradas nesta árvore, que tem somente cinco nós, comparados aos 18 nós da árvore da Figura 3.44.

**Quadro 3.10**

$x_1$	$x_3$	$x_6$	$x_7$	$x_2$	$x_4$	$s_1$	$s_4$	
$\bar{z}$	0	0	0	0	$36\frac{69}{70}$	$5\frac{8}{35}$	$\frac{1}{14}$	$1\frac{1}{5}$
$x_1$	1	0	0	0	$2\frac{51}{70}$	$-3\frac{23}{35}$	$\frac{5}{14}$	$-\frac{1}{5}$
$x_3$	0	1	0	0	$-\frac{33}{35}$	$4\frac{3}{35}$	$-\frac{2}{7}$	$\frac{1}{5}$
$s_2$	0	0	1	0	$\frac{2}{35}$	$\frac{3}{35}$	$-\frac{2}{7}$	$-\frac{4}{5}$
$s_3$	0	0	0	1	$\frac{11}{14}$	$\frac{3}{7}$	$\frac{1}{14}$	$-1$

**Quadro 3.11**

	$x_1$	$x_3$	$s_3$	$s_2$	$s_1$	$x_2$	$x_4$	$s_4$	$s_5$
$\bar{z}$	0	0	0	0	0	$36\frac{1}{5}$	$4\frac{4}{5}$	$1\frac{1}{5}$	$1$
$x_1$	1	0	0	0	0	$-1\frac{1}{5}$	$-5\frac{4}{5}$	$-\frac{1}{5}$	$5$
$x_3$	0	1	0	0	0	$2\frac{1}{5}$	$5\frac{4}{5}$	$\frac{1}{5}$	$-4$
$s_3$	0	0	1	0	0	$3\frac{1}{5}$	$1\frac{4}{5}$	$-\frac{4}{5}$	$\frac{9}{5}$
$s_2$	0	0	0	1	0	0	0	$-1$	1
$s_1$	0	0	0	0	1	11	6	0	$-14$

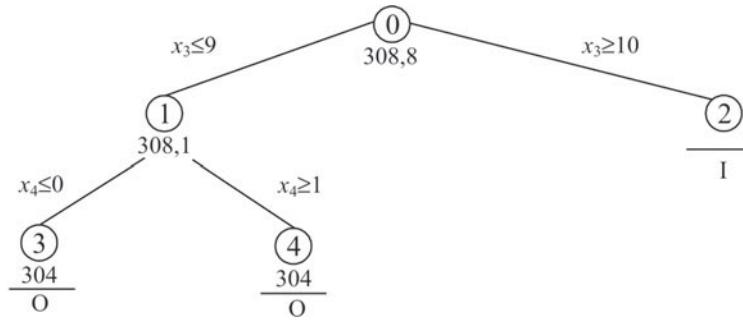
### Algoritmo branch-and-cut

O algoritmo *branch-and-cut* é semelhante ao algoritmo *B&B*, com a inclusão de  $k$  cortes em cada nó cuja relaxação linear é factível, tal que  $k$  é um parâmetro.<sup>73</sup> A inicialização do algoritmo tem uma fase de pré-processamento, que é explicada a seguir.

*Passo 0* (Inicialização). Faça  $\bar{z} = \infty$ ,  $z^* = -\infty$ ,  $\mathbf{x}^* = \emptyset$ . Pré-processe o problema inicial e coloque-o na lista  $L = \{P\}$ .

*Passo 1* (Seleção de nó). Selecione o nó ativo  $i$ , associado ao problema  $P^i$ , da lista de nós ativos. Se a lista estiver vazia, vá para o *Passo 7*.

<sup>73</sup> Uma aplicação de um algoritmo *branch-and-cut* para rearranjos de fragmentos de DNA pode ser vista em Ferreira *et al.* (2002).



**Figura 3.45** Árvore B&B com cortes no nó 0 e busca pelo nó de maior limitante superior.

*Passo 3* (Teste de eliminação 1). Se a região factível de  $PL^i$  estiver vazia, vá para o *Passo 1*.

*Passo 4* (Corte). Tente eliminar a solução ótima de  $PL^i$ . Se isso não for possível, faça  $k = 0$ , e vá para o *Passo 5*. Caso contrário, adicione  $k$  cortes a  $PL^i$  de modo a obter a formulação  $PL^{ik}$ .

*Passo 5* (Teste de eliminação 2). Se o valor  $\bar{z}^{ik}$  da solução ótima de  $PL^{ik}$  for tal que  $\bar{z}^{ik} \leq z^*$ , vá para o *Passo 1*.

*Passo 6* (Teste de eliminação 3). Se a solução ótima  $\bar{x}^{ik}$  de  $PL^{ik}$  for inteira com valor  $\bar{z}^{ik}$ , e se  $\bar{z}^{ik} > z^*$ , atualize  $\mathbf{x}^*$  e  $z^*$ . Elimine nós ativos  $i$  da lista  $L$ , tais que  $\bar{z}^i \leq z^*$ , e volte para o *Passo 1*.

*Passo 7* (Ramificação). Selecione uma variável da solução ótima  $\bar{x}^{ik}$  de  $PL^{ik}$  com valor não inteiro e divida  $P^{ik}$  em dois problemas. Adicione estes problemas à lista  $L$  e vá para o *Passo 1*.

*Passo 8* (Fim). Se  $z^* = -\infty$ , não existe solução factível; caso contrário, a solução incumbente  $\mathbf{x}^*$  é uma solução ótima.

### 3.10\* PRÉ-PROCESSAMENTO

Pacotes comerciais de otimização possuem um módulo de pré-processamento que verifica a formulação original de um problema. A meta é detectar rapidamente variáveis e restrições redundantes e apertar os limitantes de variáveis. Se o problema resultante de programação linear/inteira é menor/mais apertado, então, provavelmente, será resolvido com mais rapidez. Isso é fundamental em problemas grandes, pois o método *branch-and-cut* pode exigir a resolução de dezenas ou centenas de milhares de problemas de programação linear.

O exemplo a seguir ilustra o pré-processamento em programação linear.

#### Exemplo 3.24

$$\begin{aligned}
 \max \quad & 4x_1 + 3x_2 - 2x_3 \\
 \text{s.t.} \quad & 8x_1 - 5x_2 + 10x_3 \leq 20 \\
 & 10x_1 + 4x_2 - 2x_3 \geq 10 \\
 & x_1 + x_2 + x_3 \leq 5 \\
 & 0 \leq x_1 \leq 3; \quad 0 \leq x_2 \leq 1; \quad 1 \leq x_3
 \end{aligned}$$

*Apertando limitantes.* Isolando a variável  $x_1$  na primeira restrição e usando os limitantes  $x_2 \leq 1$  e  $-x_3 \leq -1$  obtém-se

$$8x_1 \leq 20 + 5x_2 - 10x_3 \leq 20 + 5 \times 1 - 10 \times 1 = 15$$

e o limitante apertado é  $x_1 \leq \frac{15}{8}$ .

Isolando a variável  $x_3$ , obtém-se

$$10x_3 \leq 20 + 5x_2 - 8x_1 \leq 20 + 5 \times 1 - 5 \times 0 = 25$$

e o limitante apertado é  $x_3 \leq \frac{5}{2}$ .

Isolando a variável  $x_2$ , obtém-se

$$5x_2 \geq 8x_1 + 10x_3 - 20 \geq 8 \times 0 + 10 \times 1 - 20 = -10$$

Neste caso, o limitante  $x_2 \geq 0$  não muda.

Isolando agora  $x_1$  na segunda restrição, obtém-se

$$10x_1 \geq 10 + 2x_3 - 4x_2 \geq 10 + 2 \times 1 - 4 \times 1 = 8$$

e o limitante apertado é  $x_1 \geq \frac{4}{5}$ . A segunda restrição não causa mudança nos limitantes de  $x_2$  e  $x_3$ .

Como alguns limitantes foram alterados, retorna-se à primeira restrição. Isolando a variável  $x_3$ , obtém-se

$$10x_3 \leq 20 + 5x_2 - 8x_1 \leq 20 + 5 \times 1 - 8 \times \frac{4}{5} = \frac{93}{5}$$

e o novo limitante apertado é  $x_3 \leq \frac{93}{50}$ . Os demais limitantes não mais se alteram.

### *Restrições redundantes*

Usando os novos limitantes na terceira restrição, segue-se que

$$\frac{15}{8} + 1 + \frac{93}{50} < 5$$

que implica que esta restrição é redundante e pode ser descartada. O problema torna-se, então,

$$\begin{aligned} \max \quad & 4x_1 + 3x_2 - 2x_3 \\ & 8x_1 - 5x_2 + 10x_3 \leq 20 \\ & 10x_1 + 4x_2 - 2x_3 \geq 10 \\ & \frac{4}{5} \leq x_1 \leq \frac{15}{8}; \quad 0 \leq x_2 \leq 1; \quad 1 \leq x_3 \leq \frac{93}{50} \end{aligned}$$

### *Fixação de variáveis*

Note que o acréscimo de  $x_2$  torna as restrições mais folgadas, e como seu coeficiente na função objetivo é positivo, isso sugere que  $x_2$  assume o valor de seu limitante superior, igual a 1. Multiplique a segunda restrição por -1, e sejam  $u_1$  e  $u_2$  variáveis duais não-negativas associadas às duas restrições. Seja  $u_3$  a variável dual não-negativa associada à restrição  $x_2 \leq 1$ . Tem-se, então, a seguinte restrição do problema dual

$$-5u_1 - 4u_2 + u_3 \geq 3$$

A desigualdade acima implica que  $u_3 > 0$  e, pelas folgas complementares, tem-se que  $x_2 = 1$ . De forma semelhante, ao se decrescer  $x_3$ , tem-se que as restrições ficam mais folgadas. Como seu coeficiente na função objetivo é negativo, isso sugere que  $x_3$  assume o valor de seu limitante inferior, igual a 1. Seja  $u_4$  a variável dual não-positiva associada à restrição  $x_3 \geq 1$ . Se  $u_4 = 0$  na solução dual ótima, então, pelas folgas complementares, tem-se que o valor ótimo de  $x_3$  situa-se no intervalo  $[1, \frac{93}{50}]$ . No entanto,  $x_3 > 1$  é um absurdo, pois o coeficiente de  $x_3$  na função objetivo é negativo e o valor de  $x_3$  pode ser reduzido sem violar as restrições. Se  $u_4 < 0$  na solução dual ótima, então, pelas folgas complementares,  $x_3 = 1$ . O problema de programação linear reduz-se, então, a

$$\max \{4x_1 : \frac{4}{5} \leq x_1 \leq \frac{15}{8}\}$$

Essas idéias estão formalizadas a seguir.

**Proposição 3.3** Considere o conjunto

$$S = \{x : a_0x_0 + \sum_{j=1}^n a_jx_j \leq b, \quad l_j \leq x_j \leq u_j, \quad j = 1, \dots, n\}$$

- *Limitantes em variáveis.* Se  $a_0 > 0$ , então

$$x_0 \leq (b - \sum_{j:a_j>0} a_jl_j - \sum_{j:a_j<0} a_ju_j) / a_0$$

e, se  $a_0 < 0$ , então

$$x_0 \geq (b - \sum_{j:a_j>0} a_jl_j - \sum_{j:a_j<0} a_ju_j) / a_0$$

- *Redundância.* A restrição  $a_0x_0 + \sum_{j=1}^n a_jx_j \leq b$  é redundante se

$$\sum_{j:a_j>0} a_ju_j + \sum_{j:a_j<0} a_jl_j \leq b$$

- *Infactibilidade.*  $S = \emptyset$  se

$$\sum_{j:a_j>0} a_jl_j + \sum_{j:a_j<0} a_ju_j > b$$

- *Fixação de variáveis.* Para um problema de maximização na forma

$$\max \{cx : Ax \leq b, l \leq x \leq u\}$$

se  $a_{ij} > 0, i = 1, \dots, m$  e  $c_j < 0$ , então  $x_j = l_j$ . Se  $a_{ij} \leq 0, i = 1, \dots, m$  e  $c_j > 0$ , então  $x_j = u_j$ .

Para problemas binários, buscam-se restrições lógicas envolvendo uma ou duas variáveis e adicionam-se as restrições ao problema, ou estas são usadas para fixar algumas variáveis. Este procedimento é ilustrado a seguir.

**Exemplo 3.25** Considere o conjunto de restrições envolvendo quatro variáveis 0-1:

$$\begin{aligned} 2x_1 + x_3 + x_4 &\geq 2 \\ x_1 - 2x_2 + 5x_3 + 2x_4 &\leq 4 \\ -2x_1 + 6x_2 + x_3 - x_4 &\leq 3 \\ 5x_2 - 3x_4 &\geq 0 \end{aligned}$$

Se na linha 1,  $x_3 = 0$ , isso implica  $x_1 = 1$ , que conduz à desigualdade  $x_1 + x_3 \geq 1$ . De modo análogo,  $x_1 + x_4 \geq 1$ .

Se na linha 2,  $x_3 = 1$ , isso implica  $x_2 = 1$ , que leva à desigualdade  $x_3 \leq x_2$ . Esta restrição é infactível se  $x_3 = x_4 = 1$ , que leva à desigualdade  $x_3 + x_4 \leq 1$ .

Se na linha 3,  $x_2 = 1$ , isso implica  $x_1 = 1$  e  $x_4 = 1$ . Estas implicações podem ser formuladas pelas desigualdades  $x_2 \leq x_1$  e  $x_2 \leq x_4$ . Esta restrição é infactível se  $x_2 = x_3 = 1$ , que resulta na desigualdade  $x_2 + x_3 \leq 1$ .

Se na linha 4,  $x_4 = 1$ , isso implica  $x_2 = 1$ , que leva à desigualdade  $x_2 \geq x_4$ .

*Combinação de pares de desigualdades lógicas.* Das linhas 3 e 4, tem-se  $x_2 \leq x_4$  e  $x_4 \geq x_2$ , que implica  $x_2 = x_4$ . Das linhas 2 e 3, têm-se  $x_3 \leq x_2$  e  $x_2 + x_3 \leq 1$ , que implicam  $x_3 = 0$ . A partir de  $x_1 + x_3 \geq 1$ , na linha 1, e  $x_3 = 0$ , obtém-se  $x_1 = 1$ .

*Simplificação.* Substituindo  $x_1 = 1$ ,  $x_2 = x_4$ ,  $x_3 = 0$ , todas as restrições são redundantes, restando  $x_2 \in \{0,1\}$ , portanto, as únicas soluções factíveis são  $(1,1,0,1)$  e  $(1,0,0,0)$ .

Algumas implicações lógicas são formalizadas a seguir.

**Proposição 3.4** Considere o conjunto

$$X = \{x \in B^n : \sum_{j=1}^n a_j x_j \leq b\}$$

- $X = \emptyset$  se  $b < 0$  e  $\sum_{j:a_j < 0} a_j > b$
- A restrição  $\sum_{j=1}^n a_j x_j \leq b$  é redundante se  $\sum_{j:a_j > 0} a_j \leq b$
- A restrição  $x_i + x_j \leq 1$  é válida se  $a_i + a_j > b$
- Seja  $a_k > 0$ . Se  $\sum_{j:a_j < 0} a_j + a_k > b$ , então  $x_k = 0$

### 3.11\* OUTROS MÉTODOS EM OTIMIZAÇÃO DISCRETA

Existem outros enfoques de resolução em otimização discreta para problemas que possuem uma estrutura particular, tais como os métodos de Benders, Dantzig-Wolfe (geração de colunas) e relaxação lagrangiana. Esses métodos são, em geral, ineficientes para problemas genéricos ou sem estrutura.

Para descrever os métodos de Benders e de Dantzig-Wolfe, é necessário apresentar os conceitos de pontos extremos e raios extremos, que são elementos essenciais para gerar qualquer ponto de um poliedro.

**Definição 3.6** Considere o poliedro  $P = \{\mathbf{x} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \in R_+^n\}$ , em que  $\mathbf{A}$  é uma matriz de dimensão  $m \times n$ , com posto de  $\mathbf{A}$  igual a  $m$ .

- Um ponto extremo do poliedro corresponde a um vértice ou solução básica.
- Um raio extremo de um poliedro é um vetor  $\mathbf{w}$  tal que  $n - 1$  restrições são ativas em  $\mathbf{w}$ .

**Exemplo 3.26** Seja o poliedro ilimitado definido por

$$x_1 - x_2 \geq -2$$

$$x_1 + x_2 \geq 1$$

$$x_1, x_2 \geq 0,$$

representado na Figura 3.46.

Os raios extremos são  $\mathbf{w}_1 = (1,1)$  e  $\mathbf{w}_2 = (1,0)$  e os pontos extremos são  $\mathbf{x}_1 = (0,2)$ ,  $\mathbf{x}_2 = (0,1)$  e  $\mathbf{x}_3 = (1,0)$ . O teorema a seguir mostra que qualquer elemento de um poliedro pode ser gerado a partir de seus pontos extremos e raios extremos.

**Teorema 3.1** Seja  $P = \{\mathbf{x} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \in R_+^n\}$  um poliedro,  $\mathbf{x}_i, i = 1, \dots, p$ , seus pontos extremos, e  $\mathbf{w}_j, j = 1, \dots, q$  seus raios extremos. Então, qualquer elemento  $x \in X$  pode ser escrito como

$$\mathbf{x} = \sum_{i=1}^p \alpha_i \mathbf{x}_i + \sum_{j=1}^q \beta_j \mathbf{w}_j, \quad \alpha_i \geq 0, \quad \beta_j \geq 0, \quad \sum_{i=1}^p \alpha_i = 1$$

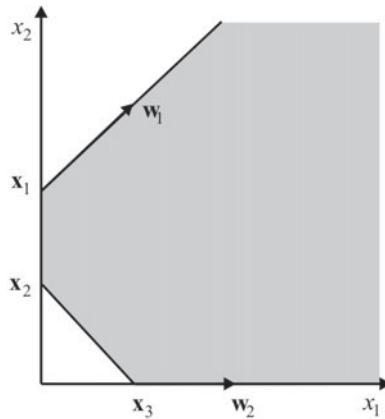
O vetor  $(2, 2)$  é um elemento do poliedro do Exemplo 3.26 e admite mais de uma representação, como, por exemplo,  $(2, 2) = \mathbf{x}_2 + \mathbf{w}_1 + \mathbf{w}_2$  e  $(2, 2) = \frac{1}{2}\mathbf{x}_2 + \frac{1}{2}\mathbf{x}_3 + \frac{3}{2}\mathbf{w}_1$ . Se o poliedro é limitado, contém somente pontos extremos e, neste caso, qualquer elemento  $x \in X$  pode ser escrito como combinação convexa dos pontos extremos, isto é,

$$\mathbf{x} = \sum_{i=1}^p \alpha_i \mathbf{x}_i, \quad \alpha_i \geq 0, \quad \sum_{i=1}^p \alpha_i = 1$$

Para ilustrar os métodos de Benders, Dantzig-Wolfe e relaxação lagrangiana, considere o problema de dimensionamento de lotes de múltiplos itens com restrição de capacidade, apresentando na Seção 3.6.1, e aqui repetido para facilitar a leitura.

$$\min \sum_{i=1}^n \sum_{t=1}^T (s_i y_{it} + h_i I_{it}) \quad (\text{DL9})$$

$$I_{it} = I_{i,t-1} + x_{it} - d_{it}, \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (\text{DL10})$$



**Figura 3.46** Pontos extremos e raios extremos de um poliedro.

$$\sum_{i=1}^n (sp_i y_{it} + b_i x_{it}) \leq C_t, \quad t = 1, \dots, T \quad (\text{DL11})$$

$$x_{it} \leq M_{it} y_{it}, \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (\text{DL12a})$$

$$M_{it} = \min\left\{\frac{C_t - sp_i}{b_i}, \sum_{\tau=t}^T d_{i\tau}\right\}, \quad i = 1, \dots, n, \quad t = 1, \dots, T \quad (\text{DL12b})$$

$$\mathbf{x} \in R_+^{nT}, \mathbf{I} \in R_+^{nT}, \mathbf{y} \in B^{nT} \quad (\text{DL13})$$

Considere o problema de programação inteira-mista

$$\begin{aligned} z &= \max \mathbf{c}^T \mathbf{x} + \mathbf{d}^T \mathbf{y} \\ \mathbf{A}\mathbf{x} + \mathbf{D}\mathbf{y} &\leq \mathbf{b} \\ \mathbf{x} &\in R_+^n, \mathbf{y} \in Z_+^p \end{aligned} \quad (3.27)$$

### Método de Benders

Este método utiliza o conceito de projeção no espaço de variáveis complicadoras para então obter um problema, em geral, com uma estrutura de fácil resolução. Considere que as variáveis  $\mathbf{y}$  em (3.27) são as variáveis complicadoras. A projeção dessas variáveis no espaço fornece o problema

$$z = \max_{\mathbf{y} \in Z_+^p} [\mathbf{d}^T \mathbf{y} + \max_{\mathbf{x} \in R_+^n} \{\mathbf{c}^T \mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b} - \mathbf{D}\mathbf{y}\}] \quad (3.28)$$

O problema nas variáveis  $\mathbf{x}$

$$h(\mathbf{y}) = \max_{\mathbf{x} \in R_+^n} \{\mathbf{c}^T \mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b} - \mathbf{D}\mathbf{y}\} \quad (3.29)$$

é um problema de programação linear parametrizado pelas variáveis  $\mathbf{y}$ , denominado subproblema de Benders. O subproblema dual é dado por

$$v(y) = \min_{\mathbf{u} \in R_+^m} \{\mathbf{u}^T (\mathbf{b} - \mathbf{D}\mathbf{y}) : \mathbf{u}^T \mathbf{A} \geq \mathbf{c}^T\} \quad (3.30)$$

Sejam  $\{\mathbf{u}_1, \dots, \mathbf{u}_p\}$  e  $\{\mathbf{w}_1, \dots, \mathbf{w}_q\}$  os conjuntos de pontos extremos e raios extremos de (3.30), respectivamente. Note que (3.29) é factível se e somente se (3.30) tem valor ótimo finito, isto é, se e somente se  $\mathbf{y}$  satisfaz as restrições

$$(\mathbf{w}_k)^T (\mathbf{b} - \mathbf{D}\mathbf{y}) \geq 0, \quad k = 1, \dots, q \quad (3.31)$$

Portanto, a projeção aplicada a (3.27) resulta em (3.28) sujeito às restrições adicionais (3.31). A seguir, a função  $h(\mathbf{y})$  em (3.29) é linearizada externamente e, devido a (3.30), seu valor é

$$\min_{1 \leq j \leq p} \{(\mathbf{u}_j)^T (\mathbf{b} - \mathbf{D}\mathbf{y})\}$$

para todo  $\mathbf{y}$  factível em (3.29) com a restrição adicional (3.31). Então, (3.28) pode ser reescrita como

$$z = \max_{\mathbf{y} \in Z_+^p} [\mathbf{d}^T \mathbf{y} + \min_{1 \leq j \leq p} \{(\mathbf{u}_j)^T (\mathbf{b} - \mathbf{D}\mathbf{y})\}] \quad (3.32)$$

Como o mínimo é o maior limitante superior, então (3.32) pode ser reescrita como

$$\begin{aligned} \max_{\sigma, \mathbf{y} \in Z_+^p} \sigma \\ \mathbf{d}^T \mathbf{y} + (\mathbf{u}_j)^T (\mathbf{b} - \mathbf{D}\mathbf{y}) \geq \sigma, \quad j = 1, \dots, p \\ (\mathbf{w}_k)^T (\mathbf{b} - \mathbf{D}\mathbf{y}) \geq 0, \quad k = 1, \dots, q \end{aligned} \quad (3.33)$$

O problema (3.33) é chamado problema mestre de Benders. Como o número de pontos extremos e raios extremos em problemas de grande porte é enorme, a relaxação é uma forma natural de resolver (3.33). Seja  $\hat{P} \subset \{1, \dots, p\}$  e  $\hat{Q} \subset \{1, \dots, q\}$ . O problema mestre relaxado é dado por

$$\begin{aligned} & \max_{\sigma, y \in Z_+^p} \sigma \\ & \mathbf{d}^T \mathbf{y} + (\mathbf{u}_j)^T (\mathbf{b} - \mathbf{Dy}) \geq \sigma, \quad j \in \hat{P} \\ & (\mathbf{w}_k)^T (\mathbf{b} - \mathbf{Dy}) \geq 0, \quad k \in \hat{Q} \end{aligned} \quad (3.34)$$

Por ser uma relaxação, segue-se que o valor ótimo de (3.34) é um limitante superior do valor ótimo de (3.33). Uma iteração  $l$  do método de Benders consiste em resolver (3.34) e obter a solução ótima  $\mathbf{y}_l$ . Esta solução é enviada ao problema (3.30), que gera um novo ponto extremo ou um novo raio extremo para o problema (3.34). Pode-se demonstrar que (3.30) não gera um ponto extremo ou raio extremo repetido.

O processo continua até que o limitante inferior seja igual ao valor de uma solução do subproblema dual. Em geral, o problema primal de programação linear (3.29) possui uma estrutura e este problema é resolvido em lugar do problema dual (3.30). As variáveis duais ótimas são obtidas a partir da base ótima do problema (3.29).

A seguir, mostra-se que, ao se fixarem as variáveis binárias  $y_{it}$  no problema de dimensionamento de lotes, obtém-se um problema de fluxo de custo mínimo em redes, apresentado na Seção 4.2.4. Uma maneira mais fácil de mostrar esse fato consiste em definir as variáveis  $z_{it} = b_i x_{it}$  e  $r_{it} = b_i I_{it}$ , de forma que, em ambas as restrições, (DL10) e (DL11), a unidade de medida é tempo.

A Figura 3.47 mostra a rede para um problema com 2 itens e três períodos, com  $I_{10} = I_{20} = 0$ . Os nós 1 e 2 correspondem aos itens 1 e 2, os nós  $M$  correspondem a nós de uma máquina, o nó  $F$  corresponde a um nó fonte e o nó  $S$  é um nó sumidouro. Em cada arco é mostrada a variável de fluxo que o atravessa, bem como o custo unitário do fluxo ( $\alpha$ ), e a capacidade do arco ( $\beta$ ), representados por  $[\alpha, \beta]$ . Por exemplo, o arco por onde passa o fluxo  $r_{11}$  tem custo unitário  $h_1/b_1$  e capacidade infinita. Os arcos entre o nó fonte e os nós  $M$  têm capacidades  $\hat{C}_1, \hat{C}_2$  e  $\hat{C}_3$ . Essas capacidades dependem dos valores das variáveis binárias  $y_{it}$ , e são dadas por  $\hat{C}_t = C_t - sp_1 y_{1t} - sp_2 y_{2t} - sp_3 y_{3t}$ . A demanda  $D$  do nó sumidouro corresponde à capacidade não utilizada, e é dada por  $\hat{C}_1 + \hat{C}_2 + \hat{C}_3 - b_1(d_{11} + d_{12} + d_{13}) - b_2(d_{21} + d_{22} + d_{23})$ , de forma que a rede é balanceada.

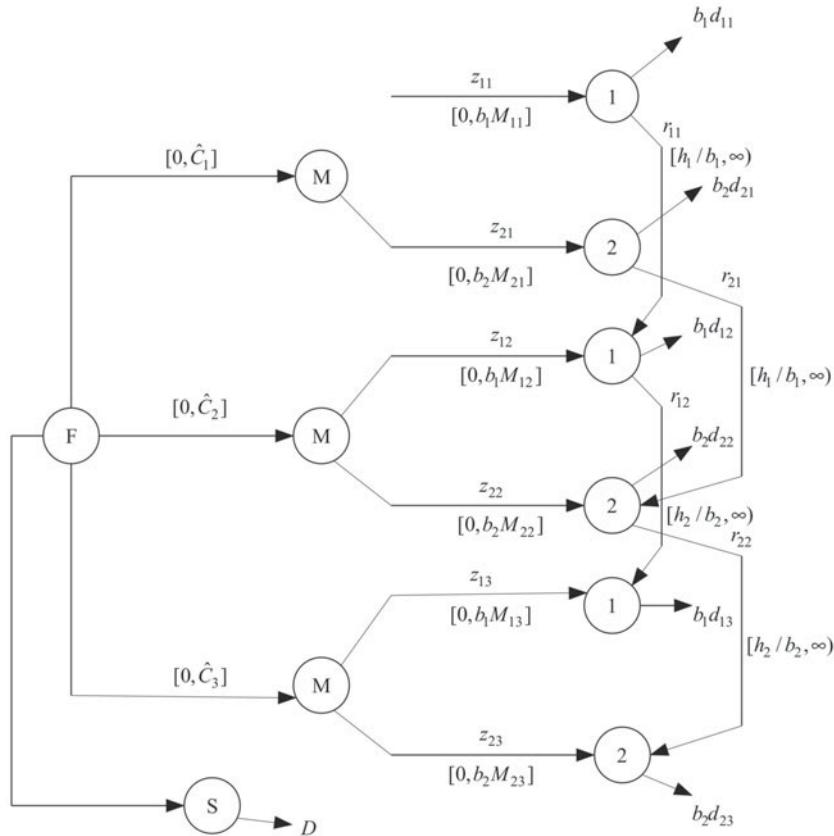
### *Método de Dantzig-Wolfe*

Este método é uma aplicação do princípio de decomposição, que envolve a resolução de diversos subproblemas de tamanho menor, tipicamente estruturados, em lugar da resolução do problema original, cujo tamanho e complexidade impedem que seja resolvido em tempo razoável.

Considere o seguinte problema de programação inteira mista

$$\begin{aligned} z = & \max \mathbf{c}^T \mathbf{x} + \mathbf{d}^T \mathbf{y} \\ & \mathbf{A}_1 \mathbf{x} + \mathbf{D}_1 \mathbf{y} \leq \mathbf{b}_1 \\ & \mathbf{A}_2 \mathbf{x} + \mathbf{D}_2 \mathbf{y} \leq \mathbf{b}_2 \\ & \mathbf{x} \in R_+^n, \mathbf{y} \in Z_+^p \end{aligned} \quad (3.35)$$

em que  $\mathbf{b}_1$  e  $\mathbf{b}_2$  têm dimensão  $(m_1 \times n)$  e  $(m_2 \times n)$ , respectivamente.



**Figura 3.47** Rede de dimensionamento de lotes.

As estruturas adequadas à decomposição são as seguintes:<sup>74</sup>

- As restrições  $\mathbf{A}_2\mathbf{x} + \mathbf{D}_2\mathbf{y} \leq \mathbf{b}_2$  são restrições complicadoras e a resolução do subproblema restrito a  $\mathbf{A}_1\mathbf{x} + \mathbf{D}_1\mathbf{y} \leq \mathbf{b}_1$  é mais fácil.
- As restrições  $\mathbf{A}_1\mathbf{x} + \mathbf{D}_1\mathbf{y} \leq \mathbf{b}_1$  têm uma estrutura de bloco diagonal e as restrições  $\mathbf{A}_2\mathbf{x} + \mathbf{D}_2\mathbf{y} \leq \mathbf{b}_2$  acoplam os blocos.
- Os subsistemas  $\mathbf{A}_1\mathbf{x} + \mathbf{D}_1\mathbf{y} \leq \mathbf{b}_1$  e  $\mathbf{A}_2\mathbf{x} + \mathbf{D}_2\mathbf{y} \leq \mathbf{b}_2$  são individualmente mais tratáveis, mas, quando presentes simultaneamente, tornam o problema bem mais difícil.

O exemplo anterior de dimensionamento de lotes enquadra-se no terceiro tipo de estrutura. As restrições (DL11), (DL12) e  $\mathbf{x} \in R_+^{nT}, \mathbf{y} \in B^{nT}$  correspondem a um subsistema que se decompõe para cada período  $t$  em um problema da mochila com variáveis binárias e contínuas. As restrições (DL10), (DL12) e (DL13) correspondem a outro subsistema que se decompõe para cada item  $i$  em um problema de dimensionamento de lotes sem restrição de capacidade.

Sejam  $\{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^K$  os pontos extremos de

$$X_2 = \{(\mathbf{x}, \mathbf{y}) : \mathbf{A}_2\mathbf{x} + \mathbf{D}_2\mathbf{y} \leq \mathbf{b}_2, \mathbf{x} \in R_+^n, \mathbf{y} \in Z_+^p\}$$

<sup>74</sup> Vanderbeck e Savelsbergh (2006).

Então,

$$X_2 = \{(\mathbf{x}, \mathbf{y}) : \mathbf{x} = \sum_{k=1}^K \alpha_k \mathbf{x}_k, \mathbf{y} = \sum_{k=1}^K \alpha_k \mathbf{y}_k, \sum_{k=1}^K \alpha_k = 1, \alpha_k \in \{0,1\}, k = 1, \dots, K\}$$

O problema mestre (*PM*) de Dantzig-Wolfe, equivalente ao problema (3.35), pode ser reescrito como

$$(PM) \quad \begin{aligned} z_{PM} &= \max \sum_{k=1}^K (\mathbf{c}^T \mathbf{x}_k) \alpha_k + \sum_{k=1}^K (\mathbf{d}^T \mathbf{y}_k) \alpha_k \\ &\sum_{k=1}^K (\mathbf{A}_1 \mathbf{x}_k) \alpha_k + \sum_{k=1}^K (\mathbf{D}_1 \mathbf{y}_k) \alpha_k \leq \mathbf{b}_1 \\ &\sum_{k=1}^K \alpha_k = 1 \\ &\alpha_k \in \{0,1\}, k = 1, \dots, K \end{aligned}$$

Considere agora a relaxação linear do problema mestre *PM*

$$(PML) \quad \begin{aligned} z_{PML} &= \max \sum_{k=1}^K (\mathbf{c}^T \mathbf{x}_k) \alpha_k + \sum_{k=1}^K (\mathbf{d}^T \mathbf{y}_k) \alpha_k \\ &\sum_{k=1}^K (\mathbf{A}_1 \mathbf{x}_k) \alpha_k + \sum_{k=1}^K (\mathbf{D}_1 \mathbf{y}_k) \alpha_k \leq \mathbf{b}_1 \\ &\sum_{k=1}^K \alpha_k = 1 \\ &\alpha_k \geq 0, k = 1, \dots, K \end{aligned} \quad (3.36)$$

Seja  $z_{PL}$  o valor ótimo da relaxação linear de (3.35). Então,  $z_{PL} \geq z_{PML}$ , pois uma solução factível da relaxação de (3.35) que não é combinação convexa dos pontos extremos de  $X$  não é factível em (3.36).

Sejam  $\{\pi_i\}_{i=1}^{m_1}$  as variáveis duais associadas às restrições (3.36), e  $\mu$  a variável dual associada à restrição (3.37). Considere o problema mestre relaxado restrito (*PMLR*)

$$(PMLR) \quad \begin{aligned} z_{PMLR} &= \max \sum_{k=1}^{\Lambda} (\mathbf{c}^T \mathbf{x}_k) \alpha_k + \sum_{k=1}^{\Lambda} (\mathbf{d}^T \mathbf{y}_k) \alpha_k \\ &\sum_{k=1}^{\Lambda} (\mathbf{A}_1 \mathbf{x}_k) \alpha_k + \sum_{k=1}^{\Lambda} (\mathbf{D}_1 \mathbf{y}_k) \alpha_k \leq \mathbf{b}_1 \\ &\sum_{k=1}^{\Lambda} \alpha_k = 1 \\ &\alpha_k \geq 0, k = 1, \dots, A \end{aligned}$$

de modo que  $\Lambda \subset K$  contém uma base para *PMLR*.

Seja  $\boldsymbol{\alpha}^*$  uma solução ótima de *PMLR* e  $(\boldsymbol{\pi}, \mu)$  uma solução dual ótima. Note que  $\boldsymbol{\alpha}^*$  também é uma solução factível de *PML* e que  $z_{PMLR} \leq z_{PML}$ . Para verificar se  $\boldsymbol{\alpha}^*$  é uma solução ótima de *PML*, deve-se verificar se, para cada coluna, isto é, para cada  $(\mathbf{x}, \mathbf{y}) \in X_2$ , se o custo reduzido  $(\mathbf{c}^T \mathbf{x} + \mathbf{d}^T \mathbf{y} - \boldsymbol{\pi}^T (\mathbf{A}_1 \mathbf{x} + \mathbf{D}_1 \mathbf{y}) - \mu) \leq 0$ . Isso é feito implicitamente por meio da resolução do seguinte subproblema:

$$\eta = \max \{(\mathbf{c}^T - \boldsymbol{\pi}^T \mathbf{A}_1) \mathbf{x} + (\mathbf{d}^T - \boldsymbol{\pi}^T \mathbf{D}_1) \mathbf{y} - \mu : (\mathbf{x}, \mathbf{y}) \in X_2\}$$

Se  $\eta \leq 0$ , então  $\alpha^*$  é uma solução ótima para  $PML$ ; caso contrário, seja  $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$  a solução ótima do subproblema acima. Introduza a coluna  $(\mathbf{c}^T \tilde{\mathbf{x}} + \mathbf{d}^T \tilde{\mathbf{y}}, \mathbf{A}_1 \tilde{\mathbf{x}} + \mathbf{D}_1 \tilde{\mathbf{y}}, 1)^T$  em  $PMLR$  e reotimize pelo método simplex.

Note que a solução ótima de  $PML$  fornece um valor  $z_{PML} \geq z$ . Neste ponto, esse limitante superior de  $z$  é usado em um procedimento *branch-and-bound*, conhecido na literatura como *branch-and-price*.

### *Relaxação lagrangiana*

Considere novamente o problema de programação inteira mista

$$\begin{aligned} z = & \max \mathbf{c}^T \mathbf{x} + \mathbf{d}^T \mathbf{y} \\ & \mathbf{A}_1 \mathbf{x} + \mathbf{D}_1 \mathbf{y} \leq \mathbf{b}_1 \\ & \mathbf{A}_2 \mathbf{x} + \mathbf{D}_2 \mathbf{y} \leq \mathbf{b}_2 \\ & \mathbf{x} \in R_+^n, \mathbf{y} \in Z_+^p \end{aligned} \quad (3.38)$$

Seja  $\mathbf{u} \in R_+^{m_2}$  o vetor de variáveis duais associadas às restrições  $\mathbf{A}_2 \mathbf{x} + \mathbf{D}_2 \mathbf{y} \leq \mathbf{b}_2$ , e defina o problema lagrangiano (*PLG*) como

$$\begin{aligned} g(\mathbf{u}) = & \max \mathbf{c}^T \mathbf{x} + \mathbf{d}^T \mathbf{y} + \mathbf{u}^T (\mathbf{b}_2 - \mathbf{A}_2 \mathbf{x} - \mathbf{D}_2 \mathbf{y}) \\ (PLG) \quad & \mathbf{A}_1 \mathbf{x} + \mathbf{D}_1 \mathbf{y} \leq \mathbf{b}_1 \\ & \mathbf{x} \in R_+^n, \mathbf{y} \in Z_+^p \end{aligned}$$

Mostra-se, a seguir, que  $g(\mathbf{u}) \geq z$ . Seja  $(\mathbf{x}^*, \mathbf{y}^*)$  uma solução ótima de (3.38). Então  $(\mathbf{x}^*, \mathbf{y}^*)$  é factível em *PLG*. Como  $\mathbf{A}_2 \mathbf{x}^* + \mathbf{D}_2 \mathbf{y}^* \leq \mathbf{b}_2$  e  $\mathbf{u} \in R_+^{m_2}$ , segue-se que  $\mathbf{c}^T \mathbf{x}^* + \mathbf{d}^T \mathbf{y}^* \leq \mathbf{c}^T \mathbf{x}^* + \mathbf{d}^T \mathbf{y}^* + \mathbf{u}^T (\mathbf{b}_2 - \mathbf{A}_2 \mathbf{x}^* - \mathbf{D}_2 \mathbf{y}^*) \leq g(\mathbf{u})$ .

O problema dual é, então, definido como

$$w = \min \{g(\mathbf{u}) : \mathbf{u} \in R_+^{m_2}\}$$

Seja  $z_{PL}$  o valor ótimo da relaxação linear de (3.38). Demonstra-se que  $w \leq z_{PL}$ , isto é, o valor ótimo do problema dual pode gerar um limitante superior melhor que o valor ótimo da relaxação linear. Em problemas em que o problema lagrangiano *PLG* é mais fácil de resolver que o problema original (3.38), a relaxação lagrangiana é, em geral, mais eficiente que a relaxação linear no método *branch-and-bound*.

No problema de dimensionamento de lotes, a relaxação lagrangiana das restrições (DL10) resulta em um problema lagrangiano que se decompõe, para cada período  $t$ , em problemas da mochila com variáveis binárias e contínuas. A relaxação lagrangiana das restrições (DL11) resulta em um problema lagrangiano que se decompõe para cada item  $i$  em um problema de dimensionamento de lotes sem restrição de capacidade.<sup>75</sup>

Isso indica que o método de Dantzig-Wolfe e a relaxação lagrangiana estão relacionados. Demonstra-se que o valor ótimo do problema dual é igual ao valor ótimo da relaxação linear do problema mestre de Dantzig-Wolfe, isto é,  $w = z_{PML}$ .

### *Heurísticas*

Heurísticas são métodos de resolução de problemas de otimização discreta que não garantem a obtenção de uma solução factível ou ótima. Nicholson (1971) propôs uma definição que expressa muito bem as características de uma heurística: é um procedimento para resolver problemas

<sup>75</sup> Os métodos de Benders e de relaxação lagrangiana são integrados no método de decomposição cruzada para a resolução do problema de dimensionamento de lotes (DL9) – (DL13); ver Souza e Armentano (1994). A relaxação lagrangiana é comumente utilizada para obtenção de limitantes para o valor ótimo de um problema; ver, por exemplo, em Campello e Maculan (1987).

por meio de um enfoque “intuitivo”, em geral racional, no qual a *estrutura* do problema possa ser interpretada e explorada inteligentemente para se obter uma solução razoável.

Para ilustrar essa definição, considere o problema da mochila 0-1, cujo modelo é apresentado na Seção 3.4. Suponha que cada item  $j$  tem um valor  $p_j$  e ocupa um volume  $a_j$  da mochila com capacidade  $b$ . Uma forma adequada para usar a informação de valor e volume dos itens consiste em listar os itens em ordem não crescente da razão  $p_j/a_j$ , isto é, o valor do item por unidade de volume. A seleção dos itens a serem incluídos na mochila segue esta ordem. O item  $j$  da lista é selecionado se sua inclusão não viola a capacidade da mochila; caso contrário, investiga-se a inclusão do item  $j + 1$ , e assim por diante. A cada inclusão de um item, a capacidade da mochila é atualizada. O procedimento é finalizado até que todos os itens sejam selecionados ou até que não caibam mais itens na mochila. Essa heurística aplicada ao Exemplo 3.19 fornece uma solução com valor 674, enquanto o valor da solução ótima é 763. Outro exemplo de um método heurístico para um problema de planejamento da produção foi descrito na Seção 1.3.

O crescimento do interesse por heurísticas ocorreu com o advento da teoria da complexidade computacional, que mostrou e ainda tem mostrado que um grande número de problemas combinatórios é intratável, como destacado na Seção 3.7. A resolução desses problemas complexos com número médio ou grande de variáveis e restrições por métodos ótimos é, em geral, inviável computacionalmente. Várias são as situações que podem tornar interessante a utilização de métodos heurísticos na resolução de um problema. Dentre estas, incluem-se:

- Situações em que um método de resolução exato não está disponível, ou está disponível mas exige um tempo computacional ou quantidade de memória além dos recursos do computador ou das necessidades da aplicação. Alguns exemplos ocorrem em problemas de programação de produção e roteamento dinâmico de veículos, em que é necessário encontrar uma solução razoável em um período de tempo muito curto. Nesses problemas, a ocorrência de um evento requer uma decisão muito rápida (por exemplo, da ordem de segundos), que, em geral, não pode ser obtida por um método ótimo.
- Situações em que pode não valer a pena o esforço e/ou custo envolvidos na geração de uma solução ótima, uma vez que esta não representará um ganho técnico ou econômico relevante em relação a uma solução heurística. Ou situações em que os dados do problema são incertos ou pouco confiáveis e a aplicação de um método exato (determinístico) pode ser pouco vantajosa em relação a um método heurístico, devido às incertezas envolvidas na solução.
- Situações em que uma heurística é utilizada em conjunto com um método exato. Por exemplo, heurísticas são usadas para gerar uma solução inicial para o método *branch-and-cut* e também para gerar soluções nos nós da árvore.

Uma vantagem de heurísticas em relação aos métodos exatos refere-se à maior flexibilidade no tratamento das características de um problema. Por exemplo, algumas características de um problema podem ser difíceis de incorporar em um modelo de otimização matemática. Além disso, as heurísticas podem oferecer mais de uma solução, permitindo ampliar as possibilidades de decisão, sobretudo quando existem fatores intangíveis que não podem ser facilmente incorporados ao modelo, mas que precisam ser considerados. Uma desvantagem das heurísticas, além de não garantirem encontrar uma solução factível ou ótima, é que em geral não é possível conhecer a qualidade da solução obtida, ou seja, quão próxima essa solução está da solução ótima.

Existem diversos tipos de heurísticas, dentre os quais se destacam: (a) as heurísticas construtivas; (b) as heurísticas de busca local; e (c) as meta-heurísticas.

- *Heurísticas construtivas*

Este tipo de heurística constrói uma solução, factível ou não, adicionando, em cada passo, um elemento da solução, tal como o valor de uma variável ou um arco ou nó em um grafo. Um exem-

plo de heurística construtiva é a *heurística gulosa*, que constrói a solução visando, a cada passo, o máximo benefício. A heurística para o problema da mochila 0-1 descrito anteriormente é um exemplo de heurística gulosa, uma vez que, a cada passo, escolhe, dentre os itens ainda não selecionados e que ainda cabem na mochila, aquele que tenha o maior valor por unidade de volume. No caso do problema do caixeiro-viajante, um exemplo de uma heurística gulosa consiste em escolher uma cidade aleatoriamente ou segundo algum critério determinístico e incorporá-la à rota. A partir daí, escolhe-se sucessivamente a cidade mais próxima da última já incorporada à rota (heurística do vizinho mais próximo). Quando todas as cidades tiverem sido incorporadas, fecha-se a rota entre a primeira e a última cidades.

- *Heurísticas de busca local*

A partir de uma solução inicial  $s$ , obtida, por exemplo, por uma heurística construtiva, associa-se uma vizinhança  $N$  de  $s$ . Cada solução  $s'$  da vizinhança  $N(s)$  é atingida a partir de  $s$  por uma operação chamada *movimento*, e a solução  $s'$  é chamada de solução *vizinha*. Seleciona-se uma solução vizinha que seja melhor que a solução corrente, e a busca prossegue iterativamente até que a vizinhança não contenha nenhuma solução melhor que a solução corrente, denominada *ótimo local* em relação à vizinhança, ou até que algum outro critério de parada tenha sido satisfeito.

Considere um exemplo do problema da mochila 0-1 com cinco itens, e uma solução  $s$  definida pelo vetor  $(1,0,0,1,1)$ , no qual 1 na posição  $i$ ,  $i = 1, \dots, 5$ , indica que o item  $i$  foi incluído na mochila, e 0 em caso contrário. Uma possibilidade é definir  $N(s)$  como todas as soluções resultantes da operação de *troca* de um dos itens na mochila por outro fora da mochila. Neste caso,  $N(s)$  incluiria, por exemplo, a solução  $s'$  definida por  $(0,0,1,1,1)$ , obtida pela operação de troca do item 1 pelo item 3. Outra possibilidade de vizinhança consiste em trocar até  $r$  itens na mochila por  $r$  itens fora da mochila. No problema do caixeiro-viajante, a vizinhança  $k$ -opt de uma solução  $s$  é composta de todas as soluções que diferem de  $s$  em exatamente  $k$  arestas. A troca de itens no problema da mochila 0-1 e a troca de  $k$  arestas no problema do caixeiro-viajante correspondem aos tipos de movimentos que definem as vizinhanças nesses exemplos.

Considere o exemplo a seguir de um caixeiro-viajante simétrico com distância euclidiana entre 12 cidades, e movimentos 2-opt. As coordenadas de cada cidade são mostradas a seguir (Figura 3.48).

$$\begin{aligned} \mathbf{x} &= [26 \quad 34 \quad 95 \quad 70 \quad 99 \quad 40 \quad 39 \quad 53 \quad 39 \quad 3 \quad 71 \quad 45] \\ \mathbf{y} &= [38 \quad 31 \quad 54 \quad 88 \quad 89 \quad 63 \quad 85 \quad 81 \quad 96 \quad 95 \quad 65 \quad 17] \end{aligned}$$

Iniciando pela cidade 1, a heurística do vizinho mais próximo gera a rota  $1 - 2 - 12 - 6 - 7 - 9 - 8 - 4 - 11 - 3 - 5 - 10 - 1$ , com distância 388,9 (Figura 3.48). Partindo-se desse ponto, usa-se a vizinhança 2-opt, e a solução vizinha com menor distância é escolhida. A partir dessa solução vizinha, busca-se outra solução vizinha com distância menor que a anterior, e assim por diante. O método pára quando não existe um vizinho melhor. Para este exemplo, a busca local encontra três soluções, mostradas a seguir e ilustradas na Figura 3.48.

Solução 1:  $1 - 2 - 12 - 6 - 5 - 3 - 11 - 4 - 8 - 9 - 7 - 10 - 1$  distância: 372,5

Solução 2:  $1 - 2 - 12 - 6 - 11 - 3 - 5 - 4 - 8 - 9 - 7 - 10 - 1$  distância: 345,1

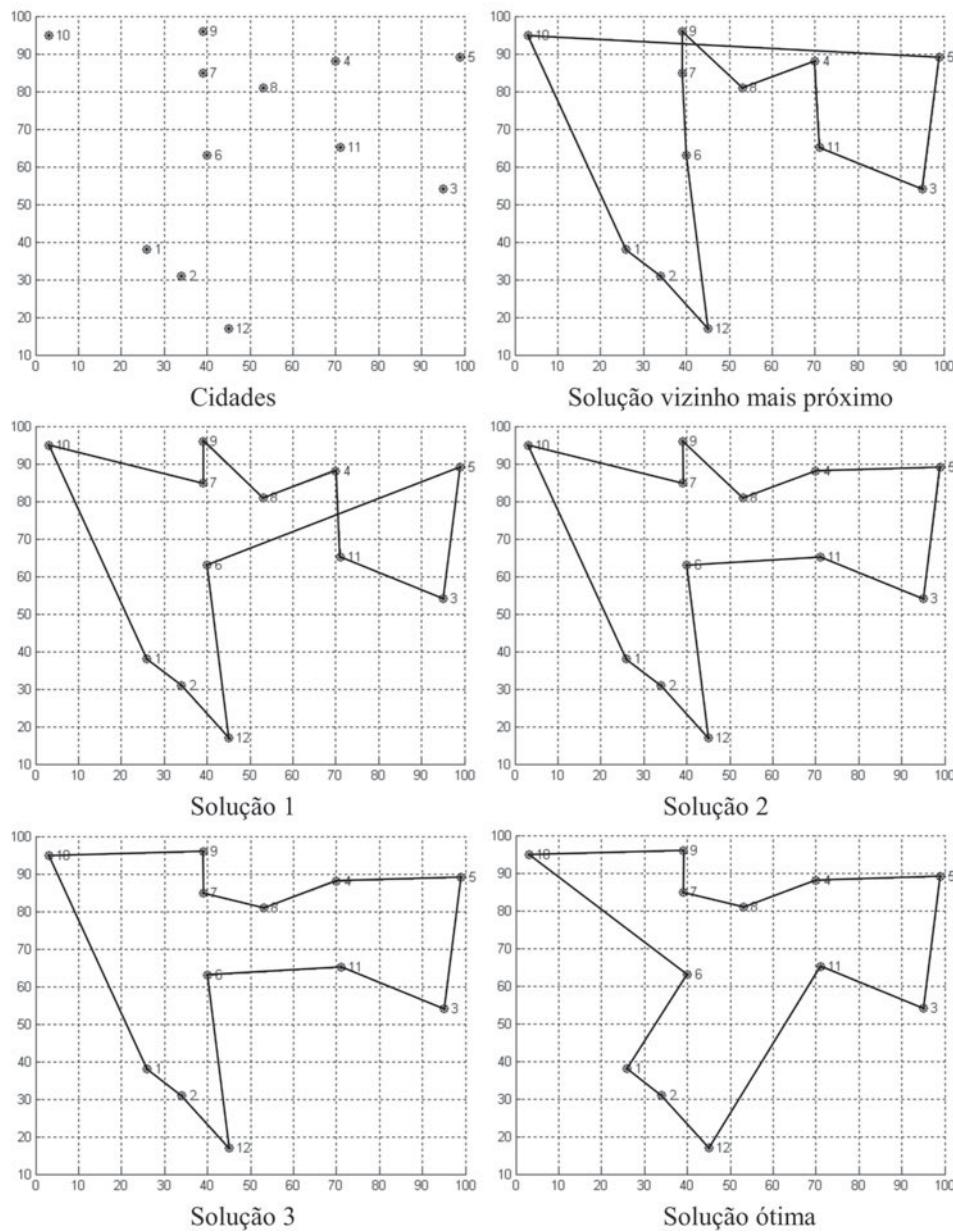
Solução 3:  $1 - 2 - 12 - 6 - 11 - 3 - 5 - 4 - 8 - 7 - 9 - 10 - 1$  distância: 337,8

A solução 3 corresponde a um mínimo local, e a solução ótima global é dada pela rota

$1 - 2 - 12 - 11 - 3 - 5 - 4 - 8 - 7 - 9 - 10 - 6 - 1$ , com distância 331,2.

- *Meta-heurísticas*

São técnicas que guiam e modificam heurísticas de forma a produzir soluções além daquelas geradas por heurísticas de busca local. Dentre as meta-heurísticas existentes, destacam-se: algo-



**Figura 3.48** Exemplo de busca local para o problema do caixeiro-viajante.

ritmo genético, busca tabu, *simulated annealing*, *scatter search*, colônia de formigas e GRASP (*greedy randomized adaptive search procedure*).<sup>76</sup>

Cada uma dessas técnicas utiliza diferentes estratégias para a exploração do espaço de busca. A busca tabu, por exemplo, usa exploração estratégica e memória flexível para guiar a busca no espaço de soluções. Por meio da exploração estratégica, determina-se uma direção de busca baseada em propriedades da solução corrente e da história da busca. A memória tem estruturas de curto e longo prazo. A memória de curto prazo armazena atributos de soluções visitadas em passado recente, e estes são utilizados para impedir que algumas soluções sejam revisitadas, evitan-

<sup>76</sup> Para detalhes destas meta-heurísticas, veja, por exemplo, Reeves (1993), Glover e Laguna (1997), Aarts *et al.* (1997), Michalewicz (1996), Blum e Roli (2003) e Resende e Ribeiro (2003). Algumas aplicações de meta-heurísticas podem ser encontradas em Almeida *et al.* (2003), Alvarenga *et al.* (2000), Alvim *et al.* (2004), Barcia *et al.* (1998), Chiyoshi e Galvão (2000), Ferreira Filho e Galvão (1999), França *et al.* (1999), Linhares *et al.* (1999), Pureza e Morabito (2006), Ravagnani *et al.* (2005), Ribeiro e Souza (2000) e Stern (1992).

do, assim, ciclagem e direcionando a busca para outras regiões. A memória de longo prazo contém uma história seletiva das soluções e seus atributos, e é usada para estimular a geração de soluções com características presentes em soluções de boa qualidade (intensificação) e com características pouco exploradas (diversificação).

Algoritmos genéticos são métodos populacionais baseados no processo biológico de seleção natural e hereditariedade, e envolvem uma evolução de populações de soluções convenientemente codificadas. Uma população é transformada em outra população por três mecanismos probabilísticos, denominados seleção, cruzamento (*crossover*) e mutação. O papel da seleção consiste em escolher soluções mais aptas de acordo com a qualidade (valor da função objetivo) e a diversidade das soluções. O cruzamento envolve a troca de partes (informação seletiva) entre soluções, visando gerar uma nova população com soluções melhores. A mutação envolve alterações estruturais nas soluções, em geral, com o objetivo de evitar uma convergência prematura do processo evolutivo.

### 3.12 NOTAS

O texto abaixo tem por objetivo prover um contexto histórico e referências adicionais que são relevantes para algumas seções.

**3.3** Otimização discreta ou programação inteira e combinatória é apresentada, em nível de graduação, em uma ou duas seções de livros de Pesquisa Operacional, tais como Hillier e Liberman (2004), Wagner (1986), Winston (2004), Rardin (1998), Murty (1995) e Taha (2002). É tratada de forma mais detalhada e em nível mais avançado no livro da Nemhauser e Wolsey (1988). O livro mais recente (Wolsey, 1998) é adequado para o nível de pós-graduação, enquanto um dos primeiros livros nesta área (Garfinkel e Nemhauser, 1972) ainda é uma referência importante.

**3.5.1** O livro editado por Toth e Vigo (2002) contém uma revisão de problemas de roteamento, métodos de resolução de problemas específicos e aplicações.

**3.5.2** Para uma revisão de localização de facilidades, veja Owen e Daskin (1998) e Klose e Drexl (2005).

**3.6.1** Variantes do modelo de dimensionamento de lotes com restrição de capacidade e métodos de resolução são revistos por Karimi *et al.* (2003). Brahimi *et al.* (2006) fazem uma revisão abrangente de problemas de dimensionamento de lotes e a seguir focam em modelos e métodos de resolução para o problema de dimensionamento de lotes de um único item.

**3.6.2** Uma revisão de problemas de dimensionamento e programação de lotes é encontrada em Drexl e Kimms (1997). Veja também Meier (2000) para problemas com tempos de preparação dependentes da seqüência.

**3.6.3** Baker (1974) e French (1982) são os livros básicos de programação (*scheduling*) da produção. Livros mais recentes são Morton e Pentico (1993) e Pinedo (1995). Blazewicz *et al.* (1996) apresentam diversos algoritmos para problemas de programação de sistemas de manufatura e de computação. O livro de Pinedo (2005) tem escopo mais amplo e cobre programação da produção e de serviços. Blazewicz *et al.* (1991) apresentam modelos matemáticos alternativos para os problemas desta seção, bem como formulações para outros problemas. O artigo de Lawler *et al.* (1993) contém uma excelente revisão de problemas em programação da produção, e Anderson *et al.* (1997) abordam métodos de busca local aplicados a estes problemas.

**3.8** A importância de formulações em programação inteira tornou-se aparente somente no final da década de 1970. O livro de Williams (1999) trata em profundidade e abrangência um tópico fundamental em otimização, que envolve a construção de modelos matemáticos, incluindo otimização discreta.

**3.9** Desenvolvimentos recentes em softwares de programação inteira, tais como CPLEX, XPRESS e LINDO têm melhorado a capacidade de resolver problemas de grande porte. Atamtürk e Savelsbergh (2005) apresentam o estado da arte de componentes principais destes softwares e discutem as opções disponíveis aos usuários para ajustar o desempenho destes, quando os ajustes padrões (*default*) não proporcionam um nível de desempenho desejado.

**3.9.1** O primeiro algoritmo de *branch-and-bound* foi apresentado por Land e Doig (1960). Neste algoritmo, uma variável com valor fracionário gera um número de nós correspondentes aos valores inteiros que a variável podia assumir. A dicotomia apresentada nesta seção foi proposta por Dakin (1965). O termo *branch-and-bound* foi introduzido por Little *et al.* (1963), que apresentaram uma aplicação bem-sucedida para o problema do caixeiro-viajante. Para um estudo computacional de métodos de seleção de variáveis a serem ramificadas e escolha de nós, veja Linderoth e Savelsbergh (1999). Uma generalização promissora foi proposta por Achterberg *et al.* (2005).

**3.9.2** O algoritmo de plano de cortes apresentado nesta seção foi proposto por Gomory (1958) e Gomory (1963). Neste último trabalho, o autor demonstra que a convergência de seu método é finita se as linhas que geram cortes forem escolhidas de forma apropriada. A demonstração de que qualquer desigualdade válida é gerada pelo procedimento de Chvátal-Gomory é atribuída a Chvátal (1973).

**3.9.3** O uso de planos de corte como ferramenta prática para solução de problemas genéricos de programação inteira foi praticamente abandonada nos anos 60 e 70. Nos anos 80, planos de corte foram utilizados somente no nó raiz da árvore *branch-and-bound*, gerando o método *cut-and-branch*. Aparentemente, o primeiro algoritmo de *branch-and-cut* foi proposto por Padberg e Rinaldi (1987) para resolver um problema do caixeiro-viajante com 532 cidades. Um sistema *branch-and-cut* é apresentado por Cordier *et al.* (1999).

**3.10** O pré-processamento é extremamente importante para uma rápida resolução de problemas de programação linear real e inteira. A Proposição 3.3 foi apresentada por Brearley *et al.* (1973). Técnicas de pré-processamento são abordadas por Savelsbergh (1994).

**3.11** O método de decomposição de Benders (1962) foi aplicado com sucesso por Geoffrion e Graves (1974) para um sistema de distribuição de múltiplos produtos. Costa (2005) provê uma revisão da aplicação do método de Benders em diversos problemas de projeto de redes.

O método de decomposição em programação linear, conhecido como método de Dantzig-Wolfe, foi proposto por Dantzig e Wolfe (1960). Gilmore e Gomory (1961, 1963) provavelmente foram os primeiros autores a usar a geração de colunas para resolver o problema mestre relaxado derivado de um problema de programação inteira, no caso, o problema de corte descrito na Seção 3.4. A demonstração da igualdade entre o valor ótimo do problema dual e o valor ótimo da relaxação linear do problema mestre de Dantzig-Wolfe é atribuída a Geoffrion (1974). O primeiro trabalho de geração de colunas em programação inteira aborda o problema de roteamento de veículos com janelas de tempo (Desrosiers *et al.*, 1984), e desde então o interesse por este método tem crescido, como mostra o livro editado por Desaulniers *et al.* (2005).

A teoria de relaxação lagrangiana em programação inteira foi proposta por Geoffrion (1974) e este enfoque tem sido extensivamente usado em métodos exatos e heurísticos para uma grande variedade de problemas de otimização combinatória.

Aparentemente, a primeira heurística de busca local foi proposta por Croes G.A. (1958), que introduziu o movimento 2-opt para o problema do caixeiro-viajante. Lin e Kernighan (1971) propuseram uma heurística de busca local para o caixeiro-viajante bastante sofisticada para a época, com estratégias inovadoras em heurísticas, utilizando movimentos  $k$ -opt,  $k = 2, 3, 4$ . Em cada iteração, o algoritmo examina se uma mudança de  $k$  para  $k+1$  resulta em uma rota menor, conhecida hoje como busca em profundidade variável. Quando o algoritmo não acha nenhum movimento que melhore a solução, ele é reiniciado de uma solução aleatória. Outra característica importante do algoritmo consiste em fixar arestas que pertencem a alguns ótimos locais. Algumas referências para meta-heurísticas são Michalewicz (1996), Reeves (1997), Glover e Laguna (1997), Aarts *et al.* (1997), Martí *et al.* (2006) e Resende e Ribeiro (2003). O livro editado por Aarts e Lenstra (1997) contém artigos sobre métodos de busca local e problemas de otimização combinatória. Ver Silver (2004) para uma revisão de métodos heurísticos e Blum (2003) para uma revisão de meta-heurísticas.

### 3.13\* EXERCÍCIOS<sup>77</sup>

**Exercício 3.1** Considere um conjunto de investimentos  $\{1, 2, 3, 4, 5, 6\}$ . Formule as seguintes restrições:

- Não é possível investir em todos os investimentos.
- Deve-se escolher pelo menos dois investimentos.
- O investimento 3 só pode ser escolhido se o investimento 1 também for escolhido.
- O investimento 2 não pode ser escolhido se o investimento 5 for escolhido.
- Os investimentos 1 e 5 devem ser escolhidos ou nenhum deles é escolhido.
- Deve-se escolher pelo menos um dos investimentos 1, 3, 4 ou pelo menos dois investimentos de 2, 3, 5, 6.
- Se os investimentos 2 e 3 forem escolhidos, então o investimento 4 também tem de ser escolhido.

**Exercício 3.2** Uma empresa tem um capital disponível e considera cinco projetos a serem executados nos próximos três anos. Se um projeto é escolhido, ele requer um investimento a cada ano, e resulta em um rendimento anual após o período de três anos. O capital disponível em cada ano é limitado. Os dados são indicados na Tabela 3.3. Formule o problema de selecionar projetos que maximizem o rendimento total.

**Tabela 3.3**

<i>Projeto</i>	<i>Investimento requerido</i>			<i>Retorno anual após três anos</i>
	<i>Ano 1</i>	<i>Ano 2</i>	<i>Ano 3</i>	
1	7	6	14	30
2	11	4	18	40
3	20	10	5	80
4	19	23	26	110
5	7	9	4	60
Capital disponível	55	60	58	

**Tabela 3.4**

<i>Máquina</i>	<i>Custo de preparação</i>	<i>Custo de processamento</i>				
		<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
1	40	28	50	49	31	33
2	35	20	16	63	27	41

<sup>77</sup> Alguns exercícios são baseados naqueles apresentados por Winston (2004), Murty (1995), Wolsey (1998) e Pinedo (2005). Exercícios com símbolo \* são mais completos

**Exercício 3.3** Considere cinco itens que podem ser processados em qualquer uma de duas máquinas. Se uma máquina é usada, incorre-se em um custo de preparação. O custo de processamento de cada item depende da máquina utilizada. Esses custos são mostrados na Tabela 3.4. Formule o problema de determinar qual máquina usar para processar todos os itens.

**Exercício 3.4** Seis itens podem ser colocados em um contêiner. O peso  $w_i$  (em toneladas), o volume  $v_i$  (em  $m^3$ ), e o valor  $r_i$  (em unidades de R\$1.000) de cada unidade do item  $i$  são dados na Tabela 3.5. Existem cinco cópias do item 1, três cópias do item 2, e para os demais itens existe um número ilimitado de cópias. O contêiner suporta uma carga máxima de 90 toneladas, e tem um volume de  $50m^3$ . Formule o problema de maximizar o valor da carga no contêiner.

**Exercício 3.5** Um treinador está tentando escalar seu time titular de basquete. Ele pode escolher entre sete jogadores classificados (em uma escala de 1 = ruim até 3 = excelente) de acordo com suas habilidades em assistência, arremesso, rebote e defesa. A Tabela 3.6 mostra a posição em que cada jogador pode jogar e sua classificação nas habilidades citadas.

**Tabela 3.5**

<i>Item i</i>	$w_i$	$v_i$	$r_i$
1	9	8	10
2	2	5	4
3	6	3	5
4	7	6	9
5	5	6	8
6	7	4	6

**Tabela 3.6**

<i>Jogador</i>	<i>Posição</i>	<i>Liderança</i>	<i>Arremesso</i>	<i>Rebote</i>	<i>Defesa</i>
1	D	3	3	1	3
2	C	2	1	3	2
3	D-A	2	3	2	2
4	A-C	1	3	3	1
5	D-A	3	3	3	3
6	A-C	3	1	2	3
7	D-A	3	2	2	1

Os cinco jogadores que compõem o time para iniciar a partida devem atender às seguintes restrições:

1. Pelo menos três jogadores devem ser capazes de jogar na defesa (D), pelo menos dois devem ser capazes de jogar no ataque (A) e pelo menos dois devem ser capazes de jogar no centro (C).
2. A média em assistência, arremesso, rebote e defesa deve ser pelo menos dois.
3. Se o jogador 3 começa a partida, então o jogador 6 não pode estar no time.
4. Se o jogador 1 começa a partida, então os jogadores 4 e 5 também devem estar no time.
5. Ou o jogador 2 ou o jogador 3 devem começar a partida.

Formule o problema de maximizar a habilidade de defesa do time que inicia a partida.

**Exercício 3.6\*** Uma linha de montagem deve ser projetada para produzir um item. Uma operação pode ser iniciada desde que as operações predecessoras tenham sido completadas. Essa informação e o tempo que um operador gasta para executar cada operação são mostrados na Tabela 3.7. Cada operador tem no máximo 45 segundos para trabalhar em um item e colocá-lo de volta na linha (tempo de ciclo). Um operador pode executar qualquer subconjunto de operações desde que o tempo de ciclo não seja excedido e as restrições de precedência de operações sejam obedecidas. Determine a designação de operações a operadores de modo a minimizar o número de operadores.

**Exercício 3.7** Em cada dia da semana, uma loja requer um número de empregados em tempo integral, de acordo com a Tabela 3.8. Cada empregado deve trabalhar cinco dias consecutivos e descansar dois. Cada empregado recebe R\$30 por dia.

**Tabela 3.7**

<i>Operação</i>	<i>Predecessores imediatos</i>	<i>Tempo em segundos</i>
1	-	15
2	-	19
3	1,2	11
4	1	17
5	3	12
6	4	21
7	5,6	10

**Tabela 3.8**

	<i>Segunda</i>	<i>Terça</i>	<i>Quarta</i>	<i>Quinta</i>	<i>Sexta</i>	<i>Sábado</i>	<i>Domingo</i>
Empregados	10	6	8	5	9	4	6

- Determine o número de empregados em tempo integral de forma a minimizar a despesa total com salários.
- Suponha que a loja permita um dia de trabalho adicional. Cada empregado ganha R\$30 por dia nos cinco primeiros dias e R\$50 no dia extra. Determine o número total de empregados para minimizar o custo total.

**Exercício 3.8** O analista de sistemas em uma universidade deseja acessar cinco diferentes arquivos espalhados em dez diferentes discos, como mostrado na Tabela 3.9. A capacidade de armazenamento requerida em cada um dos discos é: disco 1, 30GB; disco 2, 50GB; disco 3, 10GB; disco 4, 20GB; disco 5, 10GB; disco 6, 40GB; disco 7, 30GB; disco 8, 10GB; disco 9, 20GB; disco 10, 20GB.

- Formule o problema que determina o conjunto de discos com a mínima capacidade de armazenamento requerida e que contenha todos os arquivos em pelo menos um disco. Não se pode utilizar apenas parte do disco.
- Modifique sua formulação para atender à seguinte restrição: se o disco 3 ou o disco 5 são usados, então o disco 2 deve ser usado.

**Exercício 3.9** Uma empresa siderúrgica fabrica uma série de produtos a partir de lingotes que são moldados com o aço que sai dos altos-fornos. Se o lingote  $i$  é usado para fazer o produto  $j$ , incorre-se em uma perda de material  $c_{ij}$ . A empresa produz cem produtos diferentes e cada produto  $j$  só pode ser feito a partir de um lingote pertencente a um conjunto  $I_j$ . Há 400 diferentes formas de moldar os lingotes, mas, por restrições do espaço disponível para estocagem, decidiu-se que seriam mantidos estoques apenas de seis tipos de lingotes. Formule o problema de selecionar os seis tipos de lingotes que minimizem a perda total de material.

**Exercício 3.10** Uma empresa tem quatro grupos de fornecedores que fornecem materiais elétricos (ME), materiais hidráulicos (MH), bens perecíveis (BP) e serviços terceirizados (ST), respectivamente. A empresa recebe produtos ou serviços de cada grupo fornecedor. O pagamento pode ser feito imediatamente ou adiado. Para isso, a empresa considera a possibilidade de utilizar quatro bancos, com o seguinte custo anual de manutenção de contas em cada banco: B1, R\$60 mil; B2, R\$50 mil; B3, R\$20 mil; B4, R\$30 mil. Os bancos arcaram com a dívida pelo número de dias indicado na Tabela 3.10. Após isso, o saque é feito na conta da empresa. A quantia emitida por banco não pode exceder R\$85 mil. Assuma que o dinheiro investido pela empresa renda 20% ao ano.

**Tabela 3.9**

Arquivo	Disco									
	1	2	3	4	5	6	7	8	9	10
1	x	x		x	x			x	x	
2	x		x							
3		x			x		x			x
4			x			x		x		
5	x	x		x		x	x		x	x

A dívida com cada grupo fornecedor é mostrada na Tabela 3.11. Formule o problema de escolha de bancos a serem utilizados pela empresa, e a quantia a ser enviada de cada banco para cada grupo fornecedor de forma a maximizar o lucro da empresa.

**Exercício 3.11\*** Existem  $n$  regiões em um bairro, tal que a região  $j$ ,  $j = 1, \dots, n$  tem  $a_j$  clientes (por exemplo, alunos) que requerem um serviço. Existem  $m$  locais candidatos a instalações (escolas) para fornecer este serviço. Se uma instalação é localizada no local  $i$ , então ela pode atender no máximo  $k_i$  clientes,  $i = 1, \dots, m$ . O número máximo de instalações que podem ser construídas é  $p$ . A distância (ou tempo de viagem) entre a região  $j$  e o local  $i$  é  $d_{ij}$ , e a distância máxima ou tempo que um cliente pode percorrer é  $D$ .

- Formule o problema de encontrar  $p$  locais para construir as instalações e as alocações de regiões a instalações construídas, de forma a maximizar o número total de clientes alocados a instalações dentro da distância  $D$ .
- Formule o problema de encontrar  $p$  locais para construir as instalações e as alocações de regiões a instalações construídas, de forma a minimizar a distância total de todos os clientes alocados a instalações fora da região de cobertura  $D$ .

**Exercício 3.12\*** Considere  $n$  tarefas a serem processadas em  $m$  máquinas paralelas. Cada tarefa gasta uma unidade de tempo de processamento na máquina em que pode ser processada. Seja

$$a_{ij} = \begin{cases} 1 & \text{se a máquina } i \text{ pode processar a tarefa } j \\ 0 & \text{caso contrário} \end{cases}$$

**Tabela 3.10**

Fornecedor	B1	B2	B3	B4
ME	8	3	5	6
MH	7	5	4	2
BP	6	9	5	12
ST	8	3	6	7

**Tabela 3.11**

Fornecedor	Dívida (R\$)
ME	30.000
MH	50.000
BP	60.000
ST	80.000

Determine uma alocação factível e balanceada de tarefas a máquinas, isto é, a diferença entre instantes de término de processamento entre quaisquer duas máquinas é no máximo uma unidade de tempo.

**Exercício 3.13** Um grande banco tem informação de contas armazenada em uma rede de computadores interconectados. Cada agência do banco deve ser designada a um processador. São dados:

$p_i$  = demanda de processamento da agência  $i$

$v_i$  = demanda de memória da agência  $i$

$t_i$  = capacidade de comunicação requerida pela agência  $i$

$P_j, V_j, T_j$  = capacidade de processamento, memória e comunicação, respectivamente, do processador  $j$

$k_j$  = número máximo de agências alocadas ao processador  $j$

$c_{ij}$  = custo da designação da agência  $i$  ao processador  $j$

Formule o problema de designação de custo mínimo de agências a processadores.

**Exercício 3.14** Uma empresa requer um número de empregados  $r_t$  em cada mês  $t$  de um horizonte de planejamento de dez meses. A empresa pode contratar em qualquer mês para preencher suas necessidades naquele mês ou em meses futuros. In corre-se em um custo fixo  $s_t$  de contratação no mês  $t$ , independentemente do número de empregados contratados. Se uma pessoa é contratada no mês  $t_1$  para atender necessidades do mês  $t_2$  tal que  $t_2 > t_1$ , então esta pessoa é paga como empregado excedente em cada mês  $t$ , de modo que  $t_1 \leq t \leq t_2$  com custo mensal adicional de  $c_t$ . Os dados são indicados na Tabela 3.12. O custo total consiste no custo fixo de contratação nos meses e no custo excedente de empregado. Formule o problema de recrutamento de empregados ao longo do horizonte de planejamento de forma a minimizar o custo total.

**Exercício 3.15\*** Considere que cinco tarefas  $j = 1, \dots, 5$  devem ser executadas. Cada tarefa tem um tempo de processamento  $p_j$  e requer  $W_{ij}$  operadores do tipo  $i$ ,  $i = 1, 2$ , mostrados na Tabela 3.13.

**Tabela 3.12**

Mês $t$	1	2	3	4	5	6	7	8	9	10
$r_t$	20	10	8	14	25	25	11	9	8	6
$s_t$	50	51	52	63	67	61	49	52	68	54
$c_t$	7	5	8	6	4	7	7	9	4	5

**Tabela 3.13**

Tarefa	1	2	3	4	5
$p_j$	17	7	13	9	7
$W_{1j}$	2	3	7	4	4
$W_{2j}$	0	6	9	0	5

O número total de operadores do tipo 1,  $W_1$ , é três e do tipo 2,  $W_2$ , é sete. As tarefas têm relação de precedência, de acordo com a Tabela 3.14. Sugestão: defina uma variável binária  $x_{jt}$ , que assume o valor 1 se o processamento da tarefa termina no instante  $t$ , e 0 em caso contrário.

- Desenhe um diagrama de Gantt com um programa factível de execução das cinco tarefas
- Formule o problema de programação de  $n$  tarefas que minimize o instante de término de processamento das tarefas (*makespan*).

**Exercício 3.16** Uma loja de eletrodomésticos que atende parte do estado de São Paulo detectou que a demanda de seus clientes varia drasticamente em relação à velocidade com que são atendidos. São vendidos eletrodomésticos em seis cidades: São Paulo, São José dos Campos, Santos, Jundiaí, São Carlos e Rio Claro. A venda anual de eletrodomésticos depende da existência ou não de um representante do fabricante a menos de 100km da cidade, como mostra a Tabela 3.15. A loja considera a alocação de representantes em Rio Claro, Santos, São José dos Campos e São Paulo. A distância (em km) entre as cidades é dada na Tabela 3.16. Formule o problema de maximização do lucro da loja.

**Exercício 3.17** Um caminhão de entrega de óleo de uma empresa contém cinco compartimentos, com capacidade para 2.700, 2.800, 1.100, 1.800 e 3.400 galões de combustível, respectivamente. A empresa deve entregar três tipos de combustível A, B e C para um cliente. Parte da demanda pode não ser atendida, mas, neste caso, a empresa deve arcar com os custos. As demandas, penalidades por galão não entregue e o número máximo de galões de demanda não atendida são descritos na Tabela 3.17. Cada compartimento pode levar apenas um tipo de combustível. Formule o problema de carregar o caminhão de forma a minimizar os custos de não-atendimento da demanda.

**Tabela 3.14**

Tarefa	Predecessores imediatos	Sucessores imediatos
1	–	4
2	–	5
3	–	5
4	1	–
5	2,3	–

**Tabela 3.15**

Representante a menos de 100km?	Vendas					
	São Paulo	S.J. Campos	Santos	Jundiaí	São Carlos	Rio Claro
Sim	700	1000	900	800	400	450
Não	500	750	700	450	200	300

**Exercício 3.18** No tratamento de tumores cerebrais com radiação, os médicos desejam bombardear o tecido que contém os tumores com o máximo de radiação possível, respeitando o limite máximo que o tecido normal pode receber sem ser danificado. Em uma simplificação, suponha seis tipos de feixes de radiação. Cada feixe possui uma intensidade e um alvo diferente no tecido. A região que contém um tumor pode ser dividida em seis regiões: três normais e três com tumor. Para cada feixe, a Tabela 3.18 indica a quantidade de radiação que atinge cada uma das regiões.

**Exercício 3.19** A nova política implementada em uma cidade do interior de Minas Gerais impõe que 20% dos alunos matriculados em cada escola de ensino médio devem ser negros. A cidade é composta por cinco bairros e duas escolas. O número de estudantes negros e brancos em cada um dos cinco bairros é mostrado na Tabela 3.19. A Tabela 3.20 mostra a distância percorrida por aluno para chegar a cada escola (em km). Assumindo que cada escola deve ter no mínimo 150 alunos, formule o problema que associa cada bairro a uma das escolas para minimizar a distância total que os estudantes da cidade devem percorrer para ir à escola.

**Exercício 3.20\*** Uma siderúrgica recebeu um pedido de 25 toneladas de aço. O carbono e o molibdênio devem ser responsáveis por 5% e 4% do peso total, respectivamente. O aço é produzido combinando-se três tipos de metal: lingotes de aço, aço de sucata e ligas de aço. Quatro lingotes estão disponíveis para compra. O peso (em toneladas), o custo por tonelada e a porcentagem de carbono e de molibdênio de cada lingote são mostrados na Tabela 3.21. O lingote deve ser comprado integralmente, mas pode-se usar apenas parte de seu peso total na mistura final. A

**Tabela 3.16**

	Rio Claro	São Paulo	S.J. Campos	Santos
Rio Claro	0	175	232	246
São Paulo	175	0	89	105
S.J. Campos	232	89	0	95
Santos	246	105	95	0
São Carlos	64	169	290	304
Jundiaí	95	60	141	131

**Tabela 3.17**

Combustível	Demanda	Custo por galão não entregue	Máximo de demanda não atendida
A	2900	10	500
B	4000	8	500
C	4900	6	500

**Tabela 3.18**

<i>Normal</i>			<i>Tumor</i>			<i>Feixe</i>
<i>I</i>	<i>2</i>	<i>3</i>	<i>I</i>	<i>2</i>	<i>3</i>	
16	12	8	20	12	6	1
12	10	6	18	15	8	2
9	8	13	13	10	17	3
4	12	12	6	18	16	4
9	4	11	13	5	14	5
8	7	7	10	10	10	6

**Tabela 3.19**

<i>Bairro</i>	<i>Brancos</i>	<i>Negros</i>
1	80	30
2	70	5
3	90	10
4	50	40
5	60	30

**Tabela 3.20**

<i>Bairro</i>	<i>Escola 1</i>	<i>Escola 2</i>
1	1,6	3,2
2	0,8	2,7
3	1,3	1,3
2	0,8	2,7
4	2,1	0,6
5	2,4	1,0

Tabela 3.22 mostra os três tipos de ligas de aço que podem ser compradas (qualquer quantidade em toneladas está disponível). A sucata pode ser comprada a R\$50 por tonelada e contém 3% de carbono e 9% de molibdênio. Formule o problema para minimizar o custo total da siderúrgica no atendimento do pedido.

**Exercício 3.21** Uma construtora considera a construção de três prédios. O tempo gasto e o número de trabalhadores necessários para completar cada um desses prédios são mostrados na Tabela 3.23. Os prédios 1, 2 e 3, quando construídos, fornecem um lucro anual de R\$70 mil, R\$50 mil e R\$60 mil, respectivamente. A construtora conta com no máximo 60 trabalhadores por ano. Apenas um prédio pode ser começado a cada ano e, além disso, o prédio 2 deve estar completo ao final do ano 4. Formule o problema que maximize o lucro total da construtora ao final do ano 4.

**Exercício 3.22** Um novo shopping acaba de ser construído e seu dono precisa determinar que tipos de lojas poderão alugar o espaço disponível. O shopping conta com 1.000m<sup>2</sup> de área total. A Tabela 3.24 mostra a área ocupada por tipo de loja, assim como o número mínimo e má-

**Tabela 3.21**

Lingote	Peso (ton)	Custo por ton (R\$)	Carbono (%)	Molibdênio (%)
1	5	150	5	3
2	3	138	4	3
3	4	115	5	4
4	6	75	3	4

**Tabela 3.22**

Liga	Custo por ton (R\$)	Carbono (%)	Molibdênio (%)
1	270	9	6
2	215	7	7
3	186	5	8

**Tabela 3.23**

Prédio	Tempo deconstrução (anos)	Número de trabalhadores
1	2	30
2	2	20
3	3	20

ximo de lojas daquele tipo que podem existir no shopping. O lucro anual de cada loja depende de quantas outras lojas do mesmo tipo existem no shopping, como mostrado na Tabela 3.25 em unidades de R\$10 mil. Cada loja paga um aluguel ao shopping correspondente a 5% do lucro anual. Formule o problema de maximização de lucro do shopping.

**Exercício 3.23** Uma termelétrica tem três caldeiras. Se uma caldeira é utilizada, ela pode produzir uma quantidade de vapor (em toneladas) entre um máximo e um mínimo, como indicado na Tabela 3.26. Essa tabela também mostra o custo de produção de 1 tonelada de vapor em

**Tabela 3.24**

<i>Tipo de loja</i>	<i>Área (m<sup>2</sup>)</i>	<i>Mínimo</i>	<i>Máximo</i>
Calçados	60	1	3
Departamentos	150	1	3
Jóias	50	1	3
Livros	70	0	3
Roupas	90	1	3

**Tabela 3.25**

<i>Tipo de Loja</i>	<i>Número de lojas</i>		
	<i>1</i>	<i>2</i>	<i>3</i>
Calçados	11	8	6
Departamentos	29	17	15
Jóias	14	12	9
Livros	22	11	7
Roupas	25	15	10

**Tabela 3.26**

<i>Caldeira</i>	<i>Mínimo (ton)</i>	<i>Máximo (ton) produzida (R\$)</i>	<i>Custo por tonelada</i>
1	500	1.000	20
2	300	900	17
3	400	800	13

cada uma das caldeiras. O vapor é utilizado na alimentação de três turbinas. A Tabela 3.27 mostra a quantidade mínima e máxima de vapor que cada turbina pode processar e a quantidade de energia liberada por tonelada de vapor processado. Essa tabela mostra, ainda, o custo do processamento de uma tonelada de vapor em cada turbina. Formule o problema que minimize o custo de produção de 8.000Kwh de energia.

**Exercício 3.24** Um ecologista brasileiro, trabalhando na Amazônia, foi contratado para realizar a divisão de parte da floresta em reservas florestais. Estudos recentes dividiram a floresta em dez regiões. O trabalho do ecologista consiste em formar cinco reservas a partir dessas informações, observando o número de predadores e presas que cada região contém. A Tabela 3.28 mostra esses dados para cada região (em milhares). Os animais não podem ser removidos de sua região, e cada reserva deve conter entre 80 mil e 130 mil animais. Sabe-se que o ideal é que o número de presas deve ser maior que o número de predadores em uma reserva, para garantir o equilíbrio entre as espécies. Formule um problema para ajudar o ecologista a formar as cinco reservas garantindo o máximo equilíbrio ambiental.

**Tabela 3.27**

Turbina	Mínimo (ton)	Máximo (ton)	Kwh por tonelada de vapor	Custo por tonelada processada (R\$)
1	300	600	4	5
2	500	800	5	7
3	600	900	6	9

**Tabela 3.28**

Região	Predadores	Presas
1	40	17
2	29	24
3	20	22
4	25	22
5	20	57
6	19	32
7	37	7
8	10	11
9	35	27
10	35	32

**Exercício 3.25** Resolva o problema de programação inteira:

$$\begin{aligned} z &= \max x_1 + x_2 \\ -2x_1 + 2x_2 &\leq 3 \\ 7x_1 + 3x_2 &\leq 22 \\ \mathbf{x} &\in \mathbb{Z}_+^2 \end{aligned}$$

- a) De forma gráfica.
- b) Pelo método *branch-and-bound*, utilizando duas estratégias de escolha de nó: em profundidade e maior limitante superior (no nó raiz, ramifique na variável  $x_1$ ).
- c) Pelo método de Gomory.

**Exercício 3.26** Resolva o problema da mochila 0-1 pelo método *branch-and-bound*:

$$\begin{aligned} z &= \max 10x_1 + 15x_2 + 36x_3 + 20x_4 + 15x_5 + 18x_6 + 20x_7 \\ 10x_1 + 17x_2 + 49x_3 + 30x_4 + 11x_5 + 21x_6 + 31x_7 &\leq 100 \\ \mathbf{x} &\in \mathbb{B}^7 \end{aligned}$$

- a) Aplique a estratégia do nó com maior limitante superior.
- b) Aplique a estratégia de busca em profundidade, escolhendo o nó com a restrição  $\leq$ .

**Exercício 3.27** Considere o problema 0-1:

$$\begin{aligned} z &= \max 5x_1 - 7x_2 - 10x_3 + 3x_4 - 5x_5 \\ x_1 + 3x_2 - 5x_3 + x_4 + 4x_5 &\leq 0 \\ -2x_1 - 6x_2 + 3x_3 - 2x_4 - 2x_5 &\leq -4 \\ 2x_2 - 2x_3 - x_4 + x_5 &\leq -2 \\ \mathbf{x} &\in \mathbb{B}^5 \end{aligned}$$

Simplifique usando desigualdades lógicas.

**Exercício 3.28** Para cada um dos quatro conjuntos abaixo, determine graficamente  $\text{conv}(X)$ .

- a)  $X = \{\mathbf{x} \in \mathbb{B}^2 : 5x_1 - 7x_2 \leq 2\}$
- b)  $X = \{(x, y), x \in \mathbb{R}_+^1, y \in \mathbb{B}^1 : x \leq 15y, x \leq 8\}$
- c)  $X = \{(x, y), x \in \mathbb{R}_+^1, y \in \mathbb{Z}_+^1 : x \leq 5y, x \leq 12\}$
- d)  $X = \{\mathbf{x} \in \mathbb{Z}_+^2 : 2x_1 + 7x_2 \leq 18, x_2 - x_1 \leq 5, 2x_1 + 9x_2 \geq 20\}$

**Exercício 3.29** Para o conjunto  $X = \{\mathbf{x} \in \mathbb{Z}_+^4 : 2x_1 + 6x_2 + 15x_3 + 10x_4 \geq 20\}$ , identifique uma desigualdade válida que corte o ponto  $\mathbf{x} = (0, 0, \frac{4}{3}, 0)$ .

**Exercício 3.30** Mostre que a desigualdade  $x_2 + x_3 + 2x_4 \leq 6$  é válida para

$$X = \{\mathbf{x} \in \mathbb{Z}_+^4 : 6x_1 + 7x_2 + 10x_3 + 14x_5 \leq 40\}$$

**Exercício 3.31** Considere os problemas de localização de facilidades com capacidade limitada sem/com fonte única, descritos na Seção 3.5.2. Mostre que a desigualdade

$$\sum_{i \in I} Q_i y_i \geq \sum_{j \in J} q_j$$

é válida.

Página deixada intencionalmente em branco

# 4

## Otimização em redes

Muitos problemas práticos de otimização podem ser melhor analisados utilizando-se uma estrutura especial denominada grafo ou rede. Problemas de otimização em redes aparecem em diversas aplicações e em diversas formas. Algumas delas envolvem o transporte ou fluxo de um item ou itens de um nó a outro na rede com um determinado objetivo. As aplicações surgem na transmissão de mensagens em redes de comunicação de dados, no envio de água em uma rede de distribuição de água, no transporte de carga em uma rede viária etc. Em outras aplicações, à primeira vista, não se observa relação direta com grafos ou redes, como no planejamento e na programação da produção, programação de projetos, programação de máquinas e de pessoal, distribuição de bens etc. Exemplos de otimização em redes nos capítulos anteriores aparecem nas Seções 2.2.2 (transporte, transbordo e designação), 2.2.4 (programação de projetos), 2.2.5 (fluxo de caixa), 3.4.5 (caixeiros-viajantes), 3.4.6 (carteiro chinês), 3.5.1 (roteamento de veículos), entre outros.

Neste capítulo, apresentamos alguns problemas que exibem esta estrutura especial, que pode ser explorada para o desenvolvimento de métodos mais eficientes de resolução. Iniciamos o capítulo apresentando algumas noções básicas de redes.

### 4.1 NOÇÕES BÁSICAS DE REDES E GRAFOS

Em muitos problemas podemos identificar conjuntos de elementos fundamentais, os quais mantêm uma relação entre si, como, por exemplo, um conjunto de cidades e um conjunto de estradas que ligam as cidades diretamente, ou um conjunto de reservatórios e bairros em uma cidade e a rede de aquedutos ligando-os, ou um conjunto de tarefas e a relação de precedência entre elas etc.

Supomos neste livro que o número de elementos fundamentais é finito, de modo que podemos enumerá-los,  $v_1, v_2, \dots, v_n$  ou, simplesmente, 1, 2, ...,  $n$ . Estes elementos fundamentais são chamados de *vértices* ou *nós*. Para indicar que o vértice  $i$  se relaciona com o vértice  $j$ , usamos o par  $(v_i, v_j)$  ou, simplesmente,  $(i, j)$ .

**Definição 4.1** (*grafo* e *rede*) Seja  $N$  um conjunto finito, cujos elementos são chamados *nós* (ou *vértices*) e  $E$  um conjunto de pares de nós, cujos elementos  $(i, j)$  são chamados *arestas*. O par  $G = (N, E)$  é chamado *grafo*. Uma *rede* é um grafo cujos nós e/ou arestas têm valores associados.

Por simplicidade, a menos que possa haver alguma confusão para o entendimento, neste livro não fazemos distinção entre os termos grafos e redes. Os dois termos são usados como sinônimos.

O número de elementos de um conjunto  $X$  é chamado *cardinalidade* de  $X$  e denotamos por  $|X|$ . Assim, se um grafo  $G = (N, E)$  tem  $n$  nós e  $m$  arestas, significa que  $|N| = n$  e  $|E| = m$  (isto é, a cardinalidade de  $N$  é  $n$  e a cardinalidade de  $E$  é  $m$ ).

Um grafo  $G = (N, E)$  pode ser representado graficamente da seguinte maneira. Cada nó  $i$  é representado por um círculo com o rótulo  $i$ , e cada aresta  $(i, j)$  por uma linha conectando os dois círculos rotulados de  $i$  e  $j$ .

**Exemplo 4.1** Considere o conjunto de nós  $N = \{1, 2, 3, 4\}$  e o conjunto de arestas  $E = \{(1, 2), (1, 4), (4, 3), (2, 3), (2, 4)\}$ . A representação gráfica do grafo  $G = (N, E)$  é ilustrada na Figura 4.1. ■

Neste exemplo, os nós  $\{1, 2, 3, 4\}$  podem representar quatro cidades e as arestas representar as estradas ligando-as diretamente, ou podem representar quatro computadores e as arestas, cabos de fibra óptica conectando-os, ou podem representar quatro componentes eletrônicos em um circuito integrado e as arestas, as ligações entre eles etc.

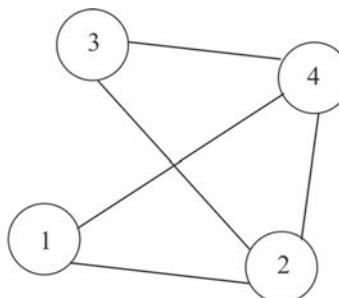
Se uma aresta  $(i, j)$  pertence a  $E$ , dizemos que a aresta  $(i, j)$  é incidente nos nós  $i$  e  $j$ , e estes nós são adjacentes. Por exemplo, no grafo do Exemplo 4.1, os nós 1 e 4 são adjacentes, a aresta  $(1, 4)$  é incidente nos nós 1 e 4; e os nós 1 e 3 não são adjacentes.

As arestas que representam as relações entre os nós podem ser orientadas ou não. Nos exemplos apresentados anteriormente, as orientações são irrelevantes. É o caso, por exemplo, de uma aresta que representa uma estrada de mão dupla entre duas cidades em que é possível ir de uma cidade a outra, e vice-versa. O mesmo não se poderia dizer se a estrada fosse de mão única. O grafo cujas arestas não têm orientação é denominado *grafo não orientado*, em contrapartida ao grafo cujas arestas são todas orientadas, que é denominado *grafo orientado*, formalizado a seguir.

**Definição 4.2** (*grafo orientado* e *rede orientada*) Um grafo  $G = (N, E)$  no qual as arestas são pares ordenados (subconjunto de  $N \times N$ ) é chamado *grafo orientado* ou *dígrafo*. Neste caso, o par ordenado  $(i, j)$  é chamado *arco*, e  $i$  é o nó inicial e  $j$  o nó final.<sup>1</sup> Uma *rede orientada* é um grafo orientado cujos nós e/ou arcos têm valores associados.

Um grafo orientado é representado graficamente de forma análoga ao grafo, porém, o arco  $(i, j)$  é representado por uma flecha de  $i$  para  $j$ , indicando a orientação relevante. Dada a orientação do arco  $(i, j)$ , dizemos que o arco é de saída (ou emergente) do nó  $i$  e de entrada no nó  $j$ . O nó inicial  $i$  é chamado de *nó cauda* do arco  $(i, j)$  e o nó final  $j$  é chamado de *nó cabeça* do arco  $(i, j)$ . O arco  $(i, j)$  é incidente nos nós  $i$  e  $j$ , e o nó  $j$  é adjacente ao nó  $i$ , mas não o reverso, a não ser que o arco  $(j, i)$  também pertença a  $E$ .

A cada aresta ou arco  $(i, j) \in E$  pode existir um ou mais valores numéricos associados (por exemplo,  $c_{ij}, l_{ij}, u_{ij}, \dots$ ) representando parâmetros relativos à aresta ou ao arco, por exemplo, custo ou comprimento da aresta ou do arco, limite inferior do fluxo sobre a aresta ou o arco, limite superior do fluxo sobre a aresta ou o arco etc. A cada nó  $i \in N$  pode existir um ou mais valores numéricos associados  $d_i$ , por exemplo, a demanda ou suprimento, a capacidade do nó etc. Os va-



**Figura 4.1** Exemplo de representação de um grafo.

<sup>1</sup> Neste caso, alguns autores preferem a notação  $G(N, A)$ , em que  $A$  é o conjunto de arcos  $(i, j)$ . Por simplicidade, neste livro usamos a notação  $G(N, E)$  para grafos orientados e não-orientados.

lores associados às arestas ou aos arcos podem ser indicados sobre eles, e os valores associados aos nós podem ser indicados ao lado deles em uma representação gráfica.

**Exemplo 4.2** Considere o conjunto de vértices  $N = \{1, 2, 3, 4\}$ , o conjunto de arcos  $E = \{(1, 2), (1, 4), (4, 3), (2, 3), (2, 4)\}$  e os valores associados a cada arco,  $c_{12} = 2$ ,  $c_{14} = 5$ ,  $c_{23} = 1$ ,  $c_{24} = 1$  e  $c_{43} = 3$ . A representação gráfica da rede orientada é ilustrada na Figura 4.2. ■

Os nós do Exemplo 4.2 podem representar quatro tarefas que devem ser executadas, e os arcos, a relação de precedência entre elas (por exemplo, a tarefa 2 só pode ser executada depois que a tarefa 1 for concluída). Os valores associados aos arcos podem representar tempos de espera, ou seja, para que uma tarefa possa ser executada, é preciso esperar um tempo após o término da tarefa que a precede (por exemplo, é preciso esperar a parede secar após a tarefa “pintura”, antes de começar a tarefa “colocar os lustres”, na construção de uma casa).

**Definição 4.3 (caminho)** Um *caminho* de um nó  $i_0$  a um nó  $i_k$  é uma seqüência de arcos  $C = \{(i_0, i_1), (i_1, i_2), (i_2, i_3), \dots, (i_{k-1}, i_k)\}$ , no qual o nó inicial de cada arco é o nó final do arco imediatamente anterior na seqüência, e  $i_0, i_1, i_2, i_3, \dots, i_k$  são todos nós distintos. Portanto, todos os arcos em um caminho estão direcionados coerentemente, apontando do nó  $i_0$  ao nó  $i_k$  (nó cabeça do arco precedente para o nó cauda do arco seguinte).

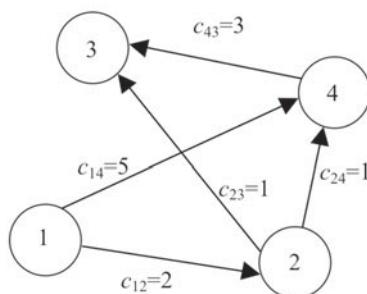
Por exemplo, no grafo da Figura 4.2, um caminho possível do nó 1 ao nó 3 seria  $\{(1, 2), (2, 4), (4, 3)\}$ .

**Definição 4.4 (cadeia)** Uma *cadeia* é uma estrutura similar à de um caminho, exceto que os arcos não precisam estar coerentemente orientados, ou seja, uma cadeia é uma seqüência de arcos de modo que cada arco tem exatamente um nó em comum com o arco imediatamente anterior na seqüência.

Por exemplo, no grafo da Figura 4.2, uma cadeia do nó 1 ao nó 4 poderia ser  $\{(1, 2), (2, 3), (4, 3)\}$ . Note que todo caminho é uma cadeia, mas nem toda cadeia é um caminho.

**Definição 4.5 (circuito, ciclo)** Um *circuito* é um caminho fechado, ou seja, é um caminho de um nó  $i_0$  a um nó  $i_k$ , em que  $i_k = i_0$ . O correspondente ao circuito, no caso da cadeia, é denominado *ciclo*, ou seja, o ciclo é uma cadeia fechada (um nó, inicial ou final, do primeiro arco da seqüência coincide com um nó, inicial ou final, do último arco da seqüência).

Na Figura 4.2, não há nenhum circuito, mas apenas ciclos, por exemplo, os ciclos  $\{(2, 3), (4, 3), (2, 4)\}$  e  $\{(1, 4), (2, 4), (1, 2)\}$ . Neste caso, dizemos que o grafo é acíclico, caso contrário seria cíclico. As definições anteriores são para caminhos, cadeias, ciclos e circuitos simples. Existem extensões para casos mais complexos mas não são consideradas neste livro.<sup>2</sup>



**Figura 4.2** Exemplo da representação gráfica de uma rede orientada.

<sup>2</sup> Estas definições podem ser encontradas, por exemplo, em Boaventura Netto (2003).

**Definição 4.6** (*grafos fraca e fortemente conectados*) Um grafo é *fracamente conectado* (ou simplesmente conectado) se existe pelo menos uma cadeia entre quaisquer dois de seus nós, e *fortemente conectado* se existe pelo menos um caminho de cada nó a todos os demais nós do grafo.

Para facilitar a apresentação e sem perda de generalidade, admite-se que os grafos considerados neste livro são sempre fracamente conectados.

**Definição 4.7** (*árvore, subgrafo e árvore geradora de um grafo*) Uma *árvore* é um grafo conectado sem ciclos. Diz-se que um grafo  $G' = (N', E')$  é um *subgrafo* de  $G = (N, E)$  se  $N' \subseteq N$  e  $E' \subseteq E$ , com a condição de que, se  $(i, j) \in E'$ , então  $i$  e  $j$  também devem pertencer a  $N'$ . Uma *árvore geradora* de um grafo  $G$  é um subgrafo de  $G$  que é uma árvore e inclui todos os nós do grafo  $G$ .

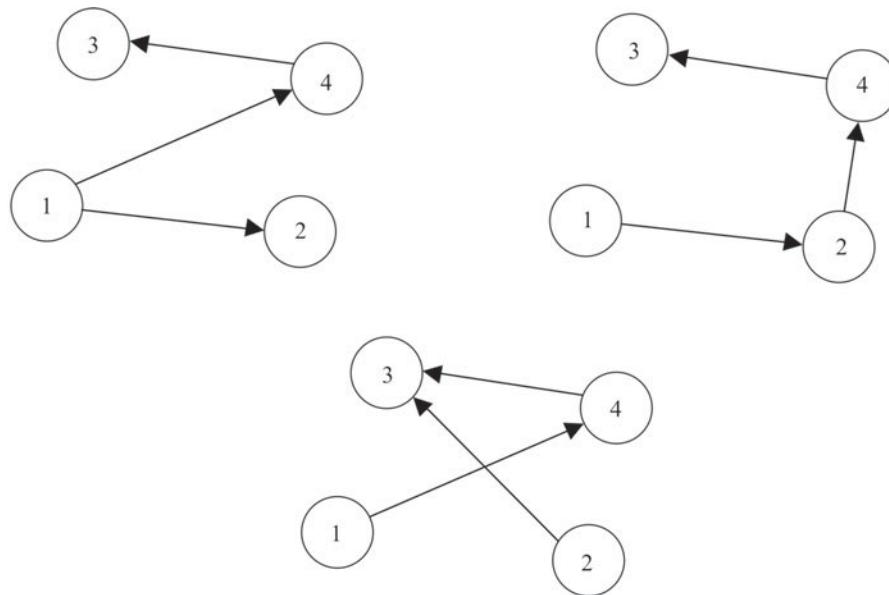
Árvores geradoras são de grande interesse no estudo de fluxos em redes e a seguir apresentamos algumas propriedades úteis para caracterizá-las.

**Propriedade 4.1** (*árvores geradoras*) Considere um grafo  $G = (N, E)$ , com  $|N| = n$  (isto é,  $G$  tem  $n$  nós), e um subgrafo  $G' = (N', E')$  de  $G$ . As seguintes afirmações são equivalentes:

- i)  $G' = (N, E')$  é uma árvore geradora de  $G$
- ii)  $|E'| = n - 1$  (isto é,  $G'$  tem  $n - 1$  arcos) e  $G'$  é conectado
- iii)  $|E'| = n - 1$  e  $G'$  não tem ciclos

Em palavras, toda árvore geradora de um grafo com  $n$  nós tem  $n - 1$  arcos e basta que seja um subgrafo conectado ou sem ciclos.

**Exemplo 4.3** Na Figura 4.3, ilustramos algumas árvores geradoras do grafo apresentado na Figura 4.2. O leitor pode verificar a validade das propriedades de árvores geradoras. ■



**Figura 4.3** Exemplos de árvores geradoras do grafo da Figura 4.2.

Para evidenciar as propriedades das árvores geradoras, na Figura 4.4, ilustramos alguns subgrafos do grafo apresentado na Figura 4.2, com  $n - 1 = 3$  arcos, mas que não são árvores (note que necessariamente formam ciclos e são desconectados. Por quê?).

#### *Representação matricial de grafos*

Um grafo também pode ser representado por matrizes de diversas formas. Essas matrizes são úteis na formalização de modelos matemáticos de certos problemas que podem ser representados por grafos. Tais representações também podem ser úteis para armazenar o grafo (ou rede) no computador.

**Definição 4.8** (*Matriz de adjacências*) Seja  $G = (N, E)$  um grafo em que  $|N| = n$ . A matriz  $\mathbf{M} = [m_{ij}]$   $n \times n$  tal que  $m_{ij} = 1$  se  $(i, j) \in E$  e  $m_{ij} = 0$ , caso contrário é chamada *matriz de adjacências*.

**Exemplo 4.4** Os grafos nos Exemplos 4.1 (não orientado) e 4.2 (orientado) são representados, respectivamente, pelas seguintes matrizes de adjacências:

$$\mathbf{M} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{matrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} \end{matrix}$$

$$\mathbf{M} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{matrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} \end{matrix}$$

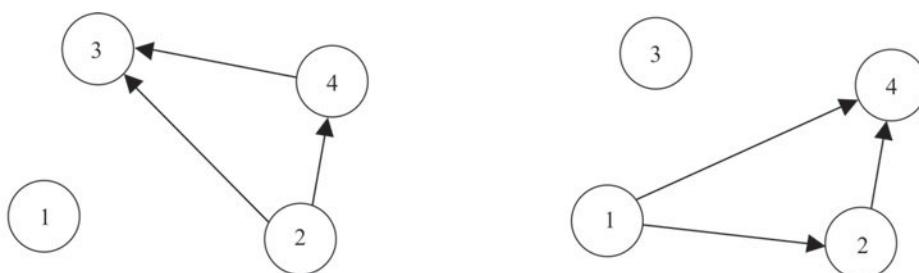
Note que a matriz de adjacências de grafos não orientados é sempre simétrica ( $\mathbf{M} = \mathbf{M}^T$ ). No caso de uma rede, com o valor  $c_{ij} \neq 0$  associado ao arco  $(i, j)$ , podemos colocar essa informação em  $\mathbf{M}$ , fazendo  $m_{ij} = c_{ij}$  para todo  $(i, j) \in E$  e  $m_{ij} = 0$  em caso contrário.

Uma outra forma bastante utilizada para representação de um grafo é a *matriz de incidência* nó-aresta ou nó-arco.

**Definição 4.9** (*Matriz de incidência nó-aresta*) Seja  $G = (N, E)$  um grafo não orientado em que  $|N| = n$  e  $|E| = m$ . A matriz  $\mathbf{P} = [p_{ik}]$ ,  $n \times m$ , tal que

- $p_{ik} = 1$  se a aresta  $k$  é definida por  $(i, j) \in E$  ou  $(j, i) \in E$  (ou seja, a aresta  $k$  é incidente ao nó  $i$ ) e
- $p_{ik} = 0$  caso contrário

é chamada *matriz de incidência nó-aresta*.



**Figura 4.4** Exemplos de subgrafos com três arcos que não são árvores geradoras do grafo da Figura 4.2.

**Definição 4.10** (*Matriz de incidência nó-arco*) Seja  $G = (N, E)$  um grafo orientado em que  $|N| = n$  e  $|E| = m$ . A matriz  $\mathbf{P} = [p_{ik}]$ ,  $n \times m$ , tal que

- $p_{ik} = +1$  se o arco  $k$  é  $(i, j)$  (isto é, o arco  $k$  sai do nó  $i$ )
- $p_{ik} = -1$  se o arco  $k$  é  $(j, i)$  (isto é, o arco  $k$  entra no nó  $i$ ) e
- $p_{ik} = 0$  caso contrário

é chamada *matriz de incidência nó-arco*.

**Exemplo 4.5** Os grafos dos Exemplos 4.1 e 4.2 podem ser representados, respectivamente, pelas seguintes matrizes de incidência nó-aresta e nó-arco:

$$\mathbf{P} = \begin{bmatrix} (1,2) & (1,4) & (2,3) & (2,4) & (3,4) \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} (1,2) & (1,4) & (2,3) & (2,4) & (4,3) \\ 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & 0 & -1 & 1 \end{bmatrix}$$

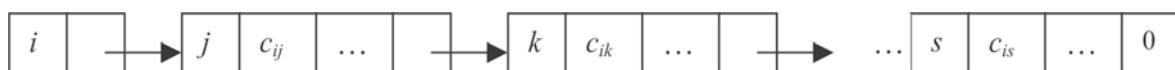
■

#### Outras representações de redes\*

Uma representação adequada de redes pode influenciar o tempo que um algoritmo requer para resolver o problema em um computador. Assim, existem outras representações de grafos (redes) em computadores que economizam espaço de memória e recuperam rapidamente informações de adjacência entre nós.<sup>3</sup>

Uma delas é por listas de adjacências de nós. Uma lista de adjacência de nós,  $A(i)$ , do nó  $i$ , é o conjunto de nós adjacentes ao nó  $i$ , ou seja,  $A(i) = \{j \in N \text{ tal que } (i, j) \in E\}$ . Uma representação da lista de adjacência de nós armazena, simplesmente, a lista de adjacência de cada nó utilizando uma lista encadeada simples. A lista encadeada referente à lista  $A(i)$  encadeia  $|A(i)|$  células, cada uma das quais com vários campos. Nestes campos, todos os dados de interesse do arco que liga o nó  $i$  ao nó  $j$  em  $A(i)$  são armazenados. Cada célula representa um arco  $(i, j)$  em que  $j \in A(i)$ . Em um dos campos da célula, armazena-se o nó  $j$  correspondente ao arco  $(i, j)$ ; em outro campo da célula, armazena-se um ponteiro que aponta para o endereço da próxima célula da lista. Se não existem mais células, o ponteiro aponta para o endereço 0. A lista inicia com uma célula mais simples, com apenas dois campos. No primeiro campo, armazena-se o próprio nó  $i$  e, no segundo campo, armazena-se um ponteiro que aponta para a primeira célula da lista de adjacências de  $i$ . Se a lista de adjacências é vazia, aponta-se para o endereço 0. A Figura 4.5 esquematiza a lista encadeada simples referente a um determinado nó  $i$ , em que  $A(i) = \{j, k, \dots, s\}$ .

A seguir, apresentamos um exemplo de representação por lista de adjacência de um grafo.



**Figura 4.5** Lista encadeada simples de um nó  $i$ .

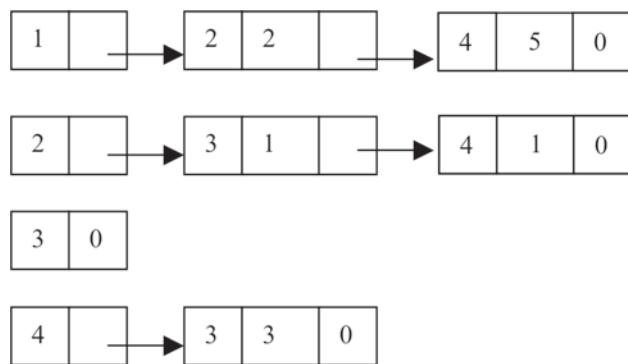
<sup>3</sup> Ver, por exemplo, Ahuja *et al.* (1993).

**Exemplo 4.6** A representação por lista de adjacência da rede do Exemplo 4.2 é apresentada na Figura 4.6. ■

Uma outra representação similar à lista de adjacência de nós é a representação por nós sucessores e predecessores. Esta representação não utiliza listas encadeadas, mas simplesmente tabelas. Inicialmente, associa-se uma seqüência única de números a cada arco do grafo, iniciando pelo número 1, e determinando uma ordem dos arcos. Estes são numerados em ordem: primeiro todos os arcos que emergem do nó 1, depois todos os arcos que emergem do nó 2, e assim por diante, até o nó  $n$  (admitindo-se um grafo com  $n$  nós, ou seja,  $|N| = n$ ). Com isso, todos os arcos que emergem de um mesmo nó têm numeração sucessiva. A ordem de numeração dos arcos que emergem de um mesmo nó não é importante. Se  $|E| = m$ , isto é, o número de arcos do grafo é  $m$ , armazenam-se as informações em uma tabela **T** de tamanho  $(m \times (k + 2))$ , em que  $k$  é o número de informações de interesse associadas aos arcos. Cada linha da tabela **T** armazena as informações sobre cada arco da lista de arcos, conforme a sua numeração; a linha 1 da tabela armazena as informações do arco 1, a linha 2 da tabela armazena as informações do arco 2, e assim por diante. A tabela **T** possui  $k + 2$  colunas, contendo as seguintes informações relevantes dos arcos: o nó cauda, o nó cabeça, a informação tipo 1 de interesse do arco, a informação tipo 2 de interesse do arco, ..., a informação tipo  $k$  de interesse do arco. Em um vetor **s**, com  $n + 1$  elementos, faz-se  $s_i$  (correspondente ao nó  $i$ ) igual ao menor número do arco dentre todos aqueles que emergem do nó  $i$ . Com isso, as informações relevantes dos arcos que emergem do nó  $i$  (conseqüentemente, os nós sucessores de  $i$ ) podem ser obtidas da tabela **T** diretamente de suas linhas  $s_i$  até  $s_{i+1} - 1$ , em que se define que  $s_{n+1} = m + 1$ . Se  $s_{i+1} - 1$  for menor que  $s_i$ , então não há arcos emergendo de  $i$ . Essa representação apresentada é por nós sucessores pelo fato de considerar os arcos que emergem de cada nó na rede. Um exemplo dessa representação é apresentado a seguir.

**Exemplo 4.7** Para representar a rede do Exemplo 4.2 por nós sucessores, precisamos numerar os arcos. Enumeremos os arcos do grafo da seguinte forma:  $(1, 2) \leftarrow$  arco 1;  $(1, 4) \leftarrow$  arco 2;  $(2, 4) \leftarrow$  arco 3;  $(2, 3) \leftarrow$  arco 4;  $(4, 3) \leftarrow$  arco 5, ou seja,  $m = 5$ . A tabela **T** correspondente à rede do Exemplo 4.2 fica, portanto,

$$\mathbf{T} = \begin{bmatrix} 1 & 2 & 2 \\ 1 & 4 & 5 \\ 2 & 4 & 1 \\ 2 & 3 & 1 \\ 4 & 3 & 3 \end{bmatrix}$$



**Figura 4.6** Lista de adjacência da rede do Exemplo 4.2.

em que temos apenas três colunas, pois há apenas uma informação de interesse correspondente a cada arco no grafo, indicado na coluna 3 de  $\mathbf{T}$ . Neste exemplo, temos 4 nós, portanto,  $n = 4$ . O vetor  $\mathbf{s}$ , com  $n + 1 = 5$  elementos, fica, portanto,  $\mathbf{s} = [1 \ 3 \ 5 \ 5 \ 6]^T$ , ou seja,  $s_1 = 1$ ,  $s_2 = 3$ ,  $s_3 = 5$ ,  $s_4 = 5$  e  $s_5 = 6$ . Com o vetor  $\mathbf{s}$ , sabemos que os dados dos arcos que emergem do nó 1 estão nas linhas  $s_1$  até  $s_2 - 1$  da tabela  $\mathbf{T}$ , isto é, linhas 1 e 2; os dados dos arcos que emergem do nó 2 estão nas linhas  $s_2$  até  $s_3 - 1$ , ou seja, linhas 3 e 4 da tabela  $\mathbf{T}$ ; os dados dos arcos que emergem do nó 3 estão nas linhas  $s_3$  até  $s_4 - 1$ , ou seja, não há arcos emergindo do nó 3, pois  $s_3 = 5$  e  $s_4 - 1 = 4$ ; os dados dos arcos que emergem do nó 4 estão nas linhas  $s_4$  até  $s_5 - 1$ , ou seja, linha 5 da tabela  $\mathbf{T}$ . ■

A coluna 1 da tabela  $\mathbf{T}$  (nós cauda) nesta representação por nós sucessores é desnecessária (redundante), pois os nós cauda podem ser obtidos pelo vetor  $\mathbf{s}$ . Entretanto, por conveniência, mantivemos essa coluna porque utilizaremos essa matriz  $\mathbf{T}$  mais adiante em uma representação mais geral.

A representação por tabelas apresentada foi por nós sucessores. De maneira similar, podemos fazer uma representação por tabelas por nós predecessores. Neste caso, a numeração dos arcos é feita pelos nós que chegam a cada nó  $i$ . Os arcos são numerados em ordem: primeiro todos os arcos que chegam ao nó 1, depois todos os arcos que chegam ao nó 2, e assim por diante, até o nó  $n$ . Com isso, todos os arcos que chegam a um mesmo nó têm numeração sucessiva. A ordem de numeração dos arcos que chegam a um mesmo nó não é importante. Uma matriz  $\mathbf{T}'$  de informações dos arcos é construída, similar à matriz  $\mathbf{T}$ . No vetor  $\mathbf{p}$ , similar a  $\mathbf{s}$ , com  $n + 1$  elementos, faz-se  $p_i$  (correspondente ao nó  $i$ ) igual ao menor número do arco dentre todos os que chegam ao nó  $i$ . Com isso, as informações relevantes dos arcos que chegam ao nó  $i$  podem ser obtidas da tabela  $\mathbf{T}'$  nas linhas  $p_i$  até  $p_{i+1} - 1$ , definindo-se  $p_{n+1} = m + 1$ . Se  $p_{i+1} - 1 < p_i$ , então não há arcos chegando ao nó  $i$ .

**Exemplo 4.8** Para representar a rede do Exemplo 4.2 por nós predecessores, precisamos numerar os arcos. Enumeremos os arcos do grafo da seguinte forma:  $(1,2) \leftarrow$  arco 1;  $(2,3) \leftarrow$  arco 2;  $(4,3) \leftarrow$  arco 3;  $(1,4) \leftarrow$  arco 4;  $(2,4) \leftarrow$  arco 5. A tabela  $\mathbf{T}'$  correspondente à rede fica, portanto,

$$\mathbf{T}' = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 3 & 1 \\ 4 & 3 & 3 \\ 1 & 4 & 5 \\ 2 & 4 & 1 \end{bmatrix}$$

e o vetor  $\mathbf{p}$  fica  $\mathbf{p} = [1 \ 1 \ 2 \ 4 \ 6]^T$ , ou seja, deste vetor, sabemos que não há arcos chegando ao nó 1, pois  $p_1 = 1$  e  $p_2 - 1 = 0$ ; os dados dos arcos que chegam ao nó 2 estão nas linhas  $p_2$  até  $p_3 - 1$ , ou seja, linha 1 da tabela  $\mathbf{T}'$ ; os dados dos arcos que chegam ao nó 3 estão nas linhas  $p_3$  até  $p_4 - 1$ , ou seja, linhas 2 e 3 da tabela  $\mathbf{T}'$ ; os dados dos arcos que chegam ao nó 4 estão nas linhas  $p_4$  até  $p_5 - 1$ , ou seja, linhas 4 e 5 da tabela  $\mathbf{T}'$ . ■

Da representação por nós sucessores, podem ser derivados os nós predecessores de cada nó no grafo, e vice-versa. Em um algoritmo, para que se ganhe em eficiência (tempo de execução), pode ser interessante ter-se rapidamente essa informação dos nós predecessores e sucessores dos nós do grafo, sem que elas precisem ser derivadas quando necessárias. Assim, para ter esse ganho de eficiência, poderíamos armazenar ambas as representações apresentadas anteriormente. Observe, pelos Exemplos 4.7 e 4.8, que as tabelas  $\mathbf{T}$  e  $\mathbf{T}'$  contêm os mesmos dados, apenas em linhas diferentes. Assim, é possível fazer uma representação compacta dessas duas representações (por nós sucessores e predecessores), armazenando a tabela  $\mathbf{T}$ , os vetores  $\mathbf{p}$  e  $\mathbf{s}$  e mais um vetor de “conversão” que identifica as linhas trocadas entre  $\mathbf{T}$  e  $\mathbf{T}'$ .

**Exemplo 4.9** O vetor de conversão das representações apresentadas nos Exemplos 4.7 e 4.8 é  $\mathbf{c} = [1 \ 4 \ 5 \ 2 \ 3]^T$ , ou seja, a linha 1 da tabela  $\mathbf{T}'$  é a linha  $c_1$  da tabela  $\mathbf{T}$ , ou seja, a própria linha 1; a linha 2 da tabela  $\mathbf{T}'$  é a linha  $c_2$  da tabela  $\mathbf{T}$ , ou seja, a linha 4; a linha 3 da tabela  $\mathbf{T}'$  é a linha  $c_3$  da tabela  $\mathbf{T}$ , ou seja, a linha 5; a linha 4 da tabela  $\mathbf{T}'$  é a linha  $c_4$  da tabela  $\mathbf{T}$ , ou seja, a linha 2; e a linha 5 da tabela  $\mathbf{T}'$  é a linha  $c_5$  da tabela  $\mathbf{T}$ , ou seja, a linha 3. ■

## 4.2 ALGUMAS APLICAÇÕES E MÉTODOS DE RESOLUÇÃO

Muitas aplicações práticas importantes podem ser modeladas como problemas de otimização em grafos e métodos de solução eficientes podem ser desenvolvidos. A seguir apresentamos alguns exemplos.

### 4.2.1 O problema de caminho mínimo

Um dos problemas mais simples em grafos é o de determinar o *caminho mínimo* entre dois nós no grafo. Este problema aparece com freqüência em aplicações práticas, tanto diretamente quanto como um subproblema de outros mais difíceis.<sup>4</sup> Para introduzir o problema, apresentamos alguns exemplos.

**Exemplo 4.10** Suponha a entrega de uma mercadoria do depósito de uma fábrica até o endereço do cliente. Qual o menor caminho a percorrer? Este problema pode ser modelado como um problema de otimização em redes. Construa um grafo cujos nós correspondem às esquinas das ruas da cidade e inclua mais dois nós correspondentes ao depósito e ao endereço do cliente. Os arcos do grafo correspondem às ruas que ligam as esquinas, e o depósito e o endereço do cliente às esquinas de suas ruas. Qualquer caminho no grafo partindo do nó do depósito ao nó do endereço do cliente corresponde a um caminho real pela cidade, do depósito ao endereço do cliente. Definindo um valor para cada arco igual ao comprimento do trecho correspondente de cada rua, então o valor do caminho é a soma dos valores dos arcos no caminho. O menor caminho a percorrer equivale, então, a encontrar no grafo o caminho de menor valor entre o nó do depósito ao nó do endereço do cliente. ■

**Exemplo 4.11** Suponha que um vendedor de material hospitalar da cidade de São Paulo pretenda viajar a Fortaleza para visitar um cliente importante. Ele deve ir de carro, utilizar as estradas principais e visitar outros clientes ao longo do caminho. O vendedor tem uma estimativa média da comissão que deve ganhar de cada um de seus clientes no caminho e deve atingir seu objetivo. Que rota ele deve escolher entre São Paulo e Fortaleza? Construa um grafo cujos nós correspondem às junções das estradas principais e os arcos correspondem às estradas que unem os respectivos nós. Faça o “comprimento” de cada arco no grafo igual ao custo líquido esperado naquele trecho de estrada correspondente. O custo líquido esperado é igual ao custo das despesas rodoviárias (supostamente conhecidas: gasolina, pedágios, manutenção etc.), menos a comissão esperada naquele trecho de estrada. O vendedor deve escolher a rota correspondente ao menor caminho neste grafo do nó que representa a junção da estrada em São Paulo ao nó que representa a junção da estrada em Fortaleza. Os valores de alguns dos arcos podem ser negativos, indicando que o vendedor espera ter lucros ao passar naqueles trechos. ■

**Exemplo 4.12** Considere uma empresa que produz determinado produto em lotes. A produção é feita em lotes pois há um custo fixo de preparação cada vez que se inicia a produção, independen-

<sup>4</sup> Ver, por exemplo, Ahuja *et al.* (1993) para uma discussão de diversas aplicações deste problema.

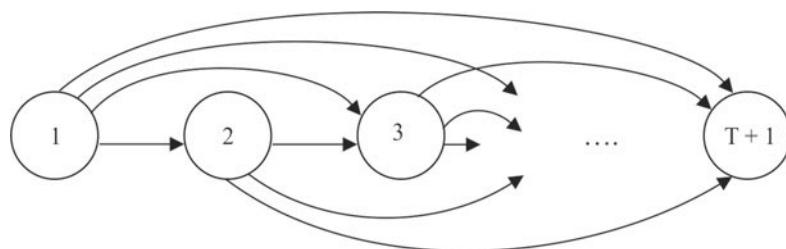
temente da quantidade a ser produzida. Admita que se pretende determinar o plano de fabricação desse produto nos próximos  $T$  períodos e que o custo unitário de produção nesses  $T$  períodos é constante. Considere, também, que há um custo para cada item do produto estocado de um período para outro. As demandas do produto em cada um dos  $T$  períodos são conhecidas e o estoque inicial do produto é nulo. Qual deve ser o plano de produção de custo mínimo desse produto?

À primeira vista, não parece ser um problema de otimização em redes. É necessária uma análise preliminar do problema para que possamos compreender que ele pode ser transformado em um problema em grafos. Observe que, como o custo de produção é igual em todos os períodos, então, se houver produção em algum período, não deve haver nada em estoque.<sup>5</sup> Isso porque sairia mais barato deixar de produzir esse excesso em algum período anterior e produzi-lo no período, uma vez que se reduziria o custo de estocagem e o custo de produção permaneceria no mesmo valor.

Tendo em vista essa observação, a produção em um determinado período  $i$ , se houver produção neste período, deverá ser igual a algumas destas possibilidades: a própria demanda do período  $i$ , ou a demanda do período  $i$  e a do período  $i + 1$  (neste caso, não teríamos produção no período  $i + 1$ ), ou a demanda dos períodos  $i, i + 1, i + 2$  (neste caso, não teríamos produção nos períodos  $i + 1, i + 2$ ), ..., ou a demanda dos períodos  $i, i + 1, i + 2, \dots, T$  (neste caso, não teríamos produção nos períodos  $i + 1, i + 2, \dots, T$ ). Correspondentemente a cada uma dessas possibilidades, incorre-se em um determinado custo dado pela soma dos custos de produção e de estocagem, dependendo de quanto for produzido.

Com as observações anteriores, é possível construir um grafo com  $T + 1$  nós (ver Figura 4.7), um para cada período e mais um nó  $T + 1$ , em que os arcos  $(i, j)$  correspondem à produção no período  $i$  de um lote de tamanho igual às demandas correspondentes aos períodos  $i, i + 1, i + 2, \dots, j - 1$ . Cada um desses arcos tem um custo associado, que é o custo de produção dessa quantidade, mais o custo de estocagem nos períodos  $i + 1, i + 2, \dots, j - 2$ . O plano de produção é dado pelo caminho mais curto do nó 1 ao nó  $T + 1$  neste grafo.<sup>6</sup>

Ilustremos a obtenção do grafo para este problema utilizando um exemplo numérico. Suponha que temos quatro períodos, o estoque inicial do produto é 0, o custo de produção é constante durante esses quatro períodos e é igual a \$2 por tonelada produzida e incorre-se em um custo fixo de preparação de \$10 cada vez que o produto é fabricado. O custo de estocagem de um mês para outro é de \$0,2 por unidade de produto armazenada. As demandas de cada período são conforme indicado na Tabela 4.1.



**Figura 4.7** Problema de planejamento da produção em lote como um problema de caminho mínimo.

<sup>5</sup> Também é válida para custos variantes no tempo; ver Wagner e Whitin (1958).

<sup>6</sup> Esta estratégia é usada para resolver vários problemas de produção; ver, por exemplo, Wagner e Whitin (1958) e Gramani e França (2006).

**Tabela 4.1**  
Demanda em cada período (em toneladas).

Período	1	2	3	4
Demandas (toneladas)	30	20	60	10

Com esses dados, o grafo correspondente tem cinco nós e os custos associados aos arcos  $(i, j)$  são calculados conforme explicação anterior. O arco  $(i, j), j = i + 1, i + 2, \dots, 5$ , corresponde à produção no período  $i$  ( $i = 1, 2, \dots, 4$ ) de um lote de tamanho igual às demandas correspondentes aos períodos  $i, i + 1, i + 2, \dots, j - 1$ , e tem um custo associado  $c_{ij}$  igual ao custo fixo de produção, mais o custo da produção dessa quantidade, mais o custo de estocagem nos períodos  $i, i + 1, i + 2, \dots, j - 2$ , ou seja,

$$c_{ij} = \$10 + \$2(d_i + d_{i+1} + \dots + d_{j-1}) + \$0,2(d_{i+1} + \dots + d_{j-1}) + \$0,2(d_{i+2} + \dots + d_{j-1}) + \dots + \$0,2(d_{j-1})$$

em que  $d_i$  é a demanda do produto no período  $i$ . Assim, para  $i = 1$ , temos

$$c_{12} = \$10 + \$2(30) + \$0,2(0) = \$70$$

$$c_{13} = \$10 + \$2(50) + \$0,2(20) = \$114$$

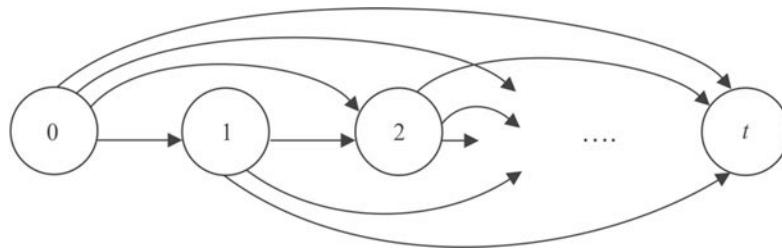
$$c_{14} = \$10 + \$2(110) + \$0,2(80) + \$0,2(60) = \$258$$

$$c_{15} = \$10 + \$2(120) + \$0,2(90) + \$0,2(70) + \$0,2(10) = \$284$$

De forma similar, calculamos  $c_{23} = \$50$ ,  $c_{24} = \$182$ ,  $c_{25} = \$206$ ,  $c_{34} = \$130$ ,  $c_{35} = \$152$ ,  $c_{45} = \$20$ . Com isso, temos determinado o grafo em que cada caminho que liga o nó 1 ao nó 5 representa um plano de produção e o menor caminho (isto é, o menos custoso) é o plano desejado. ■

**Exemplo 4.13** Considere uma empresa que tem uma linha de produção em lotes que consiste em  $t$  estágios ordenados de produção (estágio 1, estágio 2, ..., estágio  $t$ ). A fabricação de cada lote do produto inicia no estágio 1 e, sucessivamente, passa em cada um dos estágios até chegar ao  $t$ -ésimo estágio, quando o produto final é obtido. Em cada estágio, diversas operações/montagens são realizadas e, caso desejado, pode ser feita uma inspeção do produto semi-acabado naquele estágio. No último estágio, o produto acabado sempre é testado. As inspeções servem para detectar defeitos no produto. A probabilidade de se produzir um defeito é conhecida para cada estágio e produtos defeituosos são imediatamente descartados, uma vez que não é possível recuperá-los. Assim, se a inspeção em um estágio não for feita e o produto for defeituoso, as operações/montagens realizadas nos estágios subsequentes desses produtos defeituosos são perdidas e constituem despesas adicionais que poderiam ter sido evitadas caso a inspeção fosse feita. No entanto, a inspeção em cada estágio tem custos significativos. Assim, deseja-se saber em quais estágios as inspeções devem ser feitas para se incorrer no menor custo total de produção.

Novamente, o Exemplo 4.13 não parece ser um problema de otimização em redes. Mas pode ser transformado em um problema de fluxo em redes. Para este problema, pode-se construir um grafo com  $t + 1$  nós (ver Figura 4.8), um para cada estágio, e um nó 0 indicando o início da produção, antes de passar pelo estágio 1. Neste grafo, os arcos  $(i, j)$  correspondem a intervalos em que nenhuma inspeção é feita ou, de forma equivalente, aos estágios de inspeções sucessivas, ou seja, uma inspeção é feita no estágio  $i$  e a próxima inspeção é feita no estágio  $j$  e nenhuma inspeção é feita entre esses dois estágios. Temos arcos ligando cada nó  $i, i = 0, 1, 2, \dots, t - 1$ , a todos os nós  $i + 1, i + 2, \dots, t$ , correspondendo a todas as possibilidades e cada caminho ligando o nó 0 ao nó  $t$  define as inspeções em cada nó (estágio) do caminho.



**Figura 4.8** Problema de produção em estágios transformado em um problema de fluxo.

A cada um desses arcos  $(i, j)$  associamos um custo  $c_{ij}$ , dado pela soma do custo de inspeção do lote no estágio  $j$  e dos custos de processamento do lote nos estágios  $i + 1, i + 2, \dots, j$ . Observe que o tamanho do lote nestes estágios  $i + 1, i + 2, \dots, j$  não se altera, uma vez que nenhuma inspeção é feita, portanto, nenhum produto defeituoso é detectado. Com isso, o custo de produção mais baixo é dado pelo caminho mais curto que vai do nó 0 ao nó  $t$  nesse grafo.

Admitindo-se que o custo de inspecionar os itens existentes no estágio  $j$  e o custo de processamento dos itens em cada um dos estágios  $i + 1, i + 2, \dots, j$  sejam proporcionais ao número de itens processados nesses estágios, precisamos determinar qual é o tamanho do lote nesses estágios para determinar esses custos.

Seja  $p_i$  a probabilidade de produção de um defeito no estágio  $i$ . Assim, se  $L$  é o tamanho do lote inicial, após o estágio 1 temos, em média,  $L p_1$  produtos semi-acabados defeituosos, portanto, restariam  $L(1 - p_1)$  produtos sem defeito. Estes  $L(1 - p_1)$  produtos semi-acabados sem defeito passam agora pelo estágio 2, em que, em média,  $L(1 - p_1)p_2$  deles resultam defeituosos, portanto, restam  $L(1 - p_1)(1 - p_2)$  produtos sem defeito após o estágio 2. Computando-se de forma similar, para os demais estágios sucessivos, temos que, no estágio  $k$ , em média,  $L(1 - p_1)(1 - p_2) \dots (1 - p_{k-1})(1 - p_k)$  produtos resultam sem defeito. Assim, para o cálculo de  $c_{ij}$ , o tamanho do lote processado nos estágios  $i + 1, i + 2, \dots, j$  é igual ao número de produtos sem defeito após o estágio  $i$ , ou seja,  $L(1 - p_1)(1 - p_2) \dots (1 - p_{i-1})(1 - p_i)$ , uma vez que nenhuma inspeção é feita entre os estágios  $i$  e  $j$ . Com isso, os custos decorrentes podem ser computados.

Ilustremos a obtenção do grafo para este problema utilizando um exemplo numérico. Suponha que temos  $t = 4$  estágios, o lote inicial do produto é de  $L = 1.000$  unidades. As probabilidades de produção de um defeito e os custos unitários de produção em cada estágio ( $u_i$ ) estão indicados na Tabela 4.2.

**Tabela 4.2**  
Probabilidade de defeito em cada período e custos unitários de produção.

Estágio	1	2	3	4
Probabilidade de produção de um defeito ( $p_i$ )	0,03	0,02	0,06	0,01
Custo unitário de produção ( $u_i$ )	4	3	7	3

O custo de inspeção no estágio  $j$  depende de quando foi feita a última inspeção (quanto mais recente, menor o custo). Portanto, se a última inspeção ocorreu no estágio  $i$ , o custo unitário de inspeção é denotado por  $h_{ij}$ . Admitimos que, além desse custo variável de inspeção, incorre-se em um custo fixo de  $k_{ij}$  para se realizar essa inspeção no estágio  $j$ . As Tabelas 4.3 e 4.4 fornecem os valores desses custos.

Com esses dados, o grafo correspondente teria cinco nós e os custos associados aos arcos  $(i, j)$  seriam calculados conforme explicação anterior. O arco  $(i, j)$ ,  $j = i + 1, i + 2, \dots, 4$ , corresponde à inspeção do lote no estágio  $j$  ( $j = 1, 2, 3, 4$ ), nenhuma inspeção deste lote ocorreu nos estágios  $i + 1, i + 2, \dots, j - 1$ , e a última inspeção ocorreu no estágio  $i$ . Então,  $c_{ij}$  (soma dos custos de inspeção do lote no estágio  $j$  e dos custos de processamento do lote nos estágios  $i + 1, i + 2, \dots, j$ ) é dado por

$$\begin{aligned} c_{ij} &= (k_{ij} + h_{ij})L(1 - p_1)(1 - p_2) \dots (1 - p_{i-1})(1 - p_i) + \\ &\quad L(1 - p_1)(1 - p_2) \dots (1 - p_{i-1})(1 - p_i)(u_{i+1} + u_{i+2} + \dots + u_j) = \\ &= (k_{ij} + h_{ij})1.000(1 - 0,03) \dots (1 - p_{i-1})(1 - p_i) + \\ &\quad 1.000(1 - 0,03) \dots (1 - p_{i-1})(1 - p_i)(u_{i+1} + u_{i+2} + \dots + u_j) \end{aligned}$$

para  $i = 1, 2, 3$  e  $j = i + 1, \dots, 4$ . Para  $i = 0$  e  $j = 1, \dots, 4$ , temos

$$\begin{aligned} c_{0j} &= (k_{0j} + h_{0j})L + L(u_1 + u_2 + \dots + u_j) = \\ &= (k_{0j} + h_{0j})1.000 + 1.000(u_1 + u_2 + \dots + u_j) \end{aligned}$$

**Tabela 4.3**  
Custos de inspeção conforme períodos.

$h_{ij}$	0	1	2	3	4
0	-	0,3	0,6	1,1	1,3
1	-	-	0,3	0,9	1,3
2	-	-	-	0,6	1,0
3	-	-	-	-	0,3

**Tabela 4.4**  
Custos fixos de inspeção conforme períodos.

$k_{ij}$	0	1	2	3	4
0	-	100	200	310	390
1	-	-	110	210	300
2	-	-	-	90	190
3	-	-	-	-	120

Desta forma,

$$c_{01} = (k_{01} + h_{01} \cdot 1.000) + 1.000(u_1) = (100 + \$0,3 \cdot 1.000) + 1.000 (\$4) = \$4.400$$

$$c_{02} = (k_{02} + h_{02} \cdot 1.000) + 1.000 (u_1 + u_2) = (200 + \$0,6 \cdot 1.000) + 1.000 (\$4 + \$3) = \$ 7.800$$

e, de maneira similar, obtemos  $c_{03} = \$ 15.410$ ,  $c_{04} = \$ 18.690$ .

Para  $i = 1$ , temos:

$$\begin{aligned} c_{12} &= (k_{12} + h_{12} \cdot 1.000(1 - 0,03)) + 1.000(1 - 0,03)(u_2) = (110 + \$0,3 \cdot 970) \\ &\quad + 970 \cdot \$3 = \$3.311 \end{aligned}$$

$$\begin{aligned} c_{13} &= (k_{13} + h_{13} \cdot 1.000(1 - 0,03)) + 1.000(1 - 0,03)(u_2 + u_3) = (210 + \$0,9 \cdot 970) \\ &\quad + 970 \cdot (\$3 + \$7) = \$10.783 \end{aligned}$$

e, de maneira similar, obtemos  $c_{14} = \$ 14.171$ . Prosseguindo com os cálculos para os demais valores de  $i$ , obtemos  $c_{23} = \$7.314,56$ ,  $c_{24} = \$10.646,6$ ,  $c_{34} = \$3.068,7612$ . Com isso, temos determinado o grafo em que cada caminho que liga o nó 1 ao nó 5 representa um plano de inspeção e o menor caminho (isto é, o menos custoso) é o plano desejado. ■

**Exemplo 4.14** Um viajante deve decidir que itens levar para sua viagem à Europa. Ele já fez a mala que vai despachar no avião, na qual colocou todos os itens que julga indispensáveis, mas ainda restaram outros  $n$  itens que gostaria de levar, cada um deles com um certo valor de utilidade. O problema é que a companhia aérea permite apenas uma bagagem de mão com um limite de no máximo  $U$  quilos. O viajante pretende levar uma mochila como bagagem de mão e seu problema agora é decidir que itens deve levar nessa mochila de modo a maximizar o valor de utilidade total e não violar a restrição de peso da companhia aérea. Apenas uma unidade de cada item deve ser incluída na mochila (ver problema da mochila 0-1, Seção 3.4.1 – Capítulo 3), embora o viajante possa considerar a duplicação de um item como um novo item, o qual pode ter valor de utilidade igual ou menor.

À primeira vista, este exemplo também não parece ser um problema de otimização em redes. Como nos dois exemplos anteriores, é necessário um pouco de experiência na resolução de problemas de otimização e de problemas de rede para transformar este problema da mochila em um problema em grafos.

Admite-se que todos os pesos dos itens e o limite máximo  $U$  sejam inteiros. Pode-se construir um grafo com  $U + 1$  nós, um para cada unidade de capacidade preenchida da mochila e mais um nó 0 que representa a mochila vazia. A inclusão de um item na mochila é representada com um arco que une dois nós. Mais precisamente, um item de peso  $w_i$  incluído na mochila que já tem o peso acumulado  $r$  de outras inclusões é representado por um arco  $(r, s)$ , em que  $s = r + w_i$  e contribui com o valor de utilidade  $u_i$  para o viajante. Entretanto, este arco só será considerado se  $r + w_i \leq U$ , pois  $U - r$  é o espaço restante a ser preenchido.

Para a construção dos arcos do grafo, considere os itens  $i = 1, 2, \dots, n$ , um item de cada vez de forma ordenada. Iniciamos sem nenhum arco no grafo, só os  $U + 1$  nós. Para cada item  $i$  em consideração, com exceção do nó 0, consideramos arcos apenas que emergem de nós que já tenham algum arco incidente nele (isto é, que sejam nó cabeça de algum arco já existente) da iteração anterior. Isso quer dizer que adicionamos a algum preenchimento já realizado da mochila (com os itens 1, 2, ...,  $i - 1$ ), mais aquele item  $i$ . Lembre-se de que apenas uma unidade de cada item pode ser incluída na mochila. Do nó 0, sempre emergem arcos para qualquer item (equivalente a iniciar o preenchimento da mochila com o item  $i$ , sem colocar nenhum dos itens anteriores). Portanto, para  $i = 1$ , apenas um arco será definido com cauda no nó 0 e o nó cabeça em  $(0 + w_1)$  e utilidade  $u_1$ . Ao considerar o item  $i$  e um nó  $r$  (na ordem,  $r = 0, 1, \dots, U$ ) que já tenha algum arco incidente nele (isto é, algum preenchimento com itens considerados anteriormente), então

é definido um arco com cauda em  $r$  e com nó cabeça em  $(r + w_i)$ , se  $(r + w_i) \leq U$  (ou seja, se este item ainda não ultrapassa o limite de peso da mochila) e utilidade  $u_i$ . Se  $(r + w_i) > U$ , então o arco não é considerado. Um nó  $r$  em que não incide nenhum arco significa que não houve nenhum preenchimento da mochila com combinação de itens  $1, 2, \dots, i - 1$ , que soma  $r$ . Portanto, nenhum arco precisa ser definido a partir deste nó.

Após o  $n$ -ésimo item ser considerado, todos os nós do grafo que são cabeça de algum arco representam preenchimentos possíveis da mochila. Admitindo-se que as utilidades dos itens sejam todas não-negativas, sempre vale a pena colocar mais um item, se isso for possível. Assim, os enxaimentes de interesse são apenas aqueles correspondentes aos nós do grafo que são cabeça de algum arco e não têm nenhum nó sucessor. Isso porque um nó que não seja nó cabeça corresponde a um preenchimento possível da mochila que ainda pode ser melhorado com a inclusão de mais um item. A todo nó que seja cabeça de algum arco e não tenha sucessor, um arco *artificial* ligando a cabeça ao nó  $U$  com valor de utilidade zero é definido (representa um espaço vazio na mochila). O problema da mochila do viajante pode ser resolvido determinando-se o *caminho de maior utilidade* (ou caminho máximo, conforme Seção 4.2.2, adiante) entre o nó 0 e o nó  $U$  no grafo, obedecendo a algumas restrições adicionais.

Ilustremos a construção do grafo para este problema utilizando um exemplo numérico. Suponha que temos quatro itens que o viajante gostaria de levar, conforme indica a Tabela 4.5.

Suponha que o limite da bagagem seja de 6 quilos. Com isso, o grafo correspondente seria construído, passo a passo, conforme a Figura 4.9 (a), (b), (c), (d), (e) e (f). A notação  $(i, u_i)$  em cada arco da figura significa item  $i$  com utilidade  $u_i$ .

Este problema da mochila pode ser resolvido achando-se o caminho mais longo (caminho máximo) nesse grafo, do nó 0 ao nó 6, obedecendo a algumas restrições. Observe que nesse grafo existem arcos paralelos, ou seja, arcos adjacentes ao mesmo par de nós e, também, caminhos alternativos do nó 0 aos demais nós do grafo indicando que existem preenchimentos diferentes da mochila com o mesmo peso total. No caso de arcos paralelos, como o interesse é determinar o caminho máximo, o arco mais valioso adjacente ao par de nós será o utilizado caso algum arco que une esse par de nós faça parte do caminho.

Com o grafo construído é possível que o caminho mais longo do nó 0 ao nó 6 não seja uma solução factível para o problema da mochila pois pode ocorrer de essa solução utilizar mais de uma unidade de um determinado item. Por exemplo, se o limite da bagagem fosse de 4 quilos, o caminho mais longo do nó 0 ao nó 4 (ver grafo (f) da Figura 4.9) tem valor 32 e é obtido com os arcos (0,2) e (2,4) com valores utilidades de 16 cada, correspondentes ao item 3. Essa solução não é factível pois o viajante utiliza duas unidades do item 3 e dispõe-se apenas de uma unidade para incluir na mochila.

**Tabela 4.5**  
**Pesos dos itens e utilidades correspondentes.**

Item	1	2	3	4
Peso	3	2	2	4
Utilidade	25	15	16	30

Para achar a solução da mochila utilizando o grafo construído precisamos achar o caminho máximo do nó 0 ao nó 6 que utiliza no máximo uma unidade de cada item. Uma maneira de obter essa solução é enumerar, em ordem não crescente de utilidade, os caminhos máximos do nó 0 ao nó 6 no grafo (f) da Figura 4.9 até que algum deles satisfaça às condições desejadas.

É possível utilizar, entretanto, um outro grafo, similar ao construído anteriormente, mas com um número maior de nós, em que a solução do caminho máximo do nó 0 ao nó 6 nos fornece diretamente uma solução para o problema da mochila. Esse grafo é construído seguindo-se os mesmos passos de construção do grafo anterior, diferindo apenas na fase de inclusão dos arcos. Considere a inclusão de um arco correspondente a um item de peso  $w_i$ , a partir de um nó  $r$  que é nó cabeça de algum arco incidente nele, incluso em uma iteração anterior. A inclusão do arco  $(r; s)$ , em que  $s = r + w_i$ ,  $s < U$ , (e contribui com o valor de utilidade  $u_i$ ) é feita somente se o nó  $s$  não for nó cabeça de nenhum arco já incluído em iterações anteriores. Caso o nó  $s$  já seja um nó cabeça de algum arco, um novo nó é introduzido no grafo que é uma cópia de nó  $s$ . Denotamos essa cópia do nó  $s$  por  $s^i$  em que  $i = 1, 2, \dots$ , indica o número da cópia introduzida, e inclui-se o arco  $(r, s^i)$ .

Procedendo-se dessa forma, obtém-se um grafo sem arcos paralelos em que a solução do problema da mochila é obtida diretamente pelo caminho mais longo do nó 0 ao nó 6 ou uma de suas cópias. O grafo resultante correspondente ao exemplo numérico da mochila é o da Figura 4.10.

Transformamos o problema da mochila em um problema de caminho máximo no grafo. Esse grafo não tem circuitos, portanto pode ser facilmente transformado em um problema de caminho mais curto. Para isso, basta definir todos os custos no novo problema como o negativo das utilidades das arcas. Alguns algoritmos para resolução do caminho mínimo em um grafo que apresentamos mais adiante, por exemplo o algoritmo de Dijkstra, admitem que todos os comprimentos dos arcos são não-negativos. Portanto, eles não podem ser utilizados para resolver essa classe de problemas.

A transformação do problema da mochila em um problema de caminho mais curto em um grafo pode dar a impressão de que o problema da mochila é “fácil” de ser resolvido, visto que o problema de caminho mais curto, como vemos adiante, pode ser resolvido com o uso de um algoritmo relativamente simples. O esforço computacional desse algoritmo depende do tamanho do grafo (o tamanho de um grafo pode ser medido pelo número de nós e arcos), isto é, quanto maior o grafo, mais computações terão de ser feitas para encontrar uma solução para o problema. Isso quer dizer que podemos incorrer em tempos computacionais excessivamente longos na resolução de problemas da mochila com capacidade máxima  $U$  grandes, uma vez que o grafo obtido da transformação tem  $U + 1$  nós. ■

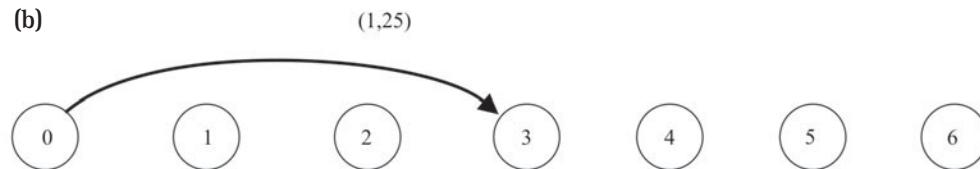
#### *Formulação matemática do problema de caminho mais curto entre dois nós*

Para escrever um modelo matemático para o problema de caminho mais curto do nó 1 ao nó  $n$  de um grafo  $G = (N, E)$ ,  $N = \{1, 2, \dots, n\}$ , observamos que este problema é um caso especial do problema de transbordo (Seção 2.2 – Capítulo 2), em que se quer transportar, ao menor custo, uma unidade de um produto produzido no nó 1 ao nó  $n$ . Os demais nós do grafo são todos de transbordo. Define-se o comprimento de cada arco como o custo no problema de transbordo e, com isso, temos a seguinte formulação para o problema de caminho mais curto. Sem perda de generalidade, consideramos o nó origem como o nó 1 e o nó destino o nó  $n$ , mas a formulação e os algoritmos que se seguem podem ser estendidos para quaisquer dois nós da rede.

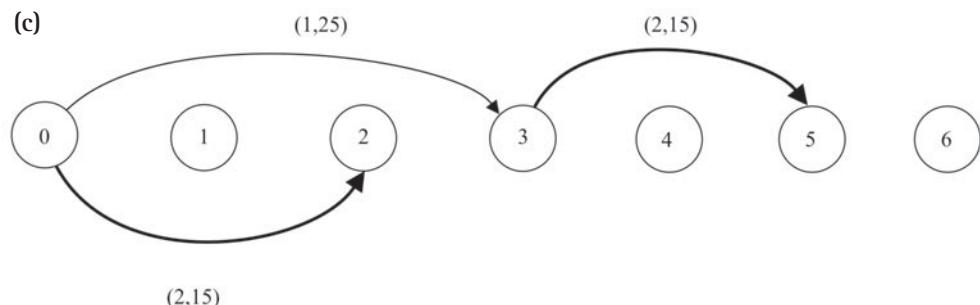
(a)



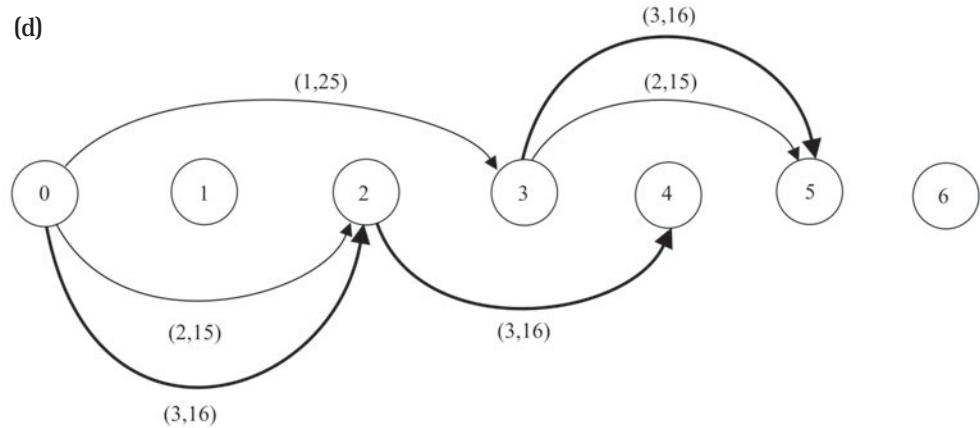
(b)



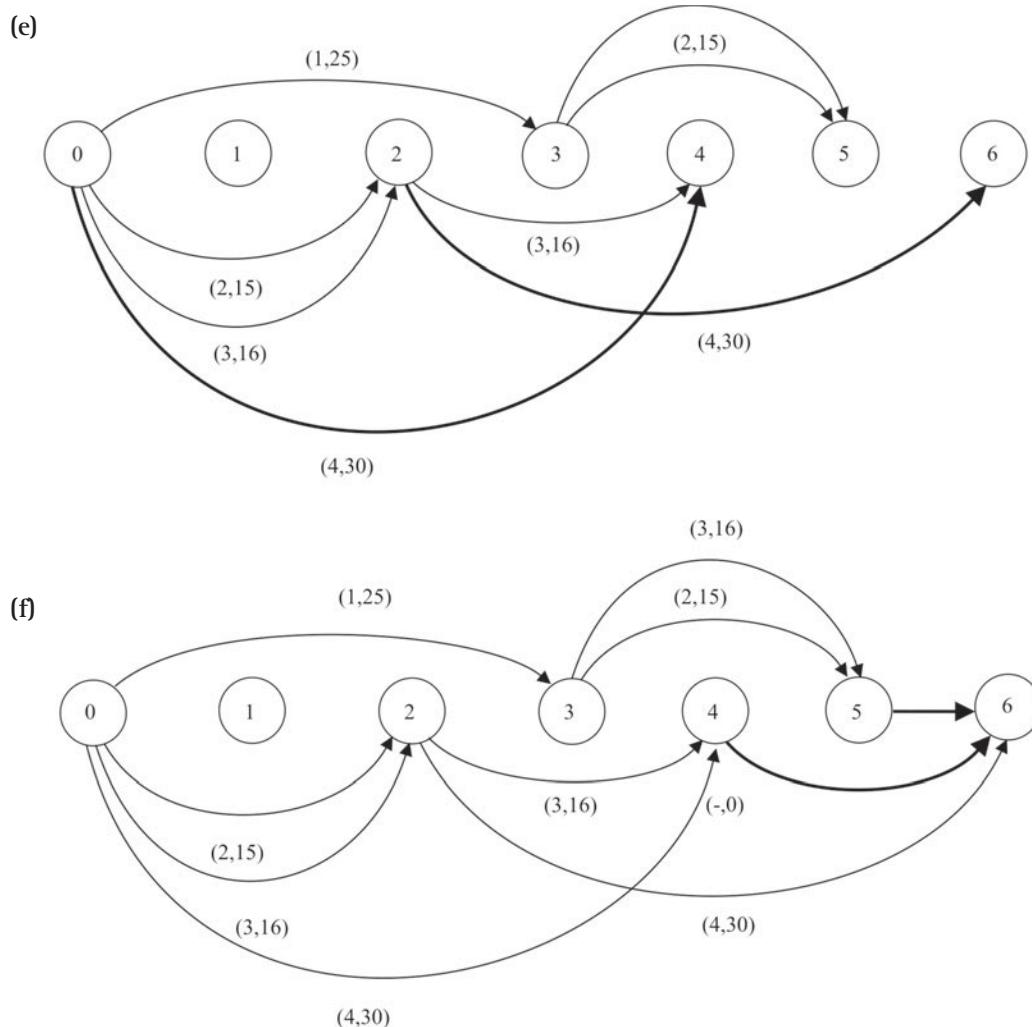
(c)



(d)



**Figura 4.9** (a) Grafo inicial; (b) Arcos devido à inserção do item 1; (c) Arcos devido à inserção do item 2; (d) Arcos devido à inserção do item 3; (e) Arcos devido à inserção do item 4; (f) Introdução de arcos artificiais para simplificação do problema.



**Figura 4.9 cont.** (a) Grafo inicial; (b) Arcos devido à inserção do item 1; (c) Arcos devido à inserção do item 2; (d) Arcos devido à inserção do item 3; (e) Arcos devido à inserção do item 4; (f) Introdução de arcos artificiais para simplificação do problema.

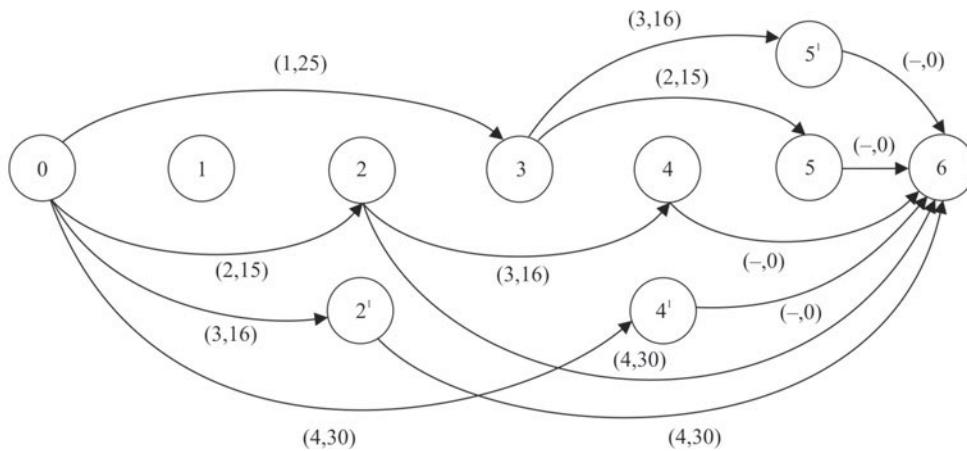
$$\text{Minimizar } f(\mathbf{x}) = \sum_{i=1}^n \sum_{j \in S(i)} c_{ij} x_{ij}$$

$$\sum_{j \in S(1)} x_{1j} = 1$$

$$\sum_{i \in P(n)} x_{in} = 1$$

$$\sum_{i \in P(j)} x_{ij} = \sum_{k \in S(j)} x_{jk}, \quad j = 2, \dots, n-1$$

$$x_{ij} \geq 0, \quad i = 1, \dots, n \text{ e } j = 1, \dots, n$$



**Figura 4.10** Grafo resultante para o problema da mochila.

em que

$S(j)$  é o conjunto dos nós sucessores de  $j$

$P(j)$  é o conjunto dos nós predecessores de  $j$

$x_{ij}$  é a quantidade transportada do produto da origem  $i$  para o destino  $j$  utilizando o arco  $(i, j)$

$c_{ij}$  é o “custo” incorrido por usar o arco  $(i, j)$

No modelo anterior, para sermos rigorosos, deveríamos impor que as quantidades transportadas nos diversos arcos fossem inteiras, para que a solução do problema realmente indicasse um caminho, ou seja,  $x_{ij} = 1$  se e somente o arco  $(i, j)$  pertencesse ao caminho. Entretanto, essa restrição não precisa ser explicitada, pois, se o modelo anterior for resolvido pelo método simplex (Capítulo 2), a solução será inteira (os valores das variáveis serão 0 ou 1) devido à estrutura particular desse modelo, como vemos mais adiante.

Para a resolução do problema de caminho mais curto, existem algoritmos alternativos mais simples e mais eficientes (tempo de execução menor no computador para se encontrar uma solução) que o método simplex e são descritos a seguir.

#### *O algoritmo de Dijkstra para determinação do caminho mínimo entre dois nós no grafo*

O algoritmo de Dijkstra encontra o menor caminho entre quaisquer dois nós da rede, quando todos os arcos têm comprimentos não-negativos. O algoritmo de Dijkstra utiliza um procedimento iterativo, determinando, na iteração 1, o nó mais próximo do nó 1, na segunda iteração, o segundo nó mais próximo do nó 1, e assim sucessivamente, até que em alguma iteração o nó  $n$  seja atingido.

Suponha que estamos em alguma iteração  $t$  (isto é, já determinamos o nó mais próximo do nó 1, o segundo nó mais próximo, ..., o  $(t - 1)$ -ésimo nó mais próximo). Como determinar o  $t$ -ésimo nó mais próximo do nó 1? O menor caminho do nó 1 a esse  $t$ -ésimo nó mais próximo, digamos, o nó  $k$  (o qual é desconhecido) deve ser um caminho que passa por algum dos  $t - 1$  nós mais próximos de 1, ou um caminho direto do nó 1 ao nó  $k$  (naturalmente, se houver o arco  $(1, k)$ ). Observe que não pode ser um caminho que passe por algum outro nó que não os  $t - 1$  nós mais

próximos pois, caso contrário, o nó  $k$  não seria o  $t$ -ésimo nó mais próximo de 1, pois o nó imediatamente anterior no caminho do nó  $k$  estaria mais próximo (ou a uma mesma distância) do nó 1, uma vez que todos os arcos têm comprimentos não-negativos. A Figura 4.11 ilustra essa argumentação.

O algoritmo de menor caminho de Dijkstra é descrito a seguir. Separaremos os nós em rotulados (conjunto  $R$ ) e não-rotulados (conjunto  $NR$ ). Os nós rotulados são aqueles cuja ordenação já foi definida, ou seja, o nó mais próximo, o segundo nó mais próximo etc. Para recuperar o caminho mais curto até um determinado nó  $k$ , guardaremos o nó anterior ao nó  $k$  no caminho, que denotamos por  $p(k)$ , isto é, o caminho de 1 ao nó  $k$  é constituído do caminho de 1 ao nó  $p(k)$  e o arco  $(p(k), k)$ . Se  $p(k) = 1$ , então o menor caminho que liga o nó 1 ao nó  $k$  é constituído tão-somente do arco  $(1, k)$ .

### Algoritmo de Dijkstra

*Dados:*

$G(N, E)$ : grafo em que  $N = \{1, 2, \dots, n\}$

$1$  : nó inicial do caminho

$n$  : nó final do caminho

$c(i, j)$  : comprimento do arco  $(i, j) \in E$  (hipótese:  $c(i, j) \geq 0$ )

*Saída:*

$d(n)$  : menor distância do nó 1 ao nó  $n$

$C$  : caminho mínimo entre o nó 1 e o nó  $n$

*Passo 1: Início*

$R = \{1\}$  : inicialmente o nó 1 é rotulado

$NR = \{2, \dots, n\}$  : os demais nós não são rotulados

$d(1) = 0$  : a distância do nó 1 ao nó 1 é zero

$p(1) = 0$  : o nó 1 é o nó inicial

Para  $i \in NR$ ,

$d(i) = +\infty$  : a distância do nó 1 aos nós não rotulados é  $+\infty$

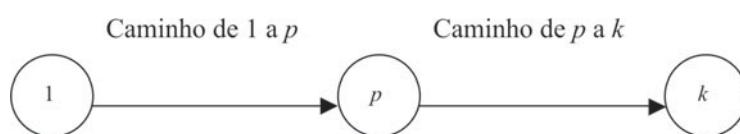
$p(i) = n + 1$  : o nó  $i$  não tem predecessor (observe que não existe nó  $n + 1$  no grafo)

$a = 1$  : último nó incluído em  $R$

*Passo 2: Para todo  $i \in NR$ , determine  $d(i) = \min\{d(i), d(a) + c(a, i)\}$  e faça  $p(i) = a$ , caso  $d(i) = d(a) + c(a, i)$ .*

Se  $d(i) = +\infty$  para todo  $i \in NR$ , então pare {não existe caminho de 1 a qualquer um dos nós em  $NR$ }.

Se não, determine  $k \in NR$  tal que  $d(k) = \min\{d(i), i \in NR\}$ . Exclua o nó  $k$  de  $NR$  (isto é,  $NR \leftarrow NR - \{k\}$ ) e inclua-o em  $R$  (isto é,  $R \leftarrow R \cup \{k\}$ ) e faça  $a = k$ .



**Figura 4.11** O nó  $p$  está necessariamente mais próximo de 1 do que o nó  $k$ .

*Passo 3:* Se  $a = n$ , então recupere o caminho mínimo  $C$  a partir dos valores armazenados em  $p(.)$ , iniciando por  $k_1 = p(n)$ , em seguida,  $k_2 = p(k_1)$ , até que o nó 1 seja atingido. Se não (isto é,  $a \neq n$ ), retorne ao *Passo 2*.

**Exemplo 4.15** Ilustramos o algoritmo de Dijkstra para achar o menor caminho do nó 5 ao nó 2 no grafo da Figura 4.12. Os valores respectivos dos comprimentos dos arcos estão indicados ao lado de cada arco.

*Passo 1:*

$$R = \{5\} \text{ (ver Figura 4.13)}$$

$$NR = \{1, 2, 3, 4, 6\}$$

$$d(5) = 0, p(5) = 0 \text{ e } d(i) = +\infty, p(i) = 7, i = 1, 2, 3, 4, 6$$

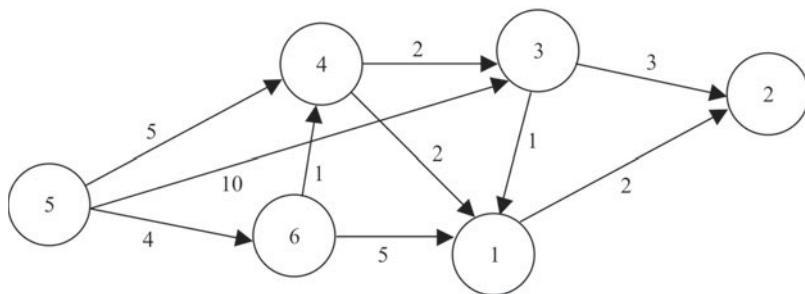
$$a = 5$$

Na Figura 4.13, o nó rotulado no passo 1 é indicado com um ‘ $\odot$ ’ próximo ao nó. O nó 6 é o próximo nó a ser rotulado e está indicado com um ‘ $\oplus$ ’. Esses mesmos símbolos são empregados nas figuras subsequentes para indicar nós rotulados e não rotulados.

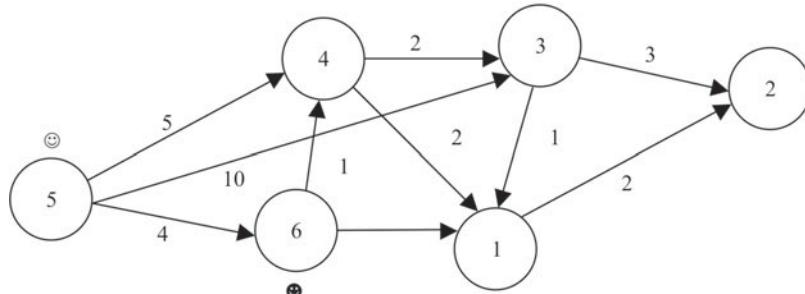
*Passo 2:*

$$d(1) = \min \{d(1), d(5) + c(5, 1)\} = \min \{+\infty, 0 + \infty\} = +\infty$$


---



**Figura 4.12** Encontrar o caminho mínimo do nó 5 ao nó 2.



**Figura 4.13** Nô rotulado: 5.

$$d(2) = \min \{d(2), d(5) + c(5, 2)\} = \min \{+\infty, 0 + \infty\} = +\infty$$

$$d(3) = \min \{d(3), d(5) + c(5, 3)\} = \min \{+\infty, 0 + 10\} = 10 \text{ e } p(3) = 5$$

$$d(4) = \min \{d(4), d(5) + c(5, 4)\} = \min \{+\infty, 0 + 5\} = 5 \text{ e } p(4) = 5$$

$$d(6) = \min \{d(6), d(5) + c(5, 6)\} = \min \{+\infty, 0 + 4\} = 4 \text{ e } p(6) = 5$$

$$k = 6$$

$R = \{5, 6\}$  (ver Figura 4.14)

$NR = \{1, 2, 3, 4\}$

$$a = 6$$

*Passo 3:* O nó 2 ( $n = 2$ , neste exemplo) não foi rotulado, portanto, retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1), d(6) + c(6, 1)\} = \min \{+\infty, 4 + 5\} = 9 \text{ e } p(1) = 6$$

$$d(2) = \min \{d(2), d(6) + c(6, 2)\} = \min \{+\infty, 4 + \infty\} = +\infty$$

$$d(3) = \min \{d(3), d(6) + c(6, 3)\} = \min \{10, 4 + \infty\} = 10$$

$$d(4) = \min \{d(4), d(6) + c(6, 4)\} = \min \{5, 4 + 1\} = 5$$

$$k = 4$$

$R = \{5, 6, 4\}$  (ver Figura 4.15)

$NR = \{1, 2, 3\}$

$$a = 4$$

*Passo 3:* O nó 2 não foi rotulado, portanto, retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1), d(4) + c(4, 1)\} = \min \{9, 5 + 2\} = 7 \text{ e } p(1) = 4$$

$$d(2) = \min \{d(2), d(4) + c(4, 2)\} = \min \{+\infty, 5 + \infty\} = +\infty$$

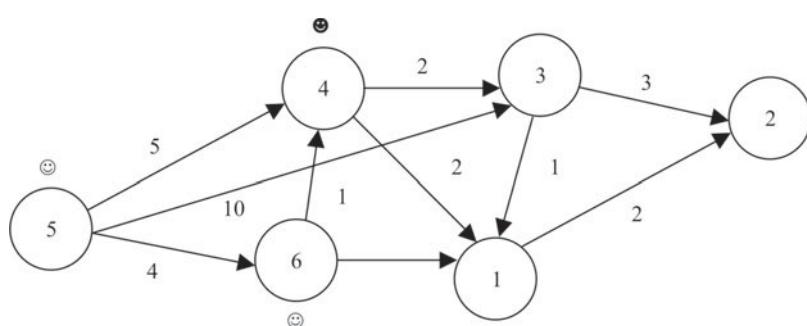
$$d(3) = \min \{d(3), d(4) + c(4, 3)\} = \min \{10, 5 + 2\} = 7 \text{ e } p(3) = 4$$

$$k = 3$$

$R = \{5, 6, 4, 3\}$  (ver Figura 4.16)

$NR = \{1, 2\}$

$$a = 3$$



**Figura 4.14** Nós rotulados: 5 e 6; caminho mínimo do nó 5 ao nó 6.

*Passo 3:* O nó 2 não foi rotulado, portanto, retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1), d(3) + c(3, 1)\} = \min \{7, 7 + 1\} = 7$$

$$d(2) = \min \{d(2), d(3) + c(3, 2)\} = \min \{+\infty, 7 + 3\} = 10 \text{ e } p(2) = 3$$

$$k = 1$$

$$R = \{5, 6, 4, 3, 1\} \text{ (ver Figura 4.17)}$$

$$NR = \{2\}$$

$$a = 1$$

*Passo 3:* O nó 2 não foi rotulado, portanto, retorna-se ao *Passo 2*.

*Passo 2:*

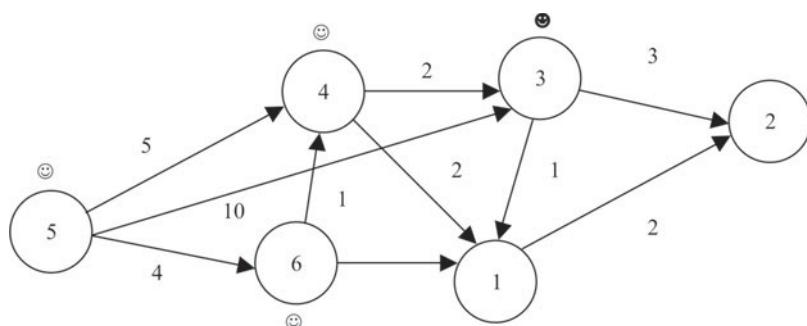
$$d(2) = \min \{d(2), d(1) + c(1, 2)\} = \min \{10, 7 + 2\} = 9, \text{ e } p(2) = 1$$

$$k = 2$$

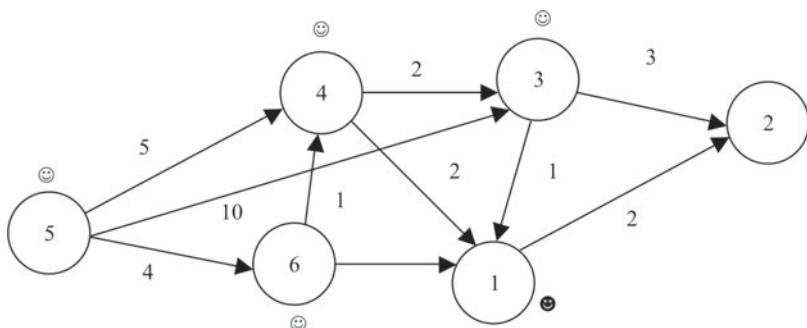
$$R = \{5, 6, 4, 1, 3, 2\} \text{ (ver Figura 4.18)}$$

$$NR = \emptyset$$

$$a = 2$$



**Figura 4.15** Nós rotulados: 5, 6 e 4; caminho mínimo do nó 5 ao nó 4.



**Figura 4.16** Nós rotulados: 5, 6, 4 e 3; caminho mínimo do nó 5 ao nó 3.

*Passo 3:* O nó 2 foi rotulado, portanto, temos:

$$p(2) = 1, p(1) = 4, p(4) = 5, p(5) = 0$$

e o caminho mínimo do nó 5 ao nó 2 é dado por:

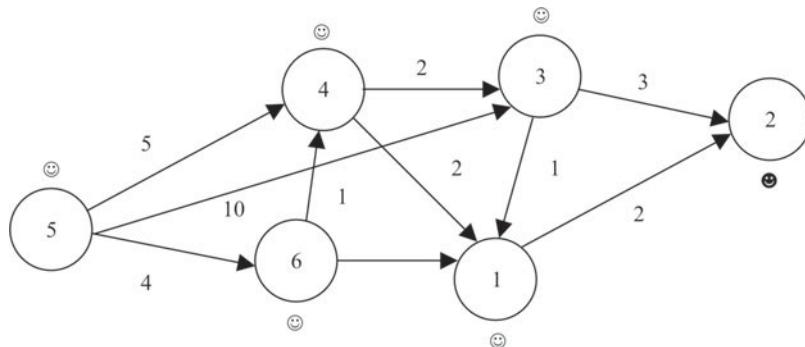
$$C = \{(5, 4), (4, 1), (1, 2)\}$$

pare

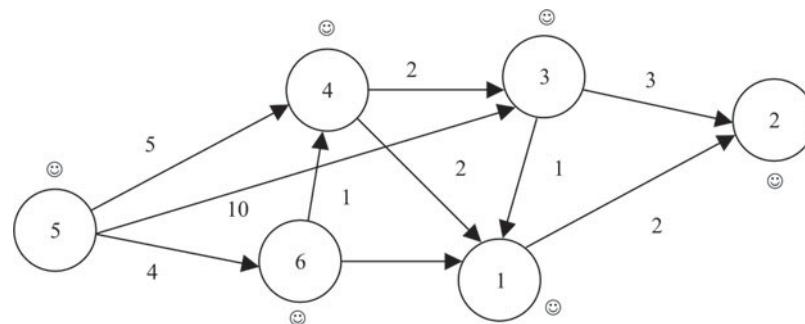
■

No caso de empates, em que existem dois nós não rotulados de mesma distância mínima, a escolha pode ser feita de maneira arbitrária. O nó não escolhido inicialmente deverá ser rotulado na iteração seguinte do *Passo 2*. Na Figura 4.19, apresentamos a árvore que contém todos os caminhos mínimos a partir do nó 5. Tais árvores são denominadas *árvores de caminho mínimo*.

Quando existem arcos com comprimentos negativos, o algoritmo de Dijkstra falha, uma vez que um caminho mais curto pode passar inicialmente por algum nó mais distante e, depois, passar por arcos negativos, de modo que, no total, a distância percorrida é menor. Por exemplo, considere a rede da Figura 4.20, em que os valores respectivos dos comprimentos dos arcos estão indicados ao lado de cada arco. Deseja-se encontrar o caminho mínimo do nó 5 ao nó 2, como antes. Pelo



**Figura 4.17** Nós rotulados: 5, 6, 4, 3 e 1; caminho mínimo do nó 5 ao nó 1.



**Figura 4.18** Nós rotulados: 5, 6, 4, 3, 1 e 2; caminho mínimo do nó 5 ao nó 2.

algoritmo de Dijkstra, o nó 4 seria rotulado com  $d(4) = 5$ , indicando erroneamente que o menor caminho do nó 1 ao nó 4 tem comprimento 5. Note que o menor caminho é ir do nó 1 ao nó 3 (de comprimento 10), para então atingir o nó 4 por um caminho de comprimento total igual a 3.

#### O algoritmo de Ford para determinação do caminho mais curto entre dois nós no grafo

O algoritmo de Ford encontra o caminho mais curto entre dois nós, mesmo que haja arcos com comprimentos negativos. Este algoritmo é uma generalização do algoritmo de Dijkstra, diferindo em alguns dos seus passos.

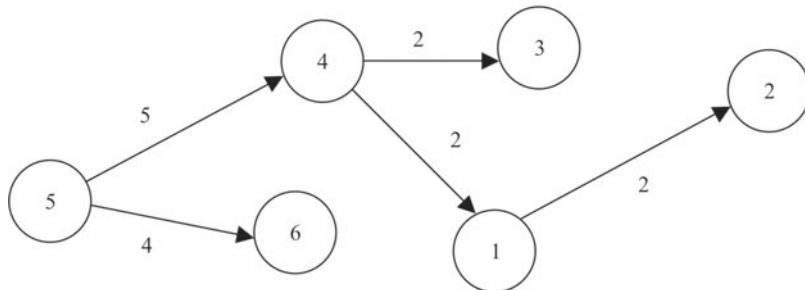
#### Algoritmo de Ford

*Dados:*

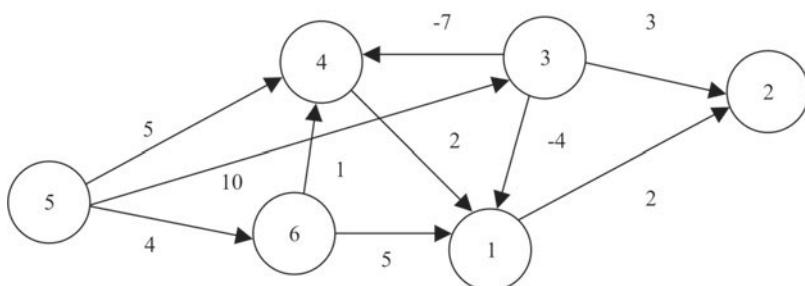
- $G(N, E)$ : grafo em que  $N = \{1, 2, \dots, n\}$
- 1           nó inicial do caminho
- $n$           nó final do caminho
- $c(i, j)$    comprimento do arco  $(i, j)$ ,  $(i, j) \in E$

*Saída:*

- $d(n)$       menor distância do nó 1 ao nó  $n$
- $C$           caminho mais curto entre o nó 1 e o nó  $n$



**Figura 4.19** Árvores que contêm todos os caminhos mínimos a partir do nó 5.



**Figura 4.20** Grafo com arcos com comprimentos negativos.

*Passo 1:* Início

$$R = \{1\}$$

$$NR = \{2, \dots, n\}$$

$$d(1) = 0, p(1) = 0 \text{ e } d(i) = +\infty, p(i) = n + 1, i \in NR$$

$$r(i) = 0, i \in NR, r(1) = 1$$

$$a = 1$$

*sinal* = 1 (*sinal* é uma variável indicadora que é igual a 0 quando todos os nós do grafo são rotulados, e é igual a 1 em caso contrário)

*Passo 2:* Para todo  $v \in N$ ,

Calcule  $d(v) = \min \{d(v), d(a) + c(a, v)\}$  e faça  $p(v) = a$ , caso  $d(v) = d(a) + c(a, v)$ .

Se  $d(v) = +\infty$  para todo  $v \in NR$ , então pare (não existe nenhum caminho de 1 a qualquer um dos nós em  $NR$ ).

Se para algum  $v \in R$ ,  $d(v)$  decresceu de valor, então exclua o nó  $v$  de  $R$  (isto é,  $R \leftarrow R - \{v\}$ ) e inclua-o em  $NR$  (isto é,  $NR \leftarrow NR \cup \{v\}$ ).

Se  $NR \neq \emptyset$ , então:

a. Determine  $k$  tal que  $d(k) = \text{mínimo } \{d(v), v \in NR\}$

b. Exclua o nó  $k$  de  $NR$  (isto é,  $NR \leftarrow NR - \{k\}$ ) e inclua-o em  $R$  (isto é,  $R \leftarrow R \cup \{k\}$ )

c.  $a = k$

d.  $r(a) \leftarrow r(a) + 1$

Se não (isto é,  $NR = \emptyset$ ), faça *sinal* = 0.

*Passo 3:* Se *sinal* = 0, então recupere o caminho mínimo  $C$  a partir dos valores armazenados em  $p(.)$ , iniciando por  $k_1 = p(n)$ , em seguida,  $k_2 = p(k_1)$ , até que o nó 1 seja atingido e pare. Se existe  $r(i)$  maior ou igual a  $n$ , então pare; existe um circuito no grafo com comprimento total negativo. Caso contrário, retorne ao *Passo 2*.

As diferenças com relação ao algoritmo de Dijkstra podem ser observadas no *Passo 2*, em que as distâncias  $d(v)$  são atualizadas para todos os nós, e não somente para os nós não rotulados. Com isso, as distâncias de alguns dos nós já rotulados podem diminuir. Caso isso ocorra, estes nós deixam de ser nós rotulados, passando a ser novamente nós não rotulados. Uma outra diferença está no *Passo 3*. O valor de *sinal* muda para 0 se todos os nós estiverem em  $R$  e não há mais nenhuma mudança nos valores de  $d(v)$  para todo  $v \in N$ . Observe que também foi incluído um contador  $r(i)$  para cada nó  $i$ , que marca quantas vezes o nó já foi rotulado. Isso porque, caso houvesse um circuito no grafo com comprimento total negativo, o algoritmo não terminaria, pois um nó  $i$  pertencente a esse circuito seria rotulado infinitas vezes, dado que sua distância  $d(i)$  sempre diminuiria após um número fixo de passos. Pode-se mostrar que, se não houver circuitos negativos em um grafo com  $n$  nós, nenhum de seus nós será rotulado mais do que  $n - 1$  vezes. Assim, se um nó é rotulado  $n$  vezes (ou mais), então existe um circuito com comprimento total negativo no grafo e o algoritmo é interrompido.

**Exemplo 4.16** Ilustramos o algoritmo de Ford para achar o menor caminho do nó 5 ao nó 2 do grafo da Figura 4.20.

*Passo 1:*

$$R = \{5\}$$

$$NR = \{1, 2, 3, 4, 6\}$$

$$d(5) = 0, p(5) = 0 \text{ e } d(i) = +\infty, p(i) = 7, i = 1, 2, 3, 4, 6$$

$$r(5) = 1, r(i) = 0, i = 1, 2, 3, 4, 6$$

$$a = 5$$

$$sinal = 1$$

*Passo 2:*

$$d(1) = \min \{d(1), d(5) + c(5,1)\} = \min \{+\infty, 0 + \infty\} = +\infty$$

$$d(2) = \min \{d(2), d(5) + c(5,2)\} = \min \{+\infty, 0 + \infty\} = +\infty$$

$$d(3) = \min \{d(3), d(5) + c(5,3)\} = \min \{+\infty, 0 + 10\} = 10, \text{ e } p(3) = 5$$

$$d(4) = \min \{d(4), d(5) + c(5,4)\} = \min \{+\infty, 0 + 5\} = 5, \text{ e } p(4) = 5$$

$$d(5) = \min \{d(5)\} = \min \{(0)\} = 0$$

$$d(6) = \min \{d(6), d(5) + c(5,6)\} = \min \{+\infty, 0 + 4\} = 4, \text{ e } p(6) = 5$$

$$k = 6$$

$$R = \{5, 6\}$$

$$r(6) = 1$$

$$NR = \{1, 2, 3, 4\}$$

$$a = 6$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 6$  para  $i = 1, 2, 3, 4, 5$  e  $6$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1), d(6) + c(6, 1)\} = \min \{+\infty, 4 + 5\} = 9 \text{ e } p(1) = 6$$

$$d(2) = \min \{d(2), d(6) + c(6, 2)\} = \min \{+\infty, 4 + \infty\} = +\infty$$

$$d(3) = \min \{d(3), d(6) + c(6, 3)\} = \min \{10, 4 + \infty\} = 10$$

$$d(4) = \min \{d(4), d(6) + c(6, 4)\} = \min \{5, 4 + 1\} = 5$$

$$d(5) = \min \{d(5), d(6) + c(6, 5)\} = \min \{0, 4 + \infty\} = 0$$

$$d(6) = \min \{d(6)\} = \min \{4\} = 4$$

$$k = 4$$

$$r(4) = 1$$

$$R = \{5, 6, 4\}$$

$$NR = \{1, 2, 3\}$$

$$a = 4$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 6$  para  $i = 1, 2, 3, 4, 5$  e  $6$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1), d(4) + c(4, 1)\} = \min \{9, 5 + 2\} = 7 \text{ e } p(1) = 4$$

$$d(2) = \min \{d(2), d(4) + c(4, 2)\} = \min \{+\infty, 5 + \infty\} = +\infty$$

$$d(3) = \min \{d(3), d(4) + c(4, 3)\} = \min \{10, 5 + \infty\} = 10$$

$$d(4) = \min \{d(4)\} = \min \{5\} = 5$$

$$d(5) = \min \{d(5), d(4) + c(4, 5)\} = \min \{0, 5 + \infty\} = 0$$

$$d(6) = \min \{d(6), d(4) + c(4, 6)\} = \min \{4, 5 + \infty\} = 4$$

$$k = 1$$

$$r(1) = 1$$

$$R = \{5, 6, 4, 1\}$$

$$NR = \{2, 3\}$$

$$a = 1$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 6$  para  $i = 1, 2, 3, 4, 5$  e  $6$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1)\} = \min \{7\} = 7$$

$$d(2) = \min \{d(2), d(1) + c(1, 2)\} = \min \{+\infty, 7 + 2\} = 9 \text{ e } p(2) = 1$$

$$d(3) = \min \{d(3), d(1) + c(1, 3)\} = \min \{10, 7 + \infty\} = 10$$

$$d(4) = \min \{d(4), d(1) + c(1, 4)\} = \min \{5, 7 + \infty\} = 5$$

$$d(5) = \min \{d(5), d(1) + c(1, 5)\} = \min \{0, 7 + \infty\} = 0$$

$$d(6) = \min \{d(6), d(1) + c(1, 6)\} = \min \{4, 7 + \infty\} = 4$$

$$k = 2$$

$$r(2) = 1$$

$$R = \{5, 6, 4, 1, 2\}$$

$$NR = \{3\}$$

$$a = 2$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 6$  para  $i = 1, 2, 3, 4, 5$  e  $6$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1), d(2) + c(2, 1)\} = \min \{7, 9 + \infty\} = 7$$

$$d(2) = \min \{d(2)\} = \min \{9\} = 9$$

$$d(3) = \min \{d(3), d(2) + c(2, 3)\} = \min \{10, 9 + \infty\} = 10$$

$$d(4) = \min \{d(4), d(2) + c(2, 4)\} = \min \{5, 9 + \infty\} = 5$$

$$d(5) = \min \{d(5), d(2) + c(2, 5)\} = \min \{0, 9 + \infty\} = 0$$

$$d(6) = \min \{d(6), d(2) + c(2, 6)\} = \min \{4, 9 + \infty\} = 4$$

$$k = 3$$

$$r(3) = 1$$

$$R = \{5, 6, 4, 1, 2, 3\}$$

$$NR = \emptyset$$

$$a = 3$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 6$  para  $i = 1, 2, 3, 4, 5$  e  $6$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1), d(3) + c(3, 1)\} = \min \{7, 10 - 4\} = 6 \text{ e } p(1) = 3$$

$$d(2) = \min \{d(2), d(3) + c(3, 2)\} = \min \{9, 10 + 2\} = 9$$

$$d(3) = \min \{d(3)\} = \min \{10\} = 10$$

$$d(4) = \min \{d(4), d(3) + c(3, 4)\} = \min \{5, 10 - 7\} = 3 \text{ e } p(4) = 3$$

$$d(5) = \min \{d(5), d(3) + c(3, 5)\} = \min \{0, 10 + \infty\} = 0$$

$$d(6) = \min \{d(6), d(3) + c(3, 6)\} = \min \{4, 10 + \infty\} = 4$$

$$R = \{5, 6, 2, 3\}$$

$$NR = \{1, 4\}$$

$$k = 4$$

$$r(4) = 2$$

$$R = \{5, 6, 2, 3, 4\}$$

$$NR = \{1\}$$

$$a = 4$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 6$  para  $i = 1, 2, 3, 4, 5$  e  $6$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1), d(4) + c(4, 1)\} = \min \{6, 3 + 2\} = 5 \text{ e } p(1) = 4$$

$$d(2) = \min \{d(2), d(4) + c(4, 2)\} = \min \{9, 3 + \infty\} = 9$$

$$d(3) = \min \{d(3), d(4) + c(4, 3)\} = \min \{10, 3 + \infty\} = 10$$

$$d(4) = \min \{d(4)\} = \min \{3\} = 3$$

$$d(5) = \min \{d(5), d(4) + c(4, 5)\} = \min \{0, 3 + \infty\} = 0$$

$$d(6) = \min \{d(6), d(4) + c(4, 6)\} = \min \{4, 3 + \infty\} = 4$$

$$k = 1$$

$$r(1) = 2$$

$$R = \{5, 6, 2, 3, 4, 1\}$$

$$NR = \emptyset$$

$$a = 1$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 6$  para  $i = 1, 2, 3, 4, 5$  e  $6$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1)\} = \min \{5\} = 5$$

$$d(2) = \min \{d(2), d(1) + c(1, 2)\} = \min \{9, 5 + 2\} = 7 \text{ e } p(2) = 1$$

$$d(3) = \min \{d(3), d(1) + c(1, 3)\} = \min \{10, 5 + \infty\} = 10$$

$$d(4) = \min \{d(4), d(1) + c(1, 4)\} = \min \{3, 5 + \infty\} = 3$$

$$d(5) = \min \{d(5), d(1) + c(1, 5)\} = \min \{0, 5 + \infty\} = 0$$

$$d(6) = \min \{d(6), d(1) + c(1, 6)\} = \min \{4, 5 + \infty\} = 4$$

$$R = \{5, 6, 3, 4, 1\}$$

$$NR = \{2\}$$

$$k = 2$$

$$r(2) = 2$$

$$R = \{5, 6, 3, 4, 1, 2\}$$

$$NR = \emptyset$$

$$a = 2$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 6$  para  $i = 1, 2, 3, 4, 5$  e  $6$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1), d(2) + c(2, 1)\} = \min \{5, 7 + \infty\} = 5$$

$$d(2) = \min \{d(2)\} = \min \{7\} = 7$$

$$d(3) = \min \{d(3), d(2) + c(2, 3)\} = \min \{10, 7 + \infty\} = 10$$

$$d(4) = \min \{d(4), d(2) + c(2, 4)\} = \min \{3, 7 + \infty\} = 3$$

$$d(5) = \min \{d(5), d(2) + c(2, 5)\} = \min \{0, 7 + \infty\} = 0$$

$$d(6) = \min \{d(6), d(2) + c(2, 6)\} = \min \{4, 7 + \infty\} = 4$$

$$sinal = 0$$

*Passo 3:* Como  $sinal = 0$ , temos:  $p(2) = 1, p(1) = 4, p(4) = 3, p(3) = 5, p(5) = 0$ , e o caminho mínimo do nó  $5$  ao nó  $2$  é dado por:  $C = \{(5, 3), (3, 4), (4, 1), (1, 2)\}$ .

pare

Consideramos, a seguir, um exemplo em que existe um circuito com comprimento total negativo no grafo.

**Exemplo 4.17** Ilustramos o algoritmo de Ford para achar o menor caminho do nó  $5$  ao nó  $2$  do grafo da Figura 4.21. Observe que existe um circuito com comprimento total negativo neste grafo.

*Passo 1:*

$$R = \{5\}$$

$$NR = \{1, 2, 3, 4\}$$

$$d(5) = 0, p(5) = 0 \text{ e } d(i) = +\infty, p(i) = 6, i = 1, 2, 3, 4$$

$$r(5) = 1, r(i) = 0, i = 1, 2, 3, 4$$

$$a = 5$$

$$sinal = 1$$

*Passo 2:*

$$d(1) = \min \{d(1), d(1) + c(5, 1)\} = \min \{+\infty, 0 + \infty\} = +\infty$$

$$d(2) = \min \{d(2), d(5) + c(5, 2)\} = \min \{+\infty, 0 + \infty\} = +\infty$$

$$d(3) = \min \{d(3), d(5) + c(5, 3)\} = \min \{+\infty, 0 + 4\} = 4 \text{ e } p(3) = 5$$

$$d(4) = \min \{d(4), d(5) + c(5, 4)\} = \min \{+\infty, 0 + \infty\} = +\infty$$

$$d(5) = \min \{d(5)\} = \min \{0\} = 0$$

$$k = 3$$

$$R = \{5, 3\}$$

$$r(3) = 1$$

$$NR = \{1, 2, 4\}$$

$$a = 3$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 5$  para  $i = 1, 2, 3, 4$  e  $5$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1), d(3) + c(3, 1)\} = \min \{+\infty, 4 + \infty\} = +\infty$$

$$d(2) = \min \{d(2), d(3) + c(3, 2)\} = \min \{+\infty, 4 + \infty\} = +\infty$$

$$d(3) = \min \{d(3)\} = 3$$

$$d(4) = \min \{d(4), d(3) + c(3, 4)\} = \min \{+\infty, 4 + 1\} = 5 \text{ e } p(4) = 3$$

$$d(5) = \min \{d(5), d(3) + c(3, 5)\} = \min \{0, 4 + \infty\} = 0$$

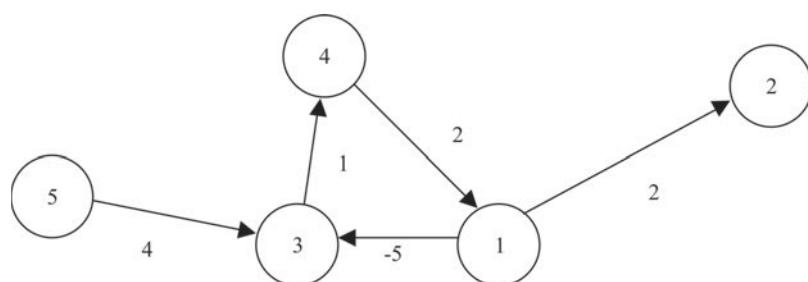
$$k = 4$$

$$r(4) = 1$$

$$R = \{5, 3, 4\}$$

$$NR = \{1, 2\}$$

$$a = 4$$



**Figura 4.21** Grafo com circuito negativo.

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 5$  para  $i = 1, 2, 3, 4$  e  $5$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1), d(4) + c(4, 1)\} = \min \{+\infty, 5 + 2\} = 7 \text{ e } p(1) = 4$$

$$d(2) = \min \{d(2), d(4) + c(4, 2)\} = \min \{+\infty, 5 + \infty\} = +\infty$$

$$d(3) = \min \{d(3), d(4) + c(4, 3)\} = \min \{4, 5 + \infty\} = 4$$

$$d(4) = \min \{d(4)\} = \min \{5\} = 5$$

$$d(5) = \min \{d(5), d(4) + c(4, 5)\} = \min \{0, 5 + \infty\} = 0$$

$$k = 1$$

$$r(1) = 1$$

$$R = \{5, 3, 4, 1\}$$

$$NR = \{2\}$$

$$a = 1$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 5$  para  $i = 1, 2, 3, 4$  e  $5$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1)\} = \min \{7\} = 7$$

$$d(2) = \min \{d(2), d(1) + c(1, 2)\} = \min \{+\infty, 7 + 2\} = 9 \text{ e } p(2) = 1$$

$$d(3) = \min \{d(3), d(1) + c(1, 3)\} = \min \{4, 7 - 5\} = 2 \text{ e } p(3) = 1$$

$$d(4) = \min \{d(4), d(1) + c(1, 4)\} = \min \{5, 7 + \infty\} = 5$$

$$d(5) = \min \{d(5), d(1) + c(1, 5)\} = \min \{0, 7 + \infty\} = 0$$

$$R = \{5, 4, 1\}$$

$$NR = \{2, 3\}$$

$$k = 3$$

$$r(3) = 2$$

$$R = \{5, 4, 1, 3\}$$

$$NR = \{2\}$$

$$a = 3$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 5$  para  $i = 1, 2, 3, 4$  e  $5$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1), d(3) + c(3, 1)\} = \min \{7, 2 + \infty\} = 7$$

$$d(2) = \min \{d(2), d(3) + c(3, 2)\} = \min \{9, 2 + \infty\} = 9$$

$$d(3) = \min \{d(3)\} = \min \{2\} = 2$$

$$d(4) = \min \{d(4), d(3) + c(3, 4)\} = \min \{5, 2 + 1\} = 3$$

$$d(5) = \min \{d(5), d(3) + c(3, 5)\} = \min \{0, 2 + \infty\} = 0$$

$$R = \{5, 1, 3\}$$

$$NR = \{2, 4\}$$

$$k = 4$$

$$r(4) = 2$$

$$R = \{5, 1, 4, 3\}$$

$$NR = \{2\}$$

$$a = 4$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 5$  para  $i = 1, 2, 3, 4$  e  $5$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1), d(4) + c(4, 1)\} = \min \{7, 3 + 2\} = 5 \text{ e } p(1) = 4$$

$$d(2) = \min \{d(2), d(4) + c(4, 2)\} = \min \{9, 3 + \infty\} = 9$$

$$d(3) = \min \{d(3), d(4) + c(4, 3)\} = \min \{2, 3 + \infty\} = 2$$

$$d(4) = \min \{d(4)\} = \min \{3\} = 3$$

$$d(5) = \min \{d(5), d(4) + c(4, 5)\} = \min \{0, 3 + \infty\} = 0$$

$$R = \{5, 4, 3\}$$

$$NR = \{1, 2\}$$

$$k = 1$$

$$r(1) = 3$$

$$R = \{5, 1, 3, 4\}$$

$$NR = \{2\}$$

$$a = 1$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 5$  para  $i = 1, 2, 3, 4$  e  $5$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1)\} = \min \{5\} = 5$$

$$d(2) = \min \{d(2), d(1) + c(1, 2)\} = \min \{9, 5 + 2\} = 7$$

$$d(3) = \min \{d(3), d(1) + c(1, 3)\} = \min \{2, 5 - 5\} = 0$$

$$d(4) = \min \{d(4), d(1) + c(1, 4)\} = \min \{3, 5 + \infty\} = 3$$

$$d(5) = \min \{d(5), d(1) + c(1, 5)\} = \min \{0, 5 + \infty\} = 0$$

$$R = \{5, 1, 4\}$$

$$NR = \{3, 2\}$$

$$k = 3$$

$$r(3) = 3$$

$$R = \{5, 3, 4, 1\}$$

$$NR = \{2\}$$

$$a = 3$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 5$  para  $i = 1, 2, 3, 4$  e  $5$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1), d(3) + c(3, 1)\} = \min \{5, 0 + \infty\} = 5$$

$$d(2) = \min \{d(2), d(3) + c(3, 2)\} = \min \{7, 0 + \infty\} = 7$$

$$d(3) = \min \{d(3)\} = \min \{0\} = 0$$

$$d(4) = \min \{d(4), d(3) + c(3, 4)\} = \min \{3, 0 + 1\} = 1$$

$$d(5) = \min \{d(5), d(3) + c(3, 5)\} = \min \{0, 0 + \infty\} = 0$$

$$R = \{5, 1, 3\}$$

$$NR = \{2, 4\}$$

$$k = 4$$

$$r(4) = 3$$

$$R = \{5, 1, 4, 3\}$$

$$NR = \{2\}$$

$$a = 4$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 5$  para  $i = 1, 2, 3, 4$  e  $5$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1), d(4) + c(4, 1)\} = \min \{5, 1 + 2\} = 3 \text{ e } p(1) = 4$$

$$d(2) = \min \{d(2), d(4) + c(4, 2)\} = \min \{7, 1 + \infty\} = 7$$

$$d(3) = \min \{d(3), d(4) + c(4, 3)\} = \min \{0, 1 + \infty\} = 0$$

$$d(4) = \min \{d(4)\} = \min \{1\} = 1$$

$$d(5) = \min \{d(5), d(4) + c(4, 5)\} = \min \{0, 1 + \infty\} = 0$$

$$R = \{5, 4, 3\}$$

$$NR = \{1, 2\}$$

$$k = 1$$

$$r(1) = 4$$

$$R = \{5, 1, 3, 4\}$$

$$NR = \{2\}$$

$$a = 1$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 5$  para  $i = 1, 2, 3, 4$  e  $5$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1)\} = \min \{3\} = 3$$

$$d(2) = \min \{d(2), d(1) + c(1, 2)\} = \min \{7, 3 + 2\} = 5$$

$$d(3) = \min \{d(3), d(1) + c(1, 3)\} = \min \{0, 3 - 5\} = -2$$

$$d(4) = \min \{d(4), d(1) + c(1, 4)\} = \min \{1, 3 + \infty\} = 1$$

$$d(5) = \min \{d(5), d(1) + c(1, 5)\} = \min \{0, 3 + \infty\} = 0$$

$$R = \{5, 1, 4\}$$

$$NR = \{3, 2\}$$

$$k = 3$$

$$r(3) = 4$$

$$R = \{5, 3, 4, 1\}$$

$$NR = \{2\}$$

$$a = 3$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 5$  para  $i = 1, 2, 3, 4$  e  $5$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1), d(3) + c(3, 1)\} = \min \{3, 0 + \infty\} = 3$$

$$d(2) = \min \{d(2), d(3) + c(3, 2)\} = \min \{5, 0 + \infty\} = 5$$

$$d(3) = \min \{d(3)\} = \min \{-2\} = -2$$

$$d(4) = \min \{d(4), d(3) + c(3, 4)\} = \min \{1, -2 + 1\} = -1$$

$$d(5) = \min \{d(5), d(3) + c(3, 5)\} = \min \{0, -2 + \infty\} = 0$$

$$R = \{5, 1, 3\}$$

$$NR = \{2, 4\}$$

$$k = 4$$

$$r(4) = 4$$

$$R = \{5, 1, 4, 3\}$$

$$NR = \{2\}$$

$$a = 4$$

*Passo 3:* Como  $sinal = 1$  e  $r(i) < 5$  para  $i = 1, 2, 3, 4$  e  $5$ , retorna-se ao *Passo 2*.

*Passo 2:*

$$d(1) = \min \{d(1), d(4) + c(4, 1)\} = \min \{3, -1 + 2\} = 1 \text{ e } p(1) = 4$$

$$d(2) = \min \{d(2), d(4) + c(4, 2)\} = \min \{5, -1 + \infty\} = 5$$

$$d(3) = \min \{d(3), d(4) + c(4, 3)\} = \min \{-2, -1 + \infty\} = -2$$

$$d(4) = \min \{d(4)\} = \min \{-1\} = -1$$

$$d(5) = \min \{d(5), d(4) + c(4, 5)\} = \min \{0, -1 + \infty\} = 0$$

$$R = \{5, 4, 3\}$$

$$NR = \{1, 2\}$$

$$k = 1$$

$$r(1) = 5$$

$$R = \{5, 1, 3, 4\}$$

$$NR = \{2\}$$

$$a = 1$$

*Passo 3:* Como  $r(1) = 5$ , pare. Existe um circuito no grafo com comprimento total negativo e o algoritmo termina sem encontrar uma solução para o problema. ■

Observe que o fato de o algoritmo terminar sem encontrar uma solução do problema indica apenas que foi detectado no grafo um circuito com comprimento total negativo. Isso não quer dizer que não exista caminho mínimo entre os nós da rede. Pode ser que exista um caminho mínimo (solução ótima finita), mas o algoritmo não é capaz de encontrá-lo ou assegurar que algum encontrado durante a execução seja o ótimo, ou pode ser que o problema não tenha solução ótima, pois o caminho entre os dois nós passa pelo circuito e, portanto, pode decrescer de valor tanto quanto se quiser. No caso do Exemplo 4.17 anterior, o problema não tem solução ótima.

Em alguns casos, há interesse em determinar as distâncias mínimas entre todos os pares de nós do grafo. Os algoritmos de Dijkstra e Ford podem ser utilizados repetidamente para se fazer essa determinação, mas apresentamos a seguir um outro algoritmo que utiliza uma abordagem diferente, recursiva, de programação dinâmica (estudada em detalhes no Capítulo 5).

#### *O algoritmo de Floyd para determinação do caminho mais curto entre todos os pares de nós no grafo\**

O algoritmo de Floyd encontra as distâncias mínimas (e os caminhos mínimos) entre todos os pares de nós do grafo. O algoritmo funciona mesmo que haja arcos com comprimentos negativos, mas não devem existir circuitos com comprimentos negativos. O algoritmo de Floyd determina as distâncias mínimas entre os pares de nós do grafo de maneira recursiva. Na iteração  $k$ , é determinado o caminho mínimo entre dois nós quaisquer utilizando somente os  $k$  primeiros nós.

Denotamos por  $d^k(i, j)$  a menor distância entre os nós  $i$  e  $j$ , dentre todos os caminhos que passam pelos primeiros  $k$  nós. Seja  $d^0(i, j)$  a distância mínima de  $i$  a  $j$  não passando por nenhum nó, ou seja  $d^0(i, j) = c(i, j)$ , caso o arco  $(i, j)$  exista; caso contrário,  $d^0(i, j) = +\infty$ . Note que  $d^k(i, j) = +\infty$  se não houver nenhum caminho entre  $i$  e  $j$  que passe apenas pelos primeiros  $k$  nós. Por definição,  $d^0(i, i) = 0$  para todo  $i$ . Como existem  $n$  nós no grafo,  $d^n(i, j)$  é a menor distância procurada entre os nós  $i$  e  $j$ .

O menor caminho de  $i$  a  $j$  que pode passar apenas pelos primeiros  $k$  nós passa ou não pelo  $k$ -ésimo nó. Se não passa pelo  $k$ -ésimo nó, então a menor distância de  $i$  a  $j$  que pode passar pelos primeiros  $k$  nós é igual à menor distância de  $i$  a  $j$  que pode passar pelos primeiros  $k - 1$  nós. Por

outro lado, se o menor caminho passa pelo  $k$ -ésimo nó, então (como não há ciclos com comprimentos negativos, por hipótese) o caminho mínimo que passa pelo  $k$ -ésimo nó é composto do caminho mínimo de  $i$  a  $k$  que pode passar apenas pelos primeiros  $k - 1$  nós, com o caminho mínimo de  $k$  a  $j$  que pode passar pelos primeiros  $k - 1$  nós. Temos, portanto, que

$$d^k(i, j) = \min\{d^{k-1}(i, j), d^{k-1}(i, k) + d^{k-1}(k, j)\}$$

Essa fórmula recursiva mostra que podemos determinar  $d^k(i, j)$  a partir de  $d^{k-1}(i, j)$ ,  $d^{k-1}(i, k)$  e  $d^{k-1}(k, j)$ , e define o algoritmo de Floyd. A fórmula recursiva anterior, utilizada para resolver o problema de caminho mínimo entre todos os pares de nós, é derivada da aplicação da técnica de programação dinâmica (apresentada no Capítulo 5) para resolução de problemas de otimização.

Como  $d^0(i, j) = c(i, j)$  para  $(i, j) \in E$  e  $d^0(i, j) = +\infty$  para  $(i, j) \notin E$ , é possível calcular  $d^1(i, j)$  para todo par  $(i, j)$ , depois  $d^2(i, j)$  a partir de  $d^1(i, j)$ , depois  $d^3(i, j)$  a partir de  $d^2(i, j)$ , e assim sucessivamente, até  $d^n(i, j)$ .

Para recuperar o caminho mínimo do nó  $i$  ao nó  $j$ , basta armazenar durante os cálculos o penúltimo nó (o nó anterior ao nó  $j$ , denotado por  $p(i, j)$ ) do caminho que fornece a menor distância. Isto é, na fórmula recursiva anterior, se  $d^k(i, j) = d^{k-1}(i, k) + d^{k-1}(k, j)$ , então  $p(i, j) = p(k, j)$ . Por exemplo,  $p(5, 2) = 3$  significa que o caminho mínimo de 5 a 2 (ainda desconhecido) passa pelo nó 3 antes de chegar no nó 2 e  $p(5, 3)$  fornece o nó anterior a 3, e assim por diante. Recuperando passo a passo os nós anteriores, recupera-se todo o caminho mínimo.

### Algoritmo de Floyd

*Dados:*

$G(N, E)$ : grafo em que  $N = \{1, 2, \dots, n\}$

$c(i, j)$  comprimento do arco  $(i, j)$ ,  $(i, j) \in E$  ( $c(i, j)$  pode ser negativo, mas o grafo não pode possuir circuito com comprimento negativo)

*Saída:*

$d(i, j)$  menor distância do nó  $i$  ao nó  $j$

$p(i, j)$  penúltimo nó intermediário no caminho mínimo do nó  $i$  ao nó  $j$

*Passo 1:*  $d^0(i, j) = c(i, j)$  se  $(i, j) \in E$   
 $d^0(i, j) = +\infty$  se  $(i, j) \notin E, i \neq j$   
 $d^0(i, i) = 0$  para  $i = 1, 2, \dots, n$   
 $p(i, j) = i$  para  $i, j = 1, 2, \dots, n$

*Passo 2:* Para  $k = 1, 2, \dots, n$ , faça:

$d^k(i, j) = \min\{d^{k-1}(i, j), d^{k-1}(i, k) + d^{k-1}(k, j)\}$   
e, se  $\{d^k(i, j) = d^{k-1}(i, k) + d^{k-1}(k, j)\}$ , então  $p(i, j) \leftarrow p(k, j)$

**Exemplo 4.18** Ilustramos o algoritmo de Floyd para o grafo da Figura 4.20. Denotemos por  $\mathbf{D}^k$  a matriz  $n \times n$  cujo elemento  $(i, j)$  é  $d^k(i, j)$ , e por  $\mathbf{P}$  a matriz dos elementos  $p(i, j)$ .

*Passo 1:*

$$\mathbf{D}^0 = \begin{bmatrix} 0 & 2 & \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty & \infty & \infty \\ -4 & 3 & 0 & -7 & \infty & \infty \\ 2 & \infty & \infty & 0 & \infty & \infty \\ \infty & \infty & 10 & 5 & 0 & 4 \\ 5 & \infty & \infty & 1 & \infty & 0 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

Passo 2:

$$\begin{aligned}
 k = 1, \quad d^l(i, j) &= \min \{d^0(i, j), d^0(i, 1) + d^0(1, j)\} \\
 d^l(1, 1) &= d^0(1, 1) = 0 \\
 d^l(1, 2) &= d^0(1, 2) = 2 \\
 d^l(1, 3) &= d^0(1, 3) = \infty \\
 d^l(1, 4) &= d^0(1, 4) = \infty \\
 d^l(1, 5) &= d^0(1, 5) = \infty \\
 d^l(1, 6) &= d^0(1, 6) = \infty \\
 d^l(2, 1) &= d^0(2, 1) = \infty \\
 d^l(2, 2) &= d^0(2, 2) = 0 \\
 d^l(2, 3) &= \min \{d^0(2, 3), d^0(2, 1) + d^0(1, 3)\} = \min \{\infty, \infty + \infty\} = \infty \\
 d^l(2, 4) &= \min \{d^0(2, 4), d^0(2, 1) + d^0(1, 4)\} = \min \{\infty, \infty + \infty\} = \infty \\
 d^l(2, 5) &= \min \{d^0(2, 5), d^0(2, 1) + d^0(1, 5)\} = \min \{\infty, \infty + \infty\} = \infty \\
 d^l(2, 6) &= \min \{d^0(2, 6), d^0(2, 1) + d^0(1, 6)\} = \min \{\infty, \infty + \infty\} = \infty \\
 d^l(3, 1) &= d^0(3, 1) = -4 \\
 d^l(3, 2) &= \min \{d^0(3, 2), d^0(3, 1) + d^0(1, 2)\} = \min \{3, -4 + 2\} = -2 \\
 \text{e } p(3, 2) &= p(1, 2) = 1 \\
 d^l(3, 3) &= d^0(3, 3) = 0 \\
 d^l(3, 4) &= \min \{d^0(3, 4), d^0(3, 1) + d^0(1, 4)\} = \min \{-7, -4 + \infty\} = -7 \\
 d^l(3, 5) &= \min \{d^0(3, 5), d^0(3, 1) + d^0(1, 5)\} = \min \{\infty, -4 + \infty\} = \infty \\
 d^l(3, 6) &= \min \{d^0(3, 6), d^0(3, 1) + d^0(1, 6)\} = \min \{\infty, -4 + \infty\} = \infty \\
 d^l(4, 1) &= d^0(4, 1) = 2 \\
 d^l(4, 2) &= \min \{d^0(4, 2), d^0(4, 1) + d^0(1, 2)\} = \min \{\infty, 2 + 2\} = 4 \\
 \text{e } p(4, 2) &= p(1, 2) = 1 \\
 d^l(4, 3) &= \min \{d^0(4, 3), d^0(4, 1) + d^0(1, 3)\} = \min \{\infty, 2 + \infty\} = \infty \\
 d^l(4, 4) &= d^0(4, 4) = 0 \\
 d^l(4, 5) &= \min \{d^0(4, 5), d^0(4, 1) + d^0(1, 5)\} = \min \{\infty, 2 + \infty\} = \infty \\
 d^l(4, 6) &= \min \{d^0(4, 6), d^0(4, 1) + d^0(1, 6)\} = \min \{\infty, 2 + \infty\} = \infty \\
 d^l(5, 1) &= d^0(5, 1) = \infty \\
 d^l(5, 2) &= \min \{d^0(5, 2), d^0(5, 1) + d^0(1, 2)\} = \min \{\infty, \infty + 2\} = \infty \\
 d^l(5, 3) &= \min \{d^0(5, 3), d^0(5, 1) + d^0(1, 3)\} = \min \{10, \infty + \infty\} = 10 \\
 d^l(5, 4) &= \min \{d^0(5, 4), d^0(5, 1) + d^0(1, 4)\} = \min \{5, \infty + \infty\} = 5 \\
 d^l(5, 5) &= d^0(5, 5) = 0 \\
 d^l(5, 6) &= \min \{d^0(5, 6), d^0(5, 1) + d^0(1, 6)\} = \min \{4, \infty + \infty\} = 4 \\
 d^l(6, 1) &= d^0(6, 1) = 5 \\
 d^l(6, 2) &= \min \{d^0(6, 2), d^0(6, 1) + d^0(1, 2)\} = \min \{\infty, 5 + 2\} = 7
 \end{aligned}$$

$$\text{e } p(6, 2) = p(1, 2) = 1$$

$$d^1(6, 3) = \min \{d^0(6, 3), d^0(6, 1) + d^0(1, 3)\} = \min \{\infty, 5 + \infty\} = \infty$$

$$d^1(6, 4) = \min \{d^0(6, 4), d^0(6, 1) + d^0(1, 4)\} = \min \{1, 5 + \infty\} = 1$$

$$d^1(6, 5) = \min \{d^0(6, 5), d^0(6, 1) + d^0(1, 5)\} = \min \{\infty, 5 + \infty\} = \infty$$

$$d^1(6, 6) = d^0(6, 6) = 0$$

De maneira similar, calcula-se para  $k = 2$ ,  $d^2(i, j) = \min \{d^1(i, j), d^1(i, 2) + d^1(2, j)\}$

$$\mathbf{D}^2 = \begin{bmatrix} 0 & 2 & \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty & \infty & \infty \\ -4 & -2 & 0 & -7 & \infty & \infty \\ 2 & 4 & \infty & 0 & \infty & \infty \\ \infty & \infty & 10 & 5 & 0 & 4 \\ 5 & 7 & \infty & 1 & \infty & 0 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 1 & 3 & 3 & 3 & 3 \\ 4 & 1 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 & 5 \\ 6 & 1 & 6 & 6 & 6 & 6 \end{bmatrix}$$

Para  $k = 3$ ,  $d^3(i, j) = \min \{d^2(i, j), d^2(i, 3) + d^2(3, j)\}$

$$\mathbf{D}^3 = \begin{bmatrix} 0 & 2 & \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty & \infty & \infty \\ -4 & -2 & 0 & -7 & \infty & \infty \\ 2 & 4 & \infty & 0 & \infty & \infty \\ 6 & 8 & 10 & 3 & 0 & 4 \\ 5 & 7 & \infty & 1 & \infty & 0 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 1 & 3 & 3 & 3 & 3 \\ 4 & 1 & 4 & 4 & 4 & 4 \\ 3 & 1 & 5 & 3 & 5 & 5 \\ 6 & 1 & 6 & 6 & 6 & 6 \end{bmatrix}$$

Para  $k = 4$ ,  $d^4(i, j) = \min \{d^3(i, j), d^3(i, 4) + d^3(4, j)\}$

$$\mathbf{D}^4 = \begin{bmatrix} 0 & 2 & \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty & \infty & \infty \\ -5 & -3 & 0 & -7 & \infty & \infty \\ 2 & 4 & \infty & 0 & \infty & \infty \\ 5 & 7 & 10 & 3 & 0 & 4 \\ 3 & 5 & \infty & 1 & \infty & 0 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 \\ 4 & 1 & 3 & 3 & 3 & 3 \\ 4 & 1 & 4 & 4 & 4 & 4 \\ 4 & 1 & 5 & 3 & 5 & 5 \\ 4 & 1 & 6 & 6 & 6 & 6 \end{bmatrix}$$

Para  $k = 5$ ,  $d^5(i, j) = \min \{d^4(i, j), d^4(i, 5) + d^4(5, j)\}$

$$\mathbf{D}^5 = \begin{bmatrix} 0 & 2 & \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty & \infty & \infty \\ -5 & -3 & 0 & -7 & \infty & \infty \\ 2 & 4 & \infty & 0 & \infty & \infty \\ 5 & 7 & 10 & 3 & 0 & 4 \\ 3 & 5 & \infty & 1 & \infty & 0 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 \\ 4 & 1 & 3 & 3 & 3 & 3 \\ 4 & 1 & 4 & 4 & 4 & 4 \\ 4 & 1 & 5 & 3 & 5 & 5 \\ 4 & 1 & 6 & 6 & 6 & 6 \end{bmatrix}$$

Para  $k = 6$ ,  $d^6(i, j) = \min \{d^5(i, j), d^5(i, 6) + d^5(6, j)\}$

$$\mathbf{D}^6 = \begin{bmatrix} 0 & 2 & \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty & \infty & \infty \\ -5 & -3 & 0 & -7 & \infty & \infty \\ 2 & 4 & \infty & 0 & \infty & \infty \\ 5 & 7 & 10 & 3 & 0 & 4 \\ 3 & 5 & \infty & 1 & \infty & 0 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 \\ 4 & 1 & 3 & 3 & 3 & 3 \\ 4 & 1 & 4 & 4 & 4 & 4 \\ 4 & 1 & 5 & 3 & 5 & 5 \\ 4 & 1 & 6 & 6 & 6 & 6 \end{bmatrix}$$

Na matriz  $\mathbf{D}^6$ , temos todas as menores distâncias entre pares de nós do grafo da Figura 4.20. Por exemplo, a menor distância do nó 3 ao nó 1 é  $-5$ , a menor distância do nó 5 ao nó 2 é  $7$  etc. Para recuperar o caminho mínimo, fazemos uso da matriz  $\mathbf{P}$ . No caso, por exemplo, no caminho mínimo do nó 3 ao nó 1, temos que o último nó intermediário no caminho é o nó  $p(3, 1) = 4$ . Do caminho mínimo do nó 4 ao nó 1, temos que o último nó intermediário no caminho é o nó  $p(4, 1) = 4$ . Portanto, o último (e único) arco do menor caminho do nó 4 ao nó 1 é  $(4, 1)$ . O último nó intermediário do menor caminho do nó 3 ao nó 4 é  $p(3, 4) = 3$ . Portanto, o último (e único) arco do menor caminho do nó 3 ao nó 4 é  $(3, 4)$ . Determinamos, assim, o caminho mais curto do nó 3 ao nó 1 dado por  $C = \{(3, 4), (4, 1)\}$ .

De maneira similar, o último nó intermediário do caminho mais curto do nó 5 ao nó 2 é  $p(5, 2) = 1$ . O último nó intermediário do caminho mais curto do nó 1 ao nó 2 é 1. Portanto, o último (e único) arco do menor caminho do nó 1 ao nó 2 é  $(1, 2)$ . O último nó intermediário do menor caminho do nó 5 ao nó 1 é  $p(5, 1) = 4$ . O último nó intermediário do menor caminho do nó 4 ao nó 1 é  $p(4, 1) = 4$ . Portanto, o último (e único) arco do menor caminho do nó 4 ao nó 1 é  $(4, 1)$ . O último nó intermediário do menor caminho do nó 5 ao nó 4 é  $p(5, 4) = 3$ . O último nó intermediário do menor caminho do nó 3 ao nó 4 é  $p(3, 4) = 3$ . Portanto, o último (e único) arco do menor caminho do nó 3 ao nó 4 é  $(3, 4)$ . O último nó intermediário do menor caminho do nó 5 ao nó 3 é  $p(5, 3) = 5$ . Portanto, o último (e único) arco do menor caminho do nó 5 ao nó 3 é  $(5, 3)$ . Assim, o caminho mais curto do nó 5 ao nó 2 é dado por  $C = \{(5, 3), (3, 4), (4, 1), (1, 2)\}$ .

■

#### 4.2.2 O problema de caminho máximo

Um outro problema simples em grafos é o de determinar *o caminho máximo entre dois nós no grafo*. A formulação matemática deste problema é igual à do problema de caminho mais curto, exceto que na função objetivo temos uma maximização, ao invés de minimização. O problema de caminho máximo aparece com freqüência em aplicações práticas, por exemplo, no problema da mochila (ver Seção 3.4 do Capítulo 3 e Seção 4.2.1, Exemplo 4.14), na programação de projetos (ver exemplo na Seção 2.2 do Capítulo 2), entre outros. Os algoritmos da Seção 4.2.1 obedecidas às condições para os quais foram desenvolvidos, podem ser utilizados para resolver o problema de caminho máximo. Observe que maximizar uma função objetivo é equivalente a minimizar o negativo desta função, portanto, o problema de caminho máximo entre dois nós no grafo pode ser transformado em um problema de caminho mínimo entre esses mesmos nós.

Considere um problema de programação de um projeto (como o exemplificado na Seção 2.2 – Capítulo 2). Considere que cada arco  $(i, j)$  no grafo corresponde a uma atividade do projeto. Assim, o nó cauda  $i$  desse arco indica o início da atividade e o nó cabeça  $j$ , o fim da atividade, e o comprimento do arco é a duração da atividade.

Iniciamos a construção do grafo com o nó 0 representando o início do projeto e um nó  $F$  que representa o final do projeto. Todos os arcos correspondentes a atividades que não exigem o término de nenhuma outra atividade anterior (isto é, atividade sem pré-requisito) têm seu nó cauda em 0. Todos os arcos correspondentes a atividades que não têm necessidade de serem completadas para que alguma outra atividade seja executada têm o seu nó cabeça em  $F$ . Os demais nós do grafo são definidos da seguinte forma. O nó cabeça de um arco correspondente a uma atividade, digamos, A, que necessariamente precisa estar completada para que uma outra atividade, digamos, B, seja iniciada, deverá ser o nó cauda da atividade B nesse grafo. Desta forma, o grafo referente às atividades do projeto fica completamente definido. O término do projeto se dá pela distância do caminho máximo do nó 0 ao nó  $F$ .

O caminho máximo nesse grafo do nó 0 ao nó  $F$  é denominado *caminho crítico*, pois qualquer atraso em alguma das atividades desse caminho implica atraso no término do projeto. Portanto, qualquer atividade desse caminho crítico é uma atividade crítica. A determinação do caminho crítico pode ser feita de maneira recursiva, a partir de um conjunto de nós cujos caminhos máximos, do nó 0 a estes nós, já tenham sido determinados.

Seja  $t(i)$  o comprimento do caminho mais longo do nó 0 ao nó  $i$ . Chamemos de nó rotulado aquele nó cujo caminho mais longo do nó 0 até ele já tenha sido determinado. Os demais nós são denominados não rotulados. Começamos com o nó 0 com  $t(0) = 0$ , ou seja, o início do projeto. O nó 0, então, é o primeiro nó rotulado. Calcula-se a seguir, sucessivamente,  $t(k)$  de algum nó  $k$  ainda não rotulado que seja nó cabeça de arcos cujos nós caudas já tenham sido todos rotulados.

Seja

$P(k)$  o conjunto dos nós predecessores do nó  $k$ , ou seja,  $P(k) = \{i \text{ tal que } (i, k) \in E\}$

$R$  o conjunto dos nós rotulados

Inicialmente,  $R = \{0\}$ ,  $P(0) = \emptyset$  (o nó inicial não tem predecessores). Se  $P(k) \subseteq R$  (isto é, todos os predecessores do nó  $k$  estão rotulados) e  $k \notin R$ , então

$$t(k) = \max_{i \in P(k)} \{t(i) + c(i, k)\}$$

ou seja, o caminho máximo de 0 até  $k$  é igual ao caminho máximo dentre todos aqueles que passam por nós caudas cujos arcos têm como nó cabeça o nó  $k$ . O procedimento continua até o momento em que não existem mais nós a rotular. O nó  $F$  é o último nó a ser rotulado, pois este nó corresponde ao final do projeto. A fórmula recursiva anterior, utilizada para resolver o problema de caminho máximo, é derivada da aplicação da técnica de programação dinâmica (apresentada no Capítulo 5) para resolução de problemas de otimização.

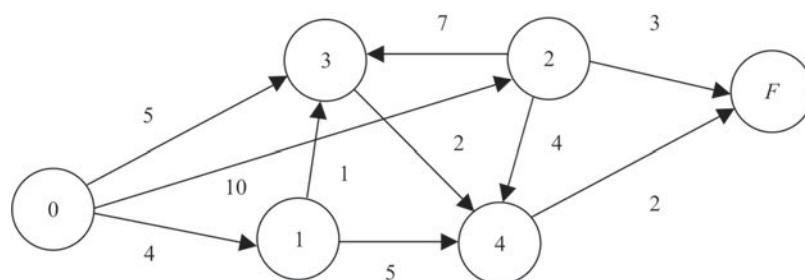
Deve-se ter em mente que este algoritmo não funciona para qualquer grafo. Por exemplo, em grafos com circuitos não se consegue rotular os nós pertencentes a um circuito, portanto, não é possível utilizar a fórmula recursiva nesses nós. Entretanto, o algoritmo é suficiente para grafos obtidos de projetos com atividades que obedecem a uma relação de ordem coerente, ou seja, um grafo sem circuitos.

**Exemplo 4.19** Ilustramos a aplicação do algoritmo de caminho mais longo para o grafo da Figura 4.22.

Temos, inicialmente,  $t(0) = 0$ ,  $R = \{0\}$  e  $NR = \{1, 2, 3, 4, F\}$ , em que  $R$  é o conjunto dos nós rotulados e  $NR$  é o conjunto dos nós não rotulados, respectivamente. Dentre os nós em  $NR$ , selecionemos algum nó  $k$  que seja nó cabeça de arcos cujos nós cauda já tenham sido todos rotulados. Temos, então, como candidatos, os nós 1, 2 e 3. Podemos escolher, por exemplo, os nós 1 ou 2. O nó 3 não poderia ser escolhido neste momento, pois  $P(3) = \{0, 1, 2\} \not\subseteq R$ , ou seja, ele não é nó cabeça de arcos cujos nós cauda já tenham sido todos rotulados.

Calculamos, inicialmente,

$$t(1) = \{t(0) + 4\} = 4, R = \{0, 1\} \text{ e } NR = \{2, 3, 4, F\}$$



**Figura 4.22** Encontrar caminho mais longo (caminho crítico) entre os nós 0 e  $F$ .

Em seguida, calculamos,

$$t(2) = \{t(0) + 10\} = 10, R = \{0, 1, 2\} \text{ e } NR = \{3, 4, F\}$$

Podemos, agora, determinar

$$t(3) = \max \{t(0) + 5, t(1) + 1, t(2) + 7\} = 17, R = \{0, 1, 2, 3\} \text{ e } NR = \{4, F\}$$

depois, determinar

$$t(4) = \max \{t(1) + 5, t(2) + 4, t(3) + 2\} = 19, R = \{0, 1, 2, 3, 4\} \text{ e } NR = \{F\}$$

e, finalmente, determinamos

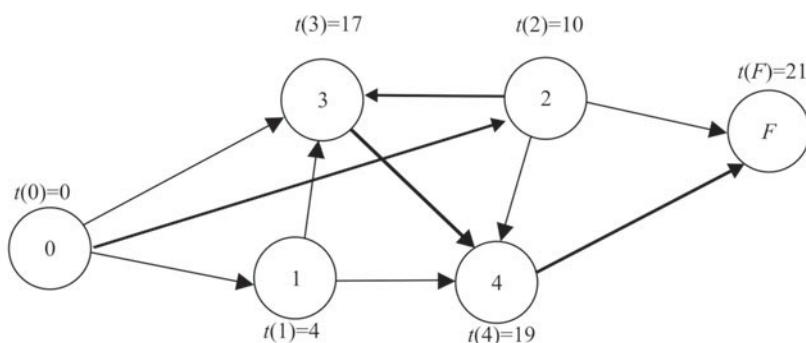
$$t(F) = \max \{t(2) + 3, t(4) + 2\} = 21, R = \{0, 1, 2, 3, 4, F\} \text{ e } NR = \emptyset$$

Portanto, o caminho mais longo de 0 a  $F$  tem comprimento 21 e é dado por  $C = \{(0, 2), (2, 3), (3, 4), (4, F)\}$ . Qualquer atraso em alguma das atividades nesse caminho (*atividades críticas*) implica um atraso do projeto. O caminho crítico do gráfico da Figura 4.22 está indicado na Figura 4.23.

Conforme mencionado na Seção 3.6.4, o algoritmo de caminho mais longo de um grafo de atividades de um projeto é conhecido como o *método de caminho crítico* ou CPM (do inglês, *critical path method*). Com o CPM é possível determinar o quanto cada atividade fora do caminho crítico de um projeto pode atrasar sem que isso afete o prazo de encerramento do projeto.

O comprimento  $t(i)$  do caminho máximo do nó 0 até um nó qualquer  $i$  de um grafo de atividades de um projeto também é conhecido como o *instante mais cedo do evento  $i$*  (*earliest event time*), ou seja, o tempo mais curto no qual o evento correspondente ao nó  $i$  pode ocorrer. Por exemplo,  $t(1) = 4$  significa que o evento 1 (fim da atividade (0, 1) e início das atividades (1, 3) e (1, 4)) não pode ser realizado antes do instante 4.

Tendo disponível  $t(i)$ , é possível determinar o instante mais tarde que uma atividade que precede imediatamente o evento  $i$  pode ocorrer sem que o evento  $i$  atrasse. Considere, por exemplo, o nó 3 do grafo de atividades da Figura 4.22. Como  $t(3) = 17$ , a atividade correspondente ao arco (1, 3) que precede imediatamente o nó 3 pode iniciar em qualquer instante anterior ou igual a  $t(3) - 1$  sem que haja atrasos no encerramento do projeto. Portanto, o evento correspondente ao nó 1 pode ocorrer em qualquer instante anterior ou igual a  $t(3) - 1 = 16$  e, com isso, não haverá atrasos no encerramento do projeto.



**Figura 4.23** Caminho mais longo (caminho crítico) do nó 0 ao nó  $F$ .

O quanto a duração de uma atividade  $(i, j)$  pode aumentar sem que, com isso, haja um atraso no início de qualquer outra atividade posterior a ela é denominada *flutuação livre* ou *folga* da atividade  $(i, j)$  e é denotada por  $FL(i, j)$  no encerramento do projeto. Se  $c(i, j)$  é a duração da atividade  $(i, j)$ ,  $FL(i, j)$  é dado por

$$FL(i, j) = t(j) - t(i) - c(i, j)$$

pois o tempo mais cedo do evento  $j$  não se altera se a duração de uma atividade  $(i, j)$  aumentar de  $\delta$ , desde que  $t(i) + c(i, j) + \delta \leq t(j)$  ou  $\delta \leq t(j) - t(i) - c(i, j)$ . Considere, por exemplo, a atividade  $(1, 3)$  da Figura 4.22. A flutuação livre de  $(1, 3)$  é igual a

$$FL(1, 3) = t(3) - t(1) - c(1, 3) = 17 - 4 - 1 = 12$$

ou seja, a duração da atividade  $(1, 3)$  pode aumentar de 12 unidades sem que com isso atrasse o evento correspondente ao nó 3 (isto é, início da atividade  $(3, 4)$  e término das atividades  $(0, 3)$ ,  $(1, 3)$  e  $(2, 3)$ ).

O tempo mais tarde que um evento correspondente a um nó  $i$  pode ocorrer sem que com isso haja atrasos no encerramento do projeto é denominado o *instante mais tarde do evento (late event time)*, e é denotado por  $LT(i)$ . Esse tempo é definido pelo menor dos tempos de todas as atividades que têm  $i$  como nó cauda. Por exemplo,  $LT(1)$  do grafo de atividades da Figura 4.22 é dado por:

$$LT(1) = \min \{LT(3) - 1, LT(4) - 5\}$$

Assim, para calcular  $LT(1)$ , precisamos dos valores de  $LT(3)$  e  $LT(4)$ , que não estão disponíveis e precisam ser calculados. Desta forma, para calcular  $LT(.)$  para os diversos nós do grafo, precisamos iniciar de trás para a frente, a partir de  $F$ , fazendo  $LT(F) = t(F)$ . Temos, para o grafo da Figura 4.22, que  $LT(F) = t(F) = 21$ . De posse de  $LT(F)$ , podemos calcular

$$LT(4) = \min \{LT(F) - 2\} = \min \{21 - 2\} = 19$$

De posse de  $LT(F)$  e  $LT(4)$ , podemos calcular

$$LT(3) = \min \{LT(4) - 2\} = \min \{19 - 2\} = 17$$

De posse de  $LT(F)$ ,  $LT(4)$  e  $LT(3)$ , podemos calcular

$$LT(2) = \min \{LT(F) - 3, LT(4) - 4, LT(3) - 7\} = \min \{21 - 3, 19 - 4, 17 - 7\} = 10,$$

$$LT(1) = \min \{LT(3) - 1, LT(4) - 5\} = \min \{17 - 1, 19 - 5\} = 14$$

De posse de  $LT(F)$ ,  $LT(4)$ ,  $LT(3)$ ,  $LT(2)$  e  $LT(1)$ , podemos calcular

$$LT(0) = \min \{LT(3) - 5, LT(2) - 10, LT(1) - 4\} = \min \{17 - 5, 10 - 10, 14 - 4\} = 0$$

o que já era esperado. De fato, para todos os nós  $i$  que pertencem ao caminho crítico, devemos ter  $LT(i) = t(i)$  pela própria definição de caminho crítico. Encontramos que  $LT(1) = 14$ , portanto, o evento correspondente ao nó 1 deve ocorrer no máximo até o instante 14 para que não haja atrasos no encerramento do projeto.

Tendo em mãos os valores de  $t(i)$  e  $LT(i)$  dos nós do grafo, é possível determinar agora a *flutuação total* de uma atividade  $(i, j)$  (denotada por  $FT(i, j)$ ), ou seja, quanto tempo a duração de uma atividade  $(i, j)$  pode aumentar sem que com isso haja um atraso no encerramento do projeto.  $FT(i, j)$  é definido por

$$FT(i, j) = LT(j) - t(i) - c(i, j)$$

uma vez que o encerramento de um projeto não se altera se a duração de uma atividade  $(i, j)$  aumentar de  $\delta$ , desde que  $t(i) + c(i, j) + \delta \leq LT(j)$  ou  $\delta \leq LT(j) - t(i) - c(i, j)$ . Considere, por exemplo, a atividade  $(0,1)$  do projeto. A flutuação total de  $(0,1)$  é

$$FT(0,1) = LT(1) - t(0) - c(0, 1) = 14 - 0 - 4 = 10$$

ou seja, a duração da atividade  $(0, 1)$  pode aumentar de 10 unidades sem que com isso o encerramento do projeto se altere.

Na prática, a duração das atividades nem sempre é conhecida com certeza. Assim, há incertezas de que o projeto se encerrará em um determinado tempo. A técnica de avaliação e revisão de programa – PERT (do inglês, *program evaluation and review technique*) modela cada uma das durações das atividades do projeto como uma variável aleatória e utiliza a teoria de probabilidades para determinar, por exemplo, as chances de um projeto ser encerrado até um determinado prazo de tempo, uma vez que não se pode ter certeza de que o projeto realmente se encerrará em uma determinada data, dada a incerteza na duração de diversas das atividades nele envolvidas.

As durações dos caminhos do nó 0 ao nó  $F$  são somas de variáveis aleatórias cuja distribuição pode ser, geralmente, aproximada por uma distribuição normal. Para determinar as chances de um projeto ser encerrado até um certo prazo é necessário obter pelo menos a média e a variância das variáveis aleatórias correspondentes às somas dos caminhos do nó 0 ao nó  $F$ . O cálculo da variância de uma variável aleatória, que é a soma de variáveis aleatórias, pode ser complicado, a menos que hipóteses simplificadoras sejam feitas. Por exemplo, pode-se admitir que as variáveis aleatórias são independentes. Essa suposição, entretanto, nem sempre condiz com a realidade.

Mesmo que se consigam determinar esses valores, dada a aleatoriedade das variáveis, não existe mais a certeza de qual deverá ser o caminho crítico. Uma suposição que normalmente é feita é de que o caminho crítico dado pelo CPM (utilizando-se o valor médio de cada uma das atividades) permanecerá crítico para o projeto. Essa suposição, entretanto, carece de suporte teórico.

#### 4.2.3 O problema da árvore geradora mínima

Um problema de interesse em grafos, também relacionado à menor distância, é o problema de determinação da árvore geradora de custo mínimo ou, simplesmente, árvore geradora mínima de um grafo. Este problema também aparece em várias aplicações práticas diretamente ou como subproblemas de outros problemas mais complexos.

**Exemplo 4.20** Considere uma empresa que deseja conectar entre si todos os diversos departamentos de sua fábrica por cabos de fibra óptica. Com isso, pretende-se ter uma transferência de dados rápida, segura e ágil entre os diversos departamentos. A empresa deseja instalar essa conexão ao menor custo possível e sabe que as despesas são proporcionais ao comprimento total do cabo de fibra óptica necessário para se realizar essas conexões. Se cada nó do grafo representa um departamento da empresa e as arestas que unem dois nós representam o cabo de fibra óptica entre os dois departamentos correspondentes, desejamos determinar apenas o conjunto de arestas na rede que conecta todos os nós e cujo comprimento total seja o menor possível.

##### *Formulação matemática do problema da árvore geradora mínima*

Para escrever um modelo matemático para o problema da árvore geradora mínima de um grafo não direcionado com  $n$  nós, podemos imaginar um problema de transporte em que existe suprimento de  $n-1$  unidades de um produto em apenas um dos nós do grafo, digamos o nó 1, e existe uma demanda de exatamente 1 unidade do produto em cada um dos demais  $n-1$  nós. O custo de transporte em cada aresta  $(i, j)$  no grafo, entretanto, diferente do que ocorre no caso do problema de transporte, independe do volume transportado do produto na aresta. Ele é zero se a

aresta não é utilizada no transporte ou  $c_{ij}$  (a distância da aresta), se a aresta é utilizada (admite-se, sem perda de generalidade, que  $c_{ij} \geq 0$  para toda aresta  $(i, j) \in E$ ). Devido a essa característica peculiar do custo na aresta tomar dois valores apenas, precisamos utilizar no modelo uma variável binária, ou seja, uma variável que toma apenas dois valores, 0 ou 1. No caso, consideraremos que essa variável toma o valor 0 quando não há fluxo na aresta e, o valor 1, caso contrário.

Adicionalmente, observe que, devido ao custo depender apenas da utilização ou não de uma aresta no grafo, qualquer um dos nós do grafo pode ser escolhido como o nó onde todo o suprimento  $n-1$  do produto está concentrado, pois a solução do problema (a árvore geradora mínima) será a mesma, independente do nó escolhido. Assim, sem perda de generalidade, apresentamos, a seguir, o modelo matemático para o problema da árvore geradora mínima considerando que o suprimento está todo concentrado no nó 1. O modelo fica,

$$\begin{aligned} \min f(\mathbf{x}, \mathbf{y}) = & \sum_{(i,j) \in E} c_{ij} y_{ij} \\ & \sum_{\{j:(1,j) \in E\}} x_{1j} = n - 1 \\ & \sum_{\{i:(i,j) \in E\}} x_{ij} - \sum_{\{k:(j,k) \in E\}} x_{jk} = 1, \quad j = 2, \dots, n \\ & x_{ij} \geq 0, \quad (i, j) \in E \\ & (n-1)y_{ij} \geq x_{ij}, \quad (i, j) \in E \\ & y_{ij} \leq x_{ij}, \quad (i, j) \in E \\ & y_{ij} \in \{0, 1\}, \quad (i, j) \in E \end{aligned}$$

em que

- $x_{ij}$  é a quantidade transportada do produto do nó  $i$  para o nó  $j$  utilizando a aresta  $(i, j)$
- $y_{ij}$  é a variável binária que indica se a aresta  $(i, j)$  é utilizada para transportar o produto do nó  $i$  para o nó  $j$ . Ela toma o valor 1 se a aresta  $(i, j)$  é utilizada e 0, caso contrário
- $c_{ij}$  é o “custo” incorrido caso a aresta  $(i, j)$  seja utilizada para se realizar este transporte do produto do nó  $i$  para o nó  $j$ .

A solução do modelo eventualmente pode não ser uma árvore, mas é fácil obter a partir dela uma solução árvore alternativa. Para outro modelo, cuja solução sempre é uma árvore, mas envolve um número exponencial de restrições, veja Bertsimas e Tsitsiklis (1997).

Observe que além da mudança da função objetivo, introduzimos mais 3 restrições (as 3 últimas) ao modelo de fluxo em redes a ser apresentado mais adiante neste capítulo. Essas restrições relacionam a variável  $x_{ij}$  com a variável correspondente  $y_{ij}$ , de modo que,  $y_{ij}$  seja 1 sempre que  $x_{ij}$  for positivo e, seja 0, caso contrário.

De fato, se  $x_{ij}$  for positivo, então para que a restrição

$$(n-1)y_{ij} \geq x_{ij}$$

seja satisfeita,  $y_{ij}$  tem de ser positivo, ou seja,  $y_{ij}$  é 1. Como temos no máximo  $n-1$  itens a transportar do nó 1 aos demais nós, em nenhum arco transportaremos mais do que  $n-1$  unidades e, portanto, essa restrição será satisfeita. Por outro lado, a restrição

$$y_{ij} \leq x_{ij},$$

assegura que  $y_{ij}$  é zero quando  $x_{ij}$  é zero. Essa restrição também garante que se  $y_{ij}$  é 1, a quantidade transportada no arco  $(i, j)$  é no mínimo 1, ou seja, esse arco será incluído na solução.

O modelo apresentado tem variáveis reais e variáveis inteiras. O método simplex poderá ser aplicado à relaxação<sup>7</sup> linear do problema, isto é, o mesmo problema sem as restrições de que as variáveis  $y_{ij}$  devam ser inteiras iguais a 0 ou 1, mas apenas que devam estar no intervalo entre 0 e 1, ou seja,  $0 \leq y_{ij} \leq 1$ . Entretanto, não se têm garantias de obtenção de uma solução inteira para as variáveis  $y_{ij}$ .

O problema da árvore geradora mínima pode ser resolvido otimamente utilizando-se um procedimento guloso no sentido de sempre escolher, sucessivamente, as arestas de menor comprimento. Este procedimento é conhecido como o *algoritmo de Kruskal*.

#### *Algoritmo de Kruskal para geração de árvore mínima*

*Dados:*

$G(N, E)$ , em que  $N = \{1, 2, \dots, n\}$

$c(i, j)$  comprimento da aresta  $(i, j)$

*Saída:*

$ST$  árvore geradora mínima

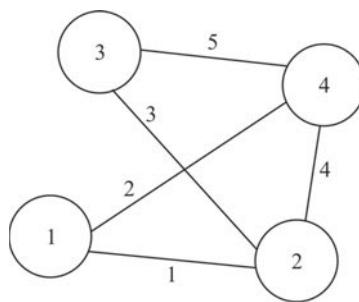
$LST$  comprimento da árvore geradora mínima.

*Passo 1:*  $LST = 0$

$C = \{1\}$  (escolha do nó 1 é arbitrária)

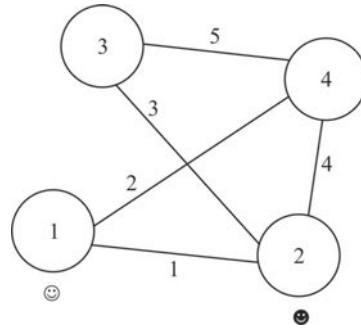
$C' = N - C$

$ST = \emptyset$



**Figura 4.24** Grafo ilustrativo para determinação da árvore geradora mínima.

<sup>7</sup> O termo “relaxação” vem do fato de se relaxar alguma restrição do problema, no caso, a restrição de integralidade das variáveis deixou de ser exigida.



**Figura 4.25** Nô inicial da árvore geradora mínima.

*Passo 2:* Enquanto ( $C' \neq \emptyset$ ), faça

Início\_Enquanto

Selecione  $j \in C'$  tal que  $c(k, j) = \min \{c(r, s) \text{ tal que } (r, s) \in E, r \in C \text{ e } s \in C'\}$ , isto é, o nó em  $C'$  mais próximo de qualquer nó em  $C$ . Seja  $k$  o nó em  $C$  mais próximo de  $j$ .

$C \leftarrow C \cup \{j\}$

$C' \leftarrow C' - \{j\}$

$LST \leftarrow LST + c(k, j)$

$ST \leftarrow ST \cup \{(k, j)\}$

Fim\_Enquanto

Ilustremos a aplicação do algoritmo de geração de árvore mínima para o grafo da Figura 4.24. Os comprimentos dos arcos  $c(i, j)$  estão indicados na figura.

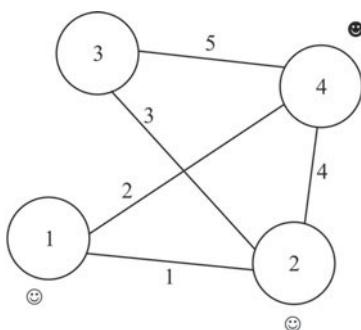
*Passo 1:*  $LST = 0$

$i = 1$

$C = \{1\}$

$C' = \{2, 3, 4\}$

$ST = \emptyset$



**Figura 4.26** Nô 2 é o mais próximo dos nós em  $C$ .

Na Figura 4.25, o nó selecionado é indicado com um ‘ $\odot$ ’. O nó que não está em  $C$  e é o mais próximo de qualquer nó no conjunto  $C$  é o próximo nó a ser inserido nesse conjunto, e está indicado com um ‘ $\oplus$ ’. Esses mesmos símbolos são empregados nas figuras subseqüentes para indicar nós em  $C$  e o nó mais próximo de qualquer nó desse conjunto.

*Passo 2:* O nó 2 é o mais próximo de qualquer nó em  $C$  e o nó 1 é o nó em  $C$  que é o mais próximo de 2.

$$C = \{1, 2\} \text{ (ver Figura 4.26)}$$

$$C' = \{3, 4\}$$

$$LST = 0 + c(1, 2) = 1$$

$$ST = \{(1, 2)\}$$

O nó 4 é o mais próximo de qualquer nó em  $C$  e o nó 1 é o nó em  $C$  que é o mais próximo de 4.

$$C = \{1, 2, 4\} \text{ (ver Figura 4.27)}$$

$$C' = \{3\}$$

$$LST = 1 + c(1, 4) = 3$$

$$ST = \{(1, 2), (1, 4)\}$$

O nó 3 é o mais próximo de qualquer nó em  $C$  e o nó 2 é o nó em  $C$  que é o mais próximo de 3.

$$C = \{1, 2, 4, 3\} \text{ (ver Figura 4.28)}$$

$$C' = \emptyset$$

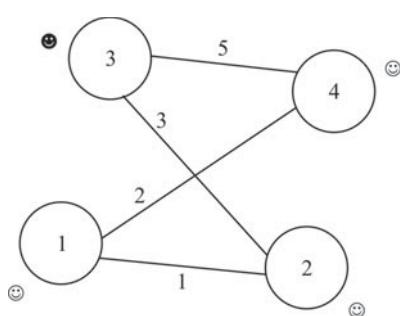
$$LST = 3 + 3 = 6$$

$$ST = \{(1, 2), (1, 4), (2, 3)\}$$

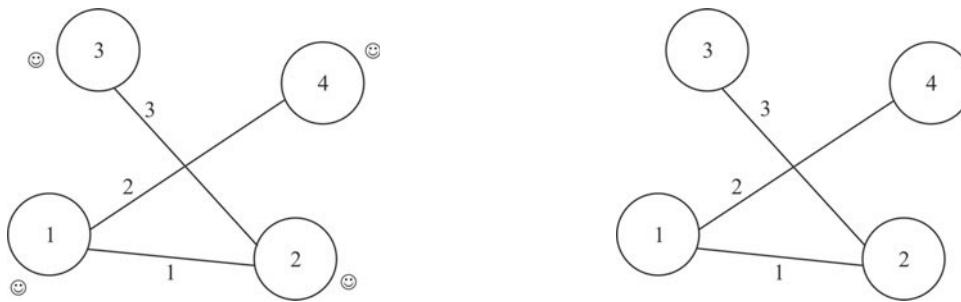
A árvore geradora mínima do grafo da Figura 4.24 é  $ST = \{(1, 2), (1, 4), (2, 3)\}$ , ilustrado na Figura 4.28, com comprimento total (soma dos comprimentos das arestas na árvore) igual a 6.

#### 4.2.4 Problemas de fluxo

Vários outros problemas de otimização em redes envolvem o transporte ou fluxo de algum item ou itens de um nó a outro na rede com um determinado objetivo. Problemas de fluxo podem ser



**Figura 4.27** Nô 4 é o mais próximo dos nós em  $C$ .



**Figura 4.28** Nô 3 é o mais próximo dos nós em  $C$  e árvore geradora mínima do grafo da Figura 4.24.

encontrados em diversas situações práticas, por exemplo, na transmissão de mensagens em redes de comunicação de dados, no envio de água em uma rede de distribuição de água, no transporte de carga em uma rede viária etc.

Considere uma rede  $G(N, E)$ . Problemas de fluxo em rede normalmente apresentam as seguintes características:

- Um nível máximo de fluxo tolerado para cada arco do grafo.
- Um nível mínimo de fluxo imprescindível para cada arco do grafo.
- Um custo por unidade de fluxo enviada para cada arco do grafo.
- Demanda ou produção (suprimento) do item (bem ou produto) em cada nó do grafo.

No caso de o objetivo ser a minimização de custos, o problema é conhecido como *problema de fluxo de custo mínimo*, que pode ser modelado como um problema de otimização linear. As variáveis de decisão são:

$x_{ij}$  fluxo (ou quantidade de itens) no arco  $(i, j)$

e os parâmetros:

$c_{ij}$  custo por unidade de fluxo no arco  $(i, j)$

$d_i$  demanda (se negativo; a rigor,  $-d_i$  é a demanda) ou suprimento (se positivo) de itens no nó  $i$

$u_{ij}$  limite máximo de fluxo no arco  $(i, j)$

$\ell_{ij}$  limite inferior, valor mínimo de fluxo no arco  $(i, j)$

Em cada nó da rede (ver Figura 4.29), o fluxo de material deve ser conservado, ou seja, o fluxo total que entra no nó  $i$  (isto é,  $\sum_{k \in P(i)} x_{ki}$ ), mais a oferta (se  $d_i > 0$ ) ou menos a demanda (se  $d_i < 0$ ), é igual ao fluxo total que sai do nó  $i$  (isto é,  $\sum_{j \in S(i)} x_{ij}$ ), ou seja,

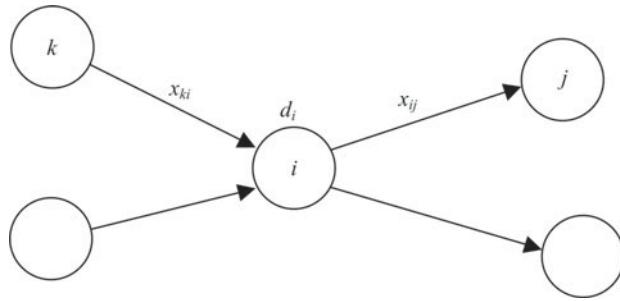
$$\sum_{k \in P(i)} x_{ki} + d_i = \sum_{j \in S(i)} x_{ij}$$

em que, como já definido anteriormente,

$S(i)$  é o conjunto dos nós sucessores de  $i$ , ou seja,  $S(i) = \{j \in N, \text{ tal que } (i, j) \in E\}$

$P(i)$  é o conjunto de nós predecessores de  $i$ , ou seja,  $P(i) = \{k \in N, \text{ tal que } (k, i) \in E\}$

Assim, o problema de fluxo a custo mínimo em uma rede de  $n$  nós pode ser modelado da seguinte forma:



**Figura 4.29** Conservação do fluxo no nó  $i$ .

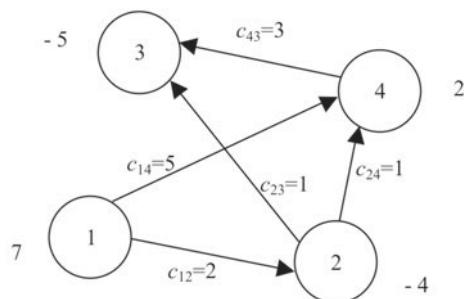
$$\text{Minimizar } f(\mathbf{x}) = \sum_{(i,j) \in E} c_{ij} x_{ij} \quad (4.1)$$

$$\sum_{j \in S(i)} x_{ij} - \sum_{k \in P(i)} x_{ki} = d_i \quad i = 1, 2, \dots, n \quad (4.2)$$

$$\ell_{ij} \leq x_{ij} \leq u_{ij} \quad \text{para todo } (i,j) \in E \quad (4.3)$$

A função  $f(\mathbf{x})$  em (4.1) nos dá o custo total referente aos fluxos por todos os arcos do grafo. As restrições em (4.2) asseguram a conservação de fluxo em todos os nós, como explicado anteriormente. As restrições em (4.3) limitam os fluxos nos arcos entre os valores mínimos e suas capacidades máximas. Podemos considerar, sem perda de generalidade, que  $\ell_{ij} = 0$  para todo arco  $(i, j)$ , uma vez que caso este valor seja diferente de zero, podemos simplesmente realizar uma mudança de variável e considerar uma nova variável  $y_{ij}$ , em que  $y_{ij} = x_{ij} - \ell_{ij}$ , de modo que  $0 \leq y_{ij} \leq u_{ij} - \ell_{ij}$ . Com isso, obtemos um modelo equivalente e com a mesma forma que o anterior, porém, o limitante inferior  $\ell_{ij}$  para a variável  $y_{ij}$  é, agora, igual a 0 para todo  $i, j$ . Note que o modelo (4.1)-(4.3) pode ser visto como um caso particular do modelo (LF16)-(LF19) da Seção 3.5.2 em que foram considerados custos fixos associados ao uso de um arco.

Cabe observar que, se colocarmos as restrições (4.2) na forma matricial  $\mathbf{Ax} = \mathbf{d}$ , a matriz  $\mathbf{A}$  é a matriz de incidência nó-arco do grafo (ver Seção 4.1, Definição 4.10). O exemplo a seguir ilustra as restrições (4.2) na forma matricial. Considere o grafo da Figura 4.30, que é idêntico ao da



**Figura 4.30** Grafo exemplo de problema de fluxo.

Figura 4.2, em que admitimos que existam suprimentos e/ou demandas nos nós, conforme valores indicados na figura.

As restrições (4.2) para este problema ficam:

$$\begin{array}{lll} \text{nó (1):} & x_{12} + x_{14} & = 7 \\ \text{nó (2):} & -x_{12} + x_{23} + x_{24} & = -4 \\ \text{nó (3):} & -x_{23} - x_{43} & = -5 \\ \text{nó (4):} & -x_{14} - x_{24} + x_{43} & = 2 \end{array}$$

ou

$$\left[ \begin{array}{ccccc} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & 0 & -1 & 1 \end{array} \right] \begin{bmatrix} x_{12} \\ x_{14} \\ x_{23} \\ x_{24} \\ x_{43} \end{bmatrix} = \begin{bmatrix} 7 \\ -4 \\ -5 \\ 2 \end{bmatrix}$$

A matriz

$$\mathbf{A} = \left[ \begin{array}{ccccc} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & 0 & -1 & 1 \end{array} \right]$$

é uma matriz de incidência nó-arco do grafo da Figura 4.30, que é a mesma encontrada no Exemplo 4.5.

Em alguns problemas de otimização em redes, deparamos com limitações de fluxo nos nós, juntamente com as limitações nos arcos. Nestes casos, podemos reescrever o problema em um outro equivalente, no qual as limitações de fluxo estão somente nos arcos. Para isso, pegue cada nó  $i$  com limitações de fluxo no grafo original e substitua-o por dois nós  $i_1$  e  $i_2$ . No novo grafo, o arco  $(k, i)$  do grafo original é substituído por um arco  $(k, i_1)$ , exatamente com os mesmos parâmetros de custos, limitações de capacidade etc., o arco  $(i, f)$  do grafo original é substituído por um arco  $(i_2, f)$ , exatamente com os mesmos parâmetros de custos, limitações de capacidade etc., e adicionamos um arco  $(i_1, i_2)$  com custo 0 e com as limitações de capacidade do nó  $i$  (ver Figura 4.31). Observe no novo grafo que o arco  $(i_1, i_2)$  restringe o fluxo do nó  $i$  conforme desejado. Assim, podemos desenvolver métodos de solução para problemas de otimização em redes apenas com limitações de capacidade nos arcos, sem perda de generalidade.

### O problema de caminho mínimo

O problema de caminho mais curto entre dois nós  $i$  e  $j$  em um grafo, discutido na Seção 4.2.1, é um caso particular do problema de fluxo a custo mínimo em uma rede em que se deseja enviar uma unidade de um produto do nó 1 ao nó  $n$  e  $c_{ij}$  é o comprimento do arco  $(i, j)$ . A formulação matemática deste problema foi apresentada na Seção 4.2.1. Neste modelo, se alterarmos o objetivo para maximização, temos a formulação matemática do problema de caminho máximo entre os nós 1 e  $n$  discutido na Seção 4.2.2.

A formulação matemática do problema do caminho mais curto do nó 1 a todos os demais nós da rede também é um caso particular do problema de fluxo. Este problema pode ser formulado considerando o problema de fluxo em que se tem um suprimento de exatamente  $n - 1$  unidades no nó 1 e uma demanda de exatamente 1 unidade em cada um dos demais nós. O modelo fica:

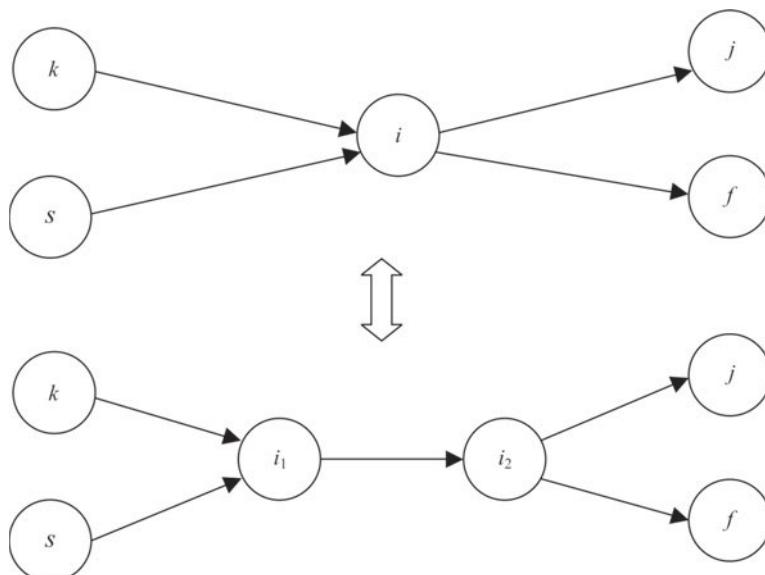
$$\begin{aligned}
 \text{Minimizar } f(\mathbf{x}) = & \sum_{(i,j) \in E} c_{ij} x_{ij} \\
 \sum_{j \in S(1)} x_{1j} - \sum_{k \in P(1)} x_{k1} &= n-1 \\
 \sum_{s \in S(j)} x_{js} - \sum_{k \in P(j)} x_{kj} &= -1, \quad j = 2, \dots, n \\
 x_{ij} \geq 0 & \quad \text{para todo } (i, j) \in E
 \end{aligned}$$

Os algoritmos para a resolução desses problemas foram apresentados nas Seções 4.2.1 e 4.2.2.

### O problema de fluxo máximo

Um outro problema muito comum de fluxo é o *problema de fluxo máximo*, que consiste em determinar o valor do maior fluxo possível que pode ser enviado de um nó a outro da rede. O problema de fluxo máximo aparece em uma empresa, por exemplo, quando se deseja determinar qual a capacidade máxima de produção de um determinado produto. O produto pode ser fabricado utilizando-se roteiros diferentes, que podem passar por diferentes centros de fabricação, cada um deles com uma certa capacidade instalada.

Sem perda de generalidade, consideramos o nó 1 como o nó origem (ou nó fonte), e o nó  $n$  como o nó para o qual se deseja enviar o fluxo máximo (o nó sorvedouro). Se denotamos por  $y$  a quantidade do produto que está sendo enviado do nó 1 ao nó  $n$ , o modelo de otimização linear para este problema é dado por:



**Figura 4.31** Construção de grafo equivalente com limitações de fluxo apenas nos arcos.

Maximizar  $y$

$$\sum_{j \in S(1)} x_{1j} - \sum_{k \in P(1)} x_{k1} = y \quad (\text{nó fonte } 1)$$

$$\sum_{j \in S(i)} x_{ij} - \sum_{k \in P(i)} x_{ki} = 0 \quad i = 2, 3, \dots, n-1$$

$$\sum_{j \in S(n)} x_{nj} - \sum_{k \in P(n)} x_{kn} = -y \quad (\text{nó sorvedouro } n)$$

$$0 \leq x_{ij} \leq u_{ij} \quad \text{para todo } (i, j) \in E$$

As equações do modelo anterior podem ser escritas com todas as variáveis no lado esquerdo:

$$\begin{aligned} \sum_{j \in S(1)} x_{1j} - \sum_{k \in P(1)} x_{k1} - y &= 0 \\ \sum_{j \in S(i)} x_{ij} - \sum_{k \in P(i)} x_{ki} &= 0 \quad i = 2, 3, \dots, n-1 \\ \sum_{j \in S(n)} x_{nj} - \sum_{k \in P(n)} x_{kn} + y &= 0 \end{aligned}$$

do que concluímos que a coluna da variável fluxo  $y$  pode ser vista como associada a um arco  $(n, 1)$ , chamado *arco de retorno*. O arco  $(n, 1)$  passa a pertencer ao grafo e basta redefinir  $S(n) = S(n) + \{1\}$ . O problema de fluxo máximo pode ser equivalentemente escrito com se segue.

Maximizar  $x_{n1}$

$$\sum_{j \in S(i)} x_{ij} - \sum_{k \in P(i)} x_{ki} = 0 \quad i = 1, 2, 3, \dots, n$$

$$0 \leq x_{ij} \leq u_{ij} \quad \text{para todo } (i, j) \in E$$

*O algoritmo de Ford e Fulkerson para resolução do problema de fluxo máximo*

Um algoritmo simples para resolução do problema de fluxo máximo tenta construir uma solução melhor para o problema a partir de uma solução já existente. Sem perda de generalidade, admitimos que o nó 1 é o nó fonte e o nó  $n$  é o nó sorvedouro do grafo, e que todos os arcos no grafo têm capacidade  $u_{ij} > 0$ . Os arcos com  $u_{ij} = 0$  obrigam que  $x_{ij} = 0$  para toda solução factível, portanto, podem ser eliminados do problema.

Suponha que temos disponível alguma solução factível com um fluxo  $\mathbf{x}^*$  indo do nó 1 ao nó  $n$  (por exemplo,  $\mathbf{x}^* = \mathbf{0}$ ). Para melhorar essa solução  $\mathbf{x}^*$ , é preciso encontrar algum caminho no grafo do nó 1 ao nó  $n$  por onde o fluxo atual pode ser incrementado. Para determinar caminhos por onde o fluxo pode ser incrementado, é conveniente classificar os arcos do grafo em duas classes:

Classe A: arcos que podem ter seu fluxo aumentado (isto é,  $x_{ij}^* < u_{ij}$ )

Classe B: arcos que podem ter seu fluxo diminuído (isto é,  $0 < x_{ij}^*$ )

Com isso, podemos construir um grafo auxiliar com os mesmos nós do grafo original e arcos definidos por:

- Para cada arco  $(i, j)$  classificado na classe A, temos um arco  $(i, j)$  no grafo auxiliar com capacidade  $(u_{ij} - x_{ij}^*)$ , ou seja, o fluxo neste arco no grafo auxiliar pode ser aumentado de até  $(u_{ij} - x_{ij}^*)$ .

- Para cada arco  $(i, j)$  classificado na classe B, temos um arco  $(j, i)$  no grafo auxiliar com capacidade  $x_{ij}^*$ , ou seja, o fluxo neste arco pode ser diminuído de até  $x_{ij}^*$ .

Um arco  $(i, j)$  pode estar ao mesmo tempo nas classes A e B. Isso ocorre quando  $0 < x_{ij}^* < u_{ij}$ , de modo que, neste caso, dois arcos,  $(i, j)$  e  $(j, i)$ , são criados no grafo auxiliar.

Suponha que neste grafo auxiliar existe um caminho do nó 1 ao nó  $n$ , como ilustrado na Figura 4.32, em que os números ao lado de cada arco indicam a capacidade máxima.

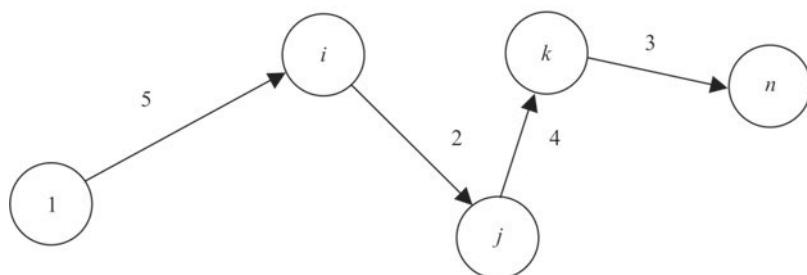
Neste caminho do grafo auxiliar, temos que o fluxo no arco  $(1, i)$  pode aumentar de cinco unidades, no arco  $(i, j)$  o fluxo pode aumentar de duas unidades, no arco  $(j, k)$  o fluxo pode aumentar de quatro unidades e, no arco  $(k, n)$  o fluxo pode aumentar de três unidades. Assim, é possível aumentar o fluxo do nó 1 ao nó  $n$  da menor das capacidades-limite dos arcos deste caminho, ou seja, duas unidades, sem que com isso se viole qualquer restrição do problema.

Suponha que esse grafo auxiliar tenha sido gerado a partir dos arcos de parte do grafo original, ilustrados na Figura 4.33. O possível aumento do fluxo descoberto a partir do grafo auxiliar da Figura 4.33 significa que no grafo original podemos incrementar de duas unidades o fluxo de 1 a  $n$ , aumentando o fluxo nos arcos  $(1, i)$ ,  $(i, j)$ ,  $(k, n)$  e diminuindo o fluxo de duas unidades no arco  $(k, j)$ .

Basicamente, o procedimento para construir soluções melhores a partir de soluções anteriores para o problema gera, a partir de um fluxo dado, o grafo auxiliar correspondente. Neste grafo auxiliar, procura-se algum caminho do nó 1 ao nó  $n$ . Determina-se o mínimo das capacidades dos arcos neste caminho (cadeia no grafo original) que nos dá o aumento possível de fluxo utilizando apenas os arcos daquela cadeia e sem que nenhuma das restrições de capacidade do problema seja violada. Aplica-se iterativamente este procedimento até que nenhum caminho do nó 1 ao nó  $n$  seja encontrado no grafo auxiliar.

Não é preciso construir explicitamente o grafo auxiliar e pode-se utilizar o grafo original, no qual se busca uma cadeia (em vez de um caminho) do nó 1 ao nó  $n$ , mas tendo em mente as características do grafo auxiliar apresentado anteriormente. A procura por uma cadeia no grafo original pode ser feita rotulando-se nós de maneira sucessiva, a partir do nó 1. Um nó  $j$  é rotulado se existir algum nó  $i$  já rotulado e existir um arco  $(i, j)$ , com a condição de que o fluxo no arco  $(i, j)$  ainda possa ser aumentado (isto é, o arco  $(i, j)$  pertence à classe A), ou se existir um arco  $(j, i)$  com a condição de que o fluxo no arco  $(j, i)$  possa ser diminuído (o arco  $(j, i)$  pertence à classe B). Rotulando sucessivamente os nós a partir do nó 1, se em algum momento rotularmos o nó  $n$ , teremos achado uma cadeia do nó 1 ao nó  $n$  por onde o fluxo pode ser aumentado.

Para recuperar essa cadeia descoberta, armazenamos, no momento que um nó  $j$  é rotulado, o nó  $i$  a partir do qual o nó  $j$  foi rotulado. Guardamos este nó  $i$  em  $p(j)$ . Armazenamos também no



**Figura 4.32** Um caminho do nó 1 ao nó  $n$ .

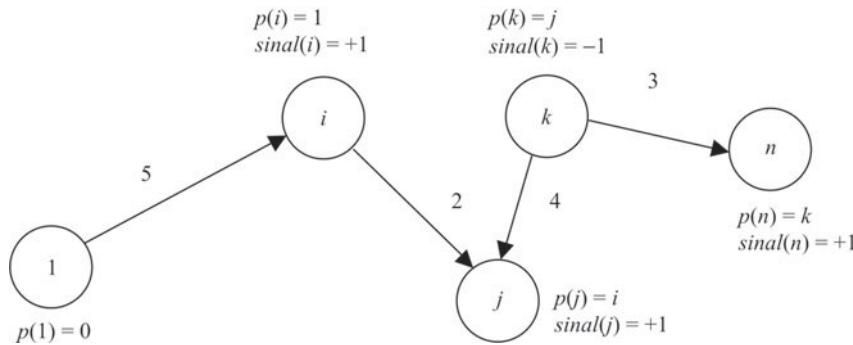


Figura 4.33 Subgrafo original.

nó  $j$  se o fluxo será aumentado ou diminuído no arco, ou seja, se é um arco  $(i, j)$  ou  $(j, i)$ , respectivamente. Isso é guardado em  $sinal(j)$ , que é feito igual a  $+1$  no caso de arco  $(i, j)$  e  $-1$  no caso de  $(j, i)$ . Na Figura 4.33, os valores de  $p(j)$  e  $sinal(j)$  para cada um dos nós da cadeia estão indicados. Para o nó 1, fazemos  $p(1) = 0$ , indicando que nenhum nó o precede.

As idéias apresentadas anteriormente são as bases para o algoritmo de Ford e Fulkerson para o problema de fluxo máximo. Sem perda de generalidade, na apresentação do algoritmo, consideramos o nó 1 como o nó fonte e o nó  $n$  como o nó sorvedouro.

#### Algoritmo de Ford e Fulkerson

*Dados:*

$G(N, E)$ , em que  $N = \{1, 2, \dots, n\}$

$u(i, j)$  capacidade máxima do arco  $(i, j)$  ( $u(i, j) > 0$ )

1 nó fonte

$n$  nó sorvedouro

*Saída:*

Fluxo máximo  $y$  do nó 1 ao nó  $n$

*Passo 1: Início*

$$y = 0 \text{ (fluxo inicial } \mathbf{x}^* = \mathbf{0})$$

$$v^+(i, j) = u(i, j) \text{ para todos os arcos } (i, j) \in E$$

$$v^-(i, j) = 0 \text{ para todos os arcos } (i, j) \in E$$

$R = \{n\}$ : conjunto de nós rotulados (artifício para que a primeira iteração a seguir seja executada. A rigor, nenhum nó está rotulado inicialmente).

*Passo 2: Enquanto  $n \in R$ , faça*

Início\_Enquanto

$$R \leftarrow \emptyset \text{ (nenhum nó está rotulado inicialmente)}$$

$$p(i) = 0 \text{ para } i \in N \text{ (}p(i)\text{ fornece o predecessor do nó na cadeia)}$$

$$R = \{1\} \text{ e } Lista = \{1\}$$

Enquanto (*Lista*  $\neq \emptyset$ ) ou ( $n \notin R$ ), faça

Início\_enquanto

- Escolha  $i \in \text{Lista}$  e faça  $\text{Lista} \leftarrow \text{Lista} - \{i\}$
- Para todo arco  $(i, j)$  com  $(v^+(i, j) > 0)$  e  $(j \notin R)$ , faça
  - $p(j) = i$
  - $sinal(j) = +1$
  - $R \leftarrow R \cup \{j\}$
  - $\text{Lista} \leftarrow \text{Lista} \cup \{j\}$
- Para todo arco  $(j, i)$  com  $(v^-(j, i) > 0)$  e  $(j \notin R)$ , faça
  - $p(j) = i$
  - $sinal(j) = -1$
  - $R \leftarrow R \cup \{j\}$
  - $\text{Lista} \leftarrow \text{Lista} \cup \{j\}$

Fim\_enquanto

Se ( $n \in R$ ), então execute o procedimento *aumente\_o\_fluxo\_e\_atualize\_fluxo\_nos\_arcos*

Fim\_Enquanto

Procedimento *aumente\_o\_fluxo\_e\_atualize\_fluxo\_nos\_arcos*

Dados:

$G(N, E)$ : grafo em que  $N = \{1, 2, \dots, n\}$

$v^+(i, j)$  capacidade máxima de aumento de fluxo no arco  $(i, j)$

$v^-(i, j)$  capacidade máxima de diminuição de fluxo no arco  $(i, j)$

$1$  nó fonte

$n$  nó sorvedouro

$y$  fluxo atual

$p(i)$  nó a partir do qual o nó  $i$  foi rotulado

$sinal(i)$  se igual a  $+1$ , indica que o arco que deve ser recuperado para determinar a cadeia  $C$  será da forma  $(p(i), i)$ ; se igual a  $-1$ , indica que o arco que deve ser recuperado para determinar a cadeia  $C$  será da forma  $(i, p(i))$ .

Saída:

Fluxo aumentado  $y$  do nó 1 ao nó  $n$

$v^+(i, j)$  valor atualizado da capacidade máxima de aumento de fluxo no arco  $(i, j)$

$v^-(i, j)$  valor atualizado da capacidade máxima de diminuição de fluxo no arco  $(i, j)$

Início\_procedimento

Passo 1: (Como o nó  $n$  está rotulado ( $n \in R$ ), então existe uma cadeia ligando o nó 1 ao nó  $n$ , a qual pode ser construída a partir dos vetores  $p(.)$  e  $sinal(.)$ . O máximo de fluxo nessa cadeia é calculado durante a busca pela cadeia).

$r = n$

$C = \emptyset$

$\delta = +\infty$

Enquanto  $r \neq 1$ , faça

Início\_Enquanto

Se  $sinal(r) = +1$  então  $C \leftarrow C \cup \{(p(r), r)\}$  e  $\delta = \min\{\delta, v^+(p(r), r)\}$   
 senão  $C \leftarrow C \cup \{(r, p(r))\}$  e  $\delta = \min\{\delta, v^-(r, p(r))\}$

$r \leftarrow p(r)$

Fim\_Enquanto

*Passo 2:*  $y \leftarrow y + \delta$

Para todo  $(i, j) \in C$ , se  $sinal(j) = +1$ , então

$v^+(i, j) \leftarrow v^+(i, j) - \delta$

$v^-(i, j) \leftarrow v^-(i, j) + \delta$

se não (isto é,  $sinal(j) = -1$ )

$v^-(i, j) \leftarrow v^-(i, j) - \delta$

$v^+(i, j) \leftarrow v^+(i, j) + \delta$

Fim\_procedimento

**Exemplo 4.21** Ilustramos a aplicação do algoritmo de Ford e Fulkerson para o grafo da Figura 4.34. Desejamos determinar qual é o maior fluxo que se pode enviar do nó 1 ao nó 6 neste grafo.

*Passo 1:* Início

$y = 0$

$v^+(1, 5) = 4, v^+(1, 2) = 10, v^+(1, 3) = 5, v^+(5, 3) = 1, v^+(5, 4) = 5$

$v^+(2, 3) = 7, v^+(2, 4) = 4, v^+(2, 6) = 3, v^+(3, 4) = 2, v^+(4, 6) = 2$

$v^-(1, 5) = 0, v^-(1, 2) = 0, v^-(1, 3) = 0, v^-(5, 3) = 0, v^-(5, 4) = 0$

$v^-(2, 3) = 0, v^-(2, 4) = 0, v^-(2, 6) = 0, v^-(3, 4) = 0, v^-(4, 6) = 0$

$R = \{6\}$

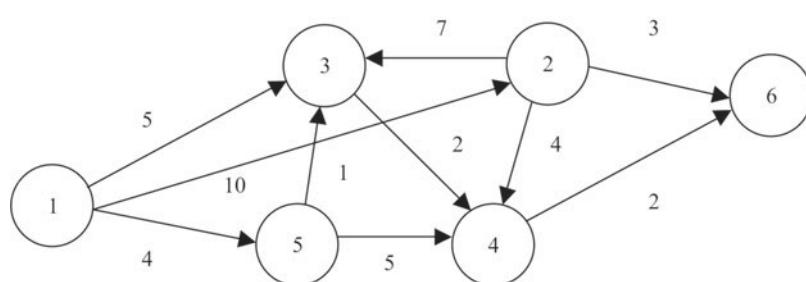
A Figura 4.35 apresenta esses valores nos arcos dos grafos.

*Passo 2:*

$R \leftarrow \emptyset$

$p(1) = 0, p(5) = 0, p(2) = 0, p(3) = 0, p(4) = 0, p(6) = 0$

$R = \{1\}$  e  $Lista = \{1\}$



**Figura 4.34** Grafo ilustrativo para aplicação do algoritmo de Ford e Fulkerson.

```

Lista = ∅
p(5) = 1
sinal(5) = +1
R = {1, 5}
Lista = {5}
p(2) = 1
sinal(2) = +1
R = {1, 5, 2}
Lista = {5, 2}
p(3) = 1
sinal(3) = +1
R = {1, 5, 2, 3}
Lista = {5, 2, 3}
Lista = {2, 3}
p(4) = 5
sinal(4) = +1
R = {5, 1, 2, 3, 4}
Lista = {2, 3, 4}
Lista = {3, 4}
p(6) = 2
sinal(6) = +1
R = {5, 1, 2, 3, 4, 6}
Lista = {3, 4, 6}

```

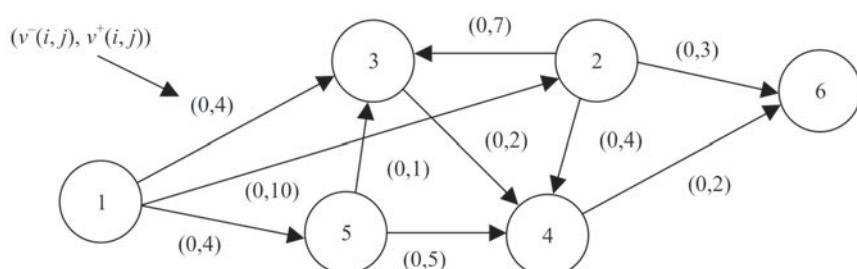
Procedimento aumente o fluxo e atualize os valores de variação de fluxo nos arcos

Início

Passo 1:

$p(6) = 2, sinal(6) = +1$   
 $p(2) = 1, sinal(2) = +1$

---



**Figura 4.35** Grafo da Figura 4.34 com os limites iniciais de  $v(i, j)$  e  $v^+(i, j)$ .

Cadeia  $C = \{(2, 6), (1, 2)\}$

Passo 2:

$$\delta = \min\{v^+(2, 6), v^+(1, 2)\} = 3$$

$$y = 0 + 3 = 3$$

$$v^+(1, 2) = 10 - 3 = 7$$

$$v^-(1, 2) = 0 + 3 = 3$$

$$v^+(2, 6) = 3 - 3 = 0$$

$$v^-(2, 6) = 0 + 3 = 3$$

Fim

$$R = \{1, 5, 2, 3, 6, 4\}$$

$$R \leftarrow \emptyset$$

$$p(1) = 0, p(5) = 0, p(2) = 0, p(3) = 0, p(4) = 0, p(6) = 0$$

$$R = \{1\} \text{ e } Lista = \{1\}$$

$$Lista = \emptyset$$

$$p(5) = 1$$

$$sinal(5) = +1$$

$$R = \{1, 5\}$$

$$Lista = \{5\}$$

$$p(2) = 1$$

$$sinal(2) = +1$$

$$R = \{1, 5, 2\}$$

$$Lista = \{5, 2\}$$

$$p(3) = 1$$

$$sinal(3) = +1$$

$$R = \{1, 5, 2, 3\}$$

$$Lista = \{5, 2, 3\}$$

$$Lista = \{2, 3\}$$

$$p(4) = 5$$

$$sinal(4) = +1$$

$$R = \{1, 5, 2, 3, 4\}$$

$$Lista = \{2, 3, 4\}$$

$$Lista = \{2, 3\}$$

$$p(6) = 4$$

$$sinal(6) = +1$$

$$R = \{1, 5, 2, 3, 4, 6\}$$

$$Lista = \{2, 3, 6\}$$

Procedimento aumente o fluxo e atualize os valores de variação de fluxo nos arcos

Início

Passo 1:

$$p(6) = 4, sinal(6) = +1$$

$$p(4) = 5, sinal(4) = +1$$

$$p(5) = 1, sinal(5) = +1$$

Cadeia  $C = \{(1, 5), (5, 4), (4, 6)\}$

*Passo 2:*

$$\delta = \min \{v^+(1, 5), v^+(5, 4), v^+(4, 6)\} = 2$$

$$y = 3 + 2 = 5$$

$$v^+(1, 5) = 4 - 2 = 2$$

$$v^-(1, 5) = 0 + 2 = 2$$

$$v^+(5, 4) = 5 - 2 = 3$$

$$v^-(5, 4) = 0 + 2 = 2$$

$$v^+(4, 6) = 2 - 2 = 0$$

$$v^-(4, 6) = 0 + 2 = 2$$

Fim

$$R = \{1, 5, 2, 3, 4, 6\}$$

$$R \leftarrow \emptyset$$

$$p(1) = 0, p(5) = 0, p(2) = 0, p(3) = 0, p(4) = 0, p(6) = 0$$

$$R = \{1\} \text{ e } Lista = \{1\}$$

$$Lista = \emptyset$$

$$p(5) = 1$$

$$sinal(5) = +1$$

$$R = \{1, 5\}$$

$$Lista = \{5\}$$

$$p(2) = 1$$

$$sinal(2) = +1$$

$$R = \{1, 5, 2\}$$

$$Lista = \{5, 2\}$$

$$p(3) = 1$$

$$sinal(3) = +1$$

$$R = \{1, 5, 2, 3\}$$

$$Lista = \{5, 2, 3\}$$

$$Lista = \{2, 3\}$$

$$p(4) = 5$$

$$sinal(4) = +1$$

$$R = \{1, 5, 2, 3, 4\}$$

$$Lista = \{2, 3, 4\}$$

$$Lista = \{3, 4\}$$

$$Lista = \{4\}$$

$$Lista = \emptyset$$

Fim

O fluxo máximo encontrado do nó 1 ao nó 6 é  $y = 5$ . ■

No algoritmo de Ford e Fulkerson, o término ocorre quando, após algumas iterações, não se consegue mais rotular o nó  $n$ , ou seja, não se encontra mais nenhuma cadeia do nó 1 ao nó  $n$ .

Isso quer dizer que não se pode mais aumentar o fluxo em quaisquer dos arcos que tenham os nós caudas rotulados e os nós cabeças não rotulados. Esta observação relaciona o fluxo máximo ao conceito de corte em grafos.

**Definição 4.11 (corte)** Um corte no grafo  $G(N, E)$  é uma partição do conjunto de nós  $N$  em dois subconjuntos  $R$  e  $N - R$ , o qual denotamos por  $[R, N - R]$ .

Cada corte define um conjunto de arcos que tem uma de suas extremidades (cabeça ou cauda) no conjunto  $R$  e a outra no conjunto  $N - R$  (cauda ou cabeça). Alternativamente, este conjunto de arcos também define um corte, uma vez que, se excluirmos todos esses arcos de  $G(N, E)$ , o grafo se divide em dois grafos desconexos, o primeiro com o conjunto  $R$  de nós e o segundo com o conjunto  $N - R$  de nós.

**Exemplo 4.22** Considere o grafo da Figura 4.34. Seja  $R = \{1, 5, 3\}$ ; consequentemente,  $N - R = \{2, 4, 6\}$ . Essa partição é um corte deste grafo. O conjunto de arcos desse corte  $[R, N - R]$  é  $\{(5, 4), (1, 2), (3, 4), (2, 3)\}$ . Observe que, se excluirmos do grafo da Figura 4.34 apenas alguns desses arcos, mas não todos, o grafo não se desconecta. ■

**Definição 4.12 (corte  $1 - n$ )** Um corte  $[R, N - R]$  é chamado  $1 - n$  se  $1 \in R$  e  $n \in N - R$ , em que o nó  $1$  é o nó fonte e o nó  $n$  o nó sorvedouro.

Um corte  $1 - n$  divide o grafo em dois subgrafos em que o nó  $1$  (fonte) está em um subgrafo e o nó  $n$  (sorvedouro) no outro. O corte do Exemplo 4.22 é um corte  $1 - 6$ .

Seja  $(R, N - R)$  o subconjunto de arcos de  $E$  dado por  $(R, N - R) = \{(i, j) \in E \text{ tal que } i \in R \text{ e } j \notin R\}$ , isto é, os arcos no corte orientados de  $R$  para  $N - R$ . A capacidade de um corte  $1 - n$   $[R, N - R]$  é dada por:

$$U[R, N - R] = \sum_{(i, j) \in (R, N - R)} u_{ij}$$

isto é, a soma das capacidades dos arcos no corte orientados de  $R$  para  $N - R$ . Por exemplo, o corte  $1 - 6$  do Exemplo 4.22 tem capacidade  $U[\{1, 5, 3\}, \{2, 4, 6\}] = u_{12} + u_{54} + u_{34} = 17$ .

Seja  $[R, N - R]$  um corte  $1 - n$ . Qualquer fluxo do nó  $1$  para o nó  $n$  tem de passar por pelo menos um arco de  $(R, N - R)$ . Desta forma, vale a seguinte propriedade.

**Propriedade 4.2** Seja  $R \subseteq N$  tal que  $1 \in R$  e  $n \notin R$ . Então, o fluxo máximo factível de  $1$  a  $n$  neste grafo é menor ou igual a  $U[R, N - R]$ .

Observe que, se todos os arcos em  $(R, N - R)$  forem excluídos do grafo  $G(N, E)$ , não haveria nenhum caminho de  $1$  para  $n$  e, portanto, o fluxo máximo de  $1$  para  $n$  seria nulo. A propriedade 4.2 é válida para qualquer corte que separa  $1$  de  $n$ , inclusive para o corte de capacidade mínima desse grafo. Portanto, o máximo fluxo de  $1$  a  $n$  tem como limitante superior a menor capacidade dentre os cortes que separam esses dois nós.

**Propriedade 4.3** O fluxo máximo de  $1$  a  $n$  no grafo  $G(N, E)$  é igual à menor capacidade de todos os cortes  $1 - n$  de  $G(N, E)$ .

O algoritmo de Ford e Fulkerson determina esse corte mínimo quando não consegue mais rotular o nó  $n$ . Ao final do algoritmo, temos um conjunto de nós rotulados que contém o nó  $1$  e um conjunto de nós não rotulados que contém o nó  $n$ , e não é possível aumentar o fluxo em nenhum dos arcos com cauda no conjunto rotulado e cabeça no conjunto não rotulado. Logo, todos esses arcos estão com o fluxo na sua capacidade máxima. Com o conjunto de nós rotulados e o con-

junto de nós não rotulados, identificamos facilmente os arcos desse corte no grafo que estão com o fluxo em sua capacidade máxima e que constituem um corte  $1 - n$  com capacidade mínima.

Por exemplo, no caso ilustrado anteriormente, o algoritmo terminou rotulando os nós  $\{1, 5, 2, 3, 4\}$ , ou seja,  $R = \{1, 2, 3, 4, 5\}$ . Portanto, o nó não rotulado é apenas o nó 6, ou seja,  $N - R = \{6\}$ . Os arcos em  $(\{1, 2, 3, 4, 5\}, \{6\})$  são, portanto,  $\{(2, 6), (4, 6)\}$ . A capacidade  $U[\{1, 2, 3, 4, 5\}, \{6\}]$  desse corte é  $3 + 2 = 5$ , exatamente o fluxo máximo encontrado.

### O problema de transporte

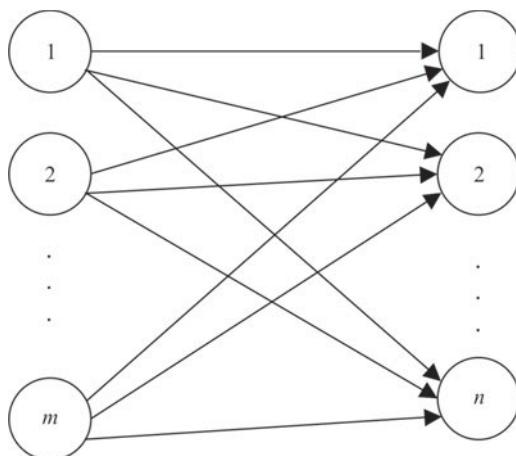
Um outro problema comum de fluxo é o *problema de transporte*, apresentado na Seção 2.2.2 do Capítulo 2. O problema de transporte é um caso particular do problema de fluxo em redes a um custo mínimo (4.1) – (4.3), no qual os nós do grafo são partitionados em dois grupos: o primeiro é formado apenas de nós caudas (chamados nós produtores) e o segundo apenas de nós cabeças (chamados nós consumidores), conforme a Figura 4.36. No problema de transporte (ver Figura 4.36), não existem nós *intermediários* no grafo, isto é, nós por onde apenas o fluxo de produtos passa (nós sem demanda ou suprimento de itens).

A formulação matemática do problema de transporte apresentado no Capítulo 2 é:

$$\begin{aligned} \text{Minimizar } f(\mathbf{x}) &= \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \sum_{j=1}^n x_{ij} &\leq a_i \quad i = 1, \dots, m \\ \sum_{i=1}^m x_{ij} &= b_j \quad j = 1, \dots, n \\ x_{ij} &\geq 0, \quad i = 1, \dots, m \text{ e } j = 1, \dots, n \end{aligned}$$

em que temos  $m$  origens e  $n$  destinos; o custo de transportar uma unidade da origem  $i$  para o destino  $j$  é  $c_{ij}$ ; a disponibilidade do produto na localidade  $i$  é  $a_i$ ; e a demanda no destino  $j$  é  $b_j$ .

Se  $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ , então tudo que se dispõe nas diversas origens terá de ser transportado aos destinos. Portanto, a desigualdade na formulação matemática pode ser trocada por uma igualdade, ficando o modelo em conformidade com a formulação geral do problema de fluxo apresentada



**Figura 4.36** Grafo do problema de transporte.

anteriormente. Caso o total da quantidade ofertada seja superior ao total da quantidade demandada (razão da desigualdade na formulação do problema de transporte apresentado), pode-se, sem perda de generalidade, introduzir um novo ponto destino fictício, nó  $n + 1$ , com demanda  $(\sum_{i=1}^m a_i - \sum_{j=1}^n b_j) > 0$ , com os custos dos diversos pontos de suprimento a este ponto destino fictício  $n + 1$  iguais a zero (isto é,  $c_{i(n+1)} = 0$  para  $i = 1, 2, \dots, m$ ).

Assim, temos assegurado, neste novo problema, que o total da quantidade ofertada é igual ao total da quantidade demandada. Portanto, a desigualdade nas restrições pode ser trocada por uma igualdade. A quantidade  $x_{i(n+1)} > 0$ , isto é, o fluxo do nó  $i$  para o nó fictício  $n + 1$  pode ser interpretada como um excesso de oferta no nó  $i$  que não é enviada e, portanto, permanece estocada no nó  $i$ .

Dada a estrutura especial do problema de transporte, o método simplex (da Seção 2.6) aplicado a este problema é simplificado limitando-se os cálculos a operações de subtração e adição. Um algoritmo especializado (baseado no simplex) para a resolução do problema de transporte pode ser encontrado em Bazaraa *et al.* (1990).

Quando existem nós *intermediários* no grafo, temos o chamado problema de transbordo, que também foi descrito na Seção 2.2.2 do Capítulo 2. O problema de transbordo também é um problema de fluxo em redes. Nos nós de transbordo, os produtos apenas passam, ou seja, a quantidade transportada destes nós aos demais nós do grafo é igual à quantidade de produto que chega a este nó. Matematicamente,  $\sum_i x_{ij} = \sum_k x_{jk}$ , para toda localidade  $j$  de transbordo. Essas restrições adicionadas à formulação matemática do problema estão em conformidade com a formulação geral do problema de fluxo em rede (4.1)–(4.3). Usando um artifício, é sempre possível transformar o problema do transbordo em um problema de transporte, conforme discutido em Bazaraa *et al.* (1990).

#### 4.3\* O MÉTODO SIMPLEX PARA REDES<sup>8</sup>

Como vimos, o modelo linear geral para o problema de fluxo a custo mínimo em uma rede é dado por (4.1)–(4.3) e a matriz  $\mathbf{A}$  associada ao conjunto de restrições (4.2) é uma matriz de incidência nó-arco do grafo, cada linha corresponde a um nó da rede e cada coluna a um arco da rede. Cada coluna correspondente a um arco  $(i, j)$  possui apenas dois coeficientes não nulos, um coeficiente igual a +1 na linha correspondente ao nó  $i$  e o outro igual a -1 na linha correspondente ao nó  $j$ . Portanto, a coluna  $k$ , correspondente ao arco  $(i, j)$  nessa matriz  $\mathbf{A}$ , é igual a  $\mathbf{e}_i - \mathbf{e}_j$ , em que  $\mathbf{e}_i$  corresponde a uma coluna de  $n$  elementos com todos eles iguais a zero, exceto o elemento  $i$ , que tem valor igual a 1.

Tomemos, por exemplo,  $\mathbf{A}$  como a matriz de incidência nó-arco  $\mathbf{P}$  do grafo da Figura 4.2, dada no Exemplo 4.4:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & 0 & -1 & 1 \end{bmatrix}$$

O arco  $(1, 4)$  do grafo da Figura 4.2, por exemplo, corresponde à segunda coluna de  $\mathbf{A}$ . Essa coluna, correspondente a esse arco e denotada por  $\mathbf{a}_{(1,4)}$ , é igual a

$$\mathbf{a}_{(1,4)} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{e}_1 - \mathbf{e}_4$$

<sup>8</sup> É aconselhável a leitura prévia das Seções 2.6.1 e 2.6.2.

Devido à particularidade da matriz  $\mathbf{A}$ , se somarmos todas as suas  $n$  linhas, temos como resultado uma linha de zeros, o que implica que o conjunto das  $n$  linhas desta matriz é linearmente dependente (uma das linhas pode ser escrita como uma combinação não-nula das demais  $n - 1$  linhas). Conseqüentemente, uma base da matriz  $\mathbf{A}$  não tem  $n$  colunas; uma base desta matriz  $\mathbf{A}$  poderá ter no máximo  $n - 1$  colunas (lembre-se que  $n$  é o número de nós do grafo, igual ao número de linhas da matriz de incidência nó-arcos).

Admitimos de início, sem perda de generalidade, que os grafos considerados neste capítulo são conectados. Uma árvore geradora de  $G$  é formada de  $(n - 1)$  arcos. Um grafo conectado  $G$  possui sempre uma árvore geradora. Para mostrar isso, basta eliminar sucessivamente arcos do grafo, de modo que o grafo resultante não se desconecte (sempre que houver ciclos isto é possível) e até que resultem apenas  $n - 1$  arcos, o que forma uma árvore.

Tomemos qualquer árvore geradora desse grafo  $G$ . Se pegarmos a submatriz  $\mathbf{B}$  de  $\mathbf{A}$  correspondente aos arcos desta árvore geradora,  $\mathbf{B}$  é uma matriz de tamanho  $n \times (n - 1)$ . Como os arcos de uma árvore geradora não formam ciclos, existe, portanto, pelo menos um nó nesta árvore com apenas um arco nele incidente. Se  $r_1$  for este nó, na linha correspondente a  $r_1$  da matriz  $\mathbf{B}$  temos apenas um elemento diferente de 0, ou seja,  $\mathbf{B}$  tem a forma

$$\mathbf{B} = \begin{bmatrix} \pm 1 & .. & 0 & .. & \pm 1 \\ : & .. & : & .. & : \\ 0 & .. & \pm 1 & .. & 0 \\ : & .. & : & .. & : \\ 0 & .. & \mp 1 & .. & \mp 1 \end{bmatrix} \quad \leftarrow r_1\text{-ésima linha com um elemento } \neq 0$$

$\downarrow$

$j\text{-ésima coluna}$

Com a notação utilizada  $\pm 1$ , queremos indicar que este elemento tem valor  $+1$  ou  $-1$ , dependendo do arco ter este nó correspondente à linha da matriz como nó cauda ou nó cabeça, respectivamente. Como o outro elemento não-nulo na mesma coluna da matriz tem o sinal trocado, ele está denotado por  $\mp 1$ .

Podemos, então, permutar as linhas e colunas de  $\mathbf{B}$  de modo a obter uma nova matriz  $\mathbf{B}^1$ , em que esse elemento diferente de zero fique posicionado em sua primeira linha e primeira coluna, ou seja,  $\mathbf{B}^1$  tem a forma

$$\mathbf{B}^1 = \begin{bmatrix} \pm 1 & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \ddots & \dots & \vdots \\ 0 & \pm 1 & \dots & \dots & \pm 1 \\ \vdots & \vdots & \ddots & \dots & \vdots \\ \mp 1 & 0 & \dots & \dots & \mp 1 \end{bmatrix}$$

$r_1$ -ésima linha foi para a 1ª linha

$j$ -ésima coluna foi para a 1ª coluna

Por conveniência, reescrevemos essa matriz na forma

$$\mathbf{B}^1 = \begin{bmatrix} \pm 1 & \mathbf{0} \\ s & \mathbf{B}^{1*} \end{bmatrix}$$

Se eliminarmos da árvore geradora o nó  $r_1$  e o arco incidente nele, o grafo resultante continua sendo uma árvore. A matriz nó-arco correspondente a esta nova árvore (com  $n - 1$  nós) é obtida simplesmente eliminando-se a primeira linha e a primeira coluna de  $\mathbf{B}^1$ . Portanto, essa matriz é dada por  $\mathbf{B}^{1*}$ .

Nessa nova árvore, similarmente ao exposto anteriormente, existe pelo menos um nó com apenas um arco que nele incide. Se  $r_2$  for este nó, na linha correspondente a  $r_2$  da matriz  $\mathbf{B}^{1*}$  temos apenas um elemento diferente de 0. Assim, podemos permutar as linhas e colunas de  $\mathbf{B}^{1*}$  de modo a obter uma matriz em que esse elemento diferente de zero esteja em sua primeira linha e primeira coluna. Na matriz inicial, essas permutações colocam esse elemento diferente de zero na segunda linha e segunda coluna, formando uma matriz  $\mathbf{B}^2$  com a forma:

$$\mathbf{B}^2 = \begin{bmatrix} \pm 1 & 0 & \mathbf{0} \\ s_1 & \pm 1 & \mathbf{0} \\ s_2 & p & \mathbf{B}^{2*} \end{bmatrix}$$

Prosseguindo de maneira similar, eliminando linhas e colunas, obtemos  $\mathbf{B}^{3*}$ ,  $\mathbf{B}^{4*}$  ... chegando finalmente a uma matriz  $\mathbf{B}^{(n-2)*}$  com a forma  $\begin{bmatrix} \pm 1 \\ \vdots \\ \pm 1 \end{bmatrix}$ . Ao realizarmos essas permutações de linhas e colunas na matriz  $\mathbf{B}$  em correspondência àquelas feitas nas diversas iterações anteriores, chegamos a uma matriz da forma

$$\mathbf{B}^{n-1} = \begin{bmatrix} \pm 1 & 0 & .. & 0 \\ s_1 & \pm 1 & .. & 0 \\ .. & .. & .. & 0 \\ s_{n-2} & p_{n-3} & .. & \pm 1 \\ s_{n-1} & p_{n-2} & .. & \mp 1 \end{bmatrix}$$

Se eliminarmos a última linha dessa matriz, obtemos uma matriz triangular inferior com uma diagonal com todos os elementos diferentes de 0. Essa matriz tem, portanto, determinante diferente de zero, ou seja, tem posto  $n - 1$  ou, em outras palavras, o conjunto formado pelas  $n - 1$  colunas é linearmente independente.

Para fins de ilustração, consideremos novamente a matriz de incidência nó-arco  $\mathbf{P}$  do grafo da Figura 4.2.

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & 0 & -1 & 1 \end{bmatrix}$$

Tomemos, por exemplo, a árvore geradora do grafo correspondente à matriz  $\mathbf{P}$  ilustrada na Figura 4.37. A submatriz  $\mathbf{B}$  de  $\mathbf{P}$  correspondente aos arcos dessa árvore geradora é

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & -1 \\ -1 & 0 & 1 \end{bmatrix}$$

Selecionemos um nó nessa árvore com apenas um arco nele incidente, o nó 1, por exemplo. Na linha correspondente ao nó 1 (linha 1), temos apenas um elemento diferente de 0, que já está na coluna 1. Portanto, não precisamos permutar linhas e colunas neste caso. A matriz  $\mathbf{B}^1$  é a própria matriz  $\mathbf{B}$ .

Eliminamos da árvore geradora o nó 1 (ver Figura 4.38) e o arco incidente nele (1, 4), e observamos que o grafo resultante continua sendo uma árvore. A matriz nó-arco correspondente a essa nova árvore é obtida simplesmente eliminando-se a primeira linha e a primeira coluna de  $\mathbf{B}^1$ .

De modo semelhante ao ocorrido anteriormente, existe pelo menos um nó nessa árvore com apenas um arco que nele incide, por exemplo, o nó 2. Na linha correspondente ao nó 2, temos apenas um elemento diferente de 0, que já está na linha 1 e coluna 1 da submatriz. Portanto, novamente não precisamos permutar linhas e colunas neste caso.

Eliminamos da árvore geradora o nó 2 (ver Figura 4.39) e o arco incidente nele (2, 3), e observamos que o grafo resultante continua sendo uma árvore. A matriz nó-arco correspondente a essa nova árvore é obtida simplesmente eliminando-se a primeira linha e a primeira coluna da submatriz.

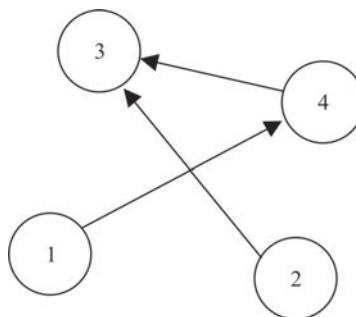
De modo semelhante ao ocorrido anteriormente, existe pelo menos um nó nessa árvore com apenas um arco que nele incide, por exemplo, o nó 2. Na linha correspondente ao nó 2, temos apenas um elemento diferente de 0, que já está na linha 1 e coluna 1 da submatriz. Portanto, novamente não precisamos permutar linhas e colunas neste caso e chegamos à submatriz

$$\mathbf{B}^{(n-2)*} = \mathbf{B}^{2*} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

Ao realizarmos as mesmas operações de permutação de linhas e colunas na matriz  $\mathbf{B}$ , em correspondência àquelas feitas nas diversas iterações anteriores (no caso, nenhuma permutação foi realizada) e eliminarmos a última linha desta matriz, obtemos uma matriz triangular inferior com a diagonal com todos os elementos diferentes de 0:

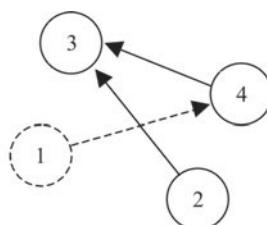
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & -1 \\ * & * & * \end{bmatrix}$$

Essa matriz tem determinante igual a  $-1$  (diferente de zero), ou seja, a matriz tem posto 3 ( $n - 1$ ). Conseqüentemente, as três colunas de  $\mathbf{B}$  formam um conjunto linearmente independente.



**Figura 4.37** Uma árvore geradora do grafo da Figura 4.2.

$$\begin{bmatrix} * & * & * \\ * & 1 & 0 \\ * & -1 & -1 \\ * & 0 & 1 \end{bmatrix}$$



**Figura 4.38** Eliminação do nó 1 da árvore geradora da Figura 4.37.

Vimos que a matriz de incidência nó-arco de um grafo conectado com  $n$  nós tem posto  $n - 1$ , ou, em outras palavras, no máximo  $n - 1$  colunas formam um conjunto linearmente independente. Por conveniência, adicionamos à matriz de incidência nó-arco do grafo mais uma coluna da matriz identidade, por exemplo,  $\mathbf{e}_1$ . Podemos imaginar que esta coluna representa mais um arco do grafo, cujo nó cabeça é algum lugar indefinido no espaço e cujo nó cauda é o nó 1 (no caso de adicionarmos a coluna  $\mathbf{e}_1$ ). Este nó, no qual esse arco artificial incide, é denominado *nó raiz*. Assim, se adicionamos a coluna  $\mathbf{e}_1$ , o nó raiz é o nó 1, se adicionamos a coluna  $\mathbf{e}_i$ , o nó raiz será o nó  $i$ . A inclusão desta nova coluna não afeta em nada a solução do problema, pois o fluxo nesse arco é sempre igual a zero e seu custo correspondente é nulo.

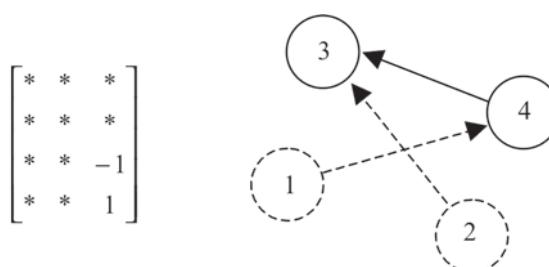
A nova matriz  $\mathbf{A}$  aumentada com mais essa coluna tem a vantagem de ter sempre  $n$  colunas linearmente independentes e, mais ainda, essa nova coluna está sempre entre as  $n$  colunas linearmente independentes. Se repetirmos o mesmo procedimento visto, pegando as colunas correspondentes a qualquer árvore geradora do grafo mais a coluna  $\mathbf{e}_1$  e realizarmos as mesmas operações para obtenção da matriz triangular inferior, verificamos novamente que o determinante desta matriz quadrada  $n \times n$  ( $n$  linhas e  $n$  colunas) é diferente de zero. A árvore geradora considerada foi qualquer uma; desta forma, concluímos que qualquer árvore geradora de  $G$  corresponde a uma base de  $\mathbf{A}$  aumentada e, portanto, corresponde a uma solução básica do método simplex para redes.

Pode-se mostrar também o inverso, ou seja, qualquer base de  $\mathbf{A}$  aumentada também corresponde a uma árvore geradora de  $G$ . Uma base de  $\mathbf{A}$  aumentada é composta de  $n$  colunas linearmente independentes, sendo uma delas  $\mathbf{e}_1$ , e mais  $n - 1$  colunas que formam um conjunto linearmente independente. Considere as  $n - 1$  colunas de  $\mathbf{A}$  pertencentes à base. Os arcos associados a essas colunas podem ou não formar ciclos. Caso não formem ciclos, como temos  $n$  nós, eles formam uma árvore geradora de  $G$ , por definição. Caso formem algum ciclo, pode-se mostrar que as colunas dos arcos que formam este ciclo constituem um conjunto linearmente dependente, pois qualquer um dos arcos pode ser posto como uma combinação dos demais arcos do ciclo, contrariando a hipótese de serem parte de uma base.

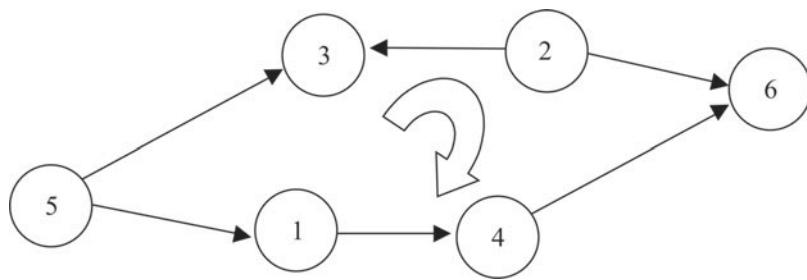
Considere, por exemplo, o ciclo da Figura 4.40. Se estipularmos uma orientação qualquer ao ciclo, por exemplo, a horária, podemos escrever a seguinte combinação linear das colunas da matriz de incidência nó arco correspondentes aos arcos deste grafo, em que multiplicamos por  $+1$  o arco que segue a mesma orientação do ciclo e por  $-1$  em caso contrário. Temos  $\mathbf{a}_{(5,3)} - \mathbf{a}_{(2,3)} + \mathbf{a}_{(2,6)} - \mathbf{a}_{(4,6)} - \mathbf{a}_{(1,4)} - \mathbf{a}_{(5,1)} = (\mathbf{e}_5 - \mathbf{e}_3) - (\mathbf{e}_2 - \mathbf{e}_3) + (\mathbf{e}_2 - \mathbf{e}_6) - (\mathbf{e}_4 - \mathbf{e}_6) - (\mathbf{e}_1 - \mathbf{e}_4) - (\mathbf{e}_5 - \mathbf{e}_1) = \mathbf{0}$ .

Portanto, o conjunto formado por essas colunas é linearmente dependente. Sempre que um conjunto de arcos forma um ciclo é possível escrever uma relação de dependência como a ilustrada no caso particular citado.

Como vimos no Capítulo 2, o método simplex inicia com uma solução básica factível, determinada por uma partição básica factível  $\mathbf{A} = [\mathbf{B} \ \mathbf{N}]$  ou, em outras palavras, as colunas de  $\mathbf{B}$  são



**Figura 4.39** Eliminação do nó 2 da árvore geradora da Figura 4.38.



**Figura 4.40** Um ciclo.

linearmente independentes e a solução básica é dada pela solução do sistema  $\mathbf{B}\hat{\mathbf{x}}_B = \mathbf{b}$ , e  $\hat{\mathbf{x}}_N = \mathbf{0}$ . Se  $\hat{\mathbf{x}}_B \geq \mathbf{0}$ , então a solução básica é factível, isto é, satisfaz todas as restrições do problema. Estendemos o método simplex desenvolvido no Capítulo 2 para o modelo de fluxo em redes a custo mínimo (4.1)–(4.3), cujas variáveis são canalizadas:  $\ell_k \leq x_k \leq u_k$ . As variáveis não-básicas são fixadas em um de seus limites:  $\hat{x}_k = \ell_k$  ou  $u_k$  e as variáveis básicas são determinadas pela solução do sistema:  $\mathbf{B}\hat{\mathbf{x}}_B = \mathbf{b} - \mathbf{N}\hat{\mathbf{x}}_N$  (a solução deste sistema básico para o cálculo das variáveis básicas é feito por simples substituições, devido à estrutura triangular de  $\mathbf{B}$ ). Se as variáveis básicas estiverem entre seus limites (isto é, se  $x_i$  é uma variável básica e  $\ell_i \leq \hat{x}_i \leq u_i$ ), então a solução básica é factível (de modo semelhante ao caso estudado no Capítulo 2, em que  $\mathbf{x} \geq \mathbf{0}$ ; esta solução básica factível corresponde a um vértice da região factível quando se tem a canalização das variáveis).

A cada iteração do simplex verifica-se se a solução básica factível atual é ótima, isto é, se satisfaz as condições de otimalidade. Para testar a otimalidade da solução, é necessário o cálculo dos custos relativos  $c_k - \boldsymbol{\lambda}^T \mathbf{a}_k$  das variáveis não-básicas, em que  $\boldsymbol{\lambda}$  é o vetor multiplicador do simplex dado pela solução do sistema  $\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B$  (o vetor multiplicador simplex  $\boldsymbol{\lambda}$  é calculado por simples substituições, devido à estrutura triangular de  $\mathbf{B}$ ). Se as condições de otimalidade não forem satisfeitas por alguma variável não-básica, digamos,  $x_k$ , então a solução atual pode ser melhorada ao se alterar o valor dessa variável, que pode ser aumentada ou diminuída.

A variação máxima no valor de  $x_k$ , que denotamos por  $\theta$ , isto é,  $x_k = \hat{x}_k \pm \theta$ , depende da direção simplex que é solução do sistema  $\mathbf{By} = \mathbf{a}_k$ , que indica como as variáveis básicas são alteradas: se  $x_k = \hat{x}_k + \theta$ , então  $\mathbf{x}_B = \hat{\mathbf{x}}_B - \theta \mathbf{y}$  ou, se  $x_k = \hat{x}_k - \theta$ , então  $\mathbf{x}_B = \hat{\mathbf{x}}_B + \theta \mathbf{y}$  (a direção simplex  $\mathbf{y}$  pode ser facilmente calculada devido à estrutura triangular de  $\mathbf{B}$ , mas é implicitamente calculada, como veremos adiante). A variável que atinge primeiro o seu limite (inferior ou superior) estabelece a máxima variação possível para  $\theta$  e torna-se uma variável não-básica. Após mudar os valores das variáveis conforme a máxima variação possível determinada, obtém-se uma nova solução básica factível melhor que a anterior. Este ciclo corresponde a uma iteração simplex. A partir dessa nova solução básica, uma nova iteração simplex é realizada até que as condições de otimalidade sejam verificadas ou uma solução ilimitada seja identificada.

No caso do modelo de fluxos em redes (4.1) – (4.3), dada uma solução básica factível (como vimos, ela está associada a uma árvore geradora do grafo), devemos calcular os multiplicadores simplex para determinar o custo relativo das variáveis não-básicas. Seja  $\boldsymbol{\lambda}^T = [\lambda_1 \lambda_2 \dots \lambda_n]$  o vetor de multiplicadores simplex relativo à base factível atual. Em uma iteração qualquer do simplex, o custo relativo da variável  $x_{ij}$  é dado por  $\hat{c}_{ij} = c_{ij} - \boldsymbol{\lambda}^T \mathbf{a}_{ij}$  ou, simplesmente,  $\hat{c}_{ij} = c_{ij} - \lambda_i + \lambda_j$ , uma vez que  $\mathbf{a}_{ij} = \mathbf{e}_i - \mathbf{e}_j$ .

Por definição, o vetor multiplicador simplex é dado por  $\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{c}_B$  ou  $\mathbf{c}_B - \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{0}$ . Em outras palavras, os custos relativos das variáveis básicas são iguais a zero, ou seja,

$$\hat{c}_{ij} = c_{ij} - \lambda_i + \lambda_j = 0 \quad (4.4)$$

para cada variável básica. Lembre-se de que uma das colunas de  $\mathbf{B}$  é o vetor  $\mathbf{e}_1$ , que introduzimos para completar a base. A essa coluna associamos uma variável (fluxo sobre o arco raiz, que será sempre zero), que denotamos por  $x_a$  e seu custo é definido por  $c_a = 0$  (este valor é arbitrário e pode ser qualquer valor, sem que os cálculos dos custos relativos das variáveis não-básicas sejam alterados). Para a coluna básica  $\mathbf{e}_1$ , temos  $\hat{c}_a = c_a - \boldsymbol{\lambda}^T \mathbf{e}_1 = 0 - \lambda_1 = 0$ , ou seja,  $\lambda_1 = 0$ . Com o valor de  $\lambda_1 = 0$ , os demais valores de  $\lambda_j, j \neq 1$ , ficam univocamente determinados pelo sistema de equações (4.4). Note, de (4.4), que a variável  $\lambda_j$  é facilmente determinada se o valor de  $\lambda_j$  já tiver sido determinado, ou vice-versa. Ou seja, o vetor multiplicador simplex é determinado por substituições recursivas, começando pelo nó raiz que tem a variável  $\lambda_1 = 0$ . Isso é possível porque a base é triangular (veja argumentação anterior para mostrar que as colunas associadas a uma árvore geradora formam um conjunto linearmente independente). A resolução do exemplo numérico adiante esclarece este assunto.

De posse de  $\boldsymbol{\lambda}$ , os custos relativos das variáveis não-básicas (correspondentes a arcos não pertencentes à árvore geradora) podem ser facilmente calculados. O custo relativo da variável não-básica indica se é ou não vantajoso introduzir essa variável (esse arco) na base. As condições para verificar se é vantajoso introduzir uma determinada variável não-básica na base atual são (estratégia simplex para variáveis canalizadas):

- Se  $\hat{c}_{ij} < 0$  e  $\hat{x}_{ij} = \ell_{ij}$ , então aumentar a variável  $x_{ij}: x_{ij} = \hat{x}_{ij} + \theta, \theta \geq 0$ , mantendo as demais variáveis não-básicas inalteradas, faz decrescer a função objetivo:  $f(\mathbf{x}) = f(\hat{\mathbf{x}}) + \theta \hat{c}_{ij}$  (análogo a (2.36) no Capítulo 2).
- Se  $\hat{c}_{ij} > 0$  e  $\hat{x}_{ij} = u_{ij}$ , então diminuir a variável  $x_{ij}: x_{ij} = \hat{x}_{ij} - \theta, \theta \geq 0$ , mantendo as demais variáveis não-básicas inalteradas, faz decrescer a função objetivo:  $f(\mathbf{x}) = f(\hat{\mathbf{x}}) - \theta \hat{c}_{ij}$ .

A condição de *optimalidade* é satisfeita por uma solução básica factível se, para todo arco no seu limite inferior (ou seja,  $\hat{x}_{ij} = \ell_{ij}$ ), então  $\hat{c}_{ij} \geq 0$  e, para todo arco no seu limite superior (ou seja,  $\hat{x}_{ij} = u_{ij}$ ), então  $\hat{c}_{ij} \leq 0$  (esta é a extensão da condição de optimalidade na propriedade 2.3 do Capítulo 2, para variáveis canalizadas).

### Troca de base no método simplex em redes

No simplex (Seção 2.6 – Capítulo 2), substituímos uma coluna básica por uma não-básica, formando uma nova base. Vimos que o vetor multiplicador simplex é facilmente calculado e podemos identificar qual coluna (arco) deve entrar na base (árvore). Precisamos calcular qual a direção simplex e, a partir dela, o novo valor da variável não-básica que entra na base, além da variável básica que deixa a base.

Essas operações a realizar, no caso de um grafo, são bem simples. Sabemos que, no grafo, cada base corresponde a uma árvore geradora e cada coluna corresponde a um arco. A introdução de um arco (a variável não-básica a entrar) na árvore geradora atual (base atual) forma um ciclo que precisa ser desfeito, pois a uma base corresponde sempre uma árvore geradora, que não possui ciclos. Assim, a variável a sair da base deve ser um dos arcos desse ciclo (único) formado, uma vez que o ciclo indica que os arcos são linearmente dependentes (dizemos que um conjunto de arcos é linearmente dependente se as colunas associadas a eles formam um conjunto de vetores linearmente dependentes). Para determinar qual deles deverá sair, basta garantir o balanceamento de fluxo em cada nó.

Se orientarmos o ciclo formado de acordo com o novo arco introduzido (nova coluna não-básica), se o fluxo neste novo arco aumenta de  $\theta$ , então todos os arcos que têm a mesma orienta-

ção do ciclo têm o valor aumentado de  $\theta$  (isto é,  $y_i = -1$ ) e os arcos com orientação contrária à do ciclo têm o fluxo diminuído de  $\theta$  (isto é,  $y_i = 1$ ). Os demais arcos permanecem com seus fluxos inalterados. Com isso, a conservação de fluxo em todos os nós do grafo é mantida (o sistema  $\mathbf{Ax} = \mathbf{b}$  é satisfeito). O aumento do fluxo no novo arco é limitado pelo fluxo adicional que pode ser enviado pelos arcos que têm a mesma orientação do ciclo ou pelo máximo valor de fluxo que pode ser diminuído dos arcos com orientação contrária ao ciclo. O arco que atinge primeiro algum desses limitantes é o arco que sai da base (o ciclo é quebrado e o fluxo sobre o arco que sai da base está em um de seus limites) e os demais arcos formam uma nova árvore geradora (a nova base). Essa variação máxima define o novo valor  $\theta$  da variável que entra na base.

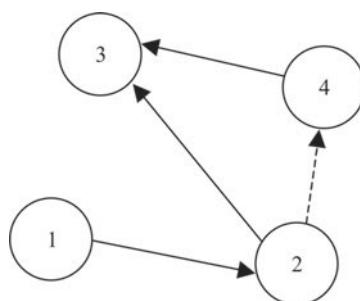
Considere, por exemplo, que a base atual de um determinado problema de fluxo corresponde à árvore geradora indicada na Figura 4.37 e que a variável a entrar na base corresponde ao arco  $(2, 4)$ , indicado de forma tracejada na Figura 4.41.

Com a introdução desse arco, forma-se o ciclo  $\{(2, 4), (4, 3), (2, 3)\}$  e é preciso retirar algum arco do grafo para que uma nova base seja formada. Como vimos anteriormente, os arcos que formam ciclos no grafo são linearmente dependentes (isto é, as colunas correspondentes formam um conjunto de vetores linearmente dependentes). Portanto, algum arco do ciclo deverá ser retirado com a entrada do novo arco  $(2, 4)$ .

Aumentando o fluxo no arco  $(2, 4)$  de  $\theta$ , para que o equilíbrio de fluxo seja mantido no nó 4, é preciso aumentar de  $\theta$  o fluxo no arco  $(4, 3)$ . Com isso, para que o equilíbrio de fluxo seja mantido no nó 3, é preciso diminuir de  $\theta$  o fluxo no arco  $(2, 3)$ . Com isso, obtemos o equilíbrio de fluxo também no nó 2. O valor de  $\theta$  pode ser aumentado desde que não ultrapasse a capacidade máxima no arco  $(2, 4)$  ou a capacidade máxima no arco  $(4, 3)$  e também não reduza além do mínimo tolerável a capacidade mínima no arco  $(2, 3)$ . O limite mais restrito fornece o valor da variação de fluxo no ciclo, bem como a variável a sair da base (o arco a ser eliminado para que se obtenha uma nova árvore geradora). O arco eliminado tem o fluxo em um de seus limites: máximo ou mínimo. Com a nova base, novos custos relativos são calculados e o processo é repetido até que a condição de otimalidade seja satisfeita.

#### *Base inicial*

No método simplex, move-se de uma solução básica para outra melhor até que não existam mais soluções básicas melhores. Para aplicar o método simplex, necessitamos de uma solução básica inicial factível e obter essa base nem sempre é simples (ver Seção 2.8 – Capítulo 2). No problema de fluxo a custo mínimo, um artifício possível para se ter uma base inicial factível consiste



**Figura 4.41** Arco  $(2,4)$  entrando na base.

em introduzir arcos artificiais no grafo que conectam nós fontes (com  $d_i > 0$ ) aos nós sorvedouros (com  $d_i < 0$ ). Nos arcos artificiais, não há limite de capacidade, mas o custo unitário de fluxo deve ser muito alto, de modo a assegurar que, em uma solução ótima, nenhum desses arcos terá fluxo positivo (esta estratégia corresponde ao método  $M$ -grande; ver Seção 2.8 – Capítulo 2).

Um arco artificial é considerado de cada vez, calculando-se o fluxo que deve passar nele, que é dado pelo mínimo entre a quantidade disponível no nó fonte e a quantidade demandada no nó sorvedouro. Por exemplo, se  $i$  é tal que  $d_i > 0$  (nó fonte) e  $j$  é tal que  $d_j < 0$  (nó sorvedouro), então o fluxo no arco artificial  $(i, j)^a$  é  $x_{ij}^a = \min\{d_i, -d_j\}$ . A demanda do nó sorvedouro e a oferta do nó fonte são atualizadas para satisfazer o equilíbrio de fluxo nesses nós (isto é,  $d_i \leftarrow d_i - x_{ij}^a$ ,  $d_j \leftarrow d_j + x_{ij}^a$ ), e mais um novo arco artificial é considerado e seu fluxo é calculado de maneira similar. Isso é realizado de maneira sucessiva até que todas as demandas dos nós sorvedouros sejam satisfeitas.

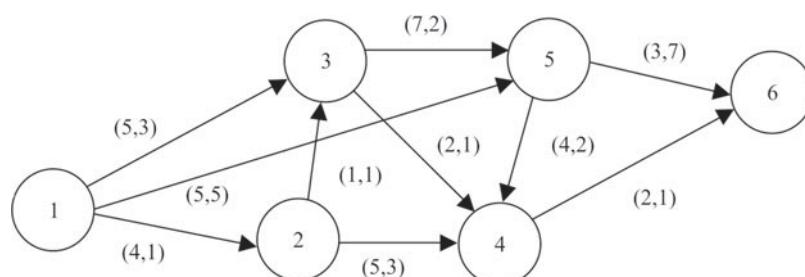
A árvore geradora (uma base) inicial é obtida pelos arcos artificiais com fluxos positivos. Caso haja nós de transbordo ( $d_k = 0$ ), então qualquer arco que incida sobre esses nós é incluído com fluxo zero, para completar a árvore geradora.

Apresentamos a seguir um exemplo numérico do método simplex em redes. Considere o grafo esquematizado na Figura 4.42. Admitimos que há disponíveis 10 unidades do produto no nó 1 (isto é,  $d_1 = 10$ ) e existe uma demanda de 4 unidades no nó 6 (isto é.,  $d_6 = -4$ ) e de 6 unidades no nó 5 ( $d_5 = -6$ ). Os limites inferiores de fluxo são iguais a zero para todos os arcos, isto é,  $\ell_{ij} = 0$ . Os limites superiores de fluxo,  $u_{ij}$ , e os custos unitários,  $c_{ij}$ , dos arcos são os indicados ao lado de cada arco no grafo. Por exemplo, a capacidade no arco  $(3, 5)$  é  $u_{35} = 7$  e custa  $c_{35} = 2$  unidades monetárias para enviar uma unidade de fluxo por este arco.

### Iteração 1 – Obtenção da base inicial

Considere inicialmente um arco artificial que une um nó com disponibilidade de produto a um nó com demanda do produto, por exemplo, o arco  $(1, 6)^a$ . Pelo arco  $(1, 6)^a$  passamos um fluxo igual ao mínimo entre a disponibilidade do produto existente no nó 1 ( $d_1 = 10$ ) e a demanda no nó 6 ( $-d_6 = 4$ ):  $x_{16}^a = \min\{d_1, -d_6\} = 4$ . A disponibilidade no nó 1 é atualizada e passa a ser de 6 unidades e a demanda no nó 6 também é atualizada e passa a ser de 0 unidade. Existe ainda um nó com demanda positiva, assim consideramos a seguir o arco  $(1, 5)^a$  e por ele passamos um fluxo igual ao mínimo entre a disponibilidade do produto existente no nó 1 ( $d_1 = 6$ , disponibilidade já atualizada) e a demanda no nó 5 ( $-d_5 = 6$ ):  $x_{15}^a = \min\{d_1, -d_5\} = 6$ .

No grafo original já existe um arco  $(1, 5)$  com capacidade de 5 unidades, insuficiente para o fluxo desejado de 6 unidades. Se a capacidade deste arco fosse suficiente, não haveria a necessi-



**Figura 4.42** Exemplo de um problema de fluxo.

dade de introduzir de um arco artificial unindo o nó 1 ao nó 5. Introduzimos, portanto, um arco artificial com capacidade alta, resultando em dois arcos ligando o nó 1 ao nó 5 no grafo estendido. Os arcos são diferenciados pela notação introduzida  $(1, 5)^a$  para o arco artificial.

A disponibilidade no nó 1 é atualizada e passa a ser de 0 unidade e a demanda no nó 5 também é atualizada e passa a ser de 0 unidade. Não existe mais nenhum nó com demanda positiva e uma árvore geradora para o grafo é construída, considerando inicialmente estes dois arcos:  $(1, 6)^a$  e  $(1, 5)^a$ , à qual três novos arcos devem ser adicionados, de modo que não formem ciclos (lembre-se que uma árvore geradora em um grafo com  $n$  nós tem  $n - 1$  arcos).

Para obter uma árvore geradora, basta adicionarmos arcos do grafo sem que ciclos sejam formados. Como os nós 2, 3 e 4 estão desconectados, podemos introduzir os arcos  $(1, 2)$ ,  $(2, 3)$  e  $(3, 4)$  para completar a árvore geradora, ilustrada na Figura 4.43. O fluxo no arco artificial  $(1, 5)^a$  é de 6 unidades, no arco artificial  $(1, 6)^a$  é de 4 unidades e nos demais arcos é de 0 unidade. Com isso, temos uma solução básica factível inicial (factível para o problema artificial que considera arcos artificiais):

#### Solução básica na iteração 1:

Variáveis básicas:  $\hat{x}_{16}^a = 4$ ,  $\hat{x}_{15}^a = 6$ ,  $\hat{x}_{12} = 0$ ,  $\hat{x}_{23} = 0$ ,  $\hat{x}_{34} = 0$

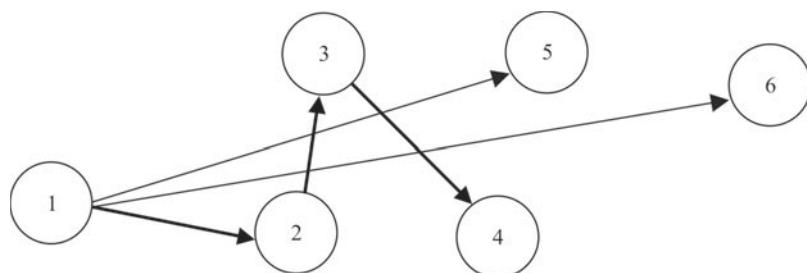
Variáveis não-básicas nulas

Atribuímos um custo bastante elevado a esses arcos artificiais, por exemplo,  $c_{ij}^a = 100$  (este valor depende dos demais custos do problema). O custo relativo  $\hat{c}_{ij} = c_{ij} - \lambda_i + \lambda_j$  para cada variável básica é igual a 0 e, além disso,  $\lambda_1 = 0$ . Portanto, os demais multiplicadores simplex podem ser obtidos por:

$$\begin{aligned} &= c_{16} - \lambda_1 + \lambda_6 = 0 \text{ ou } 0 = 100 - \lambda_1 + \lambda_6 = 0 \text{ ou } \lambda_6 = \lambda_1 - 100 = -100 \\ \hat{c}_{15} &= c_{15} - \lambda_1 + \lambda_5 = 0 \text{ ou } \lambda_5 = \lambda_1 - 100 = -100 \\ \hat{c}_{12} &= c_{12} - \lambda_1 + \lambda_2 = 0 \text{ ou } \lambda_2 = \lambda_1 - 1 = -1 \\ \hat{c}_{23} &= c_{23} - \lambda_2 + \lambda_3 = 0 \text{ ou } \lambda_3 = \lambda_2 - 1 = -1 - 1 = -2 \\ \hat{c}_{34} &= c_{34} - \lambda_3 + \lambda_4 = 0 \text{ ou } \lambda_4 = \lambda_3 - 1 = -2 - 1 = -3 \end{aligned}$$

Ou seja,  $\boldsymbol{\lambda} = (0, -1, -2, -3, -100, -100)^T$ .

Observe, pelos cálculos anteriores, que, para se obter o valor de  $\lambda_j$ , é preciso ter calculado antes o valor de  $\lambda_i$ , ou vice-versa. Utilizando a árvore correspondente à base atual, é possível ver diretamente quais valores  $\lambda_i$  ( $\lambda_j$ ) podem ser calculados tendo em vista os valores já obtidos de  $\lambda_j$  ( $\lambda_i$ ). Inicialmente temos apenas  $\lambda_1 = 0$ . Da árvore (a base) da Figura 4.43, observamos que os nós 2, 5 e 6 são adjacentes ao nó 1. Assim, tendo o valor de  $\lambda_1$ , os valores de  $\lambda_2$ ,  $\lambda_5$  e  $\lambda_6$  podem ser obti-



**Figura 4.43** Árvore geradora inicial.

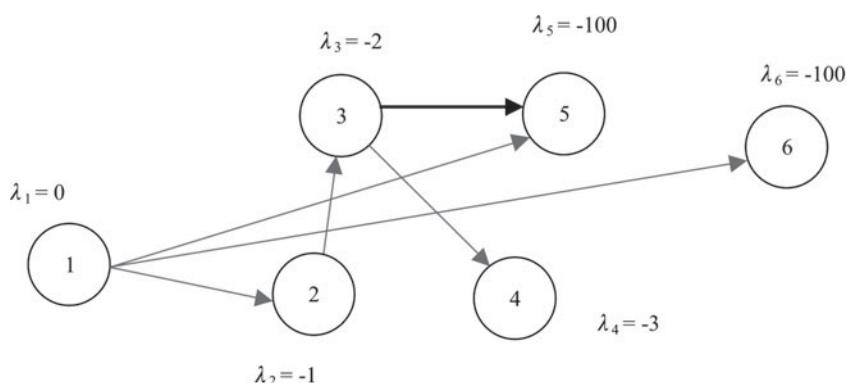
dos a seguir. Observe que, tendo apenas o valor de  $\lambda_1$ , não é possível obter os valores de  $\lambda_3$  e  $\lambda_4$ , pois os nós 3 e 4 não são adjacentes ao nó 1.  $\lambda_3$  Poderá ser calculado apenas após o valor de  $\lambda_2$  ter sido obtido, pois o nó 3 é adjacente ao nó 2, e  $\lambda_4$  poderá ser calculado apenas após o valor de  $\lambda_3$  ter sido obtido, pois o nó 4 é adjacente ao nó 3. Esse cálculo é realizado a partir do nó raiz (nó 1) em direção aos nós finais, por cadeias únicas (não há mais de uma cadeia ligando o nó 1 a qualquer nó  $i$ , pois isso caracterizaria um ciclo).

Com os multiplicadores, podemos calcular os custos relativos das variáveis não-básicas dados por  $\hat{c}_{ij} = c_{ij} - \lambda_i + \lambda_j$ . Temos que

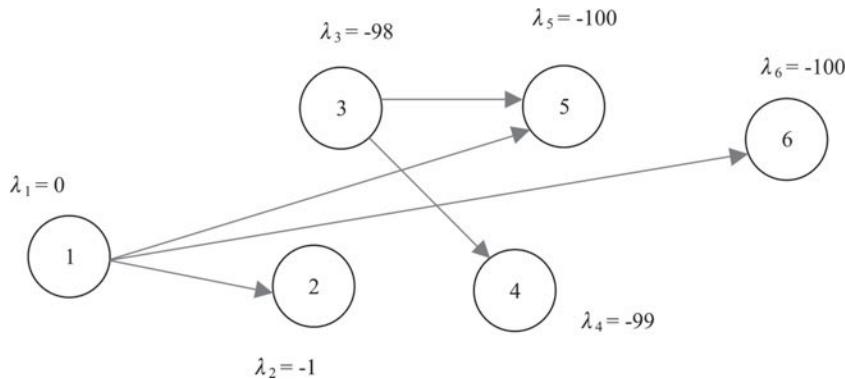
$$\begin{aligned}
 \hat{c}_{13} &= c_{13} - \lambda_1 + \lambda_3 = 3 - 0 - 2 = 1 & (\hat{x}_{13} = \ell_{13}) \\
 \hat{c}_{35} &= c_{35} - \lambda_3 + \lambda_5 = 2 + 2 - 100 = -96 & (\hat{x}_{35} = \ell_{35}) \leftarrow \\
 \hat{c}_{24} &= c_{24} - \lambda_2 + \lambda_4 = 3 + 1 - 3 = 1 & (\hat{x}_{24} = \ell_{24}) \\
 \hat{c}_{54} &= c_{54} - \lambda_5 + \lambda_4 = 2 + 100 - 3 = 99 & (\hat{x}_{54} = \ell_{54}) \\
 \hat{c}_{56} &= c_{56} - \lambda_5 + \lambda_6 = 7 + 100 - 100 = 7 & (\hat{x}_{56} = \ell_{56}) \\
 \hat{c}_{46} &= c_{46} - \lambda_4 + \lambda_6 = 1 + 3 - 100 = -96 & (\hat{x}_{46} = \ell_{46}) \\
 \hat{c}_{15} &= c_{15} - \lambda_1 + \lambda_5 = 5 - 0 - 100 = -95 & (\hat{x}_{15} = \ell_{15})
 \end{aligned}$$

Note que todas as variáveis não-básicas estão fixadas em seus limites inferiores, neste exemplo, zero. Escolhemos o arco com o menor custo relativo (regra de Dantzig). Neste caso, há um empate e escolhemos arbitrariamente o arco (3, 5).

Com a introdução deste arco (ver Figura 4.44), um ciclo é formado:  $\{(3, 5), (1, 5), (1, 2), (2, 3)\}$ . Aumentando o fluxo no arco (3, 5) de  $\theta$  para que a conservação de fluxo seja mantida no nó 5, é preciso diminuir de  $\theta$  o fluxo no arco (1, 5). Com isso, para que a conservação de fluxo seja mantida no nó 1, é preciso aumentar de  $\theta$  o fluxo no arco (1, 2). Para que a conservação de fluxo seja mantida também no nó 2, é preciso aumentar de  $\theta$  o fluxo no arco (2, 3). Com isso, obtemos a conservação de fluxo também no nó 3. O valor de  $\theta$  pode ser aumentado desde que não ultrapasse a capacidade máxima nos arcos (3, 5), (1, 2) e (2, 3) ou a capacidade mínima no arco (1, 5). O limite mais restrito fornece o valor do fluxo no novo arco e a variável a sair da base (o arco a ser eliminado para que se obtenha uma nova árvore geradora). Neste caso, temos que  $\theta$  pode ser aumentado no máximo de 1 unidade, pois a capacidade máxima no arco (2, 3) é atingida. Portanto, o arco a sair da base é (2, 3), com o fluxo no limite superior:  $x_{23} = 1$ , resultando na árvore geradora indicada na Figura 4.45.



**Figura 4.44** Arco (3,5) é introduzido.



**Figura 4.45** Nova árvore geradora resultante.

Apenas os fluxos no ciclo são alterados:  $x_{15}^a = 5$ ,  $x_{12} = 1$ ,  $x_{23} = 1$  (sai da base no limite superior),  $x_{35} = 1$  (entra na base) e os demais ficam inalterados:  $x_{16}^a = 4$ ,  $x_{34} = 0$ , como também os arcos não-básicos. Com a nova árvore (base), repetimos o procedimento anterior.

### Iteração 2:

Solução básica na iteração 2:

Variáveis básicas:  $\hat{x}_{16}^a = 4$ ,  $\hat{x}_{15}^a = 5$ ,  $\hat{x}_{12} = 1$ ,  $\hat{x}_{35} = 1$ ,  $\hat{x}_{34} = 0$

Variável não-básica em seu limite superior:  $\hat{x}_{23} = 1$ . Demais variáveis não-básicas nulas (em seus limites inferiores).

Os novos multiplicadores simplex correspondentes a esta nova base são:

$$\lambda_1 = 0$$

$$\hat{c}_{16} = c_{16} - \lambda_1 + \lambda_6 = 0 \text{ ou } 0 = 100 - \lambda_1 + \lambda_6 = 0 \text{ ou } \lambda_6 = \lambda_1 - 100 = -100$$

$$\hat{c}_{15} = c_{15} - \lambda_1 + \lambda_5 = 0 \text{ ou } \lambda_5 = \lambda_1 - 100 = -100$$

$$\hat{c}_{12} = c_{12} - \lambda_1 + \lambda_2 = 0 \text{ ou } \lambda_2 = \lambda_1 - 1 = -1$$

$$\hat{c}_{35} = c_{35} - \lambda_3 + \lambda_5 = 0 \text{ ou } \lambda_3 = 2 + \lambda_5 = -98$$

$$\hat{c}_{34} = c_{34} - \lambda_3 + \lambda_4 = 0 \text{ ou } \lambda_4 = \lambda_3 - 1 = -98 - 1 = -99$$

$$\boldsymbol{\lambda} = (0, -1, -98, -99, -100, -100)^T$$

Com os novos multiplicadores, podemos calcular os custos relativos das variáveis não-básicas dados por  $\hat{c}_{ij} = c_{ij} - \lambda_i + \lambda_j$ :

$$\hat{c}_{13} = c_{13} - \lambda_1 + \lambda_3 = 3 - 0 - 98 = -95 \quad (\hat{x}_{13} = \ell_{13})$$

$$\hat{c}_{23} = c_{23} - \lambda_2 + \lambda_3 = 1 + 1 - 98 = -96 \quad (\hat{x}_{23} = u_{23})$$

$$\hat{c}_{24} = c_{24} - \lambda_2 + \lambda_4 = 3 + 1 - 99 = -95 \quad (\hat{x}_{24} = \ell_{24})$$

$$\hat{c}_{54} = c_{54} - \lambda_5 + \lambda_4 = 2 + 100 - 99 = 3 \quad (\hat{x}_{54} = \ell_{54})$$

$$\hat{c}_{56} = c_{56} - \lambda_5 + \lambda_6 = 7 + 100 - 100 = 7 \quad (\hat{x}_{56} = \ell_{56})$$

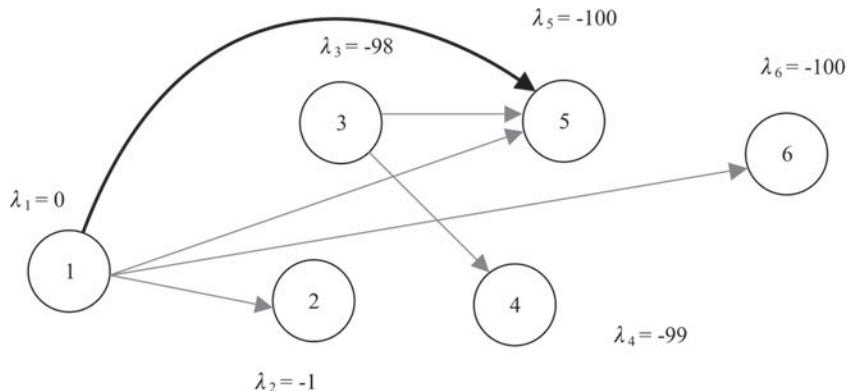
$$\hat{c}_{46} = c_{46} - \lambda_4 + \lambda_6 = 1 + 99 - 100 = 0 \quad (\hat{x}_{46} = \ell_{46})$$

$$\hat{c}_{15} = c_{15} - \lambda_1 + \lambda_5 = 5 - 0 - 100 = -95 \quad (\hat{x}_{15} = \ell_{15}) \leftarrow$$

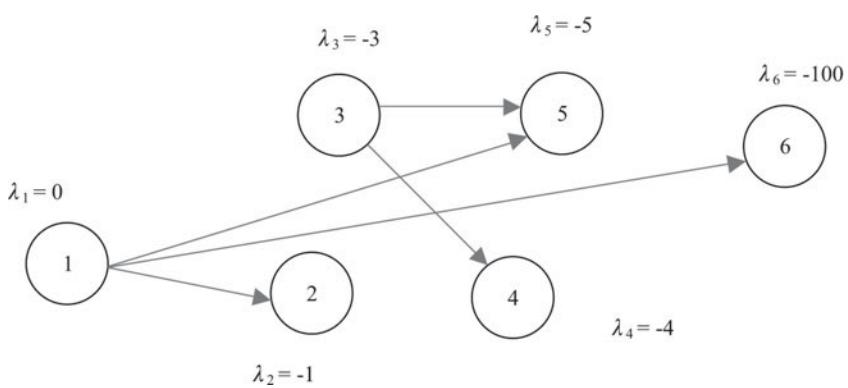
O arco (2, 3) tem custo relativo negativo e, como está em seu limite superior, satisfaz a condição de optimilidade. Há um empate com os custos relativos dos arcos (1, 3), (2, 4) e (1, 5). Arbitrariamente, escolhemos o arco (1, 5).

Com a introdução deste arco (ver Figura 4.46), um ciclo é formado:  $\{(1, 5), (1, 5)^a\}$ . Aumentando o fluxo no arco  $(1, 5)$  de  $\theta$  para que a conservação de fluxo seja mantida no nó 5, é preciso diminuir de  $\theta$  o fluxo no arco  $(1, 5)^a$ . Com isso, obtemos a conservação de fluxo também no nó 1. O valor de  $\theta$  pode ser aumentado desde que não ultrapasse a capacidade máxima no arco  $(1, 5)$  ou a capacidade mínima no arco  $(1, 5)^a$ . No caso, temos que  $\theta$  pode ser aumentado no máximo de 5 unidades, pois o limite inferior no arco  $(1, 5)^a$  é atingido e, simultaneamente, a capacidade máxima no arco  $(1, 5)$  é atingida. O arco a sair da base é o  $(1, 5)^a$ , resultando na árvore geradora indicada na Figura 4.47. Note que o arco  $(1, 5)$  poderia ter sido escolhido para quebrar o ciclo, indicando uma variável não-básica no limite superior (sai de seu limite inferior e atinge o limite superior, de modo que a solução básica muda sem que a base mude). Entretanto, é conveniente retirar arcos artificiais da árvore geradora básica.

Apenas os fluxos no ciclo são alterados:  $\hat{x}_{15}^a = 0$  (sai da base no limite inferior),  $x_{15} = 5$ . Os demais ficam inalterados.



**Figura 4.46** Ciclo formado com a introdução do arco  $(1, 5)$ .



**Figura 4.47** Nova árvore geradora após eliminação do arco  $(1, 5)^a$ .

**Iteração 3:**Solução básica na iteração 3:Variáveis básicas:  $\hat{x}_{16}^a = 4$ ,  $\hat{x}_{15} = 5$ ,  $\hat{x}_{12} = 1$ ,  $\hat{x}_{35} = 1$ ,  $\hat{x}_{34} = 0$ Variável não-básicas em seu limite superior:  $\hat{x}_{23} = 1$ . Demais variáveis não-básicas nulas (em seus limites inferiores).

Os novos multiplicadores simplex correspondentes a esta nova base são:

$$\lambda_1 = 0$$

$$\hat{c}_{16} = c_{16} - \lambda_1 + \lambda_6 = 0 \text{ ou } 0 = 100 - \lambda_1 + \lambda_6 = 0 \text{ ou } \lambda_6 = \lambda_1 - 100 = -100$$

$$\hat{c}_{15} = c_{15} - \lambda_1 + \lambda_5 = 0 \text{ ou } \lambda_5 = \lambda_1 - 5 = -5$$

$$\hat{c}_{12} = c_{12} - \lambda_1 + \lambda_2 = 0 \text{ ou } \lambda_2 = \lambda_1 - 1 = -1$$

$$\hat{c}_{35} = c_{35} - \lambda_3 + \lambda_5 = 0 \text{ ou } \lambda_3 = 2 + \lambda_5 = -3$$

$$\hat{c}_{34} = c_{34} - \lambda_3 + \lambda_4 = 0 \text{ ou } \lambda_4 = \lambda_3 - 1 = -3 - 1 = -4$$

$$\boldsymbol{\lambda} = (0, -1, -3, -4, -5, -100)^T$$

Com os novos multiplicadores, os custos relativos das variáveis não-básicas ficam (o arco (1, 5)<sup>a</sup> não precisa mais ser considerado):

$$\hat{c}_{13} = c_{13} - \lambda_1 + \lambda_3 = 3 - 0 - 3 = 0 \quad (\hat{x}_{13} = \ell_{13})$$

$$\hat{c}_{23} = c_{23} - \lambda_2 + \lambda_3 = 1 + 1 - 3 = -1 \quad (\hat{x}_{23} = u_{23})$$

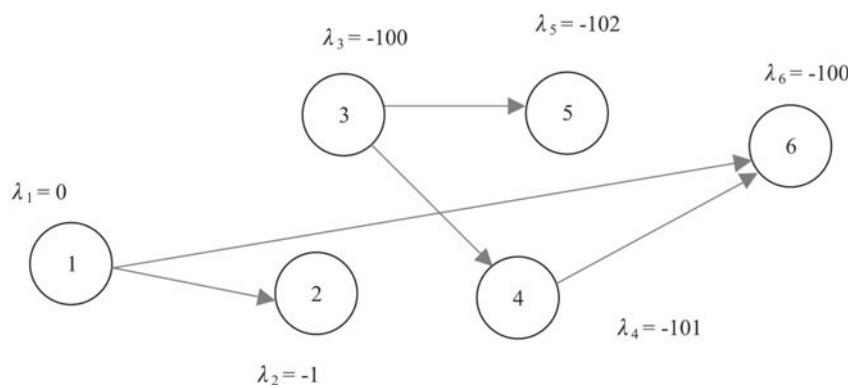
$$\hat{c}_{24} = c_{24} - \lambda_2 + \lambda_4 = 3 + 1 - 4 = 0 \quad (\hat{x}_{24} = \ell_{24})$$

$$\hat{c}_{54} = c_{54} - \lambda_5 + \lambda_4 = 2 + 5 - 4 = 3 \quad (\hat{x}_{54} = \ell_{54})$$

$$\hat{c}_{56} = c_{56} - \lambda_5 + \lambda_6 = 7 + 5 - 100 = -88 \quad (\hat{x}_{56} = \ell_{56})$$

$$\hat{c}_{46} = c_{46} - \lambda_4 + \lambda_6 = 1 + 4 - 100 = -95 \quad (\hat{x}_{46} = \ell_{46}) \leftarrow$$

O arco (4, 6) tem o menor custo relativo e está em seu limite inferior. Assim, introduzimos (4, 6) na base. Com a introdução deste arco, um ciclo é formado: {(4, 6), (1, 6), (1, 5), (3, 5), (3, 4)}. Não se pode aumentar o fluxo no arco (4, 6), pois o fluxo no arco (1, 5) já está em seu máximo (situação degenerada: um fluxo básico em um de seus limites). O fluxo no novo arco é, portanto, igual a 0 e o arco a ser eliminado para que se obtenha uma nova árvore geradora é o (1, 5), pois é ele que limita o crescimento do fluxo no ciclo, resultando na árvore geradora indicada na Figura 4.48. Neste caso, a solução básica permanece inalterada, embora a árvore geradora (base) seja alterada.

**Figura 4.48** Nova árvore geradora apóis a eliminação do arco (1, 5).

### Iteração 4:

Solução básica na iteração 4:

Variáveis básicas:  $\hat{x}_{16}^a = 4$ ,  $\hat{x}_{46} = 0$ ,  $\hat{x}_{12} = 1$ ,  $\hat{x}_{35} = 1$ ,  $\hat{x}_{34} = 0$

Variáveis não-básicas em seus limites superiores:  $\hat{x}_{23} = 1$ ,  $\hat{x}_{15} = 5$ . Demais variáveis não-básicas nulas (em seus limites inferiores).

Os novos multiplicadores simplex correspondentes a esta nova base são:

$$\lambda_1 = 0$$

$$\hat{c}_{16} = c_{16} - \lambda_1 + \lambda_6 = 0 \text{ ou } 0 = 100 - \lambda_1 + \lambda_6 = 0 \text{ ou } \lambda_6 = \lambda_1 - 100 = -100$$

$$\hat{c}_{46} = c_{46} - \lambda_4 + \lambda_6 = 0 \text{ ou } \lambda_4 = \lambda_6 - 1 = -101$$

$$\hat{c}_{12} = c_{12} - \lambda_1 + \lambda_2 = 0 \text{ ou } \lambda_2 = \lambda_1 - 1 = -1$$

$$\hat{c}_{34} = c_{34} - \lambda_3 + \lambda_4 = 0 \text{ ou } \lambda_3 = \lambda_4 + 1 = -101 + 1 = -100$$

$$\hat{c}_{35} = c_{35} - \lambda_3 + \lambda_5 = 0 \text{ ou } \lambda_5 = \lambda_3 - 2 = -102$$

$$\boldsymbol{\lambda} = (0, -1, -100, -101, -102, -100)^T$$

Com os novos multiplicadores, os custos relativos das variáveis não-básicas ficam:

$$\hat{c}_{13} = c_{13} - \lambda_1 + \lambda_3 = 3 - 0 - 100 = -97 \quad (\hat{x}_{13} = \lambda_{13}) \leftarrow$$

$$\hat{c}_{23} = c_{23} - \lambda_2 + \lambda_3 = 1 + 1 - 100 = -98 \quad (\hat{x}_{23} = \lambda_{23})$$

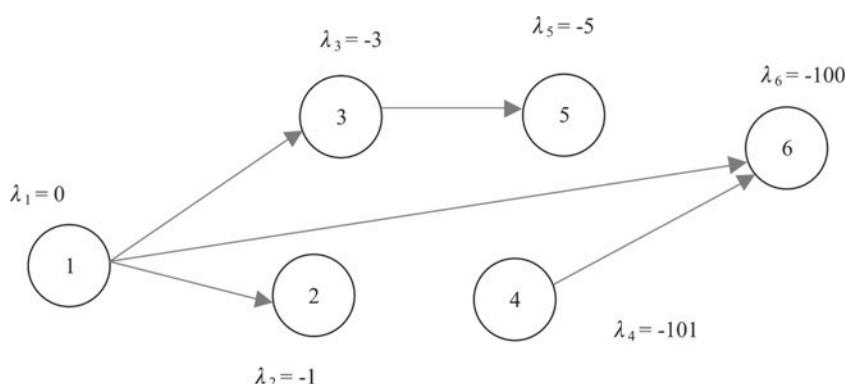
$$\hat{c}_{24} = c_{24} - \lambda_2 + \lambda_4 = 3 + 1 - 100 = -96 \quad (\hat{x}_{24} = \lambda_{24})$$

$$\hat{c}_{54} = c_{54} - \lambda_5 + \lambda_4 = 2 + 102 - 101 = 3 \quad (\hat{x}_{54} = \lambda_{54})$$

$$\hat{c}_{56} = c_{56} - \lambda_5 + \lambda_6 = 7 + 102 - 100 = 9 \quad (\hat{x}_{56} = \lambda_{56})$$

$$\hat{c}_{15} = c_{15} - \lambda_1 + \lambda_5 = 5 - 0 - 102 = -97 \quad (\hat{x}_{15} = \lambda_{15})$$

Os arcos (2, 3) e (1, 5) têm custos relativos negativos, mas estão em seus limitantes superiores, portanto, não violam a condição de optimilidade. O arco (1, 3) é aquele que mais viola a condição de optimilidade e é escolhido para entrar na base. Com a introdução deste arco, um ciclo é formado:  $\{(1, 3), (3, 4), (4, 6), (1, 6)\}$ . O fluxo no arco (1, 3) pode aumentar no máximo de 2 unidades pois o fluxo no arco (3, 4) (e também no arco (4, 6)) atinge seu limite superior. O fluxo no novo arco é, portanto, igual a 2 e (3, 4) é eliminado para quebrar o ciclo, resultando na árvore geradora indicada na Figura 4.49. Apenas os seguintes fluxos no ciclo são alterados:  $x_{13} = 2$  (entra na base),  $x_{34} = 2$  (sai da base no limite superior),  $x_{46} = 2$ ,  $x_{16}^a = 2$  e os demais permanecem inalterados.



**Figura 4.49** Nova árvore geradora após eliminação do arco (3, 4).

**Iteração 5:**Solução básica na iteração 5:Variáveis básicas:  $\hat{x}_{16}^a = 2$ ,  $\hat{x}_{46} = 2$ ,  $\hat{x}_{12} = 1$ ,  $\hat{x}_{35} = 1$ ,  $\hat{x}_{13} = 2$ Variáveis não-básicas em seus limites superiores:  $\hat{x}_{23} = 1$ ,  $\hat{x}_{15} = 5$ ,  $\hat{x}_{34} = 2$ . Demais variáveis não-básicas nulas (em seus limites inferiores).

Os novos multiplicadores simplex correspondentes a esta nova base são:

$$\lambda_1 = 0$$

$$\hat{c}_{16} = c_{16} - \lambda_1 + \lambda_6 = 0 \text{ ou } 0 = 100 - \lambda_1 + \lambda_6 = 0 \text{ ou } \lambda_6 = \lambda_1 - 100 = -100$$

$$\hat{c}_{46} = c_{46} - \lambda_4 + \lambda_6 = 0 \text{ ou } \lambda_4 = \lambda_6 - 1 = -101$$

$$\hat{c}_{12} = c_{12} - \lambda_1 + \lambda_2 = 0 \text{ ou } \lambda_2 = \lambda_1 - 1 = -1$$

$$\hat{c}_{13} = c_{13} - \lambda_1 + \lambda_3 = 0 \text{ ou } \lambda_3 = \lambda_1 - 3 = -3$$

$$\hat{c}_{35} = c_{35} - \lambda_3 + \lambda_5 = 0 \text{ ou } \lambda_5 = \lambda_3 - 2 = -5$$

$$\boldsymbol{\lambda} = (0, -1, -3, -101, -5, -100)^T$$

Com os novos multiplicadores, os custos relativos das variáveis não-básicas ficam:

$$\hat{c}_{34} = c_{34} - \lambda_3 + \lambda_4 = 1 + 3 - 101 = -97 \quad (\hat{x}_{34} = u_{34})$$

$$\hat{c}_{23} = c_{23} - \lambda_2 + \lambda_3 = 1 + 1 - 3 = -1 \quad (\hat{x}_{23} = u_{23})$$

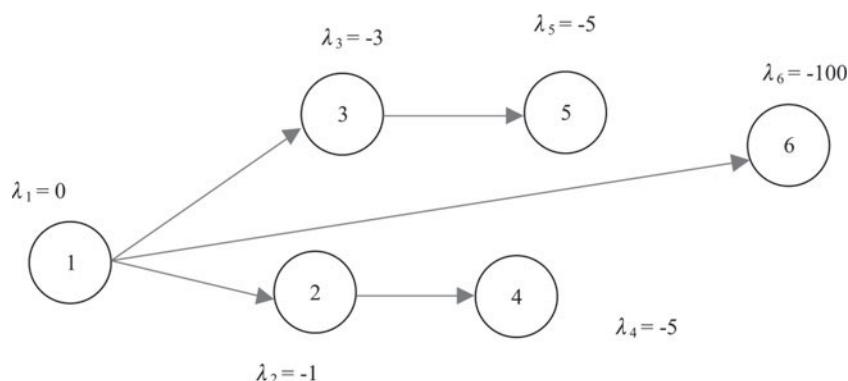
$$\hat{c}_{24} = c_{24} - \lambda_2 + \lambda_4 = 3 + 1 - 101 = -97 \quad (\hat{x}_{24} = \ell_{24}) \leftarrow$$

$$\hat{c}_{54} = c_{54} - \lambda_5 + \lambda_4 = 2 + 5 - 101 = -94 \quad (\hat{x}_{54} = \ell_{54})$$

$$\hat{c}_{56} = c_{56} - \lambda_5 + \lambda_6 = 7 + 5 - 100 = -88 \quad (\hat{x}_{56} = \ell_{56})$$

$$\hat{c}_{15} = c_{15} - \lambda_1 + \lambda_5 = 5 - 0 - 5 = 0 \quad (\hat{x}_{15} = u_{15})$$

O arco (2, 4) tem a maior violação da condição de otimalidade e é escolhido para entrar na base. Com a introdução deste arco, um ciclo é formado: {(2, 4), (4, 6), (1, 6), (1, 2)}. Não se pode circular mais fluxo no ciclo, pois o arco (4, 6) tem seu fluxo no máximo. A variação do fluxo é, portanto, igual a 0 e o arco a ser eliminado para quebrar o ciclo é (4, 6), pois é ele que limita o fluxo no ciclo, resultando na árvore geradora indicada na Figura 4.50. Neste caso, a solução básica permanece inalterada, embora a árvore geradora (base) seja alterada.

**Figura 4.50** Nova árvore geradora após eliminação do arco (4, 6).

### Iteração 6:

Solução básica na iteração 6:

Variáveis básicas:  $\hat{x}_{16}^a = 2$ ,  $\hat{x}_{24} = 0$ ,  $\hat{x}_{12} = 1$ ,  $\hat{x}_{35} = 1$ ,  $\hat{x}_{15} = 2$

Variáveis não-básicas em seus limites superiores:  $\hat{x}_{23} = 1$ ,  $\hat{x}_{15} = 5$ ,  $\hat{x}_{34} = 2$ ,  $\hat{x}_{46} = 2$ . Demais variáveis não-básicas nulas (em seus limites inferiores).

Os novos multiplicadores simplex correspondentes a esta nova base são:

$$\lambda_1 = 0$$

$$\hat{c}_{16} = c_{16} - \lambda_1 + \lambda_6 = 0 \text{ ou } 0 = 100 - \lambda_1 + \lambda_6 = 0 \text{ ou } \lambda_6 = \lambda_1 - 100 = -100$$

$$\hat{c}_{12} = c_{12} - \lambda_1 + \lambda_2 = 0 \text{ ou } \lambda_2 = \lambda_1 - 1 = -1$$

$$\hat{c}_{24} = c_{24} - \lambda_2 + \lambda_4 = 0 \text{ ou } \lambda_4 = \lambda_2 - 3 = -4$$

$$\hat{c}_{13} = c_{13} - \lambda_1 + \lambda_3 = 0 \text{ ou } \lambda_3 = \lambda_1 - 3 = -3$$

$$\hat{c}_{35} = c_{35} - \lambda_3 + \lambda_5 = 0 \text{ ou } \lambda_5 = \lambda_3 - 2 = -5$$

$$\boldsymbol{\lambda} = (0, -1, -3, -4, -5, -100)^T$$

Com os novos multiplicadores, os custos relativos das variáveis não básicas ficam:

$$\hat{c}_{34} = c_{34} - \lambda_3 + \lambda_4 = 1 + 3 - 4 = 0 \quad (\hat{x}_{34} = u_{34})$$

$$\hat{c}_{23} = c_{23} - \lambda_2 + \lambda_3 = 1 + 1 - 3 = -1 \quad (\hat{x}_{23} = u_{23})$$

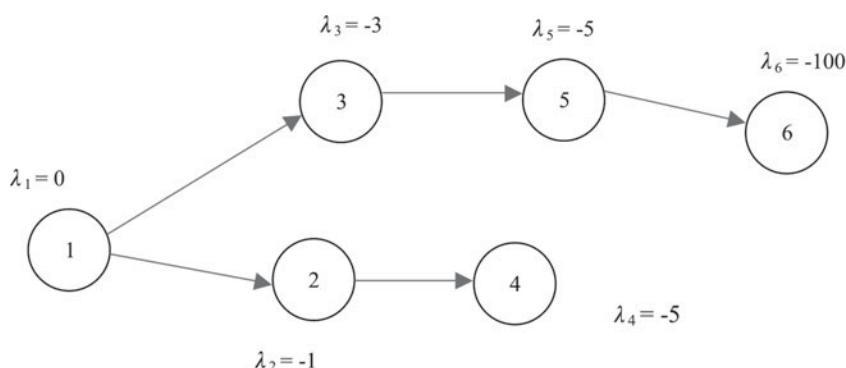
$$\hat{c}_{46} = c_{46} - \lambda_4 + \lambda_6 = 1 + 4 - 100 = -95 \quad (\hat{x}_{46} = u_{46})$$

$$\hat{c}_{54} = c_{54} - \lambda_5 + \lambda_4 = 2 + 5 - 4 = 3 \quad (\hat{x}_{54} = \ell_{54})$$

$$\hat{c}_{56} = c_{56} - \lambda_5 + \lambda_6 = 7 + 5 - 100 = -88 \quad (\hat{x}_{56} = \ell_{56}) \leftarrow$$

$$\hat{c}_{15} = c_{15} - \lambda_1 + \lambda_5 = 5 - 0 - 5 = 0 \quad (\hat{x}_{15} = u_{15})$$

O arco (5, 6) tem a maior violação da condição de otimalidade e é escolhido para entrar na base. Com a introdução deste arco, um ciclo é formado: {(5, 6), (1, 6)<sup>a</sup>, (1, 3), (3, 5)}. O fluxo no arco (5, 6) pode ser aumentado de no máximo 2 unidades, pois o fluxo no arco (1, 6)<sup>a</sup> reduz-se ao limite inferior. O fluxo no novo arco é, portanto, igual a 2 e o arco a ser eliminado para quebrar o ciclo é (1, 6)<sup>a</sup>, resultando na árvore geradora indicada na Figura 4.51. Apenas os seguintes fluxos no ciclo são alterados:  $x_{56} = 2$  (entra na base),  $x_{16}^a = 0$  (sai da base no limite inferior),  $x_{13} = 2$ ,  $x_{35} = 2$  e os demais permanecem inalterados.



**Figura 4.51** Nova árvore geradora após eliminação do arco (1, 6)<sup>a</sup>.

**Iteração 7:**Solução básica na iteração 7:Variáveis básicas:  $\hat{x}_{56} = 2$ ,  $\hat{x}_{24} = 0$ ,  $\hat{x}_{12} = 1$ ,  $\hat{x}_{35} = 3$ ,  $\hat{x}_{13} = 4$ Variáveis não-básicas em seus limites superiores:  $\hat{x}_{23} = 1$ ,  $\hat{x}_{15} = 5$ ,  $\hat{x}_{34} = 2$ ,  $\hat{x}_{46} = 2$ . Demais variáveis não-básicas nulas (em seus limites inferiores).

Os novos multiplicadores simplex correspondentes a esta nova base são:

$$\lambda_1 = 0$$

$$\hat{c}_{12} = c_{12} - \lambda_1 + \lambda_2 = 0 \text{ ou } \lambda_2 = \lambda_1 - 1 = -1$$

$$\hat{c}_{24} = c_{24} - \lambda_2 + \lambda_4 = 0 \text{ ou } \lambda_4 = \lambda_2 - 3 = -4$$

$$\hat{c}_{13} = c_{13} - \lambda_1 + \lambda_3 = 0 \text{ ou } \lambda_3 = \lambda_1 - 3 = -3$$

$$\hat{c}_{35} = c_{35} - \lambda_3 + \lambda_5 = 0 \text{ ou } \lambda_5 = \lambda_3 - 2 = -5$$

$$\hat{c}_{56} = c_{56} - \lambda_5 + \lambda_6 = 0 \text{ ou } \lambda_6 = \lambda_5 - 7 = -12$$

$$\boldsymbol{\lambda} = (0, -1, -3, -4, -5, -12)^T$$

Com os novos multiplicadores, os custos relativos das variáveis não-básicas ficam (a variável artificial  $x_{16}^a$  é desconsiderada):

$$\hat{c}_{34} = c_{34} - \lambda_3 + \lambda_4 = 1 + 3 - 5 = -1 \quad (\hat{x}_{34} = u_{34})$$

$$\hat{c}_{23} = c_{23} - \lambda_2 + \lambda_3 = 1 + 1 - 3 = -1 \quad (\hat{x}_{23} = u_{23})$$

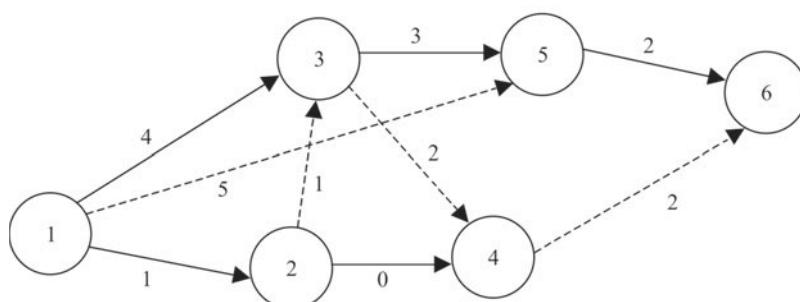
$$\hat{c}_{46} = c_{46} - \lambda_4 + \lambda_6 = 1 + 5 - 12 = -6 \quad (\hat{x}_{46} = u_{46})$$

$$\hat{c}_{54} = c_{54} - \lambda_5 + \lambda_4 = 2 + 5 - 5 = 2 \quad (\hat{x}_{54} = \ell_{54})$$

$$\hat{c}_{15} = c_{15} - \lambda_1 + \lambda_5 = 5 - 0 - 5 = 0 \quad (\hat{x}_{15} = u_{15})$$

Todos os arcos com  $\hat{c}_{ij} \leq 0$  estão com seus fluxos nos limites superiores, todos os arcos com  $\hat{c}_{ij} \geq 0$  estão com seus fluxos nos limites inferiores. Portanto, não é mais possível melhorar a solução atual. O custo da presente solução ótima indicada na Figura 4.52 é  $(5 \times 5 + 1 \times 1 + 1 \times 1 + 2 \times 1 + 2 \times 7 + 2 \times 1 + 3 \times 2 + 4 \times 3) = 63$ .**Propriedade 4.4** (propriedade de integralidade) Se os dados do problema de fluxo em redes a custo mínimo são inteiros, então todos os sistemas básicos têm solução inteira, em particular, uma solução básica ótima é inteira.

A Propriedade 4.4 segue da observação de que os sistemas básicos (para o cálculo das variáveis básicas, vetor multiplicador simplex e direção simplex) são resolvidos por substituição e somente operações de adição e subtração são necessárias, o que não altera a integralidade dos dados

**Figura 4.52** Fluxo de custo mínimo.

de entrada. Com isso, se um problema de fluxos em redes tem a restrição de integralidade, então esta restrição pode ser abandonada sem perda de generalidade. O exemplo a seguir define um caso particular de fluxos em redes, chamado problema de designação, o qual é formulado por variáveis binárias.

**Exemplo 4.23** Considere o caso de uma empresa que tem a necessidade de preencher  $n$  vagas de engenheiros em sua fábrica. A empresa seleciona  $n$  engenheiros para as vagas e deseja saber qual a melhor atribuição desses engenheiros para as vagas disponíveis. A empresa realizou testes de aptidão, dispõe do histórico escolar, experiência profissional, cartas de recomendação, perfil psicológico etc. desses  $n$  engenheiros. Com base nesses dados, a empresa elaborou um índice de capacitação/eficiência de cada engenheiro para cada uma das vagas disponíveis na fábrica:  $c_{ij}$  é a capacitação/eficiência do engenheiro  $i$  em relação à vaga  $j$ . A empresa deseja saber qual a atribuição que deve ser feita dos  $n$  engenheiros para as  $n$  vagas de modo a maximizar o índice total de capacitação/eficiência para todas as vagas da empresa.

Este problema é conhecido como o problema de designação ou atribuição (veja Seções 2.2.2 e 3.4 nos Capítulos 2 e 3) e sua formulação é dada a seguir. Seja  $x_{ij} = 1$  se o engenheiro  $i$  é designado à vaga  $j$  e  $x_{ij} = 0$ , caso contrário. A variável de decisão é, portanto, binária.

$$\begin{aligned} \text{Maximizar } f(x) &= \sum_{i,j, i \neq j} c_{ij} x_{ij} \\ \sum_{j, j \neq i} x_{ij} &= 1 \quad i = 1, 2, \dots, n \\ \sum_{k, k \neq j} x_{kj} &= 1 \quad j = 1, 2, \dots, n \\ x_{ij} &= 0 \text{ ou } 1 \quad i \text{ é um nó de oferta e } j \text{ é um nó de demanda} \end{aligned}$$

O primeiro conjunto de restrições impõe que cada engenheiro é atribuído a apenas uma das vagas; o segundo conjunto de restrições impõe que cada vaga seja preenchida apenas por um dos engenheiros. Da maneira como foi formulado, este problema não é de otimização linear e, a princípio, o método simplex não poderia ser aplicado (veja Capítulo 3 para métodos de otimização discreta). Entretanto, as soluções básicas são sempre inteiras (mesmo que  $c_{ij}$  não seja inteiro, o que faz com que apenas os vetores multiplicadores não sejam inteiros), já que as “demandas” ou “ofertas” são inteiras (iguais a  $-1$  e  $1$ , bastando multiplicar por  $-1$  as  $n$  últimas equações para que o problema fique na forma de (4.1) – (4.3)). Portanto, este problema pode ser resolvido com  $x_{ij} \geq 0$  pelo método simplex para redes, desenvolvido anteriormente, e as soluções básicas têm valores binários:  $0$  ou  $1$ .

É possível, ainda, desenvolver algoritmos especializados para o problema de designação, como, por exemplo, o método húngaro.<sup>9</sup> Inúmeros outros problemas de otimização em redes que representam problemas práticos foram extensivamente estudados e podem ser encontrados nos diversos livros e artigos especializados que tratam do assunto.<sup>10</sup>

#### 4.4 EXERCÍCIOS<sup>11</sup>

**Exercício 4.1** Modifique o algoritmo de Dijkstra para o caso de se desejar determinar o caminho mais curto do nó 1 a todos os demais nós no grafo.

9 Ver, por exemplo, Bazaraa *et al.* (1990).

10 Ver, por exemplo, Ahuja *et al.* (1993), Goldbarg e Luna (2000) e Jungnickel (1994).

11 Alguns exercícios deste capítulo foram inspirados em exercícios sugeridos em Bazaraa *et al.* (1990) e Ahuja *et al.* (1993).

**Exercício 4.2** Ache o caminho mais curto de todos os pares de nós do grafo da Figura 4.10 utilizando o algoritmo de Dijkstra.

**Exercício 4.3** Ache o caminho mais curto de todos os pares de nós do grafo da Figura 4.10 utilizando o algoritmo de Floyd.

**Exercício 4.4** Ache o caminho mais longo do nó 5 ao nó 2 do grafo da Figura 4.10.

**Exercício 4.5** Identifique o conjunto de vértices e o conjunto de arcos da rede esquematizada na Figura 4.40. Quais nós desta rede são de transbordo? Formule o problema de fluxo a custo mínimo desta rede e mostre a matriz de incidência nó-arco do problema.

**Exercício 4.6** Considere a rede com sete nós  $\{1, 2, 3, 4, 5, 6, 7\}$ , definida pela tabela a seguir.

Arco	Capacidade máxima	Arco	Capacidade máxima
(1, 2)	6	(2, 5)	4
(1, 3)	7	(2, 3)	1
(2, 4)	3	(3, 4)	2
(3, 6)	5	(4, 6)	2
(4, 5)	3	(6, 5)	2
(5, 7)	7	(6, 7)	4

- a) Esquematize a rede definida pela tabela.
- b) Determine a matriz de incidência nó-arco associada a esta rede.
- c) Ache o fluxo máximo do nó 1 ao nó 7 na rede.
- d) Identifique o corte mínimo associado ao fluxo máximo.

**Exercício 4.7** Uma companhia petrolífera tem uma programação de trabalho para os próximos seis meses em uma certa plataforma que requer um certo número de pessoas altamente qualificadas. As necessidades de pessoal qualificado são dados na seguinte tabela:

Mês	1	2	3	4	5	6
Pessoal	3	5	6	3	5	1

Os custos incorridos pela empresa são:

- Custos de transferência

A adição de mais uma pessoa qualificada na plataforma custa \$10 mil por trabalhador, e o remanejamento de um trabalhador qualificado para outra plataforma custa \$16 mil.

- A companhia incorre em um custo de \$20 mil por mês por trabalhador qualificado ocioso na plataforma e um custo de \$20 mil por mês por trabalhador, pela falta do número ne-

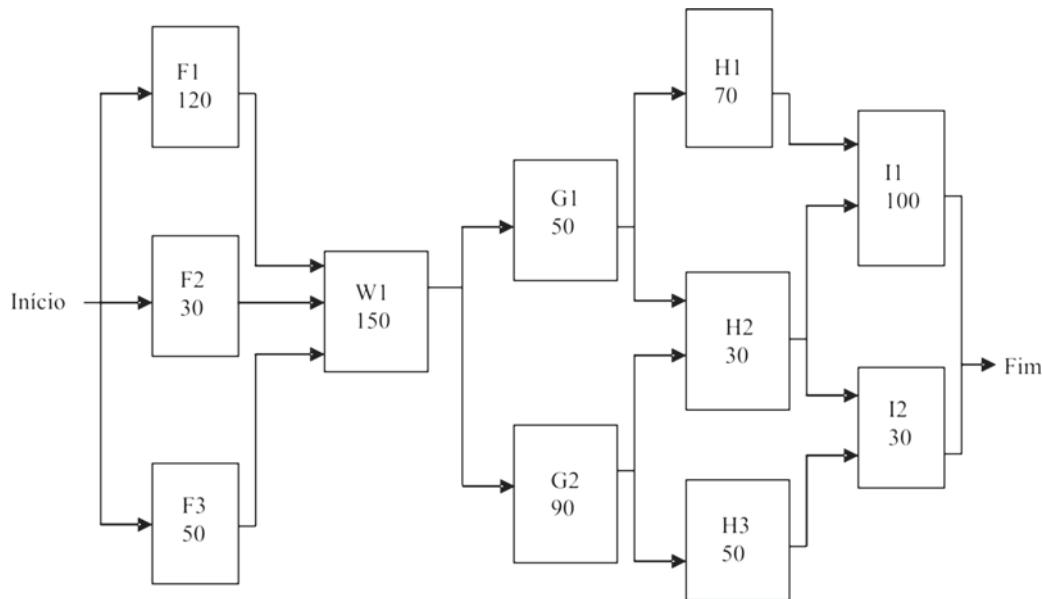
cessário na plataforma. A falta de pessoal qualificado implica que horas extras precisam ser realizadas pelos demais, e estas são limitadas a 25% das horas regulares de trabalho.

Existem ainda algumas regras de transferência que precisam ser obedecidas pela empresa, fruto de negociações com o sindicato dos trabalhadores. A empresa não pode transferir mais de que dois trabalhadores qualificados no início de cada mês e não pode remanejar para outra plataforma, no final de cada mês, mais de 1/3 dos trabalhadores qualificados trabalhando naquela plataforma.

Suponha que atualmente encontram-se dois trabalhadores na plataforma e, ao final dos próximos seis meses, necessita-se também de dois trabalhadores. O problema é determinar quantos trabalhadores a empresa deverá manter em cada um dos meses de modo a minimizar seus custos.

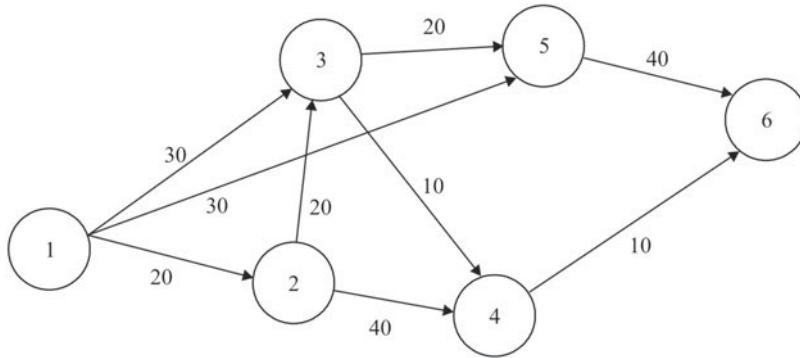
Formule este problema como um problema de caminho mínimo e resolva-o. (Dica: utilize nós em cada mês correspondendo aos possíveis números de trabalhadores qualificados.)

**Exercício 4.8** Uma empresa tem instalada uma série de equipamentos e estações de trabalho que podem ser utilizados na montagem de um novo produto. A figura a seguir esquematiza os caminhos possíveis para a montagem do produto na fábrica.



No esquema, tem-se a indicação do setor e a quantidade máxima correspondente de itens que podem ser manuseados (capacidade máxima do setor) por mês. Dadas as instalações existentes na empresa, qual é a produção máxima possível mensal do produto? Se a demanda do mercado for maior que a capacidade produtiva da fábrica, em que setor a empresa deve investir para aumentar sua capacidade?

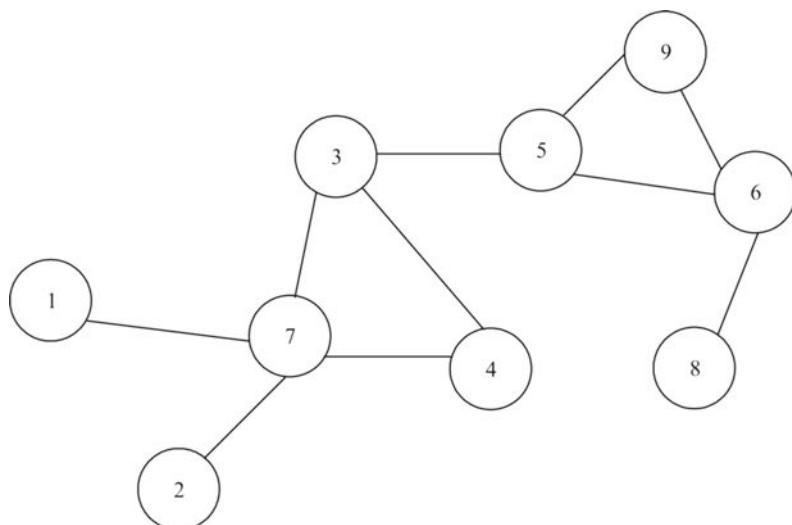
**Exercício 4.9** Suponha que uma empresa transportadora de cargas possui três caminhões que estão atualmente nas posições 1 e 2 da rede rodoviária de uma cidade esquematizada a seguir. No nó 1 estão dois caminhões da empresa e no nó 2 está o terceiro caminhão. Um novo pedido de frete foi solicitado na localidade 6. Para este pedido, são necessários três caminhões. Além disso, a carga precisa ser retirada no máximo dentro de 50 minutos, pois depois desse prazo não é mais possível estacionar o caminhão próximo do local, por problemas de tráfego. Determine se a empresa transportadora poderá atender a este pedido. Os números ao lado de cada arco representam o número de minutos para um caminhão percorrer o arco.



**Exercício 4.10** Uma empresa que consome altos níveis de energia elétrica precisa decidir de que fontes geradoras ela comprará energia para realizar sua produção. A seguinte rede representa a rede de distribuição de energia elétrica disponível no momento. Na rede, estão indicados os pontos de geração de energia, os pontos de demanda de energia da empresa e as linhas de transmissão existentes. As linhas de transmissão são não direcionadas, isto é, a energia pode fluir em ambas as direções. Os pontos 1, 2, 3 e 4 são fontes geradoras de energia. A capacidade de geração de energia em cada uma dessas fontes e o custo por unidade gerada são os seguintes:

	Fontes geradoras			
	1	2	3	4
Capacidade (Kwh)	100.000	60.000	80.000	150.000
Custo unitário por Kwh	17,5	23	21	22

Os pontos 8 e 9 são pontos de consumo com demandas 85.000Kwh e 100.000Kwh, respectivamente. As linhas de transmissão estão dimensionadas para aguentar fluxos muito altos de energia, assim, para fins práticos, não há limite na capacidade de transmissão nas linhas. No entanto, existe um custo unitário de transmissão, em cada segmento, de \$15 por 1.000Kwh. Formule o problema de distribuição de energia como um problema de fluxo em rede e resolva o problema pelo método simplex para redes.

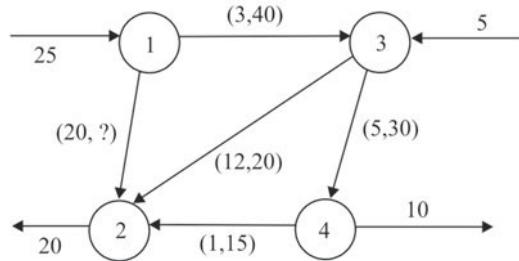


**Exercício 4.11** Resolva o seguinte problema de designação

		Tarefas		
		T1	T2	T3
Pessoas	A	7	6	5
	B	3	8	6
	C	4	9	10

como um problema de fluxos em redes.

**Exercício 4.12** Na rede a seguir, sobre os arcos, são dados os parâmetros correspondentes de custo e limite superior de fluxo. O limite inferior é 0 para todos os arcos. Aplique o método simplex para fluxo em redes e encontre a solução ótima que minimiza o custo total. Parta da seguinte base inicial: (1, 2), (1, 3) e (3, 4), com todas as variáveis não-básicas no limite inferior.



**Exercício 4.13** Duas fazendas ( $F_1, F_2$ ) produtoras de carne costumam utilizar três centros de processamento ( $P_1, P_2$  e  $P_3$ ) como intermediários para atender seus principais clientes ( $C_1, C_2$  e  $C_3$ ). Nas tabelas a seguir, são apresentados os custos de transporte por tonelada de carne entre as fazendas e os centros de processamento e entre estes e os clientes.

	$P_1$	$P_2$	$P_3$
$F_1$	10	8	$+\infty$
$F_2$	$+\infty$	6	11

	$C_1$	$C_2$	$C_3$
$P_1$	7	5	$+\infty$
$P_2$	5	4	3
$P_3$	$+\infty$	8	6

A disponibilidade de frota de caminhões limita o número de toneladas que pode ser transportado entre as fazendas, os centros de processamento e os clientes conforme os dados fornecidos nas tabelas a seguir.

	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>
F <sub>1</sub>	70	50	0
F <sub>2</sub>	0	40	60

	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>
P <sub>1</sub>	40	30	0
P <sub>2</sub>	25	36	29
P <sub>3</sub>	0	34	32

Atualmente, as fazendas estão sofrendo grande instabilidade na produção e também na aceitação de seus produtos devido a casos de febre aftosa. Modele a rede balanceada e escreva o modelo matemático para minimizar os custos de transporte quando:

- A produção de F<sub>1</sub> está entre 50 e 80 toneladas de carne, assim como a produção de F<sub>2</sub> está entre 60 e 90 toneladas de carne. Devido a obras de infra-estrutura, o centro de processamento P<sub>2</sub> não suporta processar mais 60 toneladas de carne. Já P<sub>3</sub> possui 5 toneladas de carne congelada que poderia disponibilizar a qualquer momento para atendimento dos clientes.
- A produção de F<sub>1</sub> está entre 50 e 80 toneladas de carne, assim como a produção de F<sub>2</sub> está entre 60 e 90 toneladas de carne. A demanda dos clientes, entretanto, está bastante instável. O cliente C<sub>1</sub> pode pedir entre 30 e 40, o cliente C<sub>2</sub> entre 40 e 45, e o cliente C<sub>3</sub> entre 50 e 60 toneladas de carne.

Página deixada intencionalmente em branco

# 5

## Programação dinâmica determinística

Uma das técnicas que podem ser utilizadas para resolver problemas de otimização é a *programação dinâmica*. Ela tem sido aplicada com sucesso na resolução de problemas originários das mais diversas áreas. Entre as várias aplicações, incluem-se problemas de planejamento da produção, gestão de estoques, determinação de tamanhos de lotes de produção, problemas de corte e empacotamento, alocação de recursos limitados, determinação da capacidade de expansão de usinas geradoras de energia elétrica, gestão de projetos, gestão financeira, determinação da confiabilidade de sistemas de comunicação, programação de tarefas em sistemas de manufatura flexível, manutenção e reposição de equipamentos, entre outras.

A programação dinâmica pode ser aplicada na resolução de problemas gerais de otimização linear ou não-linear, contínua ou discreta, com funções diferenciáveis ou não-diferenciáveis etc. A principal característica da programação dinâmica consiste na decomposição do problema original em uma seqüência de problemas menores e mais simples de serem resolvidos. Para alguns problemas, a maneira de realizar essa decomposição é aparente. É o caso de situações que apresentam seqüências de decisões a serem tomadas em estágios sucessivos, por exemplo, quando as decisões são definidas ao longo de uma seqüência de instantes de tempo.

### 5.1 UM EXEMPLO SIMPLES

A seguir, introduzimos a programação dinâmica por meio de um simples exemplo. A partir da Seção 5.3, formalizamos a aplicação da programação dinâmica em problemas gerais. Considere o seguinte problema:

**Exemplo 5.1** A demanda mensal de aço tipo A produzido por uma empresa siderúrgica precisa ser atendida sem atrasos, sob pena de perda de vendas. A siderúrgica não dispõe, no momento, de nenhum estoque desse tipo de aço, e tem conhecimento de que sua demanda para os próximos quatro meses ( $t = 1, 2, 3, 4$ ) é de 30, 20, 60 e 10 toneladas, respectivamente. O custo de produção do aço tipo A é constante durante os quatro meses e igual a \$2 mil por tonelada produzida, enquanto o custo de estocagem do aço de um mês para outro corresponde a 10% do custo de produção do aço armazenado. Devido a restrições tecnológicas, a produção do aço tipo A precisa ser feita em lotes e em quantidades predeterminadas múltiplas de 10 toneladas; ou seja, lotes de tamanhos 10, 20, 30 toneladas, e assim por diante. Além disso, a cada lote produzido, independentemente de seu tamanho, incorre-se em um custo fixo de preparação de \$10 mil. Qual o plano de produção de aço tipo A que minimiza os custos de produção, estocagem e preparação para esses quatro meses?

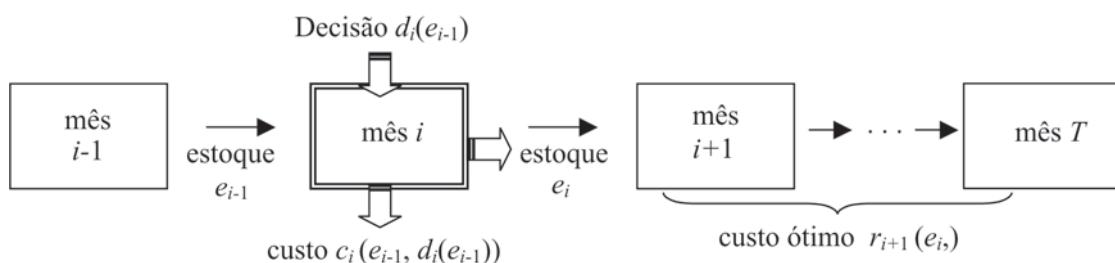
O Exemplo 5.1 é um problema de dimensionamento de lotes, já discutido nos Capítulos 2 e 3 (Seções 2.2 e 3.6.2). Nesses problemas, caso não houvesse um custo fixo de preparação, o plano de custo mínimo de produção seria o de produzir em cada mês apenas o suficiente para suprir a demanda do próprio mês; ou seja, produzir 30, 20, 60 e 10 toneladas do aço tipo A nos meses 1, 2, 3 e 4, respectivamente, isto é, aplicar uma estratégia lote por lote.<sup>1</sup> Com a aplicação desse plano, incorre-se em um custo de produção de  $\$2 \times (30 + 20 + 60 + 10) = \$240$  mil, correspondente ao menor custo possível para a produção de 120 toneladas de aço tipo A.

Como o custo unitário de produção é constante ao longo do tempo, não é vantajoso estocar nenhum material de um mês para outro, uma vez que, em adição a esse custo mínimo de produção, incorre-se também no custo de estocagem do material. Com um custo fixo de preparação de \$10 mil, haveria um adicional de \$40 mil em relação ao custo do plano anterior, devido à produção de cada um dos lotes de cada mês, totalizando  $\$240 + \$40 = \$280$  mil. A questão de interesse é a seguinte: será que existe algum plano de produção alternativo com um custo menor que \$280 mil?

No Exemplo 5.1, é necessário determinar o quanto produzir em cada um dos próximos quatro meses. O problema seria mais fácil de ser resolvido se pudesse ser decomposto em uma seqüência de problemas mais simples de um mês cada um, em que a decisão se limitasse a determinar o que produzir naquele mês. O número total de períodos (meses) do problema em que as decisões devem ser tomadas corresponde ao horizonte de planejamento. Neste exemplo, o horizonte de planejamento é igual a 4.

A quantidade de aço tipo A a ser produzida em um determinado mês depende apenas do conhecimento do estoque disponível no início deste mês. As decisões possíveis de serem tomadas são os diversos tamanhos de lotes de produção, que, juntamente com o estoque já disponível, devem ser suficientes para pelo menos atender a demanda daquele mês e, possivelmente, ser utilizado para atender as demandas dos meses subseqüentes. Note que cada uma das possíveis decisões em um dado mês leva o sistema a um certo nível de estoque no início do mês seguinte.

Considere o diagrama apresentado na Figura 5.1, em que são representados de forma genérica os meses envolvidos no planejamento da produção de um dado produto, com a possibilidade de estocagem de um mês para outro. Note que o estoque inicial do produto no mês  $i$  corresponde ao estoque final do produto no mês  $i - 1$  ( $e_{i-1}$ ).<sup>2</sup> Com base no valor de  $e_{i-1}$ , no mês  $i$  tomamos uma decisão  $d_i(e_{i-1})$  de produzir uma certa quantidade do produto. Esta decisão incorre em um custo  $c_i(e_{i-1}, d_i(e_{i-1}))$ , e em um estoque no final do mês  $i$  (e, portanto, em um estoque inicial do mês  $i + 1$ ) igual a  $e_i$ . Note que este estoque inicial do mês  $i + 1$  ( $e_i$ ) é igual ao estoque inicial do mês  $i$  ( $e_{i-1}$ ), mais a quantidade produzida no mês  $i$  ( $d_i(e_{i-1})$ ), menos a demanda no mês  $i$ .



**Figura 5.1** Meses envolvidos no planejamento da produção.

1 Também chamada *chase strategy*; ver, por exemplo, Nahmias (1995).

2 Por conveniência, neste Capítulo usamos a notação  $e_i$  para representar o estoque final do período  $i$ , em vez de  $I_i$  usada nos capítulos anteriores.

Se conhecêssemos o custo  $r_{i+1}(e_i)$  correspondente ao melhor plano de produção do mês  $i + 1$  até o último mês  $T$  do horizonte de planejamento (no Exemplo 5.1, correspondente ao mês 4), dado que o estoque inicial do mês  $i + 1$  é  $e_i$ , então seria fácil calcular o custo  $r_i(e_{i-1})$  correspondente ao melhor plano de produção do mês  $i$  até o mês  $T$ , dado que o estoque inicial do mês  $i$  é  $e_{i-1}$ . Para isso, basta determinar no mês  $i$  a decisão  $d_i(e_{i-1})$  que resulta no menor custo total do mês  $i$  até o mês  $T$ .

Seja  $c_i(e_{i-1}, d_i(e_{i-1}))$  o custo incorrido no mês  $i$  se a decisão  $d_i(e_{i-1})$  for tomada e o estoque inicial observado for  $e_{i-1}$ . A decisão  $d_i(e_{i-1})$  que resulta no menor custo total do mês  $i$  até o mês  $T$  é aquela em que  $\{c_i(e_{i-1}, d_i(e_{i-1})) + r_{i+1}(e_i)\}$  é mínimo, com  $r_{T+1}(e_T) = 0$  para qualquer valor de  $e_T$  no início do mês  $T + 1$ .

Observe que, para o cômputo da melhor decisão de um mês  $i$ , é necessário conhecer o plano ótimo a partir do mês subsequente  $i + 1$  até o último mês  $T$ . Com este objetivo, iniciamos os cálculos a partir do mês  $T$ ; após resolvemos os problemas do mês  $T$ , podemos então resolver os problemas do mês  $T - 1$ , depois os do mês  $T - 2$ , e assim por diante, até os do mês 1.

### 5.1.1 Cálculos do exemplo

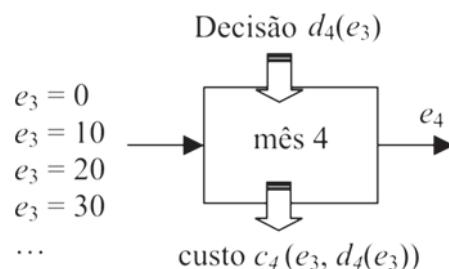
Considere, então, os cálculos para o Exemplo 5.1, onde  $T = 4$ .

#### Cálculos para o mês 4

Antes de iniciar os cálculos, é importante observar que, como a produção se faz em lotes múltiplos de 10, o estoque inicial do aço tipo A no mês 1 é de  $e_0 = 0$  toneladas, e as demandas observadas em cada um dos quatro meses também são múltiplas de 10. Assim, os possíveis valores observáveis de estoques em qualquer início de mês são também múltiplos de 10, ou seja, 0, 10, 20, e assim por diante, conforme indicado na Figura 5.2.

Suponhamos que  $e_3 = 0$ , ou seja, o estoque inicial do mês 4 é 0. Como a demanda de 10 toneladas do mês 4 precisa ser atendida, é preciso produzir no mínimo 10 toneladas. As decisões  $d_4(0)$  possíveis são, portanto, produzir lotes de 10, 20, 30, ... toneladas de aço tipo A. Produzindo-se a quantidade mínima, finaliza-se o mês 4 e inicia-se o mês 5 com um estoque  $e_4 = 0$ , a um custo  $c_4(0, 10)$  composto pela soma dos custos de preparação (\$10 mil) e de produção do lote (\$2 mil x 10), ou seja,  $c_4(0, 10) = \$10 + \$2 \times 10 = \$30$  mil. Note que não há custo de estoque neste caso.

As outras decisões possíveis levam a estoques maiores que 0 no fim do mês 4 e, portanto, custos associados superiores a \$30 mil, ou seja,  $c_4(0, d_4(0)) > \$30$  mil, para  $d_4 = 20, 30, \dots$ . Como o mês



**Figura 5.2** Possíveis níveis de estoque no início do mês 4 (Exemplo 5.1).

4 é o último mês do horizonte de planejamento, um plano de custo mínimo impõe que o estoque final deste mês seja igual a 0.<sup>3</sup> Conclui-se, portanto, que:

$$r_4(0) = \min \{c_4(0,10) + r_5(0), c_4(0,20) + r_5(10), c_4(0,30) + r_5(20), \dots\} = \$30 \text{ mil}$$

(lembmando que  $r_5(e_4) = 0$  para  $e_4 = 0, 10, 20, \dots$ ) e a decisão ótima para o problema (assinalada com “\*” e representada na Figura 5.3) com  $e_3 = 0$  é  $d_4^*(0) = 10 \text{ t}$ .

A seguir, suponhamos que  $e_3 = 10$ . Como já temos em estoque 10 toneladas do aço A, as decisões  $d_4(10)$  possíveis são produzir 0, 10, 20, 30, ... toneladas de aço A. Se não se produzir nada nesse período, chega-se ao final do mês 4 com um estoque  $e_4 = 0$ . Neste caso,  $c_4(10,0) = \$0$ . Todas as demais decisões possíveis levam a estoques maiores que 0 no final do mês 4 (ou início do mês 5) e têm custos associados superiores a \$0. Portanto,  $r_4(10) = c_4(10,0) + r_5(0) = \$0$ , e a decisão ótima para este subproblema é  $d_4^*(10) = 0 \text{ t}$ . De maneira similar, calculam-se os demais valores de  $r_4(e_3)$  para  $e_3 > 10$ . Os valores de  $r_4(e_3)$  para  $e_3 > 10$  não são explicitados aqui, mas é fácil verificar que satisfazem a relação  $r_4(e_3) \geq \$2 \text{ mil}$ , devido ao custo de estocagem ser maior ou igual a  $0,10 (10 \times \$2,00) = \$2 \text{ mil}$ .

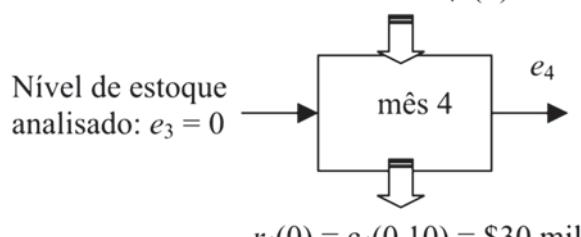
Na Tabela 5.1, apresentamos os resultados obtidos de planos ótimos para o mês 4 para os possíveis valores de  $e_3$ . Uma vez calculados os  $r_4(e_3)$  para todos os valores possíveis de  $e_3$ , passamos aos cálculos do mês 3.

### Cálculos do mês 3

Suponhamos que  $e_2 = 0$ , ou seja, o estoque inicial do mês 3 é 0. Como a demanda de 60 toneladas do mês 3 precisa ser atendida, é preciso produzir no mínimo esta quantidade. As decisões  $d_3(0)$  possíveis são, portanto, produzir 60, 70, 80, ... toneladas do aço A. Produzindo a quantidade mínima, chegamos ao mês 4 com estoque  $e_3 = 0$ , a um custo  $c_3(0,60)$  de  $c_3(0,60) = \$10 + \$2 \times 60 = \$130 \text{ mil}$ . O custo mínimo de produção do mês 4 (mês final do planejamento), obtido ao se iniciar com nível de estoque igual a 0, já foi calculado e é igual a  $r_4(0) = \$30 \text{ mil}$  (Tabela 5.1). Portanto, o custo total incorrido se decidirmos produzir 60 toneladas no mês 3 é de  $c_3(0,60) + r_4(0) = \$160 \text{ mil}$ .

Decisões possíveis:  $d_4(0) \in \{10, 20, 30, \dots\}$

Decisão ótima:  $d_4^*(0) = 10 \text{ t}$



**Figura 5.3** Decisão ótima para o mês 4 com nível de estoque inicial  $e_3 = 0$  (Exemplo 5.1).

<sup>3</sup> Em aplicações práticas, é comum o planejamento da produção se basear em um *horizonte rolante*. Somente as decisões dos meses  $1 + x$  ( $x \geq 0$  e inteiro) são de fato implementadas, enquanto as demais são refeitas em intervalos geralmente regulares. Na primeira reavaliação do plano, por exemplo, toma-se como primeiro mês o mês  $(1 + x) + 1$ , mantendo-se o tamanho do horizonte de planejamento igual a  $T$  meses. Como o mês  $T$  do plano original deixa de ser o último mês no novo plano, seu estoque final praticado não é necessariamente igual a 0.

**Tabela 5.1**  
**Planos ótimos para o mês 4 (Exemplo 5.1).**

<i>Mês 4</i>		
$e_3$	$d_4^*(e_3)$	$r_4(e_3)$
0	10	30
10	0	0
$\geq 20$	0	$\geq 2$

Se, por outro lado, produzirmos 70 toneladas no mês 3, inicia-se o mês 4 com estoque  $e_4 = 10$ . O custo total incorrido nos meses 3 e 4 consiste, portanto, dos custos de preparação (\$10 mil), produção (\$2 x 70 mil) e estoque ( $0,1 \times \$2$  mil x 10) no mês 3 adicionados a  $r_4(10)$  (que é igual a \$0, ver Tabela 5.1), ou seja,  $c_3(0,70) + r_4(10) = \$10 + \$2 \times 70 + 0,1 \times \$2 \times 10 + \$0 = \$152$  mil. De maneira similar, calculam-se os custos incorridos com os demais tamanhos possíveis de lotes de produção, obtendo-se:

- $r_3(0) = \min\{c_3(0,60) + r_4(0), c_3(0,70) + r_4(10), c_3(0,80) + r_4(20), \dots\} = \$152$  mil e  $d_3^*(0) = 70$  t.

Prosseguindo de forma similar, determina-se:

- $r_3(10) = \min\{c_3(10,50) + r_4(0), c_3(10,60) + r_4(10), c_3(10,70) + r_4(20), \dots\} = \$132$  mil e  $d_3^*(10) = 60$  t
- $r_3(20) = \min\{c_3(20,40) + r_4(0), c_3(20,50) + r_4(10), c_3(20,60) + r_4(20), \dots\} = \$112$  mil e  $d_3^*(20) = 50$  t
- $r_3(30) = \min\{c_3(30,30) + r_4(0), c_3(30,40) + r_4(10), c_3(30,50) + r_4(20), \dots\} = \$92$  mil e  $d_3^*(30) = 40$  t
- $r_3(40) = \min\{c_3(40,20) + r_4(0), c_3(40,30) + r_4(10), c_3(40,40) + r_4(20), \dots\} = \$72$  mil e  $d_3^*(40) = 30$  t
- $r_3(50) = \min\{c_3(50,10) + r_4(0), c_3(50,20) + r_4(10), c_3(50,30) + r_4(20), \dots\} = \$52$  mil e  $d_3^*(50) = 20$  t
- $r_3(60) = \min\{c_3(60,0) + r_4(0), c_3(60,10) + r_4(10), c_3(60,20) + r_4(20), \dots\} = \$30$  mil e  $d_3^*(60) = 0$  t
- $r_3(70) = \min\{c_3(70,0) + r_4(10), c_3(70,10) + r_4(20), c_3(70,20) + r_4(30), \dots\} = \$2$  mil e  $d_3^*(70) = 0$  t
- $r_3(e_2) \geq \$6$  mil e  $d_3^*(e_2) = 0$  t para  $e_2 \geq 80$

Na Tabela 5.2, apresentamos os resultados obtidos com planos ótimos para o mês 3.

O uso de um esquema gráfico pode, muitas vezes, ajudar a visualizar a aplicação da programação dinâmica na resolução de um problema. Considere o esquema apresentado na Figura 5.4, elaborado em correspondência aos cálculos efetuados para os meses 3 e 4 do Exemplo 5.1. Nesse esquema, os nós representam todas as possíveis quantidades em estoque do aço tipo A no início dos meses 3 e 4, e o estoque final resultante no mês 4 (ou estoque inicial do mês 5). Um arco ligando dois nós em meses sucessivos indica que o nível do estoque inicial no mês subsequente pode ser alcançado do nível de estoque inicial do mês anterior, por meio de uma decisão de produção factível.

Assim, por exemplo, de um nível de estoque inicial igual a 70 no mês 3, é possível obter níveis de estoque iguais a 10, 20, 30, 40, 50, ... no mês 4, produzindo-se 0, 10, 20, ... toneladas de

**Tabela 5.2**  
**Planos ótimos para o mês 3 (Exemplo 5.1).**

<i>Mês 3</i>		
$e_2$	$d_3^*(e_2)$	$r_3(e_2)$
0	70	152
10	60	132
20	50	112
30	40	92
40	30	72
50	20	52
60	0	30
70	0	2
$\geq 80$	0	$\geq 6$

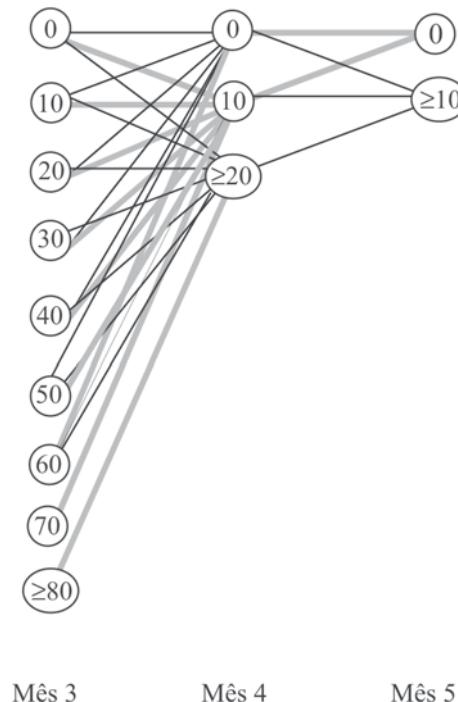
aço, respectivamente. Por outro lado, a partir de um nível de estoque inicial igual a 70 no mês 3, não é possível obter um nível de estoque inicial no mês 4 igual a 0, pois a demanda do mês 3 é de 60 toneladas, ou seja, ainda que o nível de produção no mês 3 seja 0, restarão 10 toneladas em estoque final neste mês e elas estarão presentes no início do mês 4. Note que, por essa razão, não existe um arco ligando esses dois nós no esquema da Figura 5.4.

Os arcos ressaltados na Figura 5.4 correspondem às decisões ótimas já determinadas e que dependem de cada nível de estoque inicial considerado no mês. Uma vez calculados os  $r_3(e_2)$  para todos os valores possíveis de  $e_2$ , de maneira similar passamos aos cálculos do mês 2 e, em seguida, aos cálculos do mês 1. Na Tabela 5.3, estão resumidos os resultados de planos ótimos obtidos desses meses, juntamente com os já calculados para os meses 3 e 4. Observe que, no mês 1, o cálculo a efetuar resume-se apenas a um único nível de estoque inicial fixado em 0.

O plano ótimo de produção para os quatro meses resulta em um custo total de \$266 mil. Este total é composto de custo de produção, custo fixo de preparação e custo de estocagem, respectivamente, \$240, \$20 e \$6 mil.

### 5.1.2 Determinação do plano ótimo

Observe que, no plano ótimo da Tabela 5.3, são produzidas 50 toneladas de aço tipo A no mês 1. Para determinar o que produzir nos demais meses, basta deduzir o valor do estoque inicial nos meses subsequentes, decorrentes da decisão tomada em meses anteriores, e recuperar as decisões ótimas correspondentes. Ao produzir 50 toneladas no mês 1, no início do mês 2 teremos 20 toneladas em estoque, pois a demanda do mês 1 é de 30 toneladas. Com o nível de estoque determinado, a decisão ótima no mês 2 é  $d_2^*(20) = 0$  tonelada. Portanto, nada é produzido no mês 2. Consequentemente, chegamos no início do mês 3 com 0 tonelada de estoque, uma vez que a demanda do mês 2 é de 20 toneladas. Com este nível de estoque, a decisão ótima no mês 3 é  $d_3^*(0) = 70$ . Portanto, devemos produzir 70 toneladas no mês 3. Com isso, chegamos no início do



**Figura 5.4** Possíveis níveis de estoque iniciais nos meses 3, 4 e 5 (nós) e transições factíveis (arcos) entre meses consecutivos.

mês 4 com 10 toneladas em estoque, uma vez que a demanda no mês 3 é de 60 toneladas. Com este nível de estoque, a decisão ótima no mês 4 é  $d_4^*(10) = 0$ . Portanto, não deve haver produção também no mês 4. Desta forma, temos definida toda a produção dos quatro meses do problema.

Na Figura 5.5, destacamos o caminho ótimo correspondente ao plano ótimo determinado no esquema introduzido anteriormente, que ilustra os níveis de estoque e as decisões possíveis em cada um dos quatro meses. O caminho ótimo liga os estoques iniciais que serão observados em cada um dos quatro meses do problema.

O método de cálculo ilustrado neste exemplo é denominado indução regressiva, pois começa no último período (mês 4) e transmite os resultados para o período imediatamente anterior, e assim sucessivamente, até se atingir o período 1.

## 5.2 INDUÇÃO PROGRESSIVA

De maneira análoga à indução regressiva apresentada anteriormente, é possível definir uma indução progressiva que começa no primeiro mês (mês 1) e termina no último mês do horizonte de planejamento.

Na Figura 5.6, são, mais uma vez, representados de forma genérica os meses envolvidos no planejamento da produção de um dado produto com a possibilidade de estocagem de um mês para outro. O estoque inicial no mês  $i$  é igual a  $e_{i-1}$ . Deve ser tomada uma decisão  $d_i(e_i)$  de produzir uma certa quantidade do produto no mês  $i$ , de modo que no final do mês  $i$  seja obtido um nível de estoque igual a  $e_i$ .

O estoque inicial  $e_{i-1}$  do mês  $i$ , que, com a decisão  $d_i(e_i)$ , nos leva ao final do mês  $i$  com um estoque  $e_i$ , pode ser facilmente obtido. Ele é igual ao estoque final do mês  $i$  ( $e_i$ ), menos a quantidade produzida no mês  $i$  ( $d_i(e_i)$ ), mais a demanda do mês  $i$ . Se conhecêssemos o custo  $s_{i-1}(e_{i-1})$

**Tabela 5.3**  
**Planos ótimos para os quatro meses (Exemplo 5.1).**

<i>Mês</i>											
<i>I</i>			<i>II</i>			<i>III</i>			<i>IV</i>		
$e_0$	$d_1^*(e_0)$	$r_1(e_0)$	$e_1$	$d_2^*(e_1)$	$r_2(e_1)$	$e_2$	$d_3^*(e_2)$	$r_3(e_2)$	$e_3$	$d_4^*(e_3)$	$r_4(e_3)$
0	50	266	0	20	202	0	70	152	0	10	30
			10	10	182	10	60	132	10	0	0
			20	0	152	20	50	112	$\geq 20$	0	$\geq 2$
			30	0	134	30	40	92			
			40	0	116	40	30	72			
			50	0	98	50	20	52			
			60	0	80	60	0	30			
			70	0	62	70	0	2			
			80	0	42	$\geq 80$	0	$\geq 6$			
			90	0	16						
	$\geq 100$	0		$\geq 22$							

correspondente ao melhor plano de produção a partir do início do mês 1 até o final do mês  $i - 1$ , dado que o estoque final do mês  $i - 1$  é  $e_{i-1}$ , seria fácil calcular  $s_i(e_i)$ , o custo correspondente ao melhor plano de produção do início do mês 1 até o final do mês  $i$ , dado que o estoque final do mês  $i$  é  $e_i$ . Para isso, basta determinar a decisão  $d_i(e_i)$  que resulta no menor custo total do mês 1 até o mês  $i$  com estoque final  $e_i$ . Seja  $c_i(e_i, d_i(e_i))$  o custo incorrido no mês  $i$  se a decisão  $d_i(e_i)$  for tomada e o estoque final do mês  $i$  observado for  $e_i$ . A decisão  $d_i(e_i)$  que resulta no menor custo total do mês 1 até o mês  $i$  é aquela tal que  $\{s_{i-1}(e_{i-1}) + c_i(e_i, d_i(e_i))\}$  é mínimo, com  $s_0(e_0) = 0$  para qualquer valor de  $e_0$  no final do mês 0.

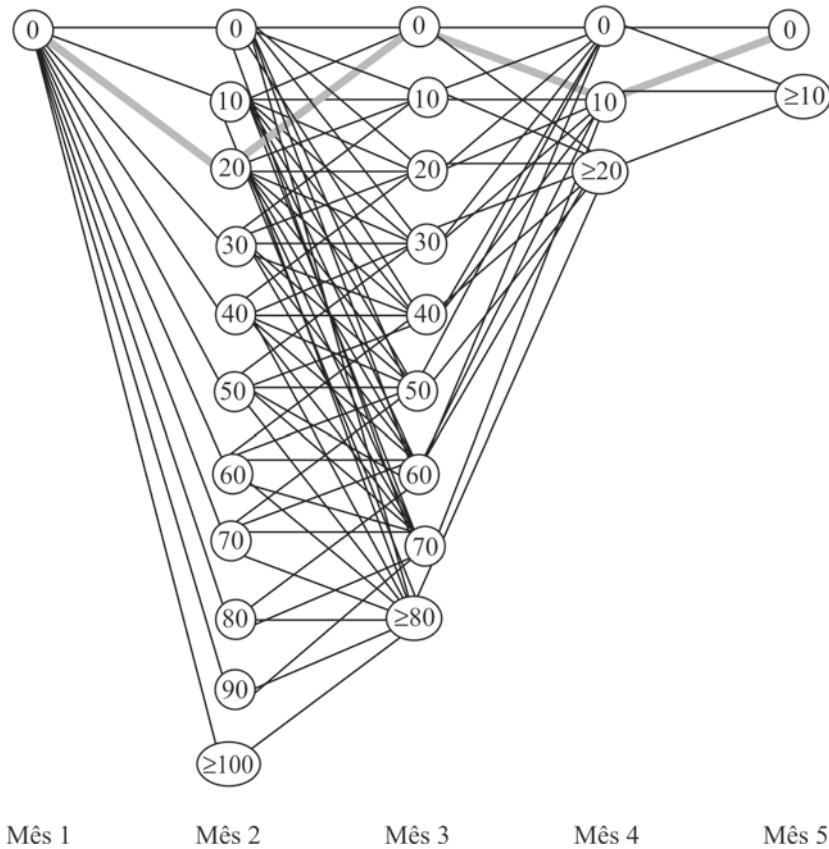
De modo semelhante ao caso de indução regressiva, observe que, para o cômputo da melhor decisão de um mês  $i$  no caso de indução progressiva, necessitamos do conhecimento do plano ótimo do mês 1 até o mês  $i - 1$ , o mês que antecede a  $i$ . Assim, iniciamos os cálculos a partir do primeiro mês. Após resolvemos os problemas do mês 1, podemos resolver os problemas do mês 2, depois os do mês 3, e assim por diante, até os do mês  $T$ .

### 5.2.1 Cálculos da indução progressiva

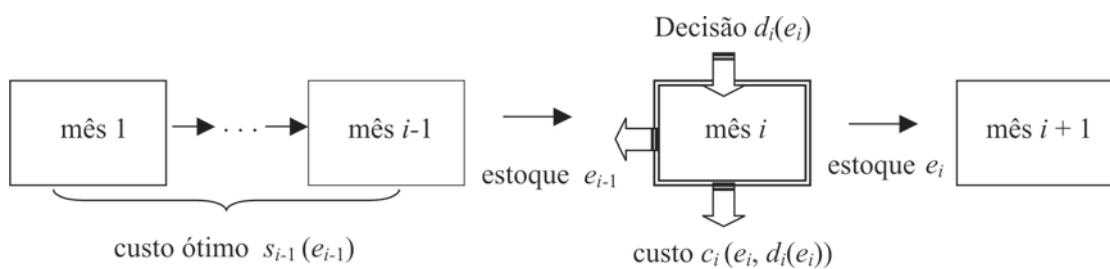
Ilustremos os cálculos da indução progressiva para o Exemplo 5.1.

#### Cálculos para o mês 1

Suponhamos que  $e_1 = 0$ , ou seja, o estoque final do mês 1 é 0. Como iniciamos o mês 1 com um estoque nulo de aço tipo A, e a demanda de 30 toneladas deste mês precisa ser atendida, a úni-



**Figura 5.5** Caminho ótimo dos estoques iniciais, correspondente ao plano ótimo determinado (Exemplo 5.1).



**Figura 5.6** Meses envolvidos no planejamento da produção.

ca decisão  $d_1(0)$  possível que nos conduz ao estoque final do mês 1 igual a 0 é produzir 30 toneladas. O custo associado a esta decisão  $c_1(0,30)$  é  $c_1(0,30) = \$10 + \$2 \times 30 = \$70$  mil. Note que não há custo de estocagem neste caso. Portanto,  $s_1(0) = s_0(e_0) + c_1(0,30) = \$70$  mil (lembrando que  $s_0(e_0) = 0$ ), e a decisão ótima para este problema é  $d_1^*(0) = 30$  t. Consideremos, agora,  $e_1 = 10$ . Para que esta quantidade em estoque ao final do período 1 seja observada, a única decisão possível é produzir 40 toneladas do produto, a um custo de  $c_1(10,40) = \$10 + \$2 \times 40 + 0,1 \times \$2 \times 10 = \$92$  mil. Portanto,

$s_1(10) = \$92$  mil e a decisão ótima para este problema é  $d_1^*(10) = 40$  t.

De maneira similar, calculam-se os demais valores de  $s_1(e_1)$  para  $e_1 > 10$ , conforme se segue:

$$\begin{aligned}
 s_1(20) &= \$114 \text{ mil e } d_1^*(20) = 50t \\
 s_1(30) &= \$136 \text{ mil e } d_1^*(30) = 60t \\
 s_1(40) &= \$158 \text{ mil e } d_1^*(40) = 70t \\
 s_1(50) &= \$180 \text{ mil e } d_1^*(50) = 80t \\
 s_1(60) &= \$202 \text{ mil e } d_1^*(60) = 90t \\
 s_1(70) &= \$224 \text{ mil e } d_1^*(70) = 100t \\
 s_1(80) &= \$246 \text{ mil e } d_1^*(80) = 110t \\
 s_1(90) &= \$268 \text{ mil e } d_1^*(90) = 120t \\
 s_1(e_2) &\geq \$290 \text{ mil e } d_1^*(e_2) = (e_2 + 30) \text{ t para } e_2 \geq 100
 \end{aligned}$$

Uma vez calculados  $s_1(e_1)$  para todos os valores possíveis de  $e_1$ , calculam-se os valores de  $s_2(e_2)$  para todos os valores possíveis de  $e_2$ .

### Cálculos para o mês 2

Tomemos, inicialmente,  $e_2 = 0$ . Considerando que a demanda de 20 toneladas do mês 2 precisa ser atendida, podemos observar este nível de estoque  $e_2$  no final do mês 2 se: com um nível de estoque inicial  $e_1 = 0$ , produzirmos 20 toneladas ou, com um nível de estoque inicial  $e_1 = 10$ , produzirmos 10 toneladas ou, com um nível de estoque inicial  $e_1 = 20$ , não produzirmos. Não existem outras possibilidades. Os custos até o mês 2 associados a cada uma dessas possíveis decisões são, respectivamente:

- $\{s_1(0) + c_2(0, d_2(0))\} = \{s_1(0) + c_2(0, 20)\} = \{s_1(0) + \$10 + \$2 \times 20\} = \{\$70 + \$50\} = \$120$  mil
- $\{s_1(10) + c_2(0, d_2(0))\} = \{s_1(10) + c_2(0, 10)\} = \{s_1(10) + \$10 + \$2 \times 10\} = \{\$92 + \$30\} = \$122$  mil
- $\{s_1(20) + c_2(0, d_2(0))\} = \{s_1(20) + c_2(0, 0)\} = \{s_1(20) + \$0\} = \{\$114 + \$0\} = \$114$  mil

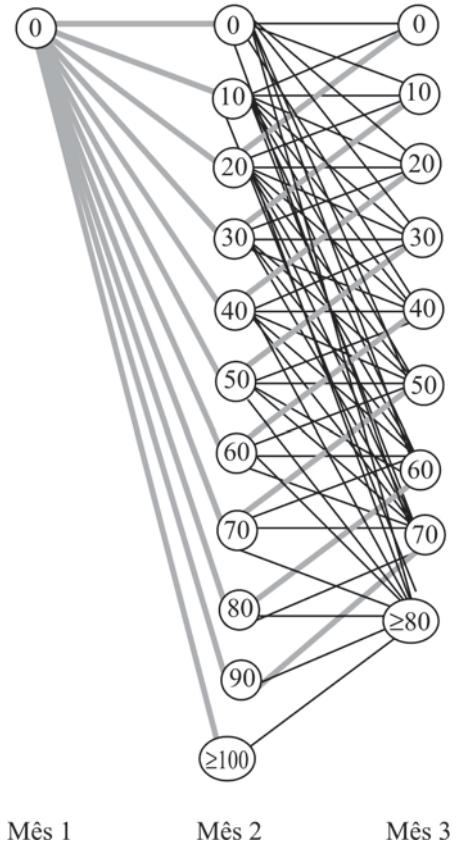
Portanto, conclui-se que:

$s_2(0) = \min\{s_1(0) + c_2(0, 20), s_1(10) + c_2(0, 10), s_1(20) + c_2(0, 0)\} = \$114$  mil, e a decisão ótima para este problema é  $d_2^*(0) = 0$  t

De maneira similar, calculam-se os demais valores de  $s_2(e_2)$  para  $e_2 = 10, 20, \dots$ , obtendo-se:

$$\begin{aligned}
 s_2(10) &= \$138 \text{ mil e } d_2^*(10) = 0t \\
 s_2(20) &= \$162 \text{ mil e } d_2^*(20) = 0t \\
 s_2(30) &= \$186 \text{ mil e } d_2^*(30) = 0t \\
 s_2(40) &= \$210 \text{ mil e } d_2^*(40) = 0t \\
 s_2(50) &= \$234 \text{ mil e } d_2^*(50) = 0t \\
 s_2(60) &= \$258 \text{ mil e } d_2^*(60) = 0t \\
 s_2(70) &= \$282 \text{ mil e } d_2^*(70) = 0t \\
 s_2(e_3) &\geq \$304 \text{ mil e } d_2^*(e_3) = 0 \text{ t para } e_3 \geq 80
 \end{aligned}$$

Na Figura 5.7, são ressaltados os arcos correspondentes às decisões ótimas já determinadas e que dependem de cada nível de estoque considerado ao final do mês. Uma vez calculados  $s_2(e_2)$  para todos os valores possíveis de  $e_2$ , são calculados, de maneira similar, os valores de  $s_3(e_3)$  para todos os valores possíveis de  $e_3$  e, em seguida,  $s_4(e_4)$  para todos os valores possíveis de  $e_4$ . Na Tabela 5.4, estão resumidos os planos ótimos dos meses 3 e 4 obtidos, juntamente com os resultados já calculados dos meses 1 e 2. O plano ótimo de produção determinado com indução progressiva resulta, como já sabido, em um custo total de \\$266 mil para os quatro meses.



**Figura 5.7** Possíveis níveis de estoque iniciais nos meses 1, 2 e 3 (nós) e transições factíveis (arcos) entre meses consecutivos.

### 5.2.2 Determinação do plano ótimo com a indução progressiva

Da Tabela 5.4, temos a indicação de que não deve haver produção de aço tipo A no mês 4. Para determinar o que produzir nos demais meses, deduz-se o valor do estoque final nos meses antecedentes, decorrentes da decisão tomada no mês posterior, e recuperam-se as decisões ótimas correspondentes. Finalizando o mês 4 com nível de estoque igual a 0 e não produzindo neste mês, iniciamos o mês 4 com 10 toneladas, ou seja, apenas com o suficiente para atender a demanda do mês. Com o nível de estoque final do mês 3 igual a 10 toneladas, obtemos a decisão ótima  $d_3^*(10) = 70$  toneladas.

Conseqüentemente, chegamos no início do mês 3 com nível de estoque igual a 0, uma vez que a demanda do mês 3 é de 60 toneladas. Com este nível de estoque, a decisão ótima no mês 2 é  $d_2^*(0) = 0$ . Portanto, iniciamos o mês 2 com 20 toneladas, correspondente à demanda deste mês. Com este nível de estoque observado, a decisão ótima no mês 1 é  $d_1^*(20) = 50$  toneladas. Com isso, temos definida toda a produção dos quatro meses do problema. O caminho ótimo correspondente ao plano ótimo determinado é destacado na Figura 5.5. O caminho ótimo liga os estoques que serão observados ao final de cada um dos quatro meses do problema.

Como pode ser observado, com a indução progressiva, os problemas (e os cálculos) são similares aos da indução regressiva, e o custo mínimo correspondente ao plano ótimo obtido, como esperado, é igual ao obtido com a indução regressiva. No exemplo apresentado, os planos ótimos obtidos são iguais, mas poderiam ser diferentes, desde que os custos mínimos fossem iguais (so-

**Tabela 5.4**  
**Planos ótimos dos quatro meses com indução progressiva (Exemplo 5.1).**

<i>Mês</i>											
<i>1</i>			<i>2</i>			<i>3</i>			<i>4</i>		
$e_1$	$d_1^*(e_1)$	$s_1(e_1)$	$e_2$	$d_2^*(e_2)$	$s_2(e_2)$	$e_3$	$d_3^*(e_3)$	$s_3(e_3)$	$e_4$	$d_4^*(e_4)$	$s_4(e_4)$
0	30	70	0	0	114	0	60	244	0	0	266
10	40	92	10	0	138	10	70	266	$\geq 10$	?	$\geq 284$
20	50	114	20	0	162	$\geq 20$	?	$\geq 284$			
30	60	136	30	0	186						
40	70	158	40	0	210						
50	80	180	50	0	234						
60	90	202	60	0	258						
70	100	224	70	0	282						
80	110	246	$\geq 80$	0	$\geq 304$						
90	120	268									
$\geq 100$	?	$\geq 290$									

luções ótimas alternativas). No uso de programação dinâmica, é preciso pensar qual indução, se a regressiva ou progressiva, melhor se adéqua à resolução do problema considerado.

## 5.3 FORMALIZAÇÃO DA PROGRAMAÇÃO DINÂMICA

### 5.3.1 Conceitos básicos

Pelos exemplos apresentados, observa-se que a simplificação na resolução do problema original é conseguida por meio da decomposição e resolução de uma seqüência de problemas mais simples. As principais características da programação dinâmica são:

#### *Estágios*

Decomposição do problema em estágios que são resolvidos seqüencialmente, um estágio de cada vez. A solução do problema em cada estágio define valores de parâmetros relevantes para o problema do estágio seguinte na seqüência. No Exemplo 5.1, os estágios representam diferentes períodos de tempo (meses). O decisor toma suas decisões em cada estágio. Assim, podemos entender como estágio cada ponto em que decisões são tomadas.

#### *Estados*

Cada estágio tem um número associado de estados. O estado do sistema é a informação necessária para a tomada de decisão. No Exemplo 5.1, o estado do sistema é o nível de estoque ob-

servado do aço tipo A. Dependendo das características do problema, existe um número maior ou menor de estados possíveis. No Exemplo 5.1, diversos níveis de estoques podem ser observados em cada estágio, portanto, diversos estados são possíveis. Caso tivessem sido impostas mais restrições ao problema, em particular, limitantes mínimos e máximos para os níveis de estoque e/ou capacidade máxima de produção, o número de estados possíveis seria menor. O conjunto de estados possíveis forma o *espaço de estados*.

### Decisões

As decisões que podem ser tomadas dependem do estado observado do sistema. Uma decisão leva (ou transforma) o estado do sistema, no qual ele se encontra no presente estágio, para um novo estado no próximo estágio, e incorre-se em um custo. No caso do problema do Exemplo 5.1, as decisões que podem ser tomadas são os diversos tamanhos de lotes de produção do aço tipo A que devem ser, pelo menos, grandes o suficiente para atender a demanda daquele mês e, possivelmente, resultar em algum excesso para atender as demandas dos meses subsequentes. A decisão tomada conduz o sistema a um novo nível de estoque no próximo estágio.

À seqüência de decisões coerentes em cada estágio do problema, dado o estado inicial, dá-se o nome de *política*. Decisões coerentes são as que respeitam as restrições do problema, ou seja, são factíveis. Uma *política ótima* é a que otimiza o objetivo desejado. No caso do problema do Exemplo 5.1, a política ótima é produzir 50 toneladas de aço tipo A no mês 1, 70 tonelada no mês 3 e 0 toneladas nos demais meses. Esta é uma solução factível para esse problema, pois atende a todas as suas restrições. Uma decisão de produzir 55 toneladas de A no mês 1, seguida de 65 toneladas no mês 3 e 0 tonelada no demais meses é uma decisão incoerente, uma vez que é uma solução infactível. Esta produção atende a demanda de todos os meses sem atraso, mas a siderúrgica não tem condições de implementá-la, pois a produção do aço tipo A se faz apenas em lotes múltiplos de 10 toneladas.

### 5.3.2 Princípio da otimalidade

Um princípio simples e intuitivo, conhecido como o *princípio da otimalidade de Bellman*,<sup>4</sup> é a base utilizada na programação dinâmica para decompor o problema. Este princípio diz que:

“Em um conjunto de decisões ótimas, tem-se a propriedade de que, qualquer que seja a primeira decisão e o estado inicial, as decisões subsequentes têm de ser ótimas com respeito ao estado resultante dessa primeira decisão.”

Este princípio é óbvio, uma vez que, se alguma das decisões subsequentes não fosse ótima com respeito ao estado resultante da primeira decisão, então poderíamos selecionar uma decisão melhor, o que contraria a hipótese inicial de que se trata de um conjunto de decisões ótimas.

No contexto do Exemplo 5.1, considere um plano ótimo de produção no qual o nível de estoque inicial no mês 3 é de 0 toneladas. Então, a parte do plano ótimo do problema que vai do período 3 até o final do período 4, com estoque inicial de 0t no mês 3, é o de menor custo possível. Isso porque, se este não fosse o caso, poderíamos substituir essa parte do plano ótimo por outro mais econômico e criariamos um plano melhor que o original, contrariando a hipótese inicial de ele ser ótimo.

Observe também que o princípio da otimalidade implica que as decisões tomadas nos diversos estágios são independentes, ou seja, a decisão em um certo estágio  $i$  é independente das demais decisões tomadas nos estágios anteriores  $i - 1, i - 2, \dots, 1$ , no caso de indução regressiva, ou nos estágios posteriores,  $i + 1, i + 2, \dots, T$ , no caso de indução progressiva.

<sup>4</sup> Richard Bellman é considerado o pai da programação dinâmica. Publicou uma coleção de artigos sobre programação dinâmica e o primeiro livro sobre a teoria da programação dinâmica nos anos 50 (ver Capítulo 1).

### 5.3.3 Otimização recursiva

Utilizando o princípio da otimalidade, decomponemos o problema de otimização original em subproblemas menores, desenvolvendo uma relação recursiva para calcular políticas ótimas, construindo-as a partir de políticas ótimas de problemas menores. Devido a essa característica recursiva, alguns autores acreditam que um nome mais apropriado para esta técnica seria *otimização recursiva*.<sup>5</sup>

Considere um problema com  $T$  estágios. Seja  $r_i(j)$  o menor custo possível em que se pode incorrer nos estágios  $i, i+1, \dots$ , até o estágio  $T$ , dado que o estado observado do sistema no estágio corrente  $i$  é  $j$ . Uma vez que estamos no estágio  $i$  e o estado do sistema é  $j$ , pode-se tomar uma decisão  $d_i(j)$  de um conjunto  $D_i(j)$  de todas as decisões viáveis que podem ser tomadas no estágio  $i$  a partir do estado  $j$ . Com esta decisão, incorre-se em um custo  $c_i(j, d_i(j))$  no estado  $j$  do estágio  $i$ , e o sistema evolui para um novo estado,  $e(d_i(j))$ , no estágio seguinte  $i+1$ . Se o valor de  $r_{i+1}(e(d_i(j)))$  é conhecido, ou seja, o menor custo possível, em que se pode incorrer nos estágios  $i+1, i+2, \dots$ , até o estágio  $T$  é conhecido, dado que o estado observado do sistema no estágio  $i+1$  é  $e(d_i(j))$ , então,  $r_i(j)$  pode ser obtido calculando-se:

$$r_i(j) = \min_{d_i(j) \in D_i(j)} \{c(d_i(j)) + r_{i+1}(e(d_i(j)))\} \quad (5.1)$$

ou seja, avaliam-se os custos de todas as decisões possíveis no estágio  $i$  dado que o estado observado é  $j$ , e toma-se a decisão de custo mínimo. No caso do Exemplo 5.1, a expressão (5.1) toma a forma:

$$r_i(e_{i-1}) = \min_{d_i(e_{i-1}) \in D_i(e_{i-1})} \{c_i(e_{i-1}, d_i(e_{i-1})) + r_{i+1}(e_{i-1} + d_i(e_{i-1}) - p_i)\}, \quad i = 1, 2, 3, 4,$$

em que  $p_i$  é a demanda de aço no período  $i$ , e  $r_5(e_4) = 0$  para qualquer valor de  $e_4$ . Note, nessa expressão, que o estado resultante no estágio seguinte  $i+1$  é:  $e_i = e_{i-1} + d_i(e_{i-1}) - p_i$ .

A expressão (5.1) foi escrita levando-se em conta que estamos considerando um problema de minimização. Ela poderia ser escrita substituindo *min* por *max* se, ao invés de minimizar custos, estivermos maximizando, por exemplo, a margem de contribuição ao lucro. A expressão (5.1) é o protótipo da equação recursiva da programação dinâmica determinística e é conhecida por *equação recursiva regressiva*. No Capítulo 6, essa fórmula é estendida para tratar problemas com incertezas.

Para a utilização da recursão (5.1), inicia-se calculando  $r_T(\cdot)$  para cada estado possível no estágio  $T$ , ou seja, no último estágio. Usando-se a recursão, determina-se  $r_{T-1}(\cdot)$  para cada estado possível no estágio  $T-1$ , armazenando a decisão ótima correspondente. Em seguida, usa-se a recursão para determinar  $r_{T-2}(\cdot)$  para cada estado possível no estágio  $T-2$ , armazenando a decisão ótima correspondente, e assim sucessivamente, até se obter  $r_1$  e escolhe-se o menor custo possível e a decisão ótima correspondente. Para recuperar as decisões tomadas que levam a esse custo, basta iniciar com a decisão ótima do estágio 1. Esta decisão leva a um estado, digamos,  $e'_2$ , no estágio 2. Então, no estágio 2, escolhe-se a decisão que incorre no custo  $r_2(e'_2)$ , e continua-se desta forma até serem determinadas todas as decisões em cada um dos  $T$  estágios do problema.

Analogamente à equação recursiva regressiva, pode-se definir uma *equação recursiva progressiva*. Denotemos por  $s_i(j)$  o menor custo possível em que se pode incorrer do estágio 1 até o estágio  $i$ , e que leva o sistema do estado inicial fixado  $e_1$ , no início do estágio 1, ao estado  $j$  observado ao final do estágio  $i$ . Dado que estamos no estágio  $i$  e observamos que o estado do sistema foi levado a  $j$ , este estado foi atingido por decorrência de uma decisão  $d_i(j)$  tomada no estágio  $i$ , cujo custo associado é  $c(d_i(j))$ , e que levou o sistema do estado  $e(d_i(j))$  do estágio anterior  $i-1$  ao estado  $j$  observado no estágio  $i$ . Portanto, de maneira similar à fórmula (5.1), o menor custo possível  $s_i(j)$  pode ser obtido calculando-se:

$$s_i(j) = \min_{d_i(j) \in D_i(j)} \{c(d_i(j)) + s_{i-1}(e(d_i(j)))\} \quad (5.2)$$

<sup>5</sup> Ver, por exemplo, Nemhauser (1966).

em que  $D_i(j)$  é o conjunto de todas as decisões que podem ser tomadas no estágio  $i$  e que conduzem ao estado observado  $j$ .

No caso do Exemplo 5.1, a expressão (5.2) toma a forma

$$s_i(e_i) = \min_{d_i(e_i) \in D_i(e_i)} \{c_i(e_i, d_i(e_i)) + s_{i-1}(e_i - d_i(e_i) + p_i)\}, \quad i = 1, 2, 3, 4,$$

em que  $s_0(e_0) = 0$  para qualquer valor de  $e_0$ . Note, nesta expressão, que o estado resultante no estágio anterior  $i-1$  é:  $e_{i-1} = e_i - d_i(e_i) + p_i$ .

Para a utilização da recursão (5.2), inicia-se calculando  $s_1(\cdot)$  para cada estado possível ao final do estágio 1, ou seja, no primeiro estágio. Usando-se a recursão, determina-se  $s_2(\cdot)$  para cada estado possível no estágio 2, armazenando-se a decisão ótima correspondente. Novamente, usa-se a recursão para determinar  $s_3(\cdot)$  para cada estado possível no estágio 3, armazenando-se a decisão ótima correspondente, e assim sucessivamente, até se chegar a  $s_T(\cdot)$ , e escolhe-se o menor custo possível e a decisão ótima correspondente. Para recuperar as decisões tomadas que levam a esse custo, basta iniciar com a decisão ótima do estágio  $T$ . Esta decisão leva a um estado, digamos,  $e'_{T-1}$ , no estágio  $T-1$ . Então, no estágio  $T-1$ , escolhe-se a decisão que incorre no custo  $S_{T-1}(e'_{T-1})$ , e continua-se desta forma até serem determinadas todas as decisões em cada um dos  $T$  estágios do problema.

A expressão (5.2) é um outro protótipo de equação recursiva da programação dinâmica e é conhecida por *equação recursiva progressiva*.

Os cálculos das equações (5.1) e (5.2) podem ser organizados em tabelas, como exemplificado nas Tabelas 5.1 e 5.2, facilitando a compreensão. A implementação das equações recursivas da programação dinâmica em um computador é simples, o que é uma vantagem sobre outras técnicas.

Os custos  $r(\cdot)$  e  $s(\cdot)$  não precisam ser lineares para a validade das expressões (5.1) e (5.2), o que é outra vantagem da programação dinâmica sobre outras técnicas. De fato, essas funções podem ser bem genéricas. O próprio Exemplo 5.1 é um caso simples em que a função custo de produção é descontínua. Além disso, os estados possíveis podem ser discretos, como no Exemplo 5.1, ou contínuos (e as decisões possíveis também em espaços contínuos).<sup>6</sup> Neste livro, tratamos apenas de programação dinâmica com estados discretos.

Um problema de otimização pode ser resolvido enumerando-se explicitamente todas as soluções factíveis. Por exemplo, no caso do problema do Exemplo 5.1, para encontrar o plano ótimo de produção, pode-se iniciar no mês 1 e examinar todas as possibilidades de produção factíveis neste mês. A cada uma dessas possibilidades, calcula-se o estoque resultante no início do mês 2 e, então, examinam-se todas as possibilidades de produção factíveis no mês 2. Cada uma dessas possibilidades de produção no mês 2, por sua vez, resulta em um estoque no início do mês 3 e, a cada uma delas, é possível enumerar todas as possibilidades de produção factíveis no mês 3, e assim sucessivamente.

Cada uma dessas combinações de produção do mês 1 ao mês 4 forma um plano factível de produção para o problema. O número total de planos factíveis cresce exponencialmente, conforme as possibilidades de produção factíveis a cada mês. Com a programação dinâmica, esses cálculos são reduzidos substancialmente, uma vez que apenas os planos ótimos de produção de cada estado em cada estágio são computados. A vantagem da programação dinâmica em termos de número de operações elementares realizadas (e, assim, do tempo computacional) sobre a enumeração explícita de todas as soluções é, portanto, aparente.

### 5.3.4 Exemplo de aplicação de programação dinâmica com custos não-lineares

Ilustramos a seguir um exemplo de aplicação de programação dinâmica para resolver um problema de otimização com custos não-lineares.<sup>7</sup>

<sup>6</sup> Ver, por exemplo, Ravindran *et al.* (1987) e Bertsekas (2005).

<sup>7</sup> Ver Arce *et al.* (2002) para outro exemplo de aplicação de programação dinâmica com custos não-lineares na minimização dos custos de ligamento e desligamento de um número de turbinas para geração de energia hidroelétrica em Itaipú.

**Exemplo 5.2** No início de cada mês, uma fábrica precisa decidir quantas unidades fabricar de um produto no período. A produção pode ser realizada em duas máquinas diferentes, com custos diferentes de produção. Se  $x$  unidades do produto são fabricadas na máquina 1, o custo de produção é dado por  $c_1(x) = 8(x + 1)^2 + 20$ ; se produzido na máquina 2, o custo de produção é dado por  $c_2(x) = 4(x + 1)^3 + 40$ . A demanda mensal do produto nos próximos três meses é de 2, 3 e 4 unidades. O espaço físico disponível para armazenamento do produto é limitado. Assim, ao final de cada mês, não é possível armazenar mais do que duas unidades do produto para o mês seguinte. As máquinas incorrem em um custo fixo cada vez que são ligadas e desligadas. No início de cada mês, a fábrica precisa decidir se vai ligar, manter ligada ou desligar cada uma das máquinas. Para ligar a máquina 1, incorre-se em um custo fixo de \$100; para desligar a máquina 1, incorre-se em um custo fixo de \$50; e para manter a máquina ligada sem produzir nada, o custo incorrido é  $c_1(0) = 28$ . Para ligar a máquina 2, incorre-se em um custo fixo de \$200; para desligar a máquina 2, incorre-se em um custo fixo de \$60; e para manter a máquina ligada sem produzir nada, o custo incorrido é  $c_2(0) = 44$ . O estoque disponível no início deste mês é 0, e apenas a máquina 2 encontra-se ligada. Qual deve ser a política de produção da empresa que minimiza os custos de produção nos próximos três meses, dado que a demanda precisa ser atendida e o custo de estocagem de um mês para outro é de \$5 por unidade?

Note que os custos deste problema são não-lineares. Deseja-se determinar o quanto produzir em cada um dos próximos três meses, e em quais máquinas. A quantidade a produzir em um determinado mês depende do conhecimento do estoque disponível no início deste mês, da demanda do mês e da limitação no número máximo de unidades que podem ser deixadas em estoque para o mês seguinte. O custo de produção depende, ainda, de quais máquinas estão ligadas e desligadas e de quanto é produzido em cada uma delas.

As decisões que podem ser tomadas em cada mês são as quantidades a serem produzidas em cada uma das máquinas 1 e 2, que, juntamente com o estoque já disponível, deve ser suficiente para pelo menos atender a demanda daquele mês e, possivelmente, ser utilizado para atender as demandas dos meses subsequentes. Cada uma das possíveis decisões em um dado mês leva o sistema a um certo nível de estoque e leva as máquinas a um certo estado, ou seja, em funcionamento ou paradas, no início do mês seguinte.

Para resolvemos este problema por programação dinâmica, utilizamos uma tripla  $(j, k, w)$  para denotar o estado do sistema no mês  $i$ , em que  $j$  representa o nível de estoque no início do mês  $i$ , e  $k$  e  $w$  indicam o funcionamento das máquinas 1 e 2, respectivamente (0 para máquina desligada e 1 para máquina ligada). Utilizamos também uma tripla  $(j, k, w)$  para denotar as decisões que podem ser tomadas:  $j$  nos dá a quantidade a ser produzida,  $k$  e  $w$  nos indicam as máquinas 1 e/ou 2 que são utilizadas na produção, respectivamente (0 para máquina não utilizada e 1 para máquina utilizada).

O estado inicial do sistema é  $(0,0,1)$ . Utilizando a recursão (5.1), iniciamos os cálculos pelo mês 3, com  $r_4(e_3) = 0$  para qualquer valor de  $e_3$ . Os estados possíveis no início do mês 3 são:  $(0,0,0)$ ,  $(0,1,0)$ ,  $(0,0,1)$ ,  $(0,1,1)$ ,  $(1,0,0)$ ,  $(1,1,0)$ ,  $(1,0,1)$ ,  $(1,1,1)$ ,  $(2,0,0)$ ,  $(2,1,0)$ ,  $(2,0,1)$ ,  $(2,1,1)$ , uma vez que não se pode estocar mais do que duas unidades do produto de um mês para outro.

Estado  $(0,0,0)$ : se o sistema está no estado  $(0,0,0)$ , as decisões que podem ser tomadas no mês 3 são: ligar somente a máquina 1 e produzir nela quatro unidades do produto (isto é, decisão  $(4,1,0)$ ); ligar somente a máquina 2 e produzir nela quatro unidades do produto (decisão  $(4,0,1)$ ); ligar as duas máquinas e produzir um total de quatro unidades do produto ao menor custo (decisão  $(4,1,1)$ ). As decisões de produzir mais do que quatro unidades não são consideradas porque as funções de custo de produção resultam em custos superiores à produção de quatro unidades.

A decisão  $(4,1,0)$  resulta em um custo de \$100 (ligar a máquina 1) + \$220 (produzir quatro unidades na máquina 1) = \$320. A decisão  $(4,0,1)$  resulta em um custo de \$200 (ligar a máquina 2) + \$540 (produzir quatro unidades na máquina 2) = \$740. A decisão  $(4,1,1)$  resulta em um custo de \$100 (ligar a máquina 1) + \$200 (ligar a máquina 2) + \$220 (menor custo de produção de

quatro unidades nas duas máquinas) = \$520. Neste caso, duas possíveis soluções de igual custo são: produzir 4 unidades na máquina 1, ou produzir três unidades na máquina 1 e uma unidade na máquina 2. Portanto,

$$\begin{aligned} r_3((0,0,0)) &= \min\{\$320 + r_4((0,1,0)), \$740 + r_4((0,0,1)), \$520 + r_4((0,1,1))\} = \\ &= \min\{\$320 + \$0, \$740 + \$0, \$520 + \$0\} = \$320 \end{aligned}$$

e a decisão ótima é (4,1,0).

Estado (0,1,0): se o sistema está no estado (0,1,0), as decisões possíveis são: permanecer com a máquina 1 ligada e produzir nela quatro unidades do produto (decisão (4,1,0)); desligar a máquina 1, ligar a máquina 2 e produzir nela quatro unidades do produto (decisão (4,0,1)); permanecer com a máquina 1 ligada, ligar a máquina 2 e produzir um total de quatro unidades do produto ao menor custo (decisão (4,1,1)). As decisões de produzir mais do que quatro unidades não são consideradas porque as funções de custo de produção resultam em custos superiores à produção de quatro unidades. Outras decisões possíveis, como permanecer com a máquina 1 ligada mas nada produzir nela, ou ligar a máquina 2 e produzir nela um total de quatro unidades do produto, também não são consideradas, pois são dominadas (são mais caras em termos de custos) por alguma das decisões já consideradas.

A decisão (4,1,0) resulta em um custo de \$0 (manter ligada a máquina 1 e utilizá-la para produção) + \$220 (produzir quatro unidades na máquina 1) = \$220. A decisão (4,0,1) resulta em um custo de \$50 (desligar a máquina 1) + \$200 (ligar a máquina 2) + \$540 (produzir quatro unidades na máquina 2) = \$790. A decisão (4,1,1) resulta em um custo de \$0 (manter a máquina 1 ligada e produzir nela) + \$200 (ligar a máquina 2) + \$220 (menor custo de produção de quatro unidades nas duas máquinas, produzindo quatro unidades na máquina 1 ou produzindo três unidades na máquina 1 e uma unidade na máquina 2) = \$520. Portanto,

$$\begin{aligned} r_3((0,1,0)) &= \min\{\$220 + r_4((0,1,0)), \$790 + r_4((0,0,1)), \$420 + r_4((0,1,1))\} = \\ &= \min\{\$220 + \$0, \$790 + \$0, \$420 + \$0\} = \$220 \end{aligned}$$

e a decisão ótima é (4,1,0).

De maneira similar, calcula-se  $r_3(e_2)$  para os demais valores de  $e_2 = (0,0,1), (0,1,1), (1,0,0), (1,1,0), (1,0,1), (1,1,1), (2,0,0), (2,1,0), (2,0,1), (2,1,1)$ . Com os  $r_3(e_2)$  calculados para todos os estados  $e_2$  possíveis, calculam-se os  $r_2(e_1)$  para todos os estados  $e_1$  possíveis e, finalmente, calcula-se  $r_1(e_0) = r_2((0,0,1))$ . Deixamos como exercício para o leitor a realização desses cálculos. ■

## 5.4 INDUÇÃO DA DECOMPOSIÇÃO NA PROGRAMAÇÃO DINÂMICA

Nos Exemplos 5.1 e 5.2, decomponemos o problema complexo do gerente de produção em uma seqüência de problemas mais simples, um ou mais de um para cada mês, seguindo os próprios períodos em que o gerente toma suas decisões. O número total de períodos (ou estágios) de um problema é definido pelos instantes no tempo em que as decisões são tomadas. Para outros problemas de otimização, a decomposição em uma seqüência de problemas mais simples não é imediata, uma vez que não é aparente como as seqüências de decisões podem ser tomadas em estágios sucessivos. O exemplo a seguir ilustra esta situação.

**Exemplo 5.3** Uma fábrica produz quatro tipos diferentes de produtos, A, B, C e D, cuja venda proporciona lucros diferenciados para a empresa. Considere que o mercado atual está aquecido e, desta forma, tudo o que puder ser produzido é vendido. Dispõe-se, entretanto, de um volume limitado de um recurso E, necessário para a produção desses produtos. Deseja-se saber quais as quantidades a produzir de cada um dos produtos, de modo a se ter o maior lucro possível. Cabe lembrar que apenas produtos completos têm valor comercial, ou seja, partes de produto não podem ser vendidas.

Este problema de otimização é conhecido na literatura como o problema da mochila unidimensional irrestrito (já descrito nos Capítulos 3 e 4), uma vez que modela o problema do mochileiro que gostaria de levar um conjunto de itens, mas não pode levá-los todos, pois sua mochila tem capacidade limitada. Portanto, ele precisa decidir quais itens e quantas unidades de cada item levar. Para isso, ele deve selecionar um subconjunto dos itens que caiba na mochila e que maximize a sua satisfação ou utilidade total.

Neste problema, não temos períodos ou instantes no tempo em que as decisões são tomadas. Assim, a decomposição deste problema de otimização em problemas mais simples, estabelecendo como as seqüências de decisões podem ser tomadas em estágios sucessivos, é menos imediata do que nos Exemplos 5.1 e 5.2. Uma possível decomposição seria considerar a inclusão de um produto de cada vez.

Inicialmente, determina-se qual seria a melhor decisão no caso de fabricar apenas um dos produtos, digamos, o produto A. A seguir, resolve-se uma nova seqüência de problemas simples; determina-se a melhor decisão se considerarmos fabricar – além do produto A – mais um produto, digamos, o produto B. Em seguida, resolve-se uma nova seqüência de problemas simples, ou seja, determina-se a melhor decisão se considerarmos fabricar – além dos produtos A e B – mais um produto, por exemplo, o produto C. Finalmente, resolve-se uma nova seqüência de problemas nos quais se determina qual seria a melhor decisão no caso de considerarmos fabricar – além dos produtos A, B e C – mais um produto, no caso, o produto D. Os estágios seriam, portanto, os seguintes:

- estágio 1 – apenas o produto A é fabricado
- estágio 2 – apenas os produtos A e B são fabricados
- estágio 3 – apenas os produtos A, B e C são fabricados
- estágio 4 – todos os produtos A, B, C e D são fabricados.

Os estados possíveis em cada estágio são as diversas quantidades (níveis) disponíveis do recurso E para a fabricação do produto adicional considerado. Podemos, portanto, utilizar uma equação recursiva similar à equação recursiva progressiva (5.2) para resolver este problema. Esta expressão pode ser escrita como:

$$l_i(j) = \max_{d_i(j) \in D_i(j)} \{t(d_i(j)) + l_{i-1}(e(d_i(j)))\}$$

em que

$l_i(j)$  é o maior lucro possível quando se considera fabricar apenas os produtos 1, 2, ...,  $i$  e se dispõe de uma quantidade  $j$  de recursos E,

$t(d_i(j))$  é o lucro obtido com a fabricação de  $d_i(j)$  unidades do produto  $i$  (estágio  $i$ ), e

$e(d_i(j))$  é a quantidade disponível do recurso disponível  $j$  que restou após a produção das  $d_i(j)$  unidades do produto  $i$ , e que pode ser utilizado na fabricação dos produtos 1, 2, ...,  $i - 1$ .

Ilustremos essa recursão por meio de um exemplo numérico. Considere que os lucros unitários associados aos produtos A, B, C e D são, respectivamente, \$15, \$50, \$79 e \$119. Para a fabricação de cada unidade desses produtos, são necessárias, respectivamente, 2, 3, 5 e 7 unidades do recurso E. Admitamos que o total de recursos E disponível é de 11 unidades.

Inicialmente definimos  $l_0(j)$  igual a 0 para todos os possíveis níveis de recursos E de 0 até 11, ou seja, não há lucro quando não se fabrica nenhum produto. Consideramos a seguir os problemas simples com apenas o produto A (estágio 1). Para cada quantidade do recurso E possível, de 0 até 11, resolve-se um problema de quanto fabricar do produto A. Determina-se:

$$l_1(j) = \max_{d_1(j) \in D_1(j)} \{t(d_1(j)) + l_0(e(d_1(j)))\} = \max_{d_1(j) \in D_1(j)} \{t(d_1(j))\}, \quad j = 0, 1, \dots, 11.$$

Por exemplo, se  $j = 7$ , ou seja, dispomos de sete unidades do recurso E, as únicas decisões possíveis são produzir 0, 1, 2 ou 3 unidades de A. Os lucros associados são \$0, \$15, \$30 e \$45, respectivamente. Portanto, a decisão que maximiza o lucro é produzir três unidades de A e  $l_1(7) = \$45$ . Para cada um dos estados possíveis, a melhor solução está apresentada na Tabela 5.5.

**Tabela 5.5**  
**Soluções ótimas dos problemas no estágio 1 (Exemplo 5.3) – inclusão do produto A.**

Quantidade disponível de recurso $j$	Produção ótima de A, $d_1^*(j)$	Lucro correspondente $l_1(j)$
0	0	0
1	0	0
2	1	15
3	1	15
4	2	30
5	2	30
6	3	45
7	3	45
8	4	60
9	4	60
10	5	75
11	5	75

Uma vez obtidos  $l_1(j)$  para todos os valores de  $j$ , consideramos a inclusão do produto B (estágio 2). Determina-se

$$l_2(j) = \max_{d_2(j) \in D_2(j)} \{t(d_2(j)) + l_1(e(d_2(j)))\}, \quad j = 0, 1, \dots, 11.$$

Resolvemos novamente uma seqüência de problemas simples em que desejamos determinar quanto produzir de B, dada uma quantidade  $j$  de recursos E disponíveis ( $j = 0, 1, 2, \dots, 11$ ). Cada unidade do produto B utiliza três unidades do recurso E. Assim, enquanto não tivermos pelo menos três unidades do recurso E, não podemos produzir B; portanto, não há decisão a tomar com respeito à produção de B. Se a quantidade de recursos disponível é menor do que três unidades, a melhor opção é continuar produzindo somente o produto A e, para isso, já temos as decisões ótimas calculadas dos problemas simples anteriores do primeiro estágio.

Quando temos três unidades do recurso E, as opções possíveis para a produção de B são: não produzir B ou produzir uma unidade do produto B, consumindo todo o recurso disponível. No primeiro caso, se não produzimos B, as três unidades do recurso E são utilizadas para a produção de A, e o melhor lucro possível para este caso foi calculado no estágio anterior ( $l_1(3) = \$15$ ). No segundo caso, consome-se todo o recurso disponível na produção de B, não restando nenhum recurso para a produção de A. O lucro associado a esta produção é de  $\$50$ . Comparando-se as duas opções possíveis, determina-se a melhor decisão, no caso,  $l_2(3) = \max \{\$0 + \$15, \$50 + \$0\} = \$50$ . Assim, a decisão ótima é  $d_2^*(3) = 1$  unidade do produto B.

De maneira similar, procede-se com o cálculo para os demais valores de recurso E. Por exemplo, para E igual a 8, pode-se produzir: 0 unidade de B, 1 unidade de B ou 2 unidades de B. No primeiro caso, restam oito unidades do recurso E para serem utilizadas da melhor maneira na produção de A, e  $l_1(8)$  já foi determinado na iteração anterior. No segundo caso, restam cinco uni-

dades do recurso E para serem utilizadas da melhor maneira na produção de A, e  $l_1(5)$  já foi determinado na iteração anterior. No terceiro caso, restam duas unidades do recurso E para serem utilizadas da melhor maneira na produção de A, e  $l_1(2)$  já foi determinado na iteração anterior. Temos, então, que:  $l_2(8) = \max \{ \$0 + \$60, \$50 + \$30, \$100 + \$15 \} = \$115$ , e  $d_2^*(8) = 2$  unidades do produto B.

Na Tabela 5.6, apresentamos a quantidade ótima de B para cada uma das quantidades possíveis do recurso E. Uma vez obtidos  $l_2(j)$  para todos os valores de  $j$ , calculamos de forma recursiva  $l_3(j)$  e, finalmente,  $l_4(j)$ , chegando às decisões ótimas apresentadas nas Tabelas 5.7 e 5.8.

O valor ótimo do problema é dado por  $l_4(11)$  e é igual a \$179. Ele é obtido utilizando-se a política ótima de produzir uma unidade do produto C, duas unidades do produto B e nada dos demais produtos A e D. Da Tabela 5.8 (estágio 4), o valor ótimo de \$179 é obtido da decisão ótima de nada produzir do produto D. A quantidade de recurso E então se mantém em 11 unidades para serem utilizadas na produção de A, B e C (estágio 3), o que, na Tabela 5.7, implica na decisão ótima de produzir uma unidade do produto C. Como o produto C consome cinco unidades de E, a quantidade disponível do recurso reduz-se a seis unidades a serem utilizadas na produção de A e B (estágio 2). Da Tabela 5.6, com o recurso disponível de seis unidades de E, a decisão ótima é produzir duas unidades do produto B. Cada produto B consome três unidades do recurso E; desta forma, o recurso disponível para a produção de A (estágio 1) é zero. Da Tabela 5.5, com o recurso E de 0 unidades, a decisão ótima é não produzir o produto A.

Note que, para resolvemos este problema da mochila, utilizamos uma equação recursiva progressiva partindo da inclusão de um único produto e, em cada estágio, consideramos a inclusão de mais um produto. De modo semelhante ao que foi feito para a resolução do problema do Exem-

**Tabela 5.6**  
**Soluções ótimas dos problemas no estágio 2 (Exemplo 5.3) – inclusão do produto B.**

Quantidade disponível de recurso $j$	Produção ótima de B, $d_2^*(j)$	Lucro correspondente $l_2(j)$
0	0	0
1	0	0
2	0	15
3	1	50
4	1	50
5	1	65
6	2	100
7	2	100
8	2	115
9	3	150
10	3	150
11	3	165

plor 5.1. de tamanho de lote de produção, também é possível realizar movimentos regressivos para resolver este problema da mochila. Uma equação recursiva regressiva para este problema é:

$$m_i(j) = \max_{d_i(j) \in D_i(j)} \{t(d_i(j)) + m_{i+1}(e(d_i(j)))\}$$

em que:

$m_i(j)$  é o maior lucro possível quando se considera fabricar apenas os produtos  $i, i+1, \dots, N$  e já foi utilizada uma quantidade  $j$  de recurso E,

$t(d_i(j))$  é o lucro obtido com a fabricação de  $d_i(j)$  unidades do produto  $i$  (estágio  $i$ ), e

$e(d_i(j))$  é a quantidade total utilizada do recurso E, incluindo a fabricação das  $d_i(j)$  unidades do produto  $i$ .

Nesta recursão, admitimos que existem  $N$  produtos. Se a equação recursiva progressiva for definida para  $i = 1, 2, \dots, N$ , definimos, por conveniência, que  $m_{N+1}(j) = 0$  para todo  $j$ .

No estágio 4 da recursão regressiva, consideramos que uma certa quantidade do recurso E já foi utilizada para a fabricação dos três primeiros produtos, digamos, os produtos A, B e C, e que o restante poderá ser utilizado para fabricar o produto D. As quantidades de recursos E já utilizadas podem variar de 0 a 11. Pela equação recursiva regressiva, temos que

$$m_4(j) = \max_{d_4(j) \in D_4(j)} \{t(d_4(j)) + m_5(e(d_4(j)))\} = \max \{t(d_4(j))\}, j = 0, 1, \dots, 11$$

Por exemplo, para  $j = 2$ , ou seja, se já utilizamos duas unidades de E, restam ainda nove unidades para a produção de D. Neste caso, as únicas decisões possíveis seriam: não produzir nenhuma unidade de D com lucro \$0 ou produzir uma unidade de D com lucro de \$119. Portanto, a melhor opção é produzir uma unidade de D. Se  $j = 5$ , ou seja, se já utilizamos cinco unidades de E, res-

**Tabela 5.7**  
**Soluções ótimas dos problemas no estágio 3 (Exemplo 5.3) – inclusão do produto C.**

Quantidade disponível de recurso $j$	Produção ótima de C, $d_3^*(j)$	Lucro correspondente $l_3(j)$
0	0	0
1	0	0
2	0	15
3	0	50
4	0	50
5	1	79
6	0	100
7	0	100
8	1	129
9	0	150
10	2	158
11	1	179

tam ainda seis unidades para a produção de D. Com este montante de recursos, não é possível produzir nenhuma unidade de D, portanto, a melhor (e única) opção para  $j = 5$  é não produzir D. Cálculos similares são realizados para os demais valores de  $j$ , chegando aos valores indicados na Tabela 5.9.

Seguindo regressivamente para o estágio 3, considera-se agora que uma certa quantidade do recurso E já foi utilizada para a fabricação dos dois primeiros produtos, digamos, A e B, e que somente o restante poderá ser utilizado para fabricar os demais produtos, no caso, C e D. As quantidades do recurso E já utilizadas variam de 0 a 11. Pela equação recursiva regressiva, temos que:

$$m_3(j) = \max_{d_3(j) \in D_3(j)} \{t(d_3(j)) + m_4(e(d_3(j)))\}, \quad j = 0, 1, \dots, 11$$

Por exemplo, se  $j = 2$ , ou seja, já utilizamos duas unidades de E, e restam ainda nove unidades para a produção de C e D. Como já calculamos quais seriam os melhores planos para a produção de D para os diversos valores já utilizados do recurso E, basta definir qual deve ser a fabricação do produto C. As únicas decisões possíveis para a produção de C são não produzir nenhuma unidade de C ou produzir uma unidade de C. Não é possível produzir mais do que uma unidade de C, pois não há recurso disponível suficiente.

Se não produzirmos nada de C, continuamos com o mesmo nível de recurso E utilizado, e a melhor política de produção de D para este valor é produzir uma unidade, por sua vez, já determinada da resolução dos problemas simples anteriores (Tabela 5.9). O lucro associado a esta opção é  $\$0 + \$119 = \$119$ . Se, por outro lado, decidirmos produzir uma unidade de C, serão consumidas cinco unidades do recurso E, totalizando sete unidades de recursos gastos. Para este valor, também já temos determinada a melhor política para a produção de D a partir da resolução dos problemas simples anteriores; da Tabela 5.9, a única opção é não produzir o produto D, resultando em um lucro de  $\$79 + \$0 = \$79$ .

**Tabela 5.8**  
**Soluções ótimas dos problemas no estágio 4 (Exemplo 5.3) – inclusão do produto D.**

Quantidade disponível de recurso $j$	Produção óptima de D, $d_4^*(j)$	Lucro correspondente $l_4(j)$
0	0	0
1	0	0
2	0	15
3	0	50
4	0	50
5	0	79
6	0	100
7	1	119
8	0	129
9	0	150
10	0	158
11	0	179

Desta forma, conclui-se que a melhor decisão, quando são disponibilizadas nove unidades do recurso E para a produção de C e D, é a de produzir uma unidade de D e não produzir C ( $\max\{ \$0 + \$119, \$79 + \$0 \} = \$119$ ). Cálculos similares para os demais valores de  $j$  resultam nos valores indicados na Tabela 5.10.

Os estados possíveis do estágio 2 são obtidos de maneira similar ao que foi feito anteriormente, considerando agora que uma certa quantidade do recurso E já foi utilizada para a fabricação do primeiro produto, digamos A, e que o restante deve ser utilizado para fabricar os demais produtos, no caso, B, C e D. Já calculamos quais seriam os melhores planos para a produção de C e D para os diversos valores já utilizados do recurso E, bastando considerar as diversas quantidades possíveis de produção de B. Os valores determinados estão indicados na Tabela 5.11.

Uma vez calculados os melhores planos para a produção de B, C e D para os diversos valores já utilizados do recurso E, basta considerar as diversas quantidades possíveis de produção de A (estágio 1). A decisão que conduz ao melhor valor de lucro é não produzir nada de A, deixando toda a quantidade do recurso E disponível para a fabricação dos produtos B, C e D (lucro ótimo igual a \$179). Das Tabelas 5.9, 5.10 e 5.11, é fácil observar que o plano ótimo associado é produzir duas unidades de B e uma unidade de C, ou seja, a mesma solução obtida com o procedimento progressivo (Tabelas 5.5, 5.6, 5.7 e 5.8).

É importante salientar que, em programação dinâmica, a decomposição do problema original em uma seqüência de problemas mais simples não é única. Para ilustrarmos este aspecto da programação dinâmica, consideremos novamente o Exemplo 5.1. Se analisarmos mais a fundo as diversas possibilidades de produção, podemos concluir que algumas delas podem ser descartadas de imediato, pois não devem fazer parte de uma política ótima, uma vez que são mais custosas que outras.

**Tabela 5.9**  
**Soluções ótimas dos problemas no estágio 4 da recursão regressiva (Exemplo 5.3) – inclusão do produto D.**

<i>Quantidade de recurso já utilizado j</i>	<i>Produção ótima de D, <math>d_4^*(j)</math></i>	<i>Lucro correspondente <math>m_4(j)</math></i>
0	1	119
1	1	119
2	1	119
3	1	119
4	1	119
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0

Por exemplo, um plano de produção em que se fabrica em um determinado mês  $i$ , e existe aço tipo A em estoque no início desse mês, não precisa ser considerado. Essa consideração é válida porque nesse problema não existem restrições de capacidade ou de nível de estoque mínimo. Assim, sempre é possível eliminar o estoque final do mês  $i - 1$  (e, por conseguinte, eliminar o custo de estocagem incorrido) produzindo-se menos no mês imediatamente anterior ao mês  $i$  em que houve produção, e mais no mês  $i$ . Considere o plano de produção em que temos  $e_{i-1}$ , o estoque existente no início do mês  $i$ , maior que zero, e  $x_i$ , a quantidade produzida do aço A no mês  $i$ , também positiva. Neste caso, podemos construir um plano alternativo menos custoso se diminuirmos de  $e_{i-1}$  a fabricação do último lote de produção imediatamente anterior ao lote do mês  $i$ , e aumentarmos a produção  $x_i$  de  $e_{i-1}$ . Com isso, continuamos atendendo a demanda de todos os meses e diminuímos o custo de estocagem, mantendo os demais custos inalterados.

Considere, por exemplo, um plano de produção para o Exemplo 5.1 em que se produz 30 toneladas no mês 1, 30 toneladas no mês 2 e 60 toneladas no mês 3. Ao final do mês 2, tem-se 10 toneladas em estoque, e há produção no mês 3. Essas 10 toneladas podem deixar de ser produzidas no mês 2 e serem produzidas no mês 3. Com isso, reduz-se o custo de estocagem no mês 2, sem prejuízo dos demais custos. Demonstra-se que a propriedade  $x_i e_{i-1} = 0$  é verdadeira, mesmo quando todos os parâmetros – custo fixo, custo unitário de produção e de estoque – são variantes no tempo. Esta é a propriedade de Wagner-Whitin<sup>8</sup> para problemas de dimensionamento de lotes.

Assim, para o Exemplo 5.1, é suficiente procurar uma política em que o lote ótimo de produção em cada mês  $i$  é zero, ou é tal que completa exatamente a demanda desse mês ou as demandas

**Tabela 5.10**  
**Soluções ótimas dos problemas no estágio 3 da recursão regressiva (Exemplo 5.3) – inclusão do produto C.**

Quantidade de recurso já utilizado $j$	Produção ótima de C, $d_3^*(j)$	Lucro correspondente $m_3(j)$
0	2	158
1	2	158
2	0	119
3	0	119
4	0	119
5	1	79
6	1	79
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0

<sup>8</sup> Ver o estudo em Wagner e Whitin (1958).

acumuladas do mês  $i$  até o mês  $i + 1, i + 2, \dots, n$ . O problema pode, então, ser formulado como um problema de caminho mais curto do nó 0 ao nó 4 em um grafo orientado acíclico, conforme esquematizado na Figura 5.8.

Este problema pode ser resolvido utilizando-se, por exemplo, uma recursão progressiva. Seja  $p_j$  o caminho de custo mínimo do nó 0 ao nó  $j$ , dado que o estoque inicial  $e_0 = 0$ . Então,

$$p_j = \min_{0 \leq i \leq j-1} \{p_i + c_{ij}\}, \quad j = 1, 2, 3, 4 \quad (5.3)$$

em que  $p_0 = 0$  e  $c_{ij}$  é o custo total incorrido devido à produção no período  $i + 1$  de um lote de tamanho  $d_{i+1} + d_{i+2} + \dots + d_j$ , e  $d_i$  é a demanda no período  $i$ . Essa recursão constitui o algoritmo de Wagner e Whitin (1958), apresentado no Capítulo 3. Usando a recursão (5.3), tem-se:

- Para  $j = 1$ :  $p_1 = p_0 + c_{01}$ ,  $c_{01} = \$10 + \$2 \times 30 = \$70$ ,  $p_1 = \$0 + \$70 = \$70$
- Para  $j = 2$ :  $p_2 = \min \{p_0 + c_{02}, p_1 + c_{12}\}$ ,  $c_{02} = \$10 + \$2 \times 50 + 0,1 \times \$2 \times 20 = \$114$ ,  $c_{12} = \$10 + \$2 \times 20 = \$50$ ,  $p_2 = \min \{\$0 + \$114, \$70 + \$50\} = \$114$
- Para  $j = 3$ :  $p_3 = \min \{p_0 + c_{03}, p_1 + c_{13}, p_2 + c_{23}\}$ ,  $c_{03} = \$10 + \$2 \times 110 + 0,1 \times \$2 \times (80 + 60) = \$258$ ,  $c_{13} = \$10 + \$2 \times 80 + 0,1 \times \$2 \times (60) = \$182$ ,  $c_{23} = \$10 + \$2 \times 60 = \$130$ ,  $p_3 = \min \{\$258, \$70 + \$182, \$114 + \$130\} = \$244$
- Para  $j = 4$ :  $p_4 = \min \{p_0 + c_{04}, p_1 + c_{14}, p_2 + c_{24}, p_3 + c_{34}\}$ ,  $c_{04} = \$10 + \$2 \times 120 + 0,1 \times \$2 \times (90 + 70 + 10) = \$284$ ,  $c_{14} = \$10 + \$2 \times 90 + 0,1 \times \$2 \times (70 + 10) = \$206$ ,  $c_{24} = \$10 + \$2 \times 70 + 0,1 \times \$2 \times 10 = \$152$ ,  $c_{34} = \$10 + \$2 \times 10 = \$30$ ,  $p_4 = \min \{\$284, \$70 + \$206, \$114 + \$152, \$244 + \$30\} = \$266$

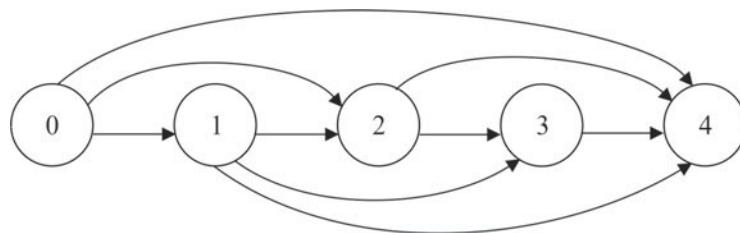
**Tabela 5.11**  
**Soluções ótimas dos problemas no estágio 2 da recursão regressiva (Exemplo 5.3) – inclusão do produto B.**

Quantidade de recurso já utilizado $J$	Produção ótima de B, $d_2^*(j)$	Lucro correspondente $m_2(j)$
0	2	179
1	1	169
2	3	150
3	1	129
4	0	119
5	2	100
6	0	79
7	1	50
8	1	50
9	0	0
10	0	0
11	0	0

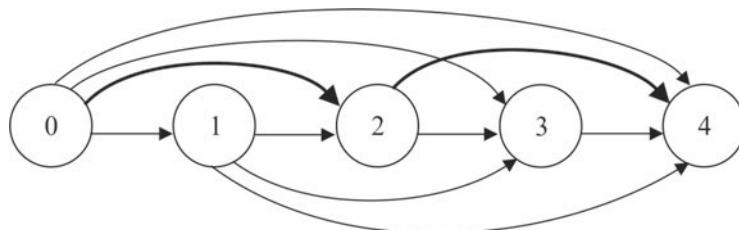
Portanto, o caminho ótimo custa \$266 mil e corresponde ao plano de se produzir um lote de tamanho  $d_3 + d_4 = 70$  no período 3 (custo associado  $p_4 = p_2 + c_{24}$ ) e um lote de tamanho  $d_1 + d_2$  no período 1 (custo associado  $p_2 = p_0 + c_{02}$ ). A Figura 5.9 ilustra o caminho de custo mínimo para este exemplo. Note que essa solução ótima encontrada é a mesma obtida no Exemplo 5.1.

Como pode ser observado, o Exemplo 5.1 foi resolvido por programação dinâmica com duas decomposições distintas, sendo que a resolução como um problema de caminho mínimo requer um esforço computacional menor (ou seja, menos cálculos) que a primeira decomposição proposta. Assim como neste caso, a existência de decomposições alternativas é verificada para muitos outros problemas de otimização. Um desafio na aplicação de programação dinâmica para a resolução de um problema de otimização é reconhecer como o problema pode ser decomposto em uma seqüência de problemas mais simples e computacionalmente mais eficientes de serem resolvidos. Desenvolver decomposições computacionalmente eficientes nem sempre é fácil, mas as chances de êxito são maiores se acumularmos experiência por meio de treinamento.

No exemplo anterior, a decomposição foi obtida devido a uma propriedade da solução ótima. A seguir é apresentada uma decomposição alternativa para o Exemplo 5.3, que utiliza uma equação recursiva mais econômica. Consideraremos um estágio para cada um dos possíveis valores  $j$  do recurso E. Em cada estágio existe um único estado, que coincide com o próprio estágio, correspondendo a termos  $j$  unidades ainda remanescentes do recurso para ser utilizado. A decisão a tomar em cada estágio é o que fabricar dentre os quatro produtos A, B, C ou D, utilizando a quantidade correspondente de recurso das  $k$  unidades existentes, sendo que o restante do recurso é utilizado da melhor forma possível para fabricar os produtos A, B, C e D. A recursão para este exemplo consiste em:



**Figura 5.8** Grafo orientado acíclico em que os arcos indicam os períodos cobertos com as produções de interesse em cada período.



**Figura 5.9** Caminho de custo mínimo.

$$\begin{aligned}
 l(j) &= \max_{i \in \{A,B,C,D\} \cap \{a_i \leq j\}} \{c(i) + l(j - a_i)\} \text{ ou} \\
 &= 0, \text{ caso não exista nenhum } i \text{ com } i \in \{A,B,C,D\} \cap \{a_i \leq j\}, \\
 j &= 1, 2, \dots; \\
 l(0) &= 0,
 \end{aligned}$$

em que

$c(i)$  é o lucro do produto  $i$ ,  $i \in \{A,B,C,D\}$ ,  
 $a_i$  é o montante do recurso E necessário à fabricação de uma unidade do produto  $i$ ,  
 $i \in \{A,B,C,D\}$ .

Aplicando esta recursão, obtemos:

- $l(0) = \$0$
- $l(1) = \max_{i \in \{A,B,C,D\} \cap \{a_i \leq 1\}} \{c(i) + l(1 - a_i)\} = \$0$  (não existe  $i$  com  $i \in \{A,B,C,D\} \cap \{a_i \leq 1\}$ )
- $l(2) = \max_{i \in \{A,B,C,D\} \cap \{a_i \leq 2\}} \{c(i) + l(2 - a_i)\} = \{c(A) + l(0)\} = \$15 + \$0 = \$15$ , com  $i^* = A$
- $l(3) = \max_{i \in \{A,B,C,D\} \cap \{a_i \leq 3\}} \{c(i) + l(3 - a_i)\} = \max \{c(A) + l(1), c(B) + l(0)\} = \max \{\$15 + \$0, \$50 + \$0\} = \$50$ , com  $i^* = B$
- $l(4) = \max_{i \in \{A,B,C,D\} \cap \{a_i \leq 4\}} \{c(i) + l(4 - a_i)\} = \max \{c(A) + l(2), c(B) + l(1)\} = \max \{\$15 + \$15, \$50 + \$0\} = \$50$ , com  $i^* = B$
- $l(5) = \max_{i \in \{A,B,C,D\} \cap \{a_i \leq 5\}} \{c(i) + l(5 - a_i)\} = \max \{c(A) + l(3), c(B) + l(2), c(C) + l(0)\} = \max \{\$15 + \$50, \$50 + \$15, \$79 + \$0\} = \$79$ , com  $i^* = C$

e assim, sucessivamente, calcula-se:

- $l(6) = \$100$ , com  $i^* = B$
- $l(7) = \$119$ , com  $i^* = D$
- $l(8) = \$129$ , com  $i^* = C$
- $l(9) = \$150$ , com  $i^* = B$
- $l(10) = \$169$ , com  $i^* = B$
- $l(11) = \$179$ , com  $i^* = B$

obtendo-se o lucro ótimo de  $\$179$ , o mesmo do Exemplo 5.3. A produção correspondente que leva a este lucro pode ser recuperada pelas decisões ótimas tomadas nos diversos subproblemas. O lucro ótimo  $l(11)$  foi obtido com a produção de 1 item B, que consome três unidades do recurso E. Com isso, os recursos disponíveis, que eram de 11 unidades, são reduzidos a  $(11 - 3) = 8$ . Para este montante de recursos, o lucro ótimo do subproblema  $l(8)$  foi obtido com a produção de 1 item C, que consome cinco unidades do recurso E. Com isso, os recursos disponíveis são reduzidos a  $(8 - 5) = 3$ . Para este montante de recursos, o lucro ótimo do subproblema  $l(3)$  foi obtido com a produção de 1 item B. Com isso, os recursos disponíveis são reduzidos a  $(3 - 3) = 0$ , não havendo mais recursos a utilizar.

Portanto, a produção correspondente ao lucro ótimo  $l(11)$  refere-se a 2 itens B e 1 item C. Note que  $l(8)$  e  $l(11)$  também poderiam ser obtidos com a escolha de  $i^* = B$  e  $i^* = C$ , respectivamente. Ainda assim, a solução ótima recuperada para este exemplo seria essencialmente a mesma daquela aqui ilustrada e obtida nas soluções anteriores. Cabe salientar que, além da preocupação com decomposições mais eficazes de programação dinâmica, na prática, também é preciso se preocupar com implementações computacionais mais eficientes, que economizem espaço de memória do computador e número de operações.<sup>9</sup>

<sup>9</sup> Outras propostas de implementações mais eficientes para o problema da mochila e variações podem ser encontradas, por exemplo, em Yanasse e Soma (1987), Yanasse et al. (2000), Soma e Toth (2002).

Os problemas de otimização considerados têm um comportamento dito determinístico, ou seja, tem-se conhecimento exato do estado para o qual o sistema evolui no próximo estágio, dado que o sistema está em um estado  $j$  de um estágio  $i$ , e uma decisão  $d_i(j)$ , entre as decisões possíveis permitidas, é tomada. Além disso, tem-se um conhecimento exato do custo incorrido com a decisão tomada. O termo determinístico serve para diferenciar de seu uso em problemas em que não se tem conhecimento exato de para onde o sistema evolui, ou do custo incorrido devido às incertezas de alguns parâmetros. Esses casos com incerteza serão analisados no Capítulo 6.

Cabe ressaltar que nem sempre a aplicação de programação dinâmica é recomendada para a solução de determinados problemas de otimização, uma vez que podem existir métodos alternativos de solução que são computacionalmente mais eficientes (exigem menos esforço computacional para resolver o problema). Por exemplo, a programação dinâmica pode ser aplicada para resolver qualquer problema de otimização linear,<sup>10</sup> mas, em geral, a resolução destes problemas é computacionalmente mais eficaz pelo método simplex apresentado no Capítulo 2.

## 5.5 O PROBLEMA DA DIMENSIONALIDADE EM PROGRAMAÇÃO DINÂMICA

Para ilustrar a aplicação da técnica de programação dinâmica, utilizamos os Exemplos 5.1 e 5.3, cujos estados do sistema foram descritos por uma única variável. Mesmo com uma única variável, o número de estados pode ser muito grande, dependendo dos diversos valores que a variável pode assumir. Uma dificuldade no emprego de programação dinâmica para a resolução de problemas práticos de otimização é a ocorrência de explosão de estados. Mais estados em cada estágio do problema implicam um número maior de cálculos a serem feitos.

Em geral, os estados de um sistema não se limitam à variação de apenas uma variável, mas de um conjunto delas, ou seja, o estado de um sistema é representado por um vetor multidimensional de dimensão  $n$ ,  $n \geq 1$ . Como no Exemplo 5.2, focamos um sistema de produção em que os estados, em cada estágio, são representados pelo número de unidades produzidas em estoque e pelas condições das máquinas 1 e 2 (em funcionamento ou não). Neste caso, temos, portanto,  $n = 3$ . Outros exemplos aparecem em problemas de corte, por exemplo, corte de peças retangulares de dimensões  $(l_i, w_i)$  a partir de chapas retangulares de dimensões  $(L, W)$  de estoque, e problemas de empacotamento, por exemplo, carregamento de caixas de dimensões  $(l_i, w_i, h_i)$  dentro de contêineres de dimensões  $(L, W, H)$ , em que o espaço de estados das formulações de programação dinâmica dependem dos tamanhos  $l_i, w_i, h_i, L, W, H$  desses itens.<sup>11</sup>

Quando se têm sistemas cujos estados são representados por variáveis multidimensionais, o número de estados possíveis tende a ser muito grande. Suponhamos, por exemplo, que cada um dos possíveis campos do vetor  $n$ -dimensional de estado possa tomar 10 possíveis valores diferentes em cada estágio. Desta maneira, o número de possíveis estados para o sistema já seria de  $10 \times 10 \times \dots \times 10 = 10^n$  estados. Se tivéssemos mais variáveis envolvidas, esse número de estados cresceria ainda mais. Com o aumento de  $n$ , o número de estados acaba facilmente chegando a valores intratáveis na prática, mesmo levando em conta a grande capacidade computacional disponível atualmente. Este problema da dimensionalidade em programação dinâmica foi denominado *maldição da dimensionalidade* por R. Bellman. Uma alternativa para tratar o problema de dimensionalidade é relaxar parte do espaço de estados usando multiplicadores de Lagrange e aplicar procedimentos do tipo otimização do subgradiente. A apresentação desse método está além do escopo deste livro.<sup>12</sup>

10 Ver, por exemplo, Taha (1976).

11 Para exemplos de aplicação de programação dinâmica em problemas de corte de fábricas de chapas duras (*hardboards*), veja Morabito e Belluzzo (2006).

12 Para um exemplo de aplicação deste método em problemas de corte de chapas, o leitor pode consultar Christofides e Hadjiconstantinou (1995).

## 5.6 EXERCÍCIOS<sup>13</sup>

**Exercício 5.1** Utilize uma recursão regressiva para determinação de um caminho ótimo para o problema do Exemplo 5.1.

**Exercício 5.2** Aplique programação dinâmica para resolver o problema do Exemplo 5.1, quando existe uma limitação de capacidade de estocagem de 10t em cada período.

**Exercício 5.3** Aplique programação dinâmica para resolver o problema de caminho mínimo do Exemplo 4.12 do Capítulo 4, com os dados numéricos fornecidos naquele exemplo.

**Exercício 5.4** No Exemplo 5.2 anterior, determine  $r_2(e_0) = r_2((0,0,1))$ .

**Exercício 5.5** Utilize programação dinâmica para determinar a solução ótima do seguinte problema:

$$\begin{aligned} \text{Maximizar } & x_1^3 + x_2^3 + x_3^3 \\ & x_1 + x_2 + x_3 \leq 11 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{aligned}$$

**Exercício 5.6** Utilize programação dinâmica para determinar a solução ótima do seguinte problema:

$$\begin{aligned} \text{Maximizar } & x_1^3 + x_2^3 + x_3^3 \\ & x_1 x_2 x_3 \leq 11 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{aligned}$$

**Exercício 5.7** Uma construtora está atualmente envolvida na construção de três grandes obras, denominadas 1, 2 e 3. Às obras 1, 2 e 3 foram designados uma certa quantidade de máquinas pesadas, um certo número de homens-hora, e caminhões e materiais diversos, com os quais a empresa estima que serão completadas em 13, 17 e 22 meses, respectivamente. A diretoria gostaria de antecipar a data de término das obras e, para isso, decidiu aplicar mais \$30 milhões. Os novos prazos para término de cada obra em função de quanto é investido estão apresentados na Tabela 5.12. Qual deve ser a quantidade a ser designada a cada obra, de modo que a construtora obtenha o máximo possível de redução nos prazos de término das três obras? Admita que apenas uma injeção de recursos em múltiplos de \$5 milhões seja possível.

**Exercício 5.8** Um engenheiro aproveitou suas férias para fazer turismo na Europa. Ele gostaria de visitar Lisboa e Paris, e comprou uma passagem aérea de ida chegando a Lisboa e com retorno saindo de Paris. Para o trecho Lisboa até Paris, o engenheiro pretende alugar um carro e escolher uma rota que passe por algumas atrações turísticas que ele gostaria de visitar. O aluguel diário do carro é de \$100. O engenheiro estudou os locais que estaria interessado em visitar e montou a Tabela 5.13 de trechos de rodovias que ele poderia utilizar. Para cada um dos trechos de rodovias, o engenheiro calculou o número de dias que demoraria para percorrê-lo e as despesas incorridas no trecho, composto de despesas com combustível, alojamento, comida, pedágios, taxas

<sup>13</sup> Alguns dos exercícios desta lista foram inspirados em exercícios em Bradley *et al.* (1977) e Taha (2003).

**Tabela 5.12**  
**Dados do Exercício 5.7**

Recursos adicionais injetados (milhões)	Obra 1	Obra 2	Obra 3
0	13	17	22
5	10	13	18
10	9	11	15
15	8	9	15
20	7	8	13
25	6	8	11
30	5	7	10

de entrada nas atrações turísticas existentes etc. (i) Determine uma rota para o engenheiro, de Lisboa a Paris, que seja a mais rápida possível. (ii) Determine uma rota para o engenheiro, de Lisboa a Paris, que seja a mais econômica possível, incluído o custo do aluguel do carro. (iii) Se o engenheiro dispuser de apenas 13 dias para percorrer o trecho de Lisboa a Paris, ele pode utilizar a rota mais econômica?

**Exercício 5.9** Um fabricante de bebidas tem contratos para entrega de 40 mil litros de seu produto principal para o final deste mês, e 280 mil litros para o final do próximo mês. Os custos de produção variam mês a mês, pois dependem de preços negociados para o fornecimento da matéria-prima do produto. O fabricante estima que os custos de produção deste e do próximo mês são:  $c_1(x_1) = 30.000 + (x_1 - 100)^2$  e  $c_2(x_2) = 30.000 + (x_2 - 80)^2$ , respectivamente, em que  $x_1$  e  $x_2$  são as quantidades produzidas em cada mês, em unidades de mil litros. Quais devem ser os valores de  $x_1$  e  $x_2$  que minimizam o custo total, dado que a demanda de cada mês tem de ser atendida sem atraso, e o custo de estocagem da produção excedente de um mês para o outro é de \$ 6 por mil litros.

**Exercício 5.10** Devido à sazonalidade de vendas dos produtos, o gerente de uma loja está decidindo quantos empregados manter em cada um dos próximos cinco meses. O número  $l_i$  de empregados considerado ideal para o próximo mês  $i$  é de 5, 7, 8, 4 e 6, para  $i = 1, 2, 3, 4$  e 5, respectivamente. Em nenhum dos próximos cinco meses o gerente pretende manter menos do que o número ideal de empregados, uma vez que ele está preocupado em servir bem os clientes da loja. Seja  $x_i$  ( $x_i \geq l_i$ ) o número de empregados que o gerente vai manter no próximo mês  $i$ . Se  $x_i > l_i$ , o gerente incorre em um custo de excesso de empregados no mês  $i$  de  $CE(x_i) = 300(x_i - l_i)$ . Por outro lado, se o gerente contrata empregados de um mês para outro, ou seja, se  $x_{i-1} < x_i$ , o gerente incorre em um custo de contratação de  $CT(x_{i-1}, x_i) = 400 + 200(x_i - x_{i-1})$ . Além disso, cada vez que o gerente dispensa empregados, ele incorre em um custo de \$100 por empregado dispensado. Determine uma solução de mínimo custo para o gerente da loja, utilizando programação dinâmica e sabendo que a loja conta hoje com cinco empregados.

**Tabela 5.13**  
**Dados do Exercício 5.8**

Origem	Destino	Dias	Custo
Lisboa	A	2	\$160
Lisboa	B	3	\$140
Lisboa	C	2	\$100
A	D	2	\$100
A	E	2	\$150
A	F	4	\$260
B	D	3	\$250
B	E	4	\$200
B	F	4	\$200
C	D	2	\$160
C	E	1	\$280
C	F	1	\$60
D	G	4	\$320
D	H	2	\$120
E	G	3	\$180
E	H	6	\$300
F	G	7	\$360
F	H	6	\$300
G	Paris	5	\$350
H	Paris	8	\$320

**Exercício 5.11** Utilize programação dinâmica para determinar a solução ótima do seguinte problema de programação linear:

$$\begin{aligned}
 & \text{Minimizar } f(x_1, x_2) = -2x_1 - x_2 \\
 & x_1 + x_2 \leq \frac{9}{2} \\
 & x_1 \leq \frac{7}{2} \\
 & x_2 \leq \frac{5}{2} \\
 & x_1 \geq 0, x_2 \geq 0
 \end{aligned}$$

**Exercício 5.12** Utilize programação dinâmica para determinar a solução ótima do Exercício 5.11 anterior, com a restrição adicional de que  $x_1$  e  $x_2$  sejam inteiros.

**Exercício 5.13** Uma empresa tem hoje em carteira quatro pedidos de clientes para seu produto. A empresa prometeu entregar os pedidos dos clientes nos dias indicados na tabela a seguir. O tempo exigido pela empresa para preparar o pedido de cada um dos clientes também está indicado na tabela. Como cada dia de atraso na entrega do produto em relação à data prometida implica perdas para os clientes, a empresa mantém um contrato padrão de multa de \$1.000 por dia de atraso. Por outro lado, para cada dia de antecipação na entrega do produto em relação à data prometida, a empresa recebe um adicional de \$200. Aplique programação dinâmica para determinar se a empresa terá de desembolsar algum dinheiro. Se a empresa tiver de desembolsar algum dinheiro, qual deve ser a ordem em que os clientes devem ser atendidos para que o total a ser desembolsado seja o menor possível? Ressalta-se que, uma vez iniciada a produção da ordem de algum cliente, a empresa não pode interromper a produção até que o pedido seja inteiramente completado, pois os custos de reiniciar a produção são muito altos.

---

**Tabela 5.14**  
**Dados do Exercício 5.13**

<i>Pedido do Cliente</i>	<i>Duração da produção (dias)</i>	<i>Data de entrega (dias a partir de hoje)</i>
1	3	5
2	9	16
3	7	11
4	5	12

# 6

## Programação dinâmica estocástica

No Capítulo 5, apresentamos a técnica de programação dinâmica para resolver um problema geral de otimização linear ou não-linear, contínua ou discreta, com funções diferenciáveis ou não diferenciáveis etc. O problema é decomposto em uma seqüência de problemas menores e mais simples de serem resolvidos, que são relacionados por uma *equação recursiva regressiva* (ver expressão (5.1) do Capítulo 5). Nesta equação, admitimos que todos os termos são conhecidos com certeza, ou seja, temos conhecimento completo do estado para o qual o sistema evolui em cada estágio do sistema, dada a decisão tomada, e do custo ou retorno incorrido.

Neste capítulo, estudamos a aplicação de programação dinâmica em situações em que não temos conhecimento exato de todos os termos da equação recursiva, razão da denominação *programação dinâmica estocástica*. A programação dinâmica estocástica tem sido aplicada em diversas áreas, por exemplo, em programação da produção, na qual a demanda em cada período não é conhecida com certeza, em investimentos de capital em que o retorno de cada investimento não é conhecido com certeza, em jogos em que as jogadas dos outros jogadores não são conhecidas com certeza, e ainda em confiabilidade de sistemas, manutenção de máquinas, distribuição de produtos etc.

Na ocorrência de incertezas, nem sempre temos conhecimento exato do estado para o qual o sistema evolui. Desta forma, uma decisão tomada em um determinado estado de um estágio não necessariamente determina, inequivocamente, o estado do sistema no próximo estágio. O custo/lucro exato incorrido pode, também, não ser totalmente conhecido. Como aplicar, então, a programação dinâmica nesses casos, ou seja, na presença de incertezas? Como os estados resultantes e os custos/lucros associados não são conhecidos com certeza em cada estágio, não é possível otimizar a soma dos custos/lucros em cada estágio.

Existem casos, entretanto, em que se conhece qual seria o estado resultante no próximo estágio, de acordo com uma certa probabilidade, ou seja, o estado resultante é governado segundo uma variável aleatória com uma certa distribuição de probabilidade. Nestes casos, pode-se otimizar o retorno esperado nos diversos estágios do problema, levando-se em conta a ordem em que as decisões são tomadas. Isto é, em cada estágio, leva-se em conta a decisão ótima que seria tomada no estágio seguinte, caso um determinado estado resulte da decisão tomada no presente estágio.

### 6.1 A EQUAÇÃO RECURSIVA REGRESSIVA

Suponha um problema com  $T$  estágios. Seja  $r_i(j)$  o menor custo esperado possível que se pode obter nos estágios  $i, i + 1, i + 2, \dots$ , até o estágio  $T$ , dado que no estágio  $i$  o estado observado do

sistema é  $j$ . Uma vez que estamos no estágio  $i$  e o estado do sistema é  $j$ , pode-se tomar uma decisão  $d_i(j)$  de um conjunto  $D_i(j)$  de todas as decisões factíveis no estágio  $i$  a partir do estado  $j$ . Com essa decisão  $d_i(j)$  tomada, incorre-se em um custo  $c(d_i(j), s_i)$  no estado  $j$  do estágio  $i$ , que depende da decisão tomada e do valor  $s_i$  de uma variável aleatória  $S_i$ . Com essa decisão, o sistema evolui para um novo estado  $e(d_i(j), s_i)$  no estágio seguinte  $i + 1$ .

Denotemos por  $E[X]$  o valor esperado de uma variável aleatória  $X$ . Se o valor de  $r_{i+1}(e(d_i(j), s_i))$  é conhecido, ou seja, o menor custo esperado possível que se pode obter nos estágios  $i + 1, i + 2, \dots$ , até o estágio  $T$  é conhecido, dado que o estado observado do sistema no estágio  $i + 1$  é  $e(d_i(j), s_i)$ , então  $r_i(j)$  pode ser obtido calculando-se:

$$r_i(j) = \min_{d_i(j) \in D_i(j)} E[c(d_i(j), S_i) + r_{i+1}(e(d_i(j), S_i))] \quad (6.1)$$

ou seja, avaliam-se os custos esperados de todas as decisões possíveis no estágio  $i$ , dado que o estado observado é  $j$ , e toma-se o menor entre eles.

O cálculo da recursão (6.1) é iniciado no último estágio  $T$ , e prossegue regressivamente até se chegar a  $r_1(j)$ . Esta recursão é similar à equação recursiva regressiva utilizada na programação dinâmica determinística (fórmula (5.1) do Capítulo 5). A diferença está na otimização do custo esperado, devido às incertezas inerentes ao sistema em consideração.

A expressão (6.1) foi escrita levando-se em conta que estamos considerando um problema de minimização de custos esperados. Ela também pode ser escrita na forma de *maximização* quando, por exemplo, em vez de minimizar custos esperados, estamos interessados em maximizar a margem de contribuição ao lucro esperado. Cabe alertar que, em problemas com incerteza, ao contrário do que foi feito no Capítulo 5 para o caso determinístico, somente a aplicação da recursão regressiva de programação dinâmica é possível. A recursão progressiva de programação dinâmica não pode ser aplicada, pois não sabemos de que estado, no estágio anterior, o presente estado se originou e, sem esta informação, não é possível realizar qualquer cálculo.

## 6.2 EXEMPLOS NUMÉRICOS

### 6.2.1 Controle de estoque

A seguir ilustramos a aplicação de programação dinâmica estocástica por meio de um exemplo de um problema de controle de estoque, em que a demanda por um produto é incerta.

**Exemplo 6.1** O dono de um pequeno estabelecimento comercial gostaria de repor o estoque de um produto sazonal que ele comercializa. A entrega desse produto é realizada apenas no início de cada semana. Uma vez encomendado o produto, sua entrega é praticamente imediata. O custo do produto é de \$100 por unidade e ele é vendido a \$200 por unidade. Existe um custo de estocagem de \$10 por unidade, por semana, para cada unidade não vendida durante a semana. Estamos, no momento, a apenas duas semanas do final da temporada e o produto, caso não seja vendido neste período, será perdido. O dono do estabelecimento está permanentemente atento a seu negócio e preocupado com sua preservação. A falta de algum produto no estabelecimento causa insatisfação e perda de confiança do cliente. O dono estima que a insatisfação e a perda de confiança do cliente podem ser avaliadas monetariamente a um custo de \$20 por unidade de demanda não satisfeita, além do lucro perdido de \$100 da venda do produto. A dificuldade do dono do estabelecimento de repor o estoque reside no fato de a demanda do produto nas próximas semanas não ser conhecida com certeza. Pelo histórico de vendas realizadas em períodos anteriores, ele chegou à conclusão de que a demanda semanal do produto, nestas duas semanas, deve obedecer à distribuição de probabilidade apresentada na Tabela 6.1. Quanto o dono do estabelecimento deve encomendar desse produto nesta semana, dado que seu estoque atual é nulo?

**Tabela 6.1**  
**Demanda do produto durante cada semana.**

Demanda	Probabilidade
0	0,05
1	0,35
2	0,30
3	0,25
4	0,05

O problema do Exemplo 6.1 pode ser resolvido por meio da recursão regressiva (6.1), escrita na forma de maximização. Seja  $r_i(j)$  o máximo retorno esperado, dado que estamos na semana (estágio)  $i$  e observamos o nível de estoque (estado)  $j$ . O retorno após a semana 2 é zero, uma vez que não há mais vendas do produto depois do final da temporada, independentemente do nível de estoque observado no início da semana 3. Portanto,  $r_3(j) = \$0$  para todo  $j$ . Para a semana 3, podemos limitar nossos cálculos aos níveis de estoque 0, 1, 2, 3 e 4 pois, como a demanda de cada semana é no máximo 4 (informação extraída da Tabela 6.1), qualquer quantidade do produto superior a 4 no início de qualquer semana implica que haverá estoque positivo desse produto para a semana seguinte.

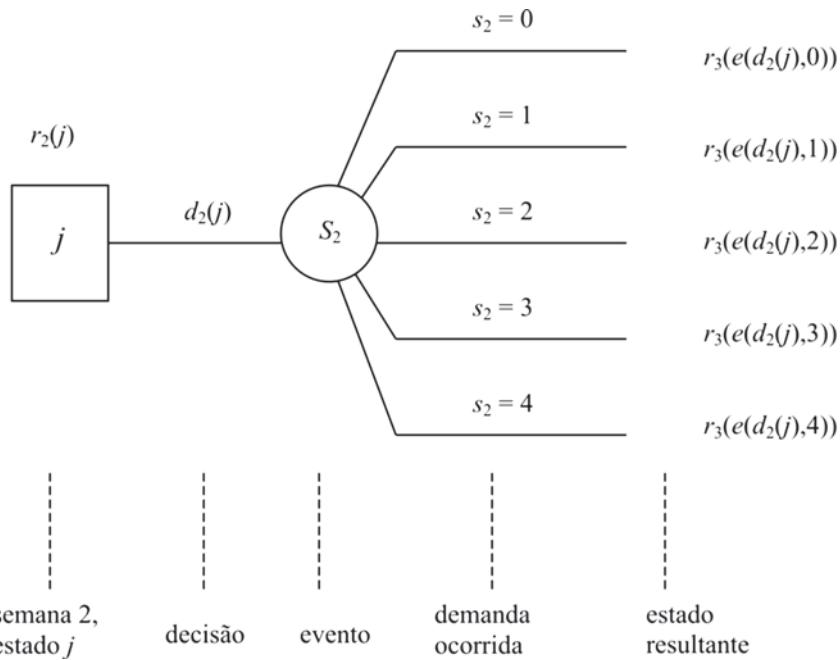
Como é possível sempre repor o estoque do produto no início de cada semana sem acréscimo de custo em relação à semana anterior, não vale a pena carregar estoque de uma semana para outra. Assim, não vale a pena repor o estoque para um nível superior a quatro unidades no início de qualquer semana. Conseqüentemente, não se observará mais do que quatro unidades do produto em estoque ao final de cada semana. De posse de  $r_3(j)$ , podemos determinar:

$$r_2(j) = \max_{d_2(j) \in D_2(j)} E[c(d_2(j), S_2) + r_3(e(d_2(j), S_2))] \quad \text{para } j = 0, 1, 2, 3 \text{ e } 4 \quad (6.2)$$

As possibilidades que podem ocorrer no estágio 2 e estado  $j$  estão ilustradas na Figura 6.1.

Ilustremos o cálculo de  $r_2(j)$  para cada um dos possíveis valores observáveis de estoque  $j$ . Tomemos inicialmente  $j = 0$ . As encomendas (decisões) possíveis são 0, 1, 2, 3 e 4, ou seja,  $D_2(j) = \{0, 1, 2, 3, 4\}$ . Consideremos o caso em que é feita uma encomenda de 0, ou seja,  $d_2(0) = 0$ . O estoque inicial zero leva somente ao estado de estoque 0 na semana seguinte, independentemente de qual seja a demanda observada  $s_2$  da semana, ou seja,  $e(d_2(0), s_2) = 0$  para todo  $s_2$ . Já temos calculado o valor de  $v_3(0) = \$0$ . Resta calcular os diversos lucros  $c(d_2(0), s_2)$  incorridos dependentes da demanda observada  $s_2$ . Apesar de o estado observado na semana seguinte ser sempre 0 neste caso, os lucros incorridos diferem, dependendo da demanda observada  $s_2$ . As demandas na semana podem ser 0, 1, 2, 3 e 4:

- Se a demanda  $s_2$  for 0, não há perda nenhuma e não há ganho algum. Isso ocorre com uma probabilidade de 0,05. Portanto,  $c(0,0) = \$0$ , com probabilidade 0,05.
- Se a demanda  $s_2$  for 1, há uma perda de lucro de  $\$100$ , mais a perda de confiança do cliente de  $\$20$ . Isso ocorre com uma probabilidade de 0,35. Portanto,  $c(0,1) = -\$120$ , com probabilidade 0,35.
- Se a demanda  $s_2$  for 2, há uma perda de lucro de  $\$200$ , mais a perda de confiança de clientes de  $\$40$ . Isso ocorre com uma probabilidade de 0,30. Portanto,  $c(0,2) = -\$240$ , com probabilidade 0,30.



**Figura 6.1** Ilustração das possibilidades existentes para o cálculo de  $r_2(j)$  em (6.2).

- Se a demanda  $s_2$  for 3, há uma perda de lucro de \$300, mais a perda de confiança de clientes de \$60. Isso ocorre com uma probabilidade de 0,25. Portanto,  $c(0,3) = -\$360$ , com probabilidade 0,25.
- Se a demanda  $s_2$  for 4, há uma perda de lucro de \$400, mais a perda de confiança do cliente de \$80. Isso ocorre com uma probabilidade de 0,05. Portanto,  $c(0,4) = -\$480$ , com probabilidade 0,05.

Portanto, o retorno esperado com essa decisão é:

$$E[c(d_2(0), S_2) + r_3(e(d_2(0), S_2))] = E[c(0, S_2) + r_3(0)] = E[c(0, S_2)] = 0,05 c(0,0) + 0,35 c(0,1) + 0,30 c(0,2) + 0,25 c(0,3) + 0,05 c(0,4) = \$0 - \$42 - \$72 - \$90 - \$24 = -\$228$$

No caso de a encomenda ser de 1 unidade, ou seja,  $d_2(0) = 1$ , temos o seguinte. O custo da encomenda é de \$100. Se as demandas na semana forem 1, 2, 3 e 4, o estoque final será 0; caso contrário, o estoque final será 1. Para qualquer um desses estoques resultantes, já calculamos que  $r_3(j) = \$0$ .

- Se a demanda  $s_2$  for 0, incorre-se em um custo de estocagem de \$10. Isso ocorre com uma probabilidade de 0,05. Portanto,  $c(1,0) = -\$100 - \$10 = -\$110$ , com probabilidade 0,05.
- Se a demanda  $s_2$  for 1, há uma receita de \$200. Isso ocorre com uma probabilidade de 0,35. Portanto,  $c(1,1) = -\$100 + \$200 = \$100$ , com probabilidade 0,35.
- Se a demanda  $s_2$  for 2, há uma receita de \$200 e uma perda de lucro de \$100, mais a perda de confiança de clientes de \$20. Isso ocorre com uma probabilidade de 0,30. Portanto,  $c(1,2) = -\$100 + \$200 - \$100 - \$20 = -\$20$ , com probabilidade 0,30.
- Se a demanda  $s_2$  for 3, há uma receita de \$100 e uma perda de lucro de \$200, mais a perda de confiança de clientes de \$40. Isso ocorre com uma probabilidade de 0,25. Portanto,  $c(1,3) = -\$100 + \$200 - \$200 - \$40 = -\$140$ , com probabilidade 0,25.

- Se a demanda  $s_2$  for 4, há uma receita de \$100 e uma perda de lucro de \$300, mais a perda de confiança do cliente de \$60. Isso ocorre com uma probabilidade de 0,05. Portanto,  $c(1,4) = -\$100 + \$200 - \$300 - \$60 = -\$260$ , com probabilidade 0,05.

O retorno esperado com essa decisão é:

$$\begin{aligned} E[c(d_2(0), S_2) + r_3(e(d_2(0), S_2))] &= E[c(1, S_2) + r_3(e(1, S_2))] = E[c(1, S_2)] = 0,05 c(1,0) \\ &+ 0,35 c(1,1) + 0,30 c(1,2) + 0,25 c(1,3) + 0,05 c(1,4) = -0,05 \$110 + 0,35 \$100 \\ &- 0,30 \$20 - 0,25 \$140 - 0,05 \$260 = -\$5,50 + \$35 - \$6 - \$35 - \$13 = -\$24,50 \end{aligned}$$

De maneira similar, são realizados os cálculos para  $d_2(0) = 2, 3$  e  $4$ . Os valores obtidos destes cálculos estão resumidos na Tabela 6.2. De posse desses valores, determina-se  $r_2(0)$ , o retorno máximo esperado quando não há nenhum produto em estoque no estágio 2. Ele é dado por:

$$\begin{aligned} r_2(0) &= \max_{d_2(0) \in D_2(0)} E[c(d_2(0), S_2) + r_3(e(d_2(0), S_2))] = \max \{-\$228, -\$24,50, +\$63,50, \\ &+ \$52,50, -\$41\} = \$63,50 \end{aligned}$$

Para obter esse retorno, a decisão ótima deve ser  $d_2^*(0) = 2$ , ou seja, encomendar duas unidades do produto. A Tabela 6.2 resume os valores dos cálculos efetuados para encontrar  $r_2(0)$ .

**Tabela 6.2**  
Estágio 2, estado  $j = 0$ .

Encomenda	Custo \$	Demandas	Probabilidade	Receita \$	Custo estocagem \$	Perda lucro \$	Perda credib. cliente \$	Receita esperada \$
0	0	0	0,05	0	0	0	0	-228
		1	0,35	0	0	-100	-20	
		2	0,30	0	0	-200	-40	
		3	0,25	0	0	-300	-60	
		4	0,05	0	0	-400	-80	
1	-100	0	0,05	0	-10	0	0	-24,50
		1	0,35	200	0	0	0	
		2	0,30	200	0	-100	-20	
		3	0,25	200	0	-200	-40	
		4	0,05	200	0	-300	-60	
2	-200	0	0,05	0	-20	0	0	63,50
		1	0,35	200	-10	-100	-20	
		2	0,30	400	0	0	0	
		3	0,25	400	0	-100	-20	
		4	0,05	400	0	-200	-40	
3	-300	0	0,05	0	-30	0	0	52,50
		1	0,35	200	-20	0	0	
		2	0,30	400	-10	0	0	
		3	0,25	600	0	0	0	
		4	0,05	600	0	-100	-20	
4	-400	0	0,05	0	-40	0	0	-41
		1	0,35	200	-30	0	0	
		2	0,30	400	-20	0	0	
		3	0,25	600	-10	0	0	
		4	0,05	800	0	0	0	

De maneira similar, pode-se calcular  $r_2(j), j = 1, 2, 3 \text{ e } 4$ , obtendo-se:

$$r_2(1) = + \$ 163,50, \text{ com } d_2^*(1) = 1$$

$$r_2(2) = + \$ 263,50, \text{ com } d_2^*(2) = 0$$

$$r_2(3) = + \$ 352,50, \text{ com } d_2^*(3) = 0$$

$$r_2(4) = + \$ 359,00, \text{ com } d_2^*(4) = 0$$

Os valores dos cálculos para obtenção desses valores são apresentados nas Tabelas 6.3, 6.4, 6.5 e 6.6, respectivamente.

Tendo calculado todos os valores de  $r_2(j), j = 0, 1, 2, 3, 4$ , podemos calcular  $r_1(0)$ , usando novamente a recursão. As decisões  $d_1^*(0)$  possíveis são ordenar 0, 1, 2, 3 e 4 unidades do produto. Da Tabela 6.7, observamos que os retornos esperados correspondentes a essas decisões são  $-\$152,50, + \$55,40, + \$179,17, + \$233,85 \text{ e } + \$223,87$ , respectivamente. Portanto, chegamos à conclusão de que  $r_1(0) = + \$233,85$ , obtido de:

$$\begin{aligned} r_1(0) &= \max_{d_1(0) \in D_1(0)} E[c(d_1(0), S_1) + r_2(e(d_1(0), S_1))] = \max \{-\$152,50, + \$55,40, + \$179,17, \\ &+ \$233,85, + \$223,87\} = \$233,85, \text{ com } d_1^*(0) = 3 \end{aligned}$$

Os cálculos para se chegar ao valor de  $r_1(0)$  estão sumarizados na Tabela 6.7.

A variável aleatória de demanda, neste exemplo, é discreta (isto é, assume apenas valores inteiros). Em outros contextos, a variável aleatória em questão pode ser contínua. A programação dinâmica estocástica também pode ser aplicada neste caso, de maneira similar. A fórmula recursiva é válida, porém o valor esperado dessa fórmula envolve cálculos integrais, em vez de

**Tabela 6.3**  
**Estágio 2, estado  $j = 1$ .**

Encomenda	Custo \$	Demanda	Probabilidade	Receita \$	Custo estocagem \$	Perda lucro \$	Perda credib. cliente \$	Receita esperada \$
0	0	0	0,05	0	-10	0	0	75,50
		1	0,35	200	0	0	0	
		2	0,30	200	0	-100	-20	
		3	0,25	200	0	-200	-40	
		4	0,05	200	0	-300	-60	
1	-100	0	0,05	0	-20	0	0	163,50
		1	0,35	200	-10	0	0	
		2	0,30	400	0	0	0	
		3	0,25	400	0	-100	-20	
		4	0,05	400	0	-200	-40	
2	-200	0	0,05	0	-30	0	0	152,50
		1	0,35	200	-20	0	0	
		2	0,30	400	-10	0	0	
		3	0,25	600	0	0	0	
		4	0,05	600	0	-100	-20	
3	-300	0	0,05	0	-40	0	0	69,50
		1	0,35	200	-30	0	0	
		2	0,30	400	-20	0	0	
		3	0,25	600	-10	0	0	
		4	0,05	800	0	0	0	

**Tabela 6.4**  
**Estágio 2, estado  $j = 2$ .**

<i>Encomenda</i>	<i>Custo \$</i>	<i>Demanda</i>	<i>Probabilidade</i>	<i>Receita \$</i>	<i>Custo estocagem \$</i>	<i>Perda lucro\$</i>	<i>Perda credib. cliente \$</i>	<i>Receita esperada \$</i>
0	0	0	0,05	0	-20	0	0	263,50
		1	0,35	200	-10	0	0	
		2	0,30	400	0	0	0	
		3	0,25	400	0	-100	-20	
		4	0,05	400	0	-200	-40	
1	-100	0	0,05	0	-30	0	0	252,50
		1	0,35	200	-20	0	0	
		2	0,30	400	-10	0	0	
		3	0,25	600	0	0	0	
		4	0,05	600	0	-100	-20	
2	-200	0	0,05	0	-40	0	0	157,50
		1	0,35	200	-30	0	0	
		2	0,30	400	-20	0	0	
		3	0,25	600	-10	0	0	
		4	0,05	800	0	0	0	

**Tabela 6.5**  
**Estágio 2, estado  $j = 3$ .**

<i>Encomenda</i>	<i>Custo \$</i>	<i>Demanda</i>	<i>Probabilidade</i>	<i>Receita \$</i>	<i>Custo estocagem \$</i>	<i>Perda lucro\$</i>	<i>Perda credib. cliente \$</i>	<i>Receita esperada \$</i>
0	0	0	0,05	0	-30	0	0	352,50
		1	0,35	200	-20	0	0	
		2	0,30	400	-10	0	0	
		3	0,25	600	0	0	0	
		4	0,05	600	0	-100	-20	
1	-100	0	0,05	0	-40	0	0	259
		1	0,35	200	-30	0	0	
		2	0,30	400	-20	0	0	
		3	0,25	600	-10	0	0	
		4	0,05	800	0	0	0	

**Tabela 6.6**  
**Estágio 2, estado  $j = 4$ .**

<i>Encomenda</i>	<i>Custo \$</i>	<i>Demanda</i>	<i>Probabilidade</i>	<i>Receita \$</i>	<i>Custo estocagem \$</i>	<i>Perda lucro\$</i>	<i>Perda credib. cliente \$</i>	<i>Receita esperada \$</i>
0	0	0	0,05	0	-40	0	0	359
		1	0,35	200	-30	0	0	
		2	0,30	400	-20	0	0	
		3	0,25	600	-10	0	0	
		4	0,05	800	0	0	0	

**Tabela 6.7**  
**Estágio 1, estado  $j = 0$ .**

<i>Encomenda</i>	<i>Custo \$</i>	<i>Demanda</i>	<i>Probabilidade</i>	<i>Receita \$</i>	<i>Custo estoquegem \$</i>	<i>Perda lucro \$</i>	<i>Perda credib. cliente \$</i>	<i>Retorno do estado resultante \$</i>	<i>Receita esperada</i>
0	0	0	0,05	0	0	0	0	75,50	-152,50
		1	0,35	0	0	-100	-20	75,50	
		2	0,30	0	0	-200	-40	75,50	
		3	0,25	0	0	-300	-60	75,50	
		4	0,05	0	0	-400	-80	75,50	
1	-100	0	0,05	0	-10	0	0	163,50	55,40
		1	0,35	200	0	0	0	75,50	
		2	0,30	200	0	-100	-20	75,50	
		3	0,25	200	0	-200	-40	75,50	
		4	0,05	200	0	-300	-60	75,50	
2	-200	0	0,05	0	-20	0	0	263,00	179,17
		1	0,35	200	-10	-100	-20	163,50	
		2	0,30	400	0	0	0	75,50	
		3	0,25	400	0	-100	-20	75,50	
		4	0,05	400	0	-200	-40	75,50	
3	-300	0	0,05	0	-30	0	0	352,00	233,85
		1	0,35	200	-20	0	0	263,00	
		2	0,30	400	-10	0	0	163,50	
		3	0,25	600	0	0	0	75,50	
		4	0,05	600	0	-100	-20	75,50	
4	-400	0	0,05	0	-40	0	0	359,00	223,87
		1	0,35	200	-30	0	0	352,50	
		2	0,30	400	-20	0	0	263,00	
		3	0,25	600	-10	0	0	163,50	
		4	0,05	800	0	0	0	75,50	

somatórias.<sup>1</sup> Além disso, os estados possíveis podem ser discretos, como no Exemplo 6.1, ou contínuos (e as decisões possíveis também em espaços contínuos).<sup>2</sup> Neste livro, tratamos apenas de programação dinâmica estocástica com estados discretos.

O problema do Exemplo 6.2 a seguir não tem um retorno monetário. Trata-se de um jogo de soma zero (do tipo do analisado na Seção 2.11 do Capítulo 2), ou seja, se alguém vence o outro perde, ou há um empate.

### 6.2.2 Jogo de soma zero

**Exemplo 6.2** (*Jogo-da-Velha*) Dados uma tabela com três linhas e três colunas, e dez peças de duas cores, sendo cinco peças de cada cor, as seguintes regras definem o jogo:

- Dois jogadores.
- A tabela está originalmente vazia, sem nenhuma peça.
- Cada jogador escolhe uma cor diferente de peças.

<sup>1</sup> Ver, por exemplo, Nemhauser (1966), White (1969) e Bertsekas (2001, 2005).

<sup>2</sup> Ver, por exemplo, Ravindran *et al.* (1987) e Bertsekas (2001, 2005).

- Em cada jogada, o jogador da vez coloca uma peça de sua cor em uma das nove posições da tabela, desde que na posição selecionada ainda não exista nenhuma peça.
- As peças são colocadas pelos jogadores de forma alternada, uma de cada vez.
- O jogo termina tão logo um jogador posicione três de suas peças em uma mesma linha, coluna ou diagonal ou se esgotem as posições livres na tabela. No primeiro caso, há um vencedor, que é o jogador que conseguiu posicionar suas três peças em uma mesma linha, coluna ou diagonal. No segundo caso, há um empate.
- O jogador que coloca a primeira peça é definido por sorteio.

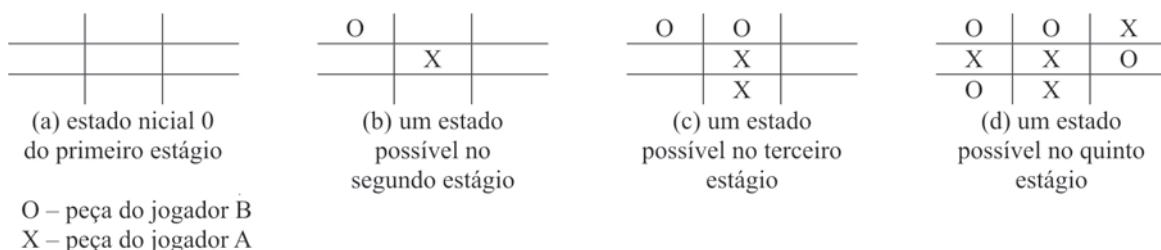
Suponha que dois jogadores A e B disputam uma partida de jogo-da-velha. A partida que A está disputando constitui um problema de otimização em que o objetivo é ganhar a partida, aumentando (otimizando) assim a sua satisfação ou auto-estima. De maneira similar, o jogador B tem seu próprio problema de otimização com o mesmo objetivo.

Para atingir o seu objetivo, cada jogador precisa achar a política ótima para seu problema de otimização, no caso, um conjunto de decisões (jogadas) que o conduzam a uma possível vitória. As decisões são tomadas em seqüência, em *estágios* sucessivos. No caso do jogador A, cada jogada ocorre após cada jogada do jogador B (de maneira similar para o problema de otimização do jogador B).

Os *estados* possíveis do sistema são todas as possíveis configurações de peças resultantes das jogadas feitas pelos jogadores A e B na tabela (inclusive a tabela vazia). Observe que os estados possíveis para cada jogador, em cada estágio, dependem de quem inicia o jogo. Se A inicia o jogo, no primeiro estágio, o estado inicial é o tabuleiro vazio. No segundo estágio para A, os estados possíveis consistem em todas as disposições possíveis de duas peças no tabuleiro, uma do jogador A e uma do jogador B. No terceiro estágio para A, os estados possíveis consistem em todas as disposições possíveis de quatro peças no tabuleiro, duas do jogador A e duas do jogador B. E assim, de maneira similar, podem ser descritos todos os estados possíveis nos estágios subsequentes.

Na Figura 6.2, exemplificamos alguns estados possíveis para um jogador A que inicia o jogo-da-velha. Como existem apenas nove posições na tabela do jogo da velha, cinco é o número máximo de estágios (de jogadas) para qualquer jogador da partida. Neste jogo, após a decisão do jogador A no estágio  $i$ , a menos que ele vença o jogo com esta jogada, ele não tem absoluta certeza de qual será o estado para o qual o sistema evoluirá no estágio  $i + 1$ , pois a jogada do jogador B independe de seu controle. No entanto, o jogador A pode sempre especificar qual seria a sua próxima decisão, caso o resultado da jogada de B resultasse em um determinado estado.

Apliquemos a recursão (6.1) na resolução do problema de otimização do jogador A, supondo que ele dá início ao jogo. Vamos admitir que a satisfação causada por uma vitória no jogo tenha sido quantificada com o valor de 1 unidade, o empate com o valor de 0 unidade e uma derrota com um valor de -1 unidade. Com isso, a recursão para o jogo-da-velha fica:



**Figura 6.2** Exemplos de estados possíveis para um jogador A que inicia o jogo-da-velha.

$$r_i(j) = \max_{d_i(j) \in D_i(j)} E[c(d_i(j), S_i) + r_{i+1}(e(d_i(j), S_i))] = \max_{d_i(j) \in D_i(j)} E[r_{i+1}(e(d_i(j), S_i))] \quad (6.3)$$

pois não há custos associados às decisões em cada estágio, observado um estado, ou seja,  $c(d_i(j), S_i) = 0$ . Os demais termos nessa recursão têm o seguinte significado no contexto deste jogo:

$r_i(j)$  é a satisfação máxima esperada para o jogador A do resultado final do jogo, para cada estágio  $i$  e estado  $j$

$e(d_i(j), S_i)$  é o estado para o qual o jogo evoluirá, governado pela variável aleatória  $S_i$ , dado que o jogador A fez a jogada  $d_i(j)$  no estágio  $i$  e o estado do sistema era  $j$

No último estágio 5, quando se tem a definição do resultado do jogo, os valores  $r_5(j)$  são iguais a +1, 0 e -1, dependendo se o jogador A ganha, empata ou perde o jogo, respectivamente. A partir desses valores de  $r_5(j)$ , calcula-se  $r_4(j)$  para todo estado  $j$ , depois, calcula-se  $r_3(j)$  para todo estado  $j$ , e assim sucessivamente, até se chegar ao valor de  $r_1(0)$ .

Para resumir a apresentação, consideraremos apenas os três primeiros estágios do problema. É suficiente considerar os três primeiros estágios deste problema, pois o resultado final está inequivocamente definido após o segundo lance do jogador B, admitindo-se que o jogador A não comete erros grosseiros. Consideremos os estados possíveis observáveis do jogador A no estágio 3. Todas as disposições possíveis das peças do jogador A (e apenas deste jogador) no estágio 3 estão apresentadas na Figura 6.3. As demais configurações possíveis são equivalentes, por simetria, a uma das oito apresentadas.

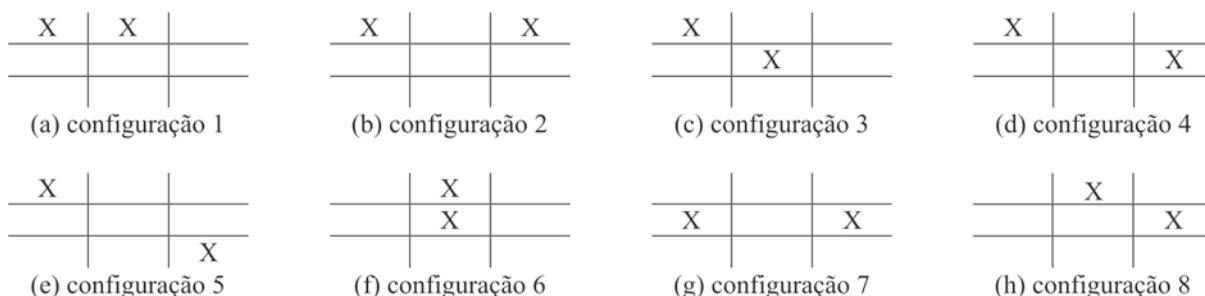
As possíveis disposições das peças de B, para a configuração 1 das peças de A da Figura 6.3, estão indicadas na Figura 6.4. Para fins de identificação dos estados, utilizamos três números separados por “.”, em que o primeiro número é o estágio, o segundo identifica a configuração das peças de A e o terceiro enumera os estados possíveis com aquela configuração. Por exemplo, 3.1.5 é o quinto estado possível da configuração 1 das peças do jogador A no estágio 3.

Observe que estados em que o jogador B não posiciona nenhuma peça na célula (1,3) (linha 1, coluna 3) da tabela são perdedores de imediato, e não estão sendo considerados por se configurarem como erros grosseiros do jogador B no estágio 2. De maneira similar, é possível identificar os estados possíveis para as demais configurações das peças de A. Elas estão indicadas nas Figuras 6.5, 6.6, 6.7, 6.8, 6.9, 6.10 e 6.11.

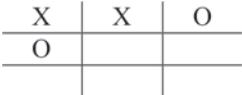
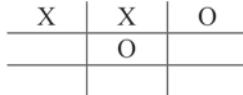
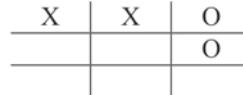
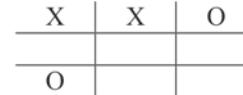
A recursão (6.3) no estágio 3, para um determinado estado  $j$  observado, fica:

$$r_3(j) = \max_{d_3(j) \in D_3(j)} E[r_4(e(d_3(j), S_3))]$$

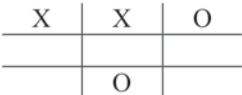
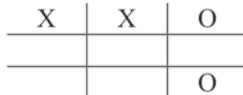
em que  $r_4(e(d_3(j), S_3))$  é a satisfação máxima esperada para o jogador A do resultado final do jogo, no estágio 4, dado que o estado é  $e(d_3(j), S_3)$ .



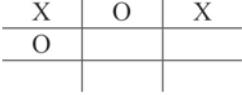
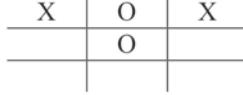
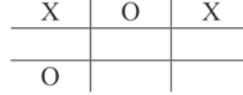
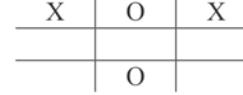
**Figura 6.3** Possíveis configurações das peças do jogador A após 2 jogadas.

			
(a) estado 3.1.1	(b) estado 3.1.2	(c) estado 3.1.3	(d) estado 3.1.4

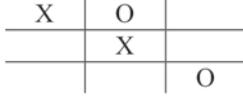
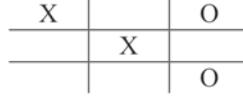
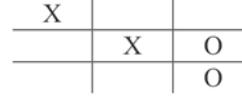
  

	
(e) estado 3.1.5	(f) estado 3.1.6

**Figura 6.4** Possíveis estados do sistema após duas jogadas e configuração 1 das peças de A.

			
(a) estado 3.2.1	(b) estado 3.2.2	(c) estado 3.2.3	(d) estado 3.2.4

**Figura 6.5** Possíveis estados do sistema após duas jogadas e configuração 2 das peças de A.

		
(a) estado 3.3.1	(b) estado 3.3.2	(c) estado 3.3.3

**Figura 6.6** Possíveis estados do sistema após duas jogadas e configuração 3 das peças de A.

Como já comentado anteriormente, o resultado final do jogo (vitória, empate, derrota) está inequivocamente definido após o segundo lance do jogador B. Isso porque, após o jogador A tomar a melhor decisão possível (posicionar a sua peça) no estágio 3, observado o estado do sistema, a configuração resultante define uma vitória, empate ou derrota de A (admitindo-se que o jogador A não comete erros grosseiros nos estágios 4 e 5). Em outras palavras,  $r_4(j)$ , a satisfação máxima esperada para o jogador A do resultado final do jogo, está determinada. Ela é conhecida para todo estado observável  $j$  no estágio 4 e toma os valores  $-1$ ,  $0$  e  $1$ , em caso de derrota, empate e vitória, respectivamente.

Analisemos, para ilustração, o estado  $j = 3.1.1$ . Na Figura 6.12, ilustramos todas as decisões viáveis possíveis para o jogador A, indicadas com um “?”.

Pode-se facilmente verificar que a configuração resultante da decisão 1 é ganhadora, pois o jogador A abre duas possibilidades para ganhar a partida no estágio 4 e o jogador B só tem a possibilidade de bloquear uma delas. Qualquer que seja a jogada de B no estágio 3, ela gera um es-

<table border="1"><tr><td>X</td><td>O</td><td>O</td></tr><tr><td></td><td></td><td>X</td></tr><tr><td colspan="3"></td></tr></table>	X	O	O			X				<table border="1"><tr><td>X</td><td>O</td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td colspan="3"></td></tr></table>	X	O		O		X				<table border="1"><tr><td>X</td><td>O</td><td></td></tr><tr><td></td><td>O</td><td>X</td></tr><tr><td colspan="3"></td></tr></table>	X	O			O	X				<table border="1"><tr><td>X</td><td>O</td><td></td></tr><tr><td></td><td>O</td><td>X</td></tr><tr><td colspan="3"></td></tr></table>	X	O			O	X				
X	O	O																																						
		X																																						
X	O																																							
O		X																																						
X	O																																							
	O	X																																						
X	O																																							
	O	X																																						
(a) estado 3.4.1	(b) estado 3.4.2	(c) estado 3.4.3	(d) estado 3.4.4																																					
<table border="1"><tr><td>X</td><td>O</td><td></td></tr><tr><td></td><td></td><td>X</td></tr><tr><td colspan="3"></td></tr></table>	X	O				X				<table border="1"><tr><td>X</td><td>O</td><td></td></tr><tr><td></td><td></td><td>X</td></tr><tr><td colspan="3"></td></tr></table>	X	O				X				<table border="1"><tr><td>X</td><td></td><td>O</td></tr><tr><td>O</td><td></td><td></td></tr><tr><td colspan="3"></td></tr></table>	X		O	O						<table border="1"><tr><td>X</td><td></td><td>O</td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td colspan="3"></td></tr></table>	X		O	O		X				
X	O																																							
		X																																						
X	O																																							
		X																																						
X		O																																						
O																																								
X		O																																						
O		X																																						
(e) estado 3.4.5	(f) estado 3.4.6	(g) estado 3.4.7	(h) estado 3.4.8																																					
<table border="1"><tr><td>X</td><td></td><td>O</td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td colspan="3"></td></tr></table>	X		O	O		X				<table border="1"><tr><td>X</td><td></td><td>O</td></tr><tr><td></td><td>O</td><td>X</td></tr><tr><td colspan="3"></td></tr></table>	X		O		O	X				<table border="1"><tr><td>X</td><td></td><td>O</td></tr><tr><td></td><td>O</td><td>X</td></tr><tr><td colspan="3"></td></tr></table>	X		O		O	X				<table border="1"><tr><td>X</td><td></td><td>O</td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td colspan="3"></td></tr></table>	X		O	O		X				
X		O																																						
O		X																																						
X		O																																						
	O	X																																						
X		O																																						
	O	X																																						
X		O																																						
O		X																																						
(i) estado 3.4.9	(j) estado 3.4.10	(k) estado 3.4.11	(l) estado 3.4.12																																					
<table border="1"><tr><td>X</td><td></td><td>O</td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td></td></tr></table>	X		O	O		X	O			<table border="1"><tr><td>X</td><td></td><td>O</td></tr><tr><td></td><td>O</td><td>X</td></tr><tr><td>O</td><td></td><td></td></tr></table>	X		O		O	X	O			<table border="1"><tr><td>X</td><td></td><td>O</td></tr><tr><td></td><td>O</td><td>X</td></tr><tr><td>O</td><td></td><td></td></tr></table>	X		O		O	X	O			<table border="1"><tr><td>X</td><td></td><td>O</td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td></td></tr></table>	X		O	O		X	O			
X		O																																						
O		X																																						
O																																								
X		O																																						
	O	X																																						
O																																								
X		O																																						
	O	X																																						
O																																								
X		O																																						
O		X																																						
O																																								
(m) estado 3.4.13	(n) estado 3.4.14	(o) estado 3.4.15	(p) estado 3.4.16																																					
<table border="1"><tr><td>X</td><td></td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td></td></tr></table>	X			O		X	O			<table border="1"><tr><td>X</td><td></td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td></td></tr></table>	X			O		X	O			<table border="1"><tr><td>X</td><td></td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td></td></tr></table>	X			O		X	O			<table border="1"><tr><td>X</td><td></td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td></td></tr></table>	X			O		X	O			
X																																								
O		X																																						
O																																								
X																																								
O		X																																						
O																																								
X																																								
O		X																																						
O																																								
X																																								
O		X																																						
O																																								
(q) estado 3.4.17	(r) estado 3.4.18	(s) estado 3.4.19	(t) estado 3.4.20																																					
<table border="1"><tr><td>X</td><td></td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td></td></tr></table>	X			O		X	O			<table border="1"><tr><td>X</td><td></td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td></td></tr></table>	X			O		X	O			<table border="1"><tr><td>X</td><td></td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td></td></tr></table>	X			O		X	O			<table border="1"><tr><td>X</td><td></td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td></td></tr></table>	X			O		X	O			
X																																								
O		X																																						
O																																								
X																																								
O		X																																						
O																																								
X																																								
O		X																																						
O																																								
X																																								
O		X																																						
O																																								
(u) estado 3.4.21																																								

**Figura 6.7** Possíveis estados do sistema após duas jogadas e configuração 4 das peças de A.

tado perdedor. Como o máximo retorno que se pode obter do jogo é uma vitória e ela é atingida com essa decisão, então:

$$r_3(3.1.1) = \max_{d_3(j) \in D_3(j)} E[r_4(e(d_3(3.1.1), S_3))] = E[r_4(e(d_3^*(3.1.1), S_3))] = 1, \text{ com } d_3^*(3.1.1) = \text{colocar peça na célula (2,2).}$$

De maneira similar, pode-se calcular  $r_3(j)$  para todos os demais estados observáveis  $j$  (e as decisões ótimas correspondentes). Se fizermos isso, chegamos à seguinte conclusão:

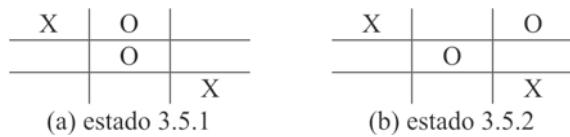
$$\begin{aligned} r_3(j) &= 1 && \text{para } j \in V^3, \text{ ou seja, vitória de A no (ou após o) estágio 3} \\ &= 0 && \text{para } j \in E^3, \text{ ou seja, empate de A no (ou após o) estágio 3} \\ &= -1 && \text{para } j \in D^3, \text{ ou seja, derrota de A no (ou após o) estágio 3} \end{aligned}$$

em que

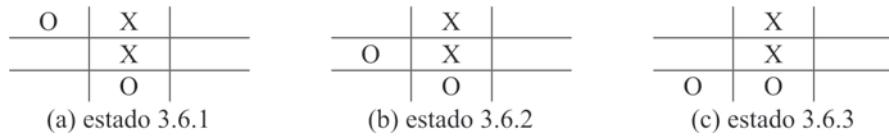
$V^3$  é o seguinte conjunto de estados: {3.1.1, 3.1.4, 3.2.1, 3.2.3, 3.2.4, 3.3.1, 3.3.3, 3.4.1, 3.4.2, 3.4.4, 3.4.5, 3.4.6, 3.4.9, 3.4.10, 3.4.11, 3.4.12, 3.4.13, 3.4.14, 3.4.16, 3.4.19, 3.5.2, 3.6.2, 3.8.1, 3.8.6, 3.8.9, 3.8.11, 3.8.12, 3.8.14, 3.8.16, 3.8.17}

$D^3$  é o seguinte conjunto de estados: {3.1.3, 3.1.6, 3.7.1, 3.7.2, 3.8.4, 3.8.20}

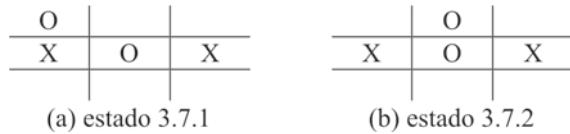
$E^3$  é o conjunto restante de estados



**Figura 6.8** Possíveis estados do sistema após duas jogadas e configuração 5 das peças de A.



**Figura 6.9** Possíveis estados do sistema após duas jogadas e configuração 6 das peças de A.



**Figura 6.10** Possíveis estados do sistema após duas jogadas e configuração 7 das peças de A.

Utilizando a recursão (6.3), podemos calcular a seguir  $r_s(j)$  para todo  $j$  do estágio 2, dado por:

$$r_2(j) = \max_{d_2(j) \in D_2(j)} E[r_3(e(d_2(j), S_2))] \quad (6.4)$$

uma vez que já são conhecidos todos os valores de  $r_3(j)$ . No estágio 2, as disposições possíveis das peças do jogador A no estágio 2 estão apresentadas na Figura 6.12. As demais configurações possíveis são equivalentes, por simetria, a uma das três apresentadas.

Os estados possíveis distintos para cada uma das configurações da peça de A estão indicados nas Figuras 6.14, 6.15 e 6.16. Os demais estados possíveis são equivalentes, por simetria, a algum desses estados apresentados. Ilustremos os cálculos de  $r_2(j)$  para o estado 2.1.1 da Figura 6.14. As decisões possíveis estão indicadas na Figura 6.17.

Cálculos para a decisão 1

Os possíveis estados após a decisão 1 são os apresentados na Figura 6.5. Admitamos, por exemplo, que todos esses quatro estados sejam eqüíprováveis (ou seja, o jogador B não antevê uma jogada adiante). Nessas condições, o valor esperado para esta decisão 1 é:

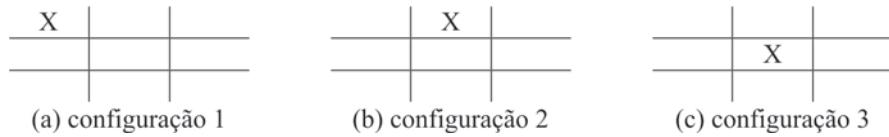
$$E[r_3(e(d_2(j), S_2))] = E[r_3(e(d_2(2.1.1), S_2))] = E[r_3(e(1, S_2))] = \frac{1}{4}(r_3(3.2.1)) + \frac{1}{4}(r_3(3.2.2)) + \frac{1}{4}(r_3(3.2.3)) + \frac{1}{4}(r_3(3.2.4)) = \frac{1}{4}(1 + 0 + 1 + 1) = 3/4$$

pois os estados 3.2.1, 3.2.3 e 3.2.4 são vitoriosos e o estado 3.2.2 leva a um empate.

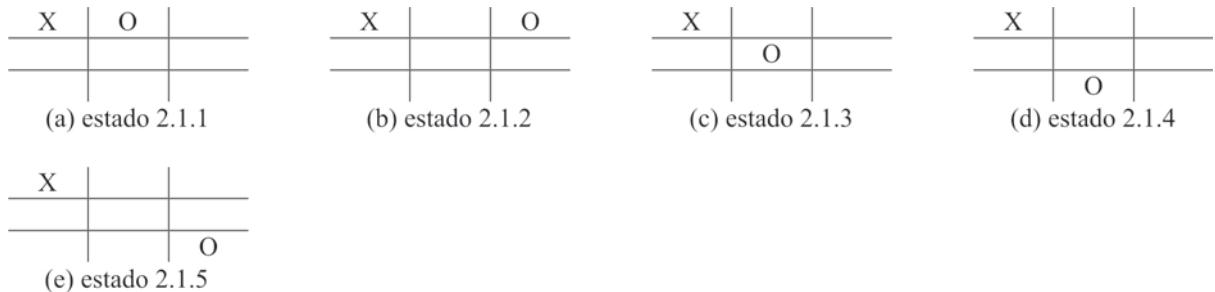
<table border="1"><tr><td>X</td><td>O</td><td>O</td></tr><tr><td></td><td></td><td>X</td></tr><tr><td></td><td></td><td></td></tr></table>	X	O	O			X				<table border="1"><tr><td>O</td><td>X</td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td></td><td></td><td></td></tr></table>	O	X		O		X				<table border="1"><tr><td>O</td><td>X</td><td>X</td></tr><tr><td></td><td>O</td><td>X</td></tr><tr><td></td><td></td><td></td></tr></table>	O	X	X		O	X				<table border="1"><tr><td>O</td><td>X</td><td></td></tr><tr><td></td><td>O</td><td>X</td></tr><tr><td></td><td></td><td></td></tr></table>	O	X			O	X			
X	O	O																																					
		X																																					
O	X																																						
O		X																																					
O	X	X																																					
	O	X																																					
O	X																																						
	O	X																																					
(a) estado 3.8.1	(b) estado 3.8.2	(c) estado 3.8.3	(d) estado 3.8.4																																				
<table border="1"><tr><td>O</td><td>X</td><td></td></tr><tr><td></td><td>O</td><td>X</td></tr><tr><td></td><td></td><td></td></tr></table>	O	X			O	X				<table border="1"><tr><td>O</td><td>X</td><td></td></tr><tr><td></td><td>O</td><td>X</td></tr><tr><td></td><td></td><td></td></tr></table>	O	X			O	X				<table border="1"><tr><td>O</td><td>X</td><td>O</td></tr><tr><td></td><td>O</td><td>X</td></tr><tr><td></td><td></td><td></td></tr></table>	O	X	O		O	X				<table border="1"><tr><td>X</td><td>O</td><td></td></tr><tr><td></td><td>O</td><td>X</td></tr><tr><td></td><td></td><td></td></tr></table>	X	O			O	X			
O	X																																						
	O	X																																					
O	X																																						
	O	X																																					
O	X	O																																					
	O	X																																					
X	O																																						
	O	X																																					
(e) estado 3.8.5	(f) estado 3.8.6	(g) estado 3.8.7	(h) estado 3.8.8																																				
<table border="1"><tr><td></td><td>X</td><td>O</td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td></td><td></td><td></td></tr></table>		X	O	O		X				<table border="1"><tr><td></td><td>X</td><td>O</td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td></td><td></td><td></td></tr></table>		X	O	O		X				<table border="1"><tr><td></td><td>X</td><td>O</td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td></td><td></td><td></td></tr></table>		X	O	O		X				<table border="1"><tr><td></td><td>X</td><td>O</td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td></td><td></td><td></td></tr></table>		X	O	O		X			
	X	O																																					
O		X																																					
	X	O																																					
O		X																																					
	X	O																																					
O		X																																					
	X	O																																					
O		X																																					
(i) estado 3.8.9	(j) estado 3.8.10	(k) estado 3.8.11	(l) estado 3.8.12																																				
<table border="1"><tr><td></td><td>X</td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td></td></tr></table>		X		O		X	O			<table border="1"><tr><td></td><td>X</td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td></td><td></td><td></td></tr></table>		X		O		X				<table border="1"><tr><td></td><td>X</td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td></td><td></td><td></td></tr></table>		X		O		X				<table border="1"><tr><td></td><td>X</td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td></td></tr></table>		X		O		X	O		
	X																																						
O		X																																					
O																																							
	X																																						
O		X																																					
	X																																						
O		X																																					
	X																																						
O		X																																					
O																																							
(m) estado 3.8.13	(n) estado 3.8.14	(o) estado 3.8.15	(p) estado 3.8.16																																				
<table border="1"><tr><td></td><td>X</td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td></td></tr></table>		X		O		X	O			<table border="1"><tr><td></td><td>X</td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td></td><td></td><td></td></tr></table>		X		O		X				<table border="1"><tr><td></td><td>X</td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td></td></tr></table>		X		O		X	O			<table border="1"><tr><td></td><td>X</td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td></td></tr></table>		X		O		X	O		
	X																																						
O		X																																					
O																																							
	X																																						
O		X																																					
	X																																						
O		X																																					
O																																							
	X																																						
O		X																																					
O																																							
(q) estado 3.8.17	(r) estado 3.8.18	(s) estado 3.8.19	(t) estado 3.8.20																																				
<table border="1"><tr><td></td><td>X</td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td>O</td></tr></table>		X		O		X	O		O	<table border="1"><tr><td></td><td>X</td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td></td><td></td><td>O</td></tr></table>		X		O		X			O	<table border="1"><tr><td></td><td>X</td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td>X</td></tr></table>		X		O		X	O		X	<table border="1"><tr><td></td><td>X</td><td></td></tr><tr><td>O</td><td></td><td>X</td></tr><tr><td>O</td><td></td><td>O</td></tr></table>		X		O		X	O		O
	X																																						
O		X																																					
O		O																																					
	X																																						
O		X																																					
		O																																					
	X																																						
O		X																																					
O		X																																					
	X																																						
O		X																																					
O		O																																					
(u) estado 3.8.21																																							

<table border="1"><tr><td>X</td><td>X</td><td>O</td></tr><tr><td>O</td><td>?</td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	X	X	O	O	?					<table border="1"><tr><td>X</td><td>X</td><td>O</td></tr><tr><td>O</td><td></td><td>?</td></tr><tr><td></td><td></td><td></td></tr></table>	X	X	O	O		?				<table border="1"><tr><td>X</td><td>X</td><td>O</td></tr><tr><td>O</td><td></td><td></td></tr><tr><td>?</td><td></td><td></td></tr></table>	X	X	O	O			?			<table border="1"><tr><td>X</td><td>X</td><td>O</td></tr><tr><td>O</td><td></td><td></td></tr><tr><td></td><td>?</td><td></td></tr></table>	X	X	O	O				?	
X	X	O																																					
O	?																																						
X	X	O																																					
O		?																																					
X	X	O																																					
O																																							
?																																							
X	X	O																																					
O																																							
	?																																						
(a) decisão 1	(b) decisão 2	(c) decisão 3	(d) decisão 4																																				
<table border="1"><tr><td>X</td><td>X</td><td>O</td></tr><tr><td>O</td><td></td><td></td></tr><tr><td>?</td><td></td><td></td></tr></table>	X	X	O	O			?																																
X	X	O																																					
O																																							
?																																							
(e) decisão 5																																							

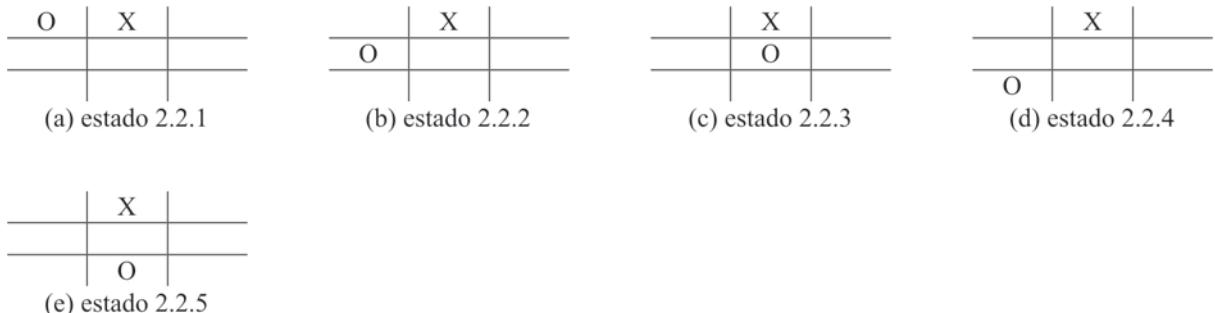
**Figura 6.12** Decisões possíveis de serem tomadas pelo jogador A no estágio 3, observado o estado 3.1.1.



**Figura 6.13** Possíveis disposições da peça do jogador A após uma jogada.



**Figura 6.14** Possíveis estados distintos do sistema após uma jogada e configuração uma da peça de A.



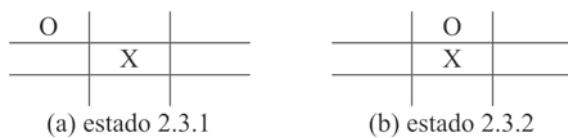
**Figura 6.15** Possíveis estados do sistema após uma jogada e configuração 2 da peça de A.

### Cálculos para a decisão 2

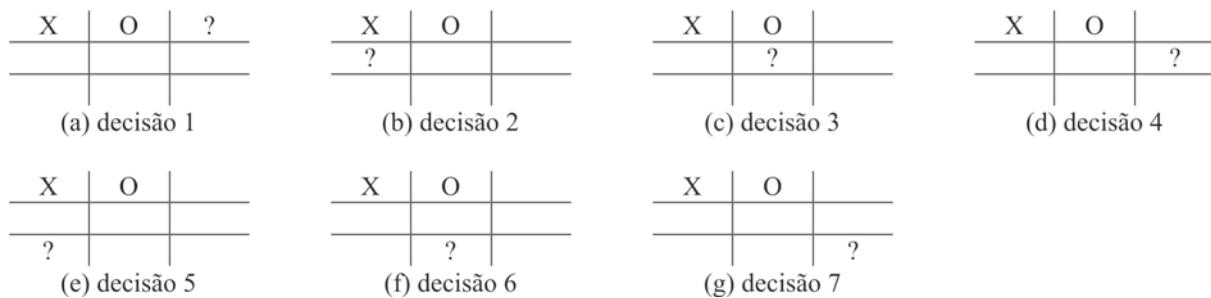
O único estado possível após a decisão 2 é o estado 3.1.1 na Figura 6.4. O valor esperado para esta decisão é  $E[r_3(e(2, S_2))]$  =  $1(r_3(3.1.1))$  = 1, pois o estado 3.1.1 é vitorioso.

### Cálculos para a decisão 3

O único estado possível após a decisão 3 é o estado 3.3.1 na Figura 6.6. O valor esperado para esta decisão é  $E[r_3(e(3, S_2))]$  =  $1(r_3(3.3.1))$  = 1, pois o estado 3.3.1 é vitorioso.



**Figura 6.16** Possíveis estados do sistema após uma jogada e configuração 3 da peça de A.



**Figura 6.17** Possíveis configurações do sistema após a decisão do jogador A no estágio 2.

Cálculos para a decisão 4

Os possíveis estados após a decisão 4 são os estados 3.4.1, 3.4.2, 3.4.3, 3.4.4, 3.4.5 e 3.4.6 na Figura 6.7. Admitamos novamente que todos esses seis estados sejam eqüíprobáveis (ou seja, o jogador B não antevê uma jogada adiante). O valor esperado para esta decisão é:

$$E[r_3(e(4,S_2))] = 1/6(r_3(3.4.1)) + 1/6(r_3(3.4.2)) + 1/6(r_3(3.4.3)) + 1/6(r_3(3.4.4)) + 1/6(r_3(3.4.5)) + 1/6(r_3(3.4.6)) = 1/6(1 + 1 + 0 + 1 + 1 + 1) = 5/6$$

pois os estados 3.4.1, 3.4.2, 3.4.4, 3.4.5 e 3.4.6 são vitoriosos e o estado 3.4.3 leva a um empate.

Cálculos para a decisão 5

O único estado possível após a decisão 5 é o estado 3.2.1 na Figura 6.5. O valor esperado para esta decisão é  $E[r_s(e(5, S_s))] = 1 (r_s(3.2.1)) = 1$ , pois o estado 3.2.1 é vitorioso.

Cálculos para a decisão 6

Os possíveis estados após a decisão 6 são os estados 3.4.2, 3.4.7, 3.4.12, 3.4.13, 3.4.14 e 3.4.15 na Figura 6.7. Admitamos novamente que todos esses seis estados sejam eqüiprováveis (ou seja, o jogador B não antevê uma jogada adiante). O valor esperado para esta decisão é:

$$E[v_3(e(6,S_2))] = 1/6(r_3(3.4.2)) + 1/6(r_3(3.4.7)) + 1/6(r_3(3.4.12)) + 1/6(r_3(3.4.13)) + 1/6(r_3(3.4.14)) + 1/6(r_3(3.4.15)) = 1/6(1 + 0 + 1 + 1 + 1 + 0) = 4/6 = 2/3$$

pois os estados 3.4.2, 3.4.12, 3.4.13 e 3.4.14 são vitoriosos e os demais estados levam a um empate.

### Cálculos para a decisão 7

O único estado possível após a decisão 7 é o estado 3.5.1 na Figura 6.9. O valor esperado para esta decisão é  $E[r_3(e(7, S_2))]$  = 1( $r_3(3.5.1)$ ) = 0, pois o estado 3.5.1 leva a um empate. Assim, para o estado 2.1.1, no estágio 2:

$$r_2(2.1.1) = \max \{3/4, 1, 1, 5/6, 1, 2/3, 0\} = 1$$

e a decisão ótima é 2, 3 ou 5. De maneira similar, pode-se calcular  $r_2(2.1.k)$ , para  $k = 2, 3, 4$  e 5;  $r_2(2.2.k)$ , para  $k = 1, 2, 3, 4$  e 5; e  $r_2(2.3.k)$ , para  $k = 1$  e 2. Após o cálculo de todos os  $r_2(j)$  para todos os  $j$ , prossegue-se utilizando a equação recursiva para se chegar a  $r_1(0)$ . A Tabela 6.8 apresenta as políticas ótimas obtidas dos estados possíveis no estágio 3, e a política ótima no estágio 2, estado 2.1.1. A tabela indica apenas uma decisão ótima em cada estado, mas cabe alertar que, para alguns estados, existem decisões ótimas alternativas.

## 6.3\* PROCESSOS MARKOVIANOS DE DECISÃO

Em muitas aplicações é relevante considerar o valor presente esperado dos retornos futuros gerados pelo processo, uma vez que  $Y$  unidades monetárias recebidas em um determinado período  $t$  têm um valor maior do que as mesmas  $Y$  unidades monetárias recebidas no período  $t + 1$ . Isso porque o poder aquisitivo de  $Y$  unidades monetárias recebidas no período  $t$  é geralmente maior do que o de  $Y$  unidades monetárias recebidas no período  $t + 1$ . Por exemplo, as  $Y$  unidades monetárias recebidas no período  $t$  poderiam ser investidas e renderiam um valor superior a  $Y$  unidades monetárias no período  $t + 1$ . O retorno recebido no período  $t + 1$  pode, então, ser corrigido de um fator  $\beta$  ( $0 < \beta \leq 1$ ) para refletir seu verdadeiro valor no período  $t$ . Levando-se em conta o valor presente na recursão (6.1), obtemos:

$$r_i(j) = \max_{d_i(j) \in D_i(j)} E[c(d_i(j), S_i) + \beta r_{i+1}(e(d_i(j), S_i))] \quad (6.5)$$

Com essa correção, se  $\beta < 1$  e, também, se os retornos esperados  $E[c(d_i(j), S_i)]$  em cada período e  $r_T(\cdot)$  são limitados por um valor, digamos,  $M$ , então o valor presente esperado  $r_i(j)$  é sempre limitado. Isso pode ser verificado, bastando observar que  $M \geq E[c(d_i(j), S_i)]$ ,

$$E[\beta r_{i+1}(e(d_i(j), S_i))] = \beta E[r_{i+1}(e(d_i(j), S_i))] \text{ e, portanto,}$$

$$r_i(j) \leq M + \max_{d_i(j) \in D_i(j)} \beta E[r_{i+1}(e(d_i(j), S_i))] \quad (6.6)$$

De maneira similar, a expressão (6.6) pode ser utilizada para limitar o valor de  $r_{i+1}(e(d_i(j), S_i))$ , obtendo-se:

$$\begin{aligned} r_i(j) &\leq M + \max_{d_i(j) \in D_i(j)} \beta E[M + \max_{d_{i+1}(j) \in D_{i+1}(e(d_i(j), S_i))} E(r_{i+2}(e(d_{i+1}(e(d_i(j), S_i)), S_{i+1})))] = M + \beta M \\ &\quad + \beta \max_{d_i(j) \in D_i(j)} E[\max_{d_{i+1}(j) \in D_{i+1}(e(d_i(j), S_i))} E(r_{i+2}(e(d_{i+1}(e(d_i(j), S_i)), S_{i+1})))] \end{aligned}$$

Aplicando o limitante (6.6) para os retornos nos demais períodos subsequentes, obtemos que:

$$\begin{aligned} r_i(j) &\leq M + \beta M + \beta^2 M + \beta^3 M + \dots + \beta^{T-i-2} M + \beta^{T-i-1} \max_{d_i(j) \in D_i(j)} E[\max_{d_{i+1}(j) \in D_{i+1}(e(d_i(j), S_i))} \\ &\quad E(\dots \max_{d_{T-1}(j) \in D_{T-1}(e(d_{T-2}(j), S_{T-1})))} E(r_T(e(\dots d_{T-1}(e(d_{T-2}(j), S_{T-1}), S_{T-1}))))]] \end{aligned}$$

A soma desses valores nunca ultrapassa  $M/(1-\beta)$ , que é o resultado da soma dos termos de uma progressão geométrica com termo inicial  $M$  e razão  $\beta < 1$ . Portanto, o retorno esperado presente é sempre limitado por  $M/(1-\beta)$ , mesmo que o número de estágios  $T$  seja grande.

Utilizamos a programação dinâmica em um problema de otimização com  $T$  estágios e, como já indicamos no Capítulo 5, se o estágio se refere a períodos no tempo,  $T$  é denominado horizonte. No problema do Exemplo 6.1, tínhamos  $T = 2$ , ou seja, o horizonte era de dois períodos.

**Tabela 6.8**  
**Políticas ótimas.**

Estágio	Estado	Decisão ótima	Retorno esperado	Estágio	Estado	Decisão ótima	Retorno esperado
3	3.1.1	(2,2)	1	3	3.1.2	(3,1)	0
3	3.1.3	(3,3)	-1	3	3.1.4	(2,2)	1
3	3.1.5	(3,3)	0	3	3.1.6	(2,3)	-1
3	3.2.1	(2,2)	1	3	3.2.2	(3,2)	0
3	3.2.3	(2,2)	1	3	3.2.4	(2,2)	1
3	3.3.1	(3,1)	1	3	3.3.2	(2,3)	0
3	3.3.3	(1,3)	1	3	3.4.1	(2,2)	1
3	3.4.2	(3,3)	1	3	3.4.3	(3,2)	0
3	3.4.4	(2,2)	1	3	3.4.5	(2,2)	1
3	3.4.6	(2,2)	1	3	3.4.7	(2,2)	0
3	3.4.8	(3,1)	0	3	3.4.9	(2,2)	1
3	3.4.10	(2,2)	1	3	3.4.11	(2,2)	1
3	3.4.12	(1,3)	1	3	3.4.13	(1,3)	1
3	3.4.14	(1,3)	1	3	3.4.15	(1,3)	0
3	3.4.16	(1,3)	1	3	3.4.17	(1,2)	0
3	3.4.18	(1,3)	0	3	3.4.19	(3,3)	1
3	3.4.20	(3,2)	0	3	3.4.17	(3,1)	0
3	3.5.1	(3,2)	0	3	3.5.2	(3,1)	1
3	3.6.1	(3,1)	0	3	3.6.2	(1,1)	1
3	3.6.3	(3,3)	0	3	3.7.1	(3,3)	-1
3	3.7.2	(3,2)	-1	3	3.8.1	(2,2)	1
3	3.8.2	(3,1)	0	3	3.8.3	(3,3)	0
3	3.8.4	(2,1)	-1	3	3.8.5	(2,2)	0
3	3.8.6	(2,2)	1	3	3.8.7	(2,2)	0
3	3.8.8	(3,1)	0	3	3.8.9	(2,2)	1
3	3.8.10	(2,2)	0	3	3.8.11	(2,2)	1
3	3.8.12	(1,3)	1	3	3.8.13	(1,1)	0
3	3.8.14	(1,3)	1	3	3.8.15	(2,2)	0
3	3.8.16	(1,3)	1	3	3.8.17	(1,3)	1
3	3.8.18	(1,1)	0	3	3.8.19	(3,3)	0
3	3.8.20	(3,2)	-1	3	3.8.21	(3,1)	0
3	2.2.1	(2,1)	1				

Existem casos em que esse horizonte  $T$  é muito grande, tanto que pode ser considerado infinito. Problemas de programação dinâmica estocástica com horizonte infinito são chamados de Processos Markovianos de Decisão.<sup>3</sup> O termo markoviano está sempre associado à idéia de dependência de dados/valores/decisões somente do período imediatamente anterior, e não de períodos mais longos.<sup>4</sup> No caso da programação dinâmica estocástica, temos que o estado do sistema  $e(d_i(j), S_i)$  no próximo período  $i + 1$  depende somente do estado atual  $j$  do período  $i$  e da decisão  $d_i(j)$  tomada no período  $i$ . Não existe nenhuma dependência de estados e decisões tomadas anteriormente ao presente período.

Procuramos na programação dinâmica uma *política ótima*, ou seja, uma seqüência de decisões coerentes que conduzem ao melhor retorno do problema. Para o caso de um processo homogêneo no tempo (o conjunto de estados possíveis e o conjunto de decisões possíveis são os mesmos em todos os períodos) e com horizonte infinito (e sob determinadas condições que são satisfeitas em uma grande gama de problemas práticos), essa política ótima existe e pode ser encontrada dentre as políticas chamadas estacionárias. Políticas estacionárias são aquelas em que, dado um estado  $j$ , a mesma decisão é tomada independentemente do período  $i$  em questão.

Sejam

$\mathbf{d}$  uma política estacionária

$\mathbf{d}^*$  uma política estacionária ótima

$R_{\mathbf{d}}(j)$  o retorno esperado de uma política estacionária  $\mathbf{d}$  de um processo markoviano de decisão, dado que estamos no estado  $j$

$R_{\mathbf{d}^*}(j)$  o retorno esperado da política estacionária ótima  $\mathbf{d}^*$  de um processo markoviano de decisão, dado que estamos no estado  $j$

Então, uma política estacionária  $\mathbf{d}^*$  é ótima (em um problema de maximização) se, para todos os estados  $j$ ,  $R_{\mathbf{d}^*}(j) \geq R_{\mathbf{d}}(j)$  para todas as políticas estacionárias possíveis  $\mathbf{d}$ . Para a determinação de uma política estacionária ótima, vamos recorrer novamente à equação recursiva (6.2) e reescrevê-la de uma forma mais conveniente.

### 6.3.1 Probabilidade de transição

Consideremos o problema do Exemplo 6.1. Nos cálculos efetuados, em cada um dos estágios do problema, o sistema estava em um dos seguintes estados possíveis  $\{0, 1, 2, 3, 4\}$ . Em um estado  $k$ , de um determinado estágio  $i$ , se uma decisão  $d_i(k)$  (pertencente a um conjunto  $D_i(k)$  de decisões) é tomada, o estado  $j$  no próximo estágio  $i + 1$  é atingido com uma determinada probabilidade  $p_{kj}(d_i(k))$ . Esta probabilidade  $p_{kj}(d_i(k))$  é denominada a *probabilidade de transição* do estado  $k$  para o estado  $j$ , dado que a decisão  $d_i(k)$  é tomada. Por exemplo, se estamos com o estoque de duas unidades no período 1 e decidimos encomendar uma unidade ( $d_1(2) = 1$ ), então a probabilidade de transição do estado 2 no período 1 para o estado 0 no período 2 é de 0,30, pois, para se chegar a este estado no período 2, bastaria ocorrer uma demanda de três ou quatro unidades no período 1, e a probabilidade de uma demanda de três ocorrer é de 0,25 e de uma demanda de quatro ocorrer é de 0,05.

Seja  $K$  o conjunto de todos os estados possíveis do processo markoviano de decisão. Em cada estágio, o processo está sempre em algum estado  $k$  pertencente a esse conjunto  $K$ . Sem perda de generalidade, digamos que  $K = \{1, 2, \dots, N\}$ , ou seja, temos  $N$  estados possíveis no processo, aos quais nos referimos como estados 1, 2, ...,  $N$ . Utilizando as probabilidades de transição  $p_{kj}(d_i(k))$  definidas anteriormente, podemos reescrever a recursão (6.5) como:

$$r_i(j) = \max_{d_i(j) \in D_i(j)} \{E[c(d_i(j), S_i)] + \beta \sum_{k=1}^N p_{kj}(d_i(j)) r_{i+1}(k)\}$$

<sup>3</sup> Existem autores que também consideram processos markovianos de decisão com horizonte finito; veja, por exemplo, Gheorghe (1990).

<sup>4</sup> Ver discussão na Seção 7.3 do Capítulo 7 de processos que não dependem da história passada do sistema (isto é, o futuro é independente da história passada), e por isso são chamados markovianos ou sem memória.

Observe que, ao admitirmos um processo homogêneo no tempo, a variável aleatória  $S_i$  e as probabilidades de transição são as mesmas para todo  $i$ . Assim, não é necessário o índice  $i$  para essa variável e para as probabilidades de transição em cada período  $i$ . Lembre-se de que  $r_i(j)$  é o retorno esperado no período  $i$ , dado que estamos no estado  $j$ , e  $R_d(j)$  é o retorno esperado de uma política estacionária  $\mathbf{d}$  de um processo markoviano de decisão, dado que estamos no estado  $j$ . Consideremos uma política estacionária  $\mathbf{d}$  qualquer. O retorno esperado desta política satisfaz:

$$r_i(j) = E[c(d(j), S)] + \beta \sum_{k=1}^N p_{jk}(d(j)) r_{i+1}(k)$$

À medida que o horizonte  $T$  cresce, os valores  $r_i(j)$  e  $r_{i+1}(j)$  devem se aproximar cada vez mais um do outro. No limite, quando  $T$  tende ao infinito, estes valores todos se igualam:  $r_i(j) = r_{i+1}(j) = R_d(j)$ . Portanto, para qualquer política estacionária  $\mathbf{d}$  de um processo markoviano de decisão, vale a equação:

$$R_d(j) = E[c(d(j), S)] + \beta \sum_{k=1}^N p_{jk}(d(j)) R_d(k) \text{ para } j = 1, 2, \dots, N \quad (6.7)$$

Uma política estacionária ótima  $\mathbf{d}^*$  é tal que  $R_{d^*}(j) \geq R_d(j)$  para todo estado  $j$  e para todas as políticas estacionárias possíveis  $\mathbf{d}$ . Podemos, então, determinar  $\mathbf{d}^*$  resolvendo o seguinte problema de otimização linear (ver Capítulo 2):<sup>5</sup>

$$\begin{aligned} (\text{PL}): \quad & \text{Minimizar } z = \sum_{j=1}^N R(j) \\ & R(j) \geq E[c(d(j), S)] + \beta \sum_{k=1}^N p_{jk}(d(j)) R(k) \end{aligned} \quad (6.8)$$

para todo estado  $j = 1, 2, \dots, N$  e cada possível  $d(j) \in D(j)$ .

Como minimizamos  $z$ , na solução ótima deste problema de programação linear, cada  $R(j)$  se iguala a  $E[c(d(j), S)] + \beta \sum_{k=1}^N p_{jk}(d(j)) R(k)$  para alguma decisão  $d^*(j) \in D(j)$ , ou seja,  $R(j)$  se iguala ao lado direito da desigualdade (6.8). Com este  $\mathbf{d}^*$ , podemos verificar que  $R_{d^*}(j) \geq R_d(j)$  para todo estado  $j$  e para todas as políticas estacionárias possíveis  $\mathbf{d}$ . Então esta política estacionária  $\mathbf{d}^*$  é ótima. Para cada um dos  $j$ ,  $j = 1, 2, \dots, N$ , a solução ótima  $R^*(j)$  do problema de programação linear nos dá  $R_{d^*}(j)$ .

### 6.3.2 Exemplo de manutenção de equipamentos

**Exemplo 6.3** (*Manutenção*) Um motorista possui um caminhão para transporte de cargas. No início de cada semana, o motorista inspeciona as condições do caminhão. Este pode estar em uma das seguintes condições: Bom (B), Médio (M), Ruim (U) ou Quebrado (Q). O caminhão se deteriora com o uso. Se sua condição na semana  $i$  é B, sua condição na semana  $i + 1$  é B, M, U ou Q, com probabilidades 0,80, 0,15, 0,04, 0,01, respectivamente. Se sua condição na semana  $i$  é M, sua condição na semana  $i + 1$  é B, M, U ou Q, com probabilidades 0,00, 0,80, 0,15, 0,05, respectivamente. Se sua condição na semana  $i$  é U, sua condição na semana  $i + 1$  é B, M, U ou Q, com probabilidades 0,00, 0,00, 0,80, 0,20, respectivamente. Se o caminhão está no estado Q, ele precisa ser consertado, pois o motorista não pode trabalhar sem o caminhão. O reparo dura duas semanas se o caminhão estiver quebrado, e o caminhão retorna na condição B. O motorista também pode optar por fazer uma manutenção preventiva quando verifica que o caminhão está na condição M ou U. No caso de decidir por uma manutenção preventiva, o reparo dura apenas uma

<sup>5</sup> Existem outros métodos para determinar políticas ótimas estacionárias de um processo markoviano de decisão; consulte, por exemplo, Winston (2004) e Gheorghe (1990).

semana, e o caminhão retorna na condição B. O motorista gostaria de saber qual política de manutenção ele deve adotar para obter o máximo rendimento possível de sua profissão. Considere que ele recebe \$1.500, \$1.100 e \$700 por semana trabalhada se seu caminhão estiver na condição B, M ou U, respectivamente; o custo de uma manutenção preventiva do caminhão é de \$500; o custo de um conserto é de \$2.000; e o banco cobra uma taxa de 3,5% de juros por semana pelo empréstimo que o motorista teve de fazer para comprar o caminhão.

No problema do Exemplo 6.3, os estados observáveis do caminhão em cada semana são B, M, U ou Q. Por conveniência, vamos denotar por C o estado em que o caminhão encontra-se em conserto. Se o estado observado do sistema é B, não há decisão a tomar. Se o estado observado do sistema é M ou U, o motorista pode tomar a decisão de continuar utilizando o caminhão ou fazer uma manutenção preventiva. Se o estado for Q, a única decisão a tomar é consertar o caminhão. Se o estado observado for C, também não há nada a fazer, pois o caminhão deverá ficar em conserto por mais uma semana.

Denotando-se as decisões “nada a fazer”, “manutenção preventiva” e “conserto” por 0, 1 e 2, respectivamente, os retornos associados a cada uma dessas decisões podem ser calculados pela equação (6.7). As políticas estacionárias possíveis para os estados observados (B,M,U,Q,C) são as seguintes: (0,0,0,2,0), (0,1,0,2,0), (0,0,1,2,0) e (0,1,1,2,0).

Usando a equação (6.7), temos, por exemplo, para a política estacionária (0,0,1,2,0):

- $R_{(0,0,1,2,0)}(B) = E[1500] + \beta\{0,80R_{(0,0,1,2,0)}(B) + 0,15 R_{(0,0,1,2,0)}(M) + 0,04 R_{(0,0,1,2,0)}(U) + 0,01 R_{(0,0,1,2,0)}(Q) + 0,00 R_{(0,0,1,2,0)}(C)\} = 1500 + 0,035\{0,80 R_{(0,0,1,2,0)}(B) + 0,15 R_{(0,0,1,2,0)}(M) + 0,04 R_{(0,0,1,2,0)}(U) + 0,01 R_{(0,0,1,2,0)}(Q)\}$
- $R_{(0,0,1,2,0)}(M) = E[1100] + \beta\{0,00 R_{(0,0,1,2,0)}(B) + 0,80 R_{(0,0,1,2,0)}(M) + 0,15 R_{(0,0,1,2,0)}(U) + 0,05 R_{(0,0,1,2,0)}(Q) + 0,00 R_{(0,0,1,2,0)}(C)\} = 100 + 0,035\{0,80 R_{(0,0,1,2,0)}(M) + 0,15 R_{(0,0,1,2,0)}(U) + 0,05 R_{(0,0,1,2,0)}(Q)\}$
- $R_{(0,0,1,2,0)}(U) = E[-500] + \beta\{1,00 R_{(0,0,1,2,0)}(B) + 0,00 R_{(0,0,1,2,0)}(M) + 0,00 R_{(0,0,1,2,0)}(U) + 0,00 R_{(0,0,1,2,0)}(C)\} = -500 + 0,035 R_{(0,0,1,2,0)}(B)$
- $R_{(0,0,1,2,0)}(Q) = E[-2000] + \beta\{0,00 R_{(0,0,1,2,0)}(B) + 0,00 R_{(0,0,1,2,0)}(M) + 0,00 R_{(0,0,1,2,0)}(U) + 1,00 R_{(0,0,1,2,0)}(C)\} = -2000 + 0,035 R_{(0,0,1,2,0)}(C)$
- $R_{(0,0,1,2,0)}(C) = E[0] + \beta\{1,00 R_{(0,0,1,2,0)}(B) + 0,00 R_{(0,0,1,2,0)}(M) + 0,00 R_{(0,0,1,2,0)}(U) + 0,00 R_{(0,0,1,2,0)}(C)\} = 0,035 R_{(0,0,1,2,0)}(B)$

A solução desse sistema de equações resulta em:

$$\begin{aligned} R_{(0,0,1,2,0)}(B) &= 1.547,928 \\ R_{(0,0,1,2,0)}(M) &= 1.125,682 \\ R_{(0,0,1,2,0)}(U) &= -445,823 \\ R_{(0,0,1,2,0)}(Q) &= -1.998,104 \\ R_{(0,0,1,2,0)}(C) &= 54,177 \end{aligned}$$

Similarmente, podemos aplicar a equação (6.7) para as demais políticas (0,0,0,2,0),(0,1,0,2,0) e (0,1,1,2,0).

Para a determinação da política ótima, podemos resolver o seguinte modelo de otimização linear (PL), combinando as equações de todas as políticas (veja (6.8)):

Minimizar  $z = R(B) + R(M) + R(U) + R(Q) + R(C)$

$$R(B) \geq 1500 + 0,035\{0,80R(B) + 0,15R(M) + 0,04R(U) + 0,01R(Q)\}$$

$$R(M) \geq 1100 + 0,035\{0,80R(M) + 0,15R(U) + 0,05R(Q)\}$$

$$R(M) \geq -500 + 0,035\{1,00R(B)\}$$

$$R(U) \geq 700 + 0,035\{0,80R(U) + 0,20R(Q)\}$$

$$R(U) \geq -500 + 0,035\{1,00R(B)\}$$

$$R(Q) \geq -2000 + 0,035\{1,00R(C)\}$$

$$R(C) \geq 0,035\{1,00R(B)\}$$

A solução deste modelo (aplicando o método simplex do Capítulo 2) nos dá:

$$R(B) = 1.549,621, \text{ com } d^* = 0$$

$$R(M) = 1.131,902, \text{ com } d^* = 0$$

$$R(U) = 705,775, \text{ com } d^* = 0$$

$$R(Q) = -1.998,102, \text{ com } d^* = 2$$

$$R(C) = 54,237, \text{ com } d^* = 0$$

A política estacionária ótima é  $(0,0,0,2,0)$ , portanto, utilizar o caminhão até quebrar e, então, consertá-lo. ■

## 6.4 EXERCÍCIOS<sup>6</sup>

**Exercício 6.1** Resolva o problema do Exemplo 6.1 considerando os mesmos dados, exceto o preço de venda do produto, que agora é de \$300 por unidade.

**Exercício 6.2** Resolva o problema do Exemplo 6.1 admitindo que o dono do estabelecimento possa devolver ao produtor todo o excedente do produto sazonal após as duas semanas, mas recebendo apenas a metade do preço de compra do produto, ou seja, \$50 por unidade.

**Exercício 6.3** Qual é o número total de possíveis estados diferentes, em todos os estágios, em um jogo da velha para o problema de otimização do jogador A, dado que ele inicia a partida?

**Exercício 6.4** Calcule o valor de  $r_1(0)$  para o jogo-da-velha, admitindo que a jogada feita pelo jogador B, dentre todas as possíveis, exceto as imediatamente perdedoras, sejam todas eqüíprováveis. Apresente a tabela de políticas ótimas. Repita o cálculo admitindo que a jogada feita por B é sempre a melhor possível, dentre todas as opções possíveis, e apresente a tabela de políticas ótimas.

**Exercício 6.5** Um motorista possui um microônibus escolar para transporte de crianças. No início de cada dia, o motorista inspecciona as condições do microônibus. O veículo pode estar em uma das seguintes condições: Bom (B), Médio (M), Ruim (U) ou Quebrado (Q). O veículo se deteriora com o uso. Se sua condição no dia  $i$  é B, sua condição no dia  $i + 1$  é B, M, U ou Q, com probabilidades 0,990, 0,007, 0,002, 0,001, respectivamente. Se sua condição no dia  $i$  é M, sua condição no dia  $i + 1$  é B, M, U ou Q, com probabilidades 0,000, 0,990, 0,008, 0,002, respectivamente. Se sua condição na semana  $i$  é U, sua condição na semana  $i + 1$  é B, M, U ou Q, com probabilidades 0,000, 0,000, 0,800, 0,200, respectivamente. Caso o motorista não possa trabalhar com seu microônibus, ele precisa alugar um outro veículo para cumprir seu contrato de prestação de serviço de transporte escolar. O reparo dura dois dias se o veículo estiver quebrado, e o veículo

<sup>6</sup> Alguns dos exercícios propostos foram inspirados em exercícios em Bradley *et al.* (1977) e Winston (2004).

retorna na condição B ou M, com probabilidades 0,950 e 0,050, respectivamente. O motorista também pode optar por fazer uma manutenção preventiva quando verifica que o veículo está na condição M ou U. No caso de decidir por uma manutenção preventiva, o reparo dura apenas um dia, e o veículo retorna na condição B. O motorista gostaria de saber qual política de manutenção ele deve adotar para obter o máximo rendimento possível de sua profissão, dado que ele recebe \$150 por dia trabalhado com seu microônibus; o custo de uma manutenção preventiva do microônibus é de \$300; o custo de um conserto é de \$800; e o aluguel de um outro veículo é de \$300 por dia.

**Exercício 6.6** Uma empresa tem sua produção centralizada em São José dos Campos e possui quatro depósitos localizados em (1) São Paulo, (2) Rio de Janeiro, (3) Porto Alegre e (4) Belo Horizonte. A capacidade máxima de produção da empresa é de 12 mil peças por mês. A demanda mensal no depósito  $i$ ,  $i = 1, 2, 3, 4$ , neste mês, segue a seguinte distribuição de probabilidade:

$$\text{Prob. } (K = k) = (\lambda_i^k e^{-\lambda_i})/k!$$

ou seja, uma distribuição de Poisson com média  $\lambda_i$ .<sup>7</sup> Se a demanda no mês excede o estoque disponível, o depósito incorre em um custo de penalidade de  $\pi_i$  por mil itens em falta. O custo de transporte da fábrica ao depósito  $i$  é de  $t_i$  por mil itens transportados. O estoque atual no depósito  $i$  é de  $e_i$  mil unidades. Formule um modelo de programação dinâmica para determinar a quantidade a produzir e transportar a cada um dos depósitos neste mês, de forma a minimizar os custos esperados de transporte e de falta de produto.

**Exercício 6.7** Um dos sistemas de comunicação de um satélite consiste de três subsistemas eletrônicos ligados em série. Este sistema de comunicação falha se algum destes subsistemas falha. Para aumentar a confiabilidade do sistema, são colocados subsistemas redundantes em paralelo, passando o sistema a ter três grupos de subsistemas. Em cada grupo, apenas um dos subsistemas fica em operação e, caso ele falhe, um outro subsistema em paralelo a ele é imediatamente acionado. Devido às restrições do projeto, este sistema de comunicação pode ter um peso máximo de 12kg. Os pesos de cada um dos subsistemas, em cada grupo, e a probabilidade de falha de cada subsistema é dado na tabela a seguir.

Grupo	Peso (kg)	Probabilidade de falha de cada subsistema
1	1	0,02
2	2	0,01
3	2	0,0005

Utilize programação dinâmica para determinar quantos subsistemas, em cada grupo, o sistema deve ter para maximizar sua confiabilidade total.

**Exercício 6.8** Uma empresa fornece válvulas reguladoras de combustível para um propulsor de satélites. Devido aos altos custos do satélite e à precisão exigida, essas válvulas precisam ter qualificação espacial. A empresa possui apenas três máquinas capazes de produzir esse item com essa qualificação. As máquinas são diferentes, têm custos e desempenhos diferentes. Na tabela a se-

<sup>7</sup> Para mais detalhes desta distribuição, ver Seção 7.3 do Capítulo 7.

uir, apresentamos os custos de cada máquina e as chances de cada máquina produzir itens que satisfaçam os padrões mínimos espaciais exigidos.

Máquina	<i>Custo de preparação(\$)</i>	<i>Custo variável (\$/unidade)</i>	<i>Probabilidade de estar dentro das especificações espaciais</i>
1	200	40	0,50
2	600	80	0,80
3	1000	120	0,90

Todos os itens produzidos são testados para verificar se atendem às qualificações espaciais. Itens que não satisfazem às qualificações espaciais são descartados, pois não podem ser recuperados. Se a empresa produz mais do que quatro itens com qualificação espacial, os itens em excesso também são descartados, pois apenas quatro serão utilizados. Por outro lado, se a empresa entregar menos de quatro itens com qualificação espacial, há uma multa de \$400 por item não entregue. Utilize programação dinâmica para determinar quantos itens devem ser produzidos em cada uma das máquinas, de forma que o custo esperado da empresa seja minimizado. (Sugestão: considere cada máquina como um estágio e defina a variável de estado como o número de itens aceitáveis, ainda a serem produzidos.)

**Exercício 6.9** Um idoso tem \$150 mil de capital em sua conta bancária. Cada ano ele pode gastar qualquer volume de capital que tem disponível e investir o restante. O investimento que o idoso tem em mente é de alto risco, com 50% de chance de, ao final do ano, duplicar o valor aplicado e 50% de chance de, ao final do ano, reduzir o montante aplicado à metade do valor investido. As aplicações neste investimento são feitas anualmente, sempre em múltiplos de \$10 mil. Dado que ele acredita que ainda tem quatro anos de vida, qual deve ser sua estratégia de gastos e aplicações para que, ao final dos quatro anos, ele tenha o máximo valor esperado de gastos efetuados.

**Exercício 6.10** Uma avicultura cria frangos para abate. A avicultura precisa definir em que período deve abater seus frangos. Cada frango custa \$0,1/periódo para ser mantido e \$0,5 para ser abatido. Existem nove tamanhos padrão de frangos: 1, 2, 3, 4, 5, 6, 7, 8 e 9. Os preços de venda de cada tamanho de frango são dados na tabela a seguir:

<i>Tamanho do frango</i>	<i>Preço de venda (\$)</i>
1	0,1
2	0,2
3	0,3
4	0,4
5	1,0
6	2
7	3
8	4,5
9	4,9

De um período para outro, os frangos crescem de tamanho de acordo com uma certa probabilidade. Com os dados coletados durante vários anos, a empresa chegou às seguintes probabilidades de transição para o crescimento dos frangos (tabela a seguir):

<i>Tamanhos</i>	1	2	3	4	5	6	7	8	9
<b>1</b>	0	1	0	0	0	0	0	0	0
<b>2</b>	0	0	1	0	0	0	0	0	0
<b>3</b>	0	0	0	1	0	0	0	0	0
<b>4</b>	0	0	0	0	1	0	0	0	0
<b>5</b>	0	0	0	0	0,05	0,95	0	0	0
<b>6</b>	0	0	0	0	0	0,2	0,8	0	0
<b>7</b>	0	0	0	0	0	0	0,9	0,1	0
<b>8</b>	0	0	0	0	0	0	0	0,7	0,3
<b>9</b>	0	0	0	0	0	0	0	0	1

Da tabela, tem-se, por exemplo, que, com probabilidade 1, o frango de tamanho 0 passa para o tamanho 1 após 1 período; com probabilidade 0,8, o frango cresce do tamanho 6 para o tamanho 7; e com probabilidade 0,2, ele se mantém no mesmo tamanho 6 após 1 período. Admitindo que o fator de desconto por período seja de 0,98, deseja-se determinar qual é a estratégia de abate para a empresa que lhe dará o maior retorno financeiro.

Página deixada intencionalmente em branco

# 7

## Sistemas de filas e otimização

Filas de espera aparecem em diversos sistemas de produção, particularmente em sistemas de serviço, tais como bancos, supermercados, correios, postos de gasolina e sistemas de manufatura, por exemplo, produtos aguardando processamento em máquinas ou estações de trabalho. Também aparecem em sistemas de transporte, como em aviões esperando para aterrissar em aeroportos, navios esperando para descarregar em portos e sistemas computacionais, por exemplo, tarefas aguardando processamento em computadores ou pacotes de dados aguardando transmissão em redes computacionais. Em geral, os usuários (clientes, produtos, veículos, tarefas) desses sistemas se deslocam até os servidores (caixas de atendimento, máquinas, pistas de aterrissagem, computadores) para obter algum tipo de serviço (sistemas usuários-para-servidores).

Mas também podemos ter sistemas em que são os servidores que se deslocam para atender os usuários (sistemas servidores-para-usuários), tais como sistemas de atendimento emergencial (ambulâncias, bombeiros, viaturas de polícia), sistemas logísticos de coleta e/ou entrega (caminhões de coleta de lixo, entregadores de alimentos e de remédios em domicílio) e sistemas de manutenção em que as equipes de manutenção se deslocam (assistência técnica em domicílio). Nestes casos, as “filas” de usuários ficam dispersas (espacialmente distribuídas), ao invés de concentradas fisicamente na frente dos servidores, como nos caixas de bancos, supermercados e correios.

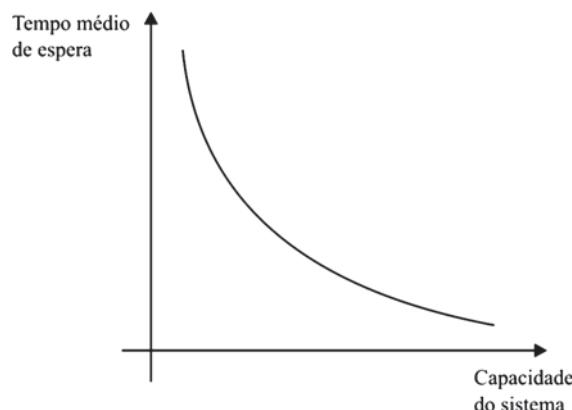
A teoria de filas (ou teoria de congestão), inicialmente motivada por aplicações em sistemas telefônicos, é um ramo da Pesquisa Operacional que estuda as relações entre as demandas em um sistema e os atrasos sofridos pelos usuários deste sistema. A formação de filas ocorre se a demanda excede a capacidade do sistema de fornecer o serviço em um certo período. A teoria de filas auxilia no projeto e na operação dos sistemas para encontrar um balanceamento adequado entre os custos de oferecer serviços no sistema (por exemplo, custos operacionais, custos de capacidade) e os custos dos atrasos sofridos pelos usuários do sistema. Dependendo do sistema, os custos associados aos atrasos podem ser muito altos, como, por exemplo, em um sistema de atendimento médico emergencial, ou nem tanto, como em um sistema de coleta de lixo.

A Figura 7.1 ilustra uma curva de *trade-off* (compromisso) entre a capacidade de um sistema (por exemplo, o número de caixas operando em um supermercado, o número de cabines abertas em uma praça de pedágio) e o tempo médio de espera dos usuários (os consumidores esperando em fila com suas compras, os veículos esperando em fila na frente das cabines). Note que o tempo de espera diminui com o aumento da capacidade, e vice-versa, e como os custos de capacidade em geral são altos, a pergunta é qual o equilíbrio ideal entre essas duas medidas para um dado sistema.

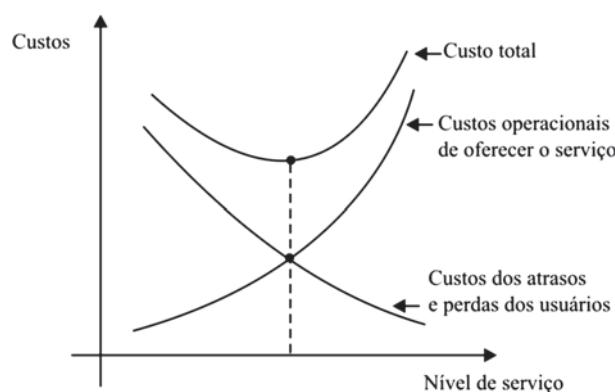
Os resultados da análise de sistemas de filas também podem ser usados em modelos de otimização, por exemplo, minimizando a soma dos custos de oferecer um nível de serviço no sistema e a soma dos custos dos atrasos ou perdas de usuários. A Figura 7.2 ilustra uma curva de custo total em função do nível de serviço do sistema (por exemplo, variando a capacidade do sistema). Note que, enquanto os custos operacionais de oferecer o serviço aumentam com o aumento do nível de serviço, os custos devido aos atrasos sofridos pelos usuários diminuem. Uma dificuldade da análise desses modelos de minimização de custo total é obter uma boa estimativa dos custos unitários de espera dos usuários.

Como é a configuração de um bom sistema de filas? Essa questão depende de certos pontos de vista. Por exemplo, do ponto de vista dos usuários, certas medidas de desempenho de interesse são: o tempo médio de serviço, o tempo médio de espera em fila até receber serviço, a probabilidade de o tempo de espera ser maior que um certo valor (digamos, 15 minutos), o número médio de usuários na fila, a probabilidade de o tamanho da fila ser maior que um certo valor (digamos, três usuários).

Do ponto de vista do gerente e dos operadores do sistema, outras medidas de interesse são: a utilização média do sistema, o número de servidores que ficam em média ocupados, o período



**Figura 7.1** Curva de *trade-off* entre capacidade do sistema e tempo de espera dos usuários.



**Figura 7.2** Curva de custo total em função do nível de serviço.

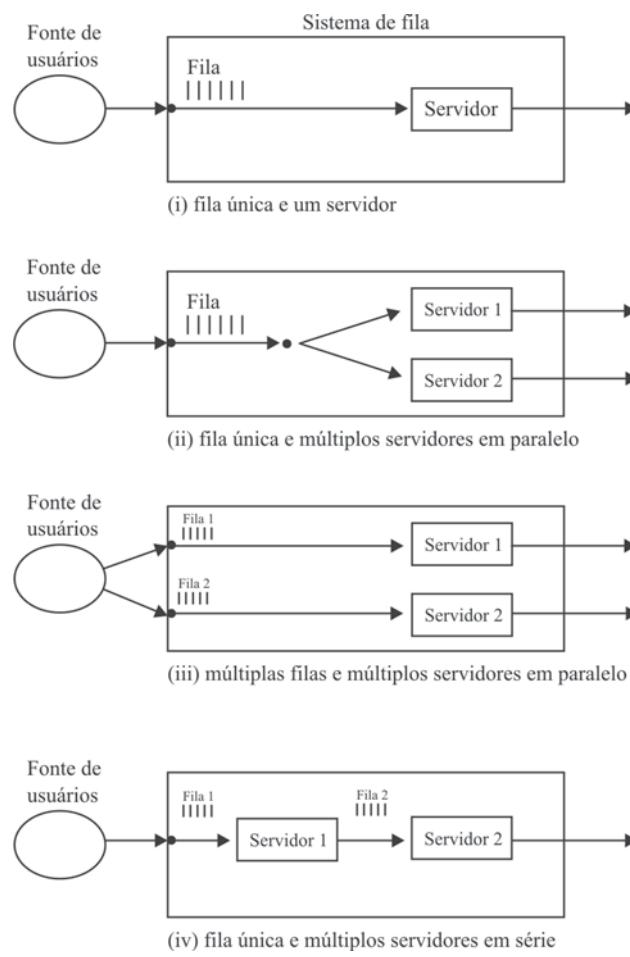
médio em que os servidores permanecem continuamente ocupados atendendo usuários sem interrupção, a probabilidade de esse período ser maior que um certo valor (digamos, 4 horas). O foco da teoria de filas é a avaliação de medidas de desempenho do sistema em função de uma dada configuração. A análise de modelos de decisão em sistemas de filas depende de boas estimativas dessas medidas.

## 7.1 DEFINIÇÃO E CLASSIFICAÇÃO DE UM SISTEMA DE FILAS

A Figura 7.3 ilustra diferentes tipos de sistemas de filas:

- (i) Fila única e um servidor.
- (ii) Fila única e múltiplos servidores em paralelo.
- (iii) Múltiplas filas e múltiplos servidores em paralelo.
- (iv) Fila única e múltiplos servidores em série.

Uma diferença é se o sistema tem um único estágio [tipos (i), (ii) e (iii)] ou múltiplos estágios [tipo (iv)]. Nos sistemas multiestágios, os usuários passam por mais de um dispositivo de serviço antes de saírem do sistema, por exemplo, no *drive-in* de certos restaurantes *fast-food* em que



**Figura 7.3** Diferentes tipos de sistemas de filas.

os usuários primeiro fazem seus pedidos e pagam, e depois recebem seus pedidos e saem do sistema. Nos sistemas monoestágios, podemos ter sistemas com fila única [tipos (i) e (ii)], por exemplo, em certas agências de bancos e correios, e sistemas com múltiplas filas em paralelo [tipo (iii)] encontrados em supermercados, balcões de atendimento dentro de restaurantes *fast-food* e praças de pedágio.

Nos sistemas do tipo (iii), ainda podemos ter as filas separadas por classe de usuários, por exemplo, filas de caixas normais e caixas rápidos (para consumidores com menos de dez itens) em supermercados, filas de passageiros de primeira classe e classe econômica de companhias aéreas em aeroportos.

Podemos também ter sistemas com múltiplas filas em um único servidor, por exemplo, no cruzamento de duas avenidas com filas de carros e filas de pedestres nas duas direções de cada avenida, todos aguardando o sinal verde no semáforo (servidor) para atravessar. Outros exemplos de múltiplas filas em um único servidor são sistemas com múltiplas classes de usuários com prioridades, como as filas de produtos com diferentes prioridades aguardando processamento na frente de uma estação de trabalho.

Sistemas de fila única do tipo (i) e (ii) contêm três elementos (ver Figura 7.3):

- Fonte de usuários.
- Fila.
- Dispositivo de serviço, contendo um ou mais servidores em paralelo.

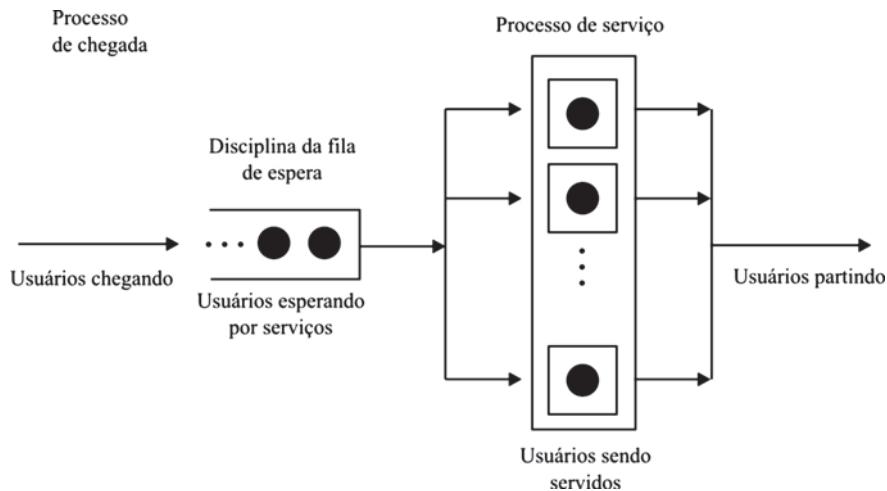
Considere, por exemplo, o sistema de fila do tipo (ii). Usuários são gerados a partir de uma fonte de usuários (por exemplo, todos os clientes de um banco), em geral suposta infinita (mesmo quando esta população é finita, mas relativamente grande). Quando um usuário chega a um dispositivo de serviço (agência do banco com vários caixas em paralelo), é imediatamente atendido por um dos servidores (caixa) ou espera em fila única até que um servidor esteja disponível. Ao completar o serviço em um servidor, o usuário sai do sistema e o servidor imediatamente inicia o atendimento a outro usuário da fila (escolhido segundo alguma disciplina de atendimento). Se a fila estiver vazia, o servidor fica desocupado aguardando a chegada de um outro usuário no sistema.

Dentro desta visão, uma rede de filas pode ser definida como um conjunto de sistemas de filas dos tipos (i) ou (ii) da Figura 7.3 interligados, em que alguns desses sistemas podem ser a fonte de usuários de outros (sistema de rede de filas). O sistema do tipo (iii) é uma rede de filas com conjuntos de sistemas do tipo (i) arranjados em paralelo, enquanto o sistema do tipo (iv) é uma rede de filas com conjuntos de sistemas do tipo (i) arranjados em série (por exemplo, uma linha de montagem com estações de trabalho em série).

Genericamente, podemos ter redes de filas combinando conjuntos de sistemas do tipo (i) e (ii) em série e em paralelo. Por exemplo, em um sistema de manufatura mais complexo, do tipo *job-shop*, em que diferentes produtos percorrem diferentes roteiros de fabricação ao longo de estações de trabalho da rede (veja, por exemplo, a Figura 7.20 na Seção 7.9). Outros exemplos de redes de filas incluem redes de transporte (por exemplo, os veículos ou cargas são os usuários e as rodovias ou ferrovias são os servidores), redes de telefonia e comunicação (por exemplo, *call centers*) e redes computacionais (por exemplo, rede *intranet*). A análise de modelos de sistemas de filas individuais serve de base para a análise de modelos de sistemas de rede de filas.

Para descrever um sistema de filas, precisamos de informação sobre três elementos (Figura 7.4):

- A entrada ou processo de chegada de usuários no sistema.
- A disciplina da fila (ordem em que os usuários em fila são atendidos).
- A saída ou processo de serviço (atendimento).



**Figura 7.4** Processo de chegada, disciplina da fila e processo de serviço.

No caso de um sistema de rede de filas, precisamos ainda de informação sobre como os sistemas de fila estão interligados e como os usuários são atribuídos aos diversos sistemas de filas (por exemplo, roteiros de fabricação dos produtos na rede de manufatura da Figura 7.20). Os modelos de filas e redes de filas são motivados por casos em que os processos de chegada ou serviço, ou ambos, são probabilísticos, resultando na formação de filas de espera e/ou perda de usuários.

#### *Processo de chegada*

O processo de chegada de usuários no sistema é descrito pelo intervalo de tempo entre chegadas sucessivas de usuários. Em geral, admitimos que não mais de um usuário pode chegar no mesmo instante; caso contrário, dizemos que pode ocorrer uma chegada em lote, por exemplo, a chegada de um casal em um restaurante (sistema com chegada em lote).

Em geral, admitimos que o processo de chegada não varia ao longo do tempo (processo estacionário); também admitimos que ele não é afetado pelo número de usuários presentes no sistema. Uma exceção é quando um usuário, ao chegar no sistema, desiste de entrar na fila caso ela seja muito longa, por exemplo, a chegada em um restaurante muito cheio (sistema com desistência na chegada). Outra exceção ocorre quando a fonte de usuários do sistema (Figura 7.3) é finita e suficientemente pequena (sistema com população finita). Por exemplo, se todos os usuários do sistema encontram-se dentro do sistema, ninguém mais pode chegar de fora, e claramente o processo de chegada depende do número de usuários presentes no sistema.

Podemos considerar todos os usuários pertencendo a uma única classe ou agrupados em múltiplas classes diferentes. Supõe-se que todos os usuários de uma mesma classe são estatisticamente idênticos, com intervalos entre chegadas modelados por uma distribuição de probabilidade.

#### *Processo de serviço*

O processo de serviço é descrito pelo tempo de serviço (atendimento) por usuário. Cada servidor não precisa ser um indivíduo, mas pode ser um grupo de pessoas ou máquinas realizando simultaneamente um serviço (por exemplo, uma equipe de manutenção). Em geral, admitimos que não mais de um usuário pode ser atendido por um servidor no mesmo instante; caso con-

trário, dizemos que pode ocorrer serviço em lote, por exemplo, um elevador que transporta várias pessoas ao mesmo tempo ou um semáforo de trânsito que permite a passagem de vários veículos (sistema com serviço em lote).

Em geral, admitimos que o processo de serviço não varia ao longo do tempo (processo estacionário) e que ele não é afetado pelo número de usuários presentes no sistema. Uma exceção é quando um servidor altera sua taxa de serviço dependendo do número de usuários esperando em fila, por exemplo, o atendente de um caixa de banco pode acelerar o atendimento se o número de clientes em fila for grande (sistema com aceleração do serviço), diferentemente de um caixa de atendimento automático.

Podemos ter o dispositivo de serviço contendo um ou mais servidores iguais em paralelo, como o do tipo (ii) da Figura 7.3. Neste caso, supõe-se que todos os servidores são estatisticamente idênticos, com tempos de serviço modelados por uma distribuição de probabilidade.

### *Disciplina da fila*

A disciplina da fila corresponde à ordem em que os usuários são selecionados da fila para atendimento. A mais comum é primeiro a chegar, primeiro a ser servido (*first come, first served – FCFS*), mas também podemos ter outras disciplinas como: último a chegar, primeiro a ser servido (*last come, first served – LCFS*) (por exemplo, em um elevador lotado, os que entraram por último em geral são os primeiros a sair), serviço em ordem aleatória (*service in random order – SIRO*) (por exemplo, em uma plataforma de embarque de uma estação de metrô), e seleção com base em alguma ordem de prioridade (por exemplo, priorizar as tarefas com menor tempo de processamento na programação da produção de uma máquina, chamada disciplina *SPT – shortest processing time*; veja Seção 3.6.3 do Capítulo 3).

No caso de sistema de fila com prioridade, podemos ter o caso com interrupção e o caso sem interrupção (também denominados prioridade de preterimento e não-preterimento). No sistema com interrupção, um usuário de maior prioridade é atendido assim que chega na fila, mesmo que um usuário de menor prioridade já esteja sendo atendido (obviamente, procura-se evitar esta regra quando o usuário é uma pessoa; exceções podem ocorrer em prontos-socorros com a chegada de pacientes em estado crítico). No sistema sem interrupção, o atendimento do usuário de menor prioridade não pode ser interrompido até ser completado.

A fila de espera pode ter tamanho (capacidade) limitado ou ilimitado, geralmente determinado pelo espaço físico disponível (por exemplo, o espaço para estoque em processo de uma máquina) (sistema com capacidade limitada). Uma vez alcançada essa capacidade, a entrada de novos usuários na fila é bloqueada. Em geral, admitimos que a capacidade é ilimitada, mesmo quando ela é limitada mas este limitante é relativamente grande. Podemos ter sistemas com servidores em paralelo com fila única, como o do tipo (ii) da Figura 7.3 (por exemplo, na maioria dos bancos e correios) ou com filas paralelas, uma para cada servidor, como o do tipo (iii) da Figura 7.3 (por exemplo, nos caixas de supermercados e cabines de pedágio). Neste caso, os usuários esperando em fila podem eventualmente trocar de fila (sistema com troca de fila), esperando reduzir seus tempos de espera, ou simplesmente desistir de esperar e abandonar a fila, se esse tempo se tornar muito longo (sistema com abandono de fila).

### *Notação de Kendall-Lee*

Todas essas variações de sistemas de filas levam a uma variedade de modelos de filas, alguns deles muito difíceis de serem tratados analiticamente (nestes casos, podemos utilizar aproximações analíticas ou modelos experimentais de simulação discreta). Para simplificar a análise dos sistemas de filas, os mais simples foram classificados conforme a notação de Kendall-Lee. Esta notação considera sistemas de fila única com um ou mais servidores idênticos em paralelo, como os ti-

pos (i) e (ii) da Figura 7.3. Ela é composta de seis características (as três primeiras foram inicialmente definidas por D. Kendall em 1953, e as três últimas foram adicionadas por A. Lee em 1968):

$A/B/m/C/K/N$

A primeira característica ( $A$ ) refere-se à distribuição de probabilidade do processo de chegada (intervalos de tempo entre chegadas sucessivas), podendo ser determinística ( $A = D$ ) ou probabilística (considerando os intervalos de tempo independentes e identicamente distribuídos – processo de renovação). Neste caso, pode ser:  $A = M$  (distribuição exponencial – markoviano<sup>1</sup>, sem memória),  $A = E_p$  (distribuição de Erlang com parâmetro de forma  $p$ ),  $A = G$  (distribuição genérica) (alguns autores adotam  $GI$  – genérica independente) etc.

A segunda característica ( $B$ ) refere-se à distribuição de probabilidade do processo de serviço (tempos de serviço), podendo ser determinística ( $B = D$ ) ou probabilística (considerando os tempos independentes e identicamente distribuídos – processo de renovação). Neste caso, pode ser:  $B = M$  (distribuição exponencial – markoviano, sem memória),  $B = E_p$  (distribuição de Erlang com parâmetro de forma  $p$ ),  $B = G$  (distribuição genérica) etc.

A terceira característica ( $m$ ) indica o número de servidores em paralelo (por exemplo,  $m = 3$ ).

A quarta característica ( $C$ ) especifica a disciplina da fila, por exemplo,  $C = FCFS$ ,  $LCFS$ ,  $SIRO$  ou  $GD$  (disciplina genérica, isto é, qualquer tipo de disciplina). Se  $C = GD$ , esta característica pode ser omitida da notação. No caso de disciplina com prioridade, é preciso distinguir classes de usuários e estabelecer a prioridade de cada classe (por exemplo, classes de tarefas em função da urgência ou prazo de entrega, classes de clientes preferenciais etc.). Neste caso, cada classe  $i$  tem sua distribuição de probabilidade do processo de chegada  $A_i$  e sua distribuição de probabilidade do processo de serviço  $B_i$ . Os casos de prioridade com e sem interrupção são denotados por  $C = PRP$  (*preemptive priority*) e  $C = NPRP$  (*nonpreemptive priority*), respectivamente.

A quinta característica ( $K$ ) indica o número máximo de usuários no sistema (incluindo os usuários em serviço e em fila) (por exemplo,  $K = 10$ ), e a sexta característica ( $N$ ) indica o tamanho da população (fonte de usuários) (por exemplo,  $N = 100$ ). Podemos ter  $K = \infty$  (sistema de capacidade ilimitada) e  $N = \infty$  (sistema de população infinita); nestes casos, essas características podem ser omitidas da notação.

Por exemplo, a notação  $M/G/3/FCFS/10/100$  indica um sistema de filas com intervalos de tempo entre chegadas exponencialmente distribuídos, tempos de serviço com distribuição genérica, três servidores idênticos em paralelo, disciplina de fila primeiro a chegar, primeiro a ser servido, capacidade limitada a 10 (no máximo de três usuários em serviço mais sete usuários em fila) e fonte de usuários com população finita igual a 100. De modo semelhante, a notação  $M/G/3$  indica um sistema igual ao anterior, a menos da disciplina de fila ( $C = GD$ ), pela capacidade ilimitada ( $K = \infty$ ) e pela população infinita ( $N = \infty$ ).

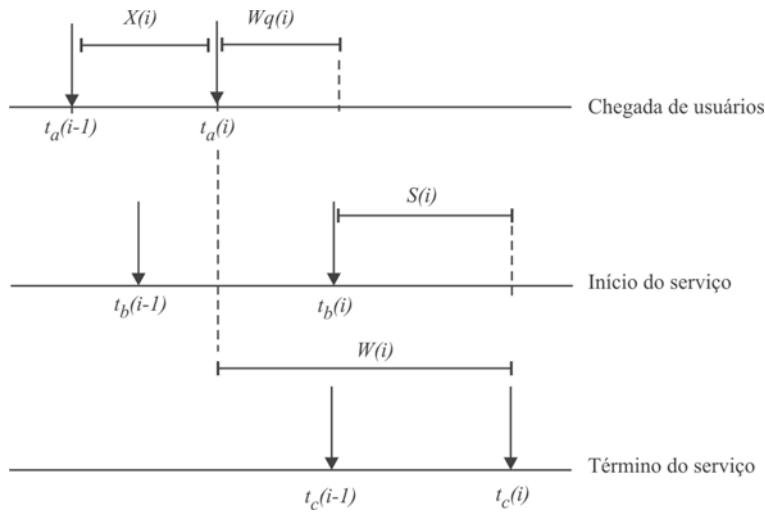
## 7.2 MEDIDAS DE DESEMPENHO DE UM SISTEMA DE FILAS

Considere um usuário qualquer em um sistema de fila do tipo  $G/G/1$ , digamos o  $i$ -ésimo usuário a chegar neste sistema a partir do instante  $t = 0$ . Definimos os seguintes instantes:

- $t_a(i)$  instante de chegada do  $i$ -ésimo usuário no sistema
- $t_b(i)$  instante de início de serviço do  $i$ -ésimo usuário no sistema
- $t_c(i)$  instante de término de serviço do  $i$ -ésimo usuário no sistema

A Figura 7.5 indica esses instantes ao longo do tempo (de modo semelhante,  $t_a(i-1)$ ,  $t_b(i-1)$  e  $t_c(i-1)$  referem-se ao  $(i-1)$ -ésimo usuário a chegar no sistema). Note que, como o usuário  $i$

<sup>1</sup> Em homenagem ao matemático russo Andrei A. Markov.



**Figura 7.5** Chegadas, inícios e términos de serviço dos usuários.

chega antes do término de serviço do usuário  $i - 1$  (isto é,  $t_a(i) < t_c(i-1)$ ), então o usuário  $i$  tem de esperar em fila antes de iniciar o serviço (supondo uma disciplina de fila sem interrupção).

Assim, definimos as seguintes medidas a partir dos instantes anteriores, também ilustradas na Figura 7.5:

$$X(i) = t_a(i) - t_a(i-1) \quad i\text{-ésimo intervalo entre chegadas dos usuários } i \text{ e } i-1$$

$$S(i) = t_c(i) - t_b(i) \quad \text{tempo de serviço do usuário } i$$

$$W_q(i) = t_b(i) - t_a(i) \quad \text{tempo de espera em fila do usuário } i$$

Note que o tempo de permanência no sistema do usuário  $i$  é:

$$W(i) = W_q(i) + S(i) = (t_b(i) - t_a(i)) + (t_c(i) - t_b(i)) = t_c(i) - t_a(i)$$

Em geral, admitimos que  $X(1), X(2), \dots, X(i), \dots$  e  $S(1), S(2), \dots, S(i), \dots$  são variáveis aleatórias contínuas independentes e identicamente distribuídas, isto é, que suas funções densidade de probabilidade sejam iguais:

$$f_{X(1)}(x) = f_{X(2)}(x) = \dots = f_{X(i)}(x) = \dots = f_X(x), \text{ com média } E(X)$$

$$f_{S(1)}(s) = f_{S(2)}(s) = \dots = f_{S(i)}(s) = \dots = f_S(s), \text{ com média } E(S)$$

Note que estamos supondo que as distribuições anteriores não se alteram com o tempo (isto é, são estacionárias) e não dependem do estado do sistema (por exemplo, do número de usuários no sistema). Logo, a taxa de chegada de usuários por unidade de tempo no sistema é:

$$\lambda = \frac{1}{E(X)}$$

e a taxa de serviço por unidade de tempo no servidor é:

$$\mu = \frac{1}{E(S)}$$

Por exemplo, se o intervalo médio entre chegadas de usuários no sistema é  $E(X) = 0,5$  hora, então chegam em média  $\lambda = 2$  usuários por hora. De modo semelhante, se o tempo médio de serviço por usuário é  $E(S) = 0,1$  hora, então o sistema é capaz de atender em média  $\mu = 10$  usuários por hora. Note que as estimativas de  $\lambda$  e  $\mu$  independem do instante de tempo e do número

de usuários no sistema. Note, também, que a taxa  $\mu$  é estimada supondo-se que o servidor está trabalhando continuamente, isto é,  $\mu$  corresponde à capacidade do servidor por unidade de tempo.

Se tivermos  $m$  servidores idênticos em paralelo no sistema de fila (por exemplo, sistema  $G/G/m$ ), então a taxa de serviço por unidade de tempo no sistema é  $m\mu$ .

O fator de utilização do sistema  $\rho$  (também chamado de razão de utilização) é definido pela razão entre a taxa de chegada de usuários no sistema e a taxa total disponível de serviço no sistema:

$$\rho = \frac{\lambda}{m\mu} = \frac{\lambda E(S)}{m}$$

Note que  $\rho$  é maior ou igual a 0, mas pode ser menor, igual ou maior que 1. Por exemplo, em um sistema  $G/G/1$  com taxa de chegada  $\lambda = 20$  usuários por hora e taxa de serviço  $\mu = 10$  usuários por hora (isto é,  $E(S) = 0,1$  hora por usuário por servidor), o fator de utilização do sistema é  $\rho = 2$ . Neste caso, a demanda média excede a capacidade média do sistema de oferecer serviço e, a menos que usuários sejam impedidos de entrar no sistema, a fila tende a crescer indefinidamente com o tempo.

O termo  $\lambda E(S)$  corresponde à carga média de trabalho que chega no sistema por unidade de tempo (também chamada intensidade de tráfego). Por exemplo,  $\lambda E(S) = 2$  horas por hora indica que em cada hora chega em média uma carga de trabalho de 2 horas (por servidor). Se o sistema tem dois servidores (por exemplo, sistema  $G/G/2$ ), então o fator de utilização do sistema é  $\rho = 1$ .

### 7.2.1 Análise em equilíbrio

A análise de muitos sistemas de filas baseia-se principalmente no comportamento do sistema em estado de equilíbrio, atingido após um tempo de operação suficientemente grande do sistema. Os modelos apresentados neste capítulo pressupõem que o sistema atinja o estado de equilíbrio ou regime permanente (*steady state*), isto é, baseiam-se no comportamento padrão do sistema em um longo período de tempo. Em outras palavras, os resultados dos modelos descrevem o comportamento médio de longo prazo de sistemas estáticos (invariante), ou seja, sistemas em que seja razoável admitir que parâmetros como a taxa média de chegada de usuários ou o tempo médio de serviço dos usuários não variam ao longo do horizonte de análise (processos de chegada e serviço estacionários). Conforme visto na seção 7.3, isso não quer dizer que essas taxas não podem variar com o estado do sistema, mas apenas que não podem variar ao longo do tempo para um mesmo estado.

Esse tipo de análise desconsidera o comportamento transitório do sistema (período de *warm-up*), que ocorre durante um período após o início de operação do sistema, quando o estado do sistema ainda depende bastante do estado inicial e do tempo decorrido desde então. Sob certas condições, é razoável esperar que, após um tempo de operação suficientemente grande, o sistema de filas atinja um equilíbrio. Sejam:

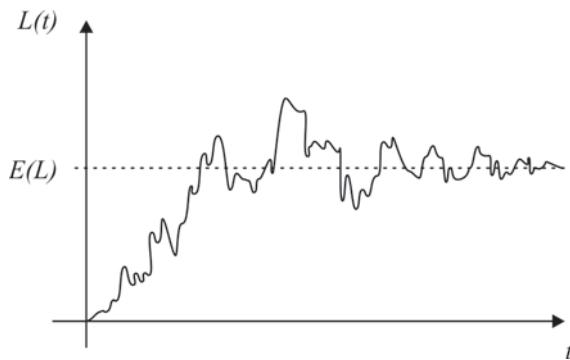
- |                          |  |
|--------------------------|--|
| $L_s(t)$                 | número de usuários em serviço no instante $t$                        |
| $L_q(t)$                 | número de usuários em fila no instante $t$                           |
| $L(t) = L_s(t) + L_q(t)$ | número de usuários (em fila e em serviço) no sistema no instante $t$ |

Para grandes valores de  $t$ , admite-se que  $L(t)$ ,  $L_s(t)$  e  $L_q(t)$  tenham funções massa de probabilidade independentes de  $t$  (estacionárias), digamos,  $p_L(l)$ ,  $p_{L_s}(l_s)$  e  $p_{L_q}(l_q)$ , com médias:

$$E(L) = \lim_{t \rightarrow \infty} E(L(t))$$

$$E(L_s) = \lim_{t \rightarrow \infty} E(L_s(t))$$

$$E(L_q) = \lim_{t \rightarrow \infty} E(L_q(t))$$



**Figura 7.6** Variação do número de usuários  $L(t)$  em torno do valor esperado  $E(L)$ .

Note que  $E(L)$  corresponde ao número médio de usuários no sistema em equilíbrio (de modo semelhante para  $E(L_s)$  e  $E(L_q)$ ). A Figura 7.6 ilustra uma possível variação de  $L(t)$  em torno de  $E(L)$  à medida que  $t$  cresce. Observe na figura que inicialmente o sistema está vazio e o número de usuários no sistema tende a ser próximo de zero. À medida que o tempo passa, esse número tende a oscilar e se estabilizar em torno do valor esperado  $E(L)$ .

De modo semelhante, é razoável supor que, para grandes valores de  $i$  (em vez de  $t$ ), o tempo de permanência no sistema e o tempo de espera na fila,  $W(i)$  e  $W_q(i)$ , têm funções densidade de probabilidade independentes de  $i$ , digamos,  $f_W(w)$  e  $f_{W_q}(w_q)$ , com médias:

$$E(W) = \lim_{i \rightarrow \infty} E(W(i))$$

$$E(W_q) = \lim_{i \rightarrow \infty} E(W_q(i))$$

(lembre-se que já estamos supondo que o tempo de serviço do usuário  $i$ ,  $S(i)$ , tem distribuição de probabilidade  $f_S(s)$  independente de  $i$ , com média  $E(S)$ .) Para muitos sistemas de filas, a condição  $\rho < 1$  é uma condição necessária para que o sistema atinja um equilíbrio, conforme visto na Seção 7.3. Uma exceção é o sistema  $D/D/1$ , que atinge um equilíbrio mesmo com  $\rho = 1$ .

### 7.2.2 Fórmula de Little

A fórmula de Little<sup>2</sup> relaciona as medidas  $E(L)$ ,  $E(L_s)$ ,  $E(L_q)$ ,  $E(W)$  e  $E(W_q)$ . Sejam:

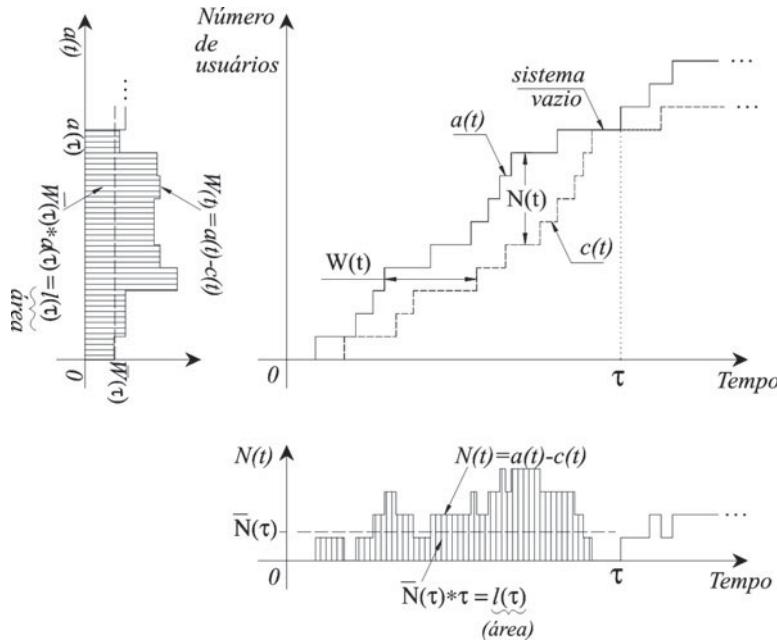
$a(t)$  número de entradas no sistema até o instante  $t$

$c(t)$  número de partidas (términos de serviço) no sistema até o instante  $t$

A Figura 7.7 ilustra esses processos de entrada  $a(t)$  e partida  $c(t)$  em um sistema de filas a partir do instante  $t = 0$  (sistema vazio). Se todos os usuários que chegam no sistema também entram no sistema (isto é, usuários que chegam não são perdidos ou impedidos de entrar no sistema), então  $a(t)$  também é o número de chegadas no sistema até o instante  $t$ . Note que, em cada instante  $t$ , o número de usuários no sistema é  $N(t) = a(t) - c(t)$  (veja a projeção horizontal da diferença entre  $a(t)$  e  $c(t)$  na Figura 7.7). De modo semelhante, o tempo de permanência do usuário  $i$  no sistema ( $W(i)$ ) está indicado na projeção vertical da diferença entre  $a(t)$  e  $c(t)$  na Figura 7.7.

Ao tomarmos um período de tempo  $\tau$  a partir de  $t = 0$ , a área  $l(\tau)$  entre  $a(t)$  e  $c(t)$  ao longo de  $0$  e  $\tau$  (isto é,  $l(\tau) = \int_0^\tau (a(t) - c(t)) dt$ ) é a mesma área das projeções horizontal e vertical (veja Figura 7.7). Logo, o número médio de usuários no sistema até  $\tau$  é:

<sup>2</sup> Mostrada por John D. Little; a demonstração está em Little (1961).



**Figura 7.7** Entradas e partidas acumuladas em um sistema de filas.

$$E(L(\tau)) = \frac{l(\tau)}{\tau} \Rightarrow l(\tau) = E(L(\tau))\tau$$

e o tempo médio de permanência no sistema até  $\tau$  é:

$$E(W(\tau)) = \frac{l(\tau)}{a(\tau)} \Rightarrow l(\tau) = E(W(\tau))a(\tau)$$

Igualando a área  $l(\tau)$  nas duas expressões anteriores, obtemos:

$$E(L(\tau)) = \frac{a(\tau)}{\tau} E(W(\tau)) = \bar{\lambda}(\tau) E(W(\tau))$$

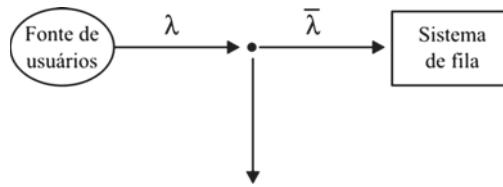
onde  $\bar{\lambda}(\tau)$  é a taxa de entrada de usuários no sistema até  $\tau$ . Conforme mencionado, após um período de tempo de operação suficientemente grande, é razoável esperar que o sistema de filas atinja um equilíbrio. Ou seja, no limite  $\tau \rightarrow \infty$ , obtemos:

$$E(L) = \bar{\lambda} E(W) \quad (7.1)$$

Esta é a chamada fórmula de Little, ou seja, o número médio de usuários no sistema  $E(L)$  é proporcional ao tempo médio de permanência dos usuários no sistema  $E(W)$ . A constante de proporcionalidade é dada pela taxa de entrada de usuários no sistema  $\bar{\lambda}$ .

Em sistemas em que nem todos os usuários que chegam também entram no sistema (por exemplo, casos em que usuários desistem ou são impedidos de entrar no sistema), a taxa de entrada  $\bar{\lambda}$  é menor que a taxa de chegada  $\lambda = \frac{1}{E(X)}$  definida anteriormente (ver Figura 7.8).

Um argumento intuitivo da fórmula de Little é que, em um sistema de filas em equilíbrio, o número médio de usuários ( $E(L)$ ) que um usuário aleatório encontra ao entrar no sistema deve ser igual ao número de usuários que ele observa ao sair do sistema, depois de ser servido. Mas o número médio de usuários que ele deixa para trás ao sair do sistema é simplesmente a taxa de entrada de usuários ( $\bar{\lambda}$ ) vezes o tempo médio que ele permaneceu (em fila e em serviço) no sistema ( $E(W)$ ).



**Figura 7.8** Taxa de chegada e taxa de entrada de usuários em um sistema de filas.

Repetindo a mesma análise anterior para:

a(t) número de entradas no sistema até o instante  $t$

b(t) número de inícios de serviço no sistema até o instante  $t$  (em vez de  $c(t)$ )

obtém-se:

$$E(L_q) = \bar{\lambda} E(W_q)$$

Além disso, repetindo a mesma análise para:

b(t) número de inícios de serviço no sistema até o instante  $t$  (em vez de  $a(t)$ )

c(t) número de término de serviço no sistema até o instante  $t$

obtém-se

$$E(L_s) = \bar{\lambda} E(S) = \frac{\bar{\lambda}}{\mu}$$

Note, nessa expressão, que, se  $\bar{\lambda} = \lambda$ , então a carga média de trabalho  $\lambda E(S) = \rho m$  também corresponde ao número médio de usuários em serviço  $E(L_s)$ , em outras palavras, ao número médio de servidores ocupados. A utilização média do sistema (proporção de servidores ocupados) é:

$$\frac{E(L_s)}{m} = \frac{\bar{\lambda} E(S)}{m}$$

que é igual ao fator de utilização  $\rho$  se  $\bar{\lambda} = \lambda$ . Se o sistema tem apenas um servidor ( $m = 1$ ), a utilização média também indica a proporção de tempo que o servidor fica ocupado. Isso pode ser mostrado com o seguinte argumento. Seja  $\tau$  um período de tempo suficientemente longo e  $P_0$  a probabilidade de o sistema estar vazio (isto é, o servidor estar desocupado). No limite  $\tau \rightarrow \infty$ , esperamos que ocorram  $\lambda\tau$  chegadas e  $\frac{\tau - \tau P_0}{E(S)}$  saídas (serviços) durante esse período, e que esses números sejam iguais:  $\lambda\tau = \frac{\tau - \tau P_0}{E(S)}$ . Segue-se que  $\lambda E(S) = 1 - P_0$ , ou seja, a utilização média do sistema corresponde à proporção de tempo que o servidor fica ocupado.

Convém salientar que a fórmula de Little é válida para qualquer processo de chegada (A), processo de serviço (B), número de servidores ( $m$ ) e disciplina de fila (C). Também é válida quando temos classes de usuários com diferentes prioridades; neste caso, a fórmula de Little pode ser aplicada para cada classe. Também é válida quando a taxa de entrada  $\bar{\lambda}$ , em vez de uma constante, depende do estado do sistema (por exemplo, do número de usuários presentes no sistema). Neste caso,  $\bar{\lambda}$  é uma média ponderada das taxas em que os usuários entram no sistema, conforme é visto na Seção 7.3.

Dados  $\bar{\lambda}$ ,  $\mu$  e apenas uma das medidas  $E(L)$ ,  $E(L_s)$ ,  $E(L_q)$ ,  $E(W)$  e  $E(W_q)$ , podemos facilmente encontrar as demais com a fórmula de Little (7.1) e as equações:

$$E(L_q) = \bar{\lambda} E(W_q) \quad (7.2)$$

$$E(L) = E(L_s) + E(L_q) = \frac{\bar{\lambda}}{\mu} + E(L_q) \quad (7.3)$$

$$E(W) = E(S) + E(W_q) = \frac{1}{\mu} + E(W_q) \quad (7.4)$$

**Exemplo 7.1** Em uma estação de trabalho, produtos chegam (e entram) em média a cada 60 minutos. A estação é composta de um grupo de pessoas, máquinas, ferramentas etc., que operam em conjunto para processar cada produto individualmente. Cada produto requer em média 50 minutos de processamento na estação. O tempo médio de permanência (*leadtime*) na estação é de 120 minutos. (i) Qual é a utilização média da estação? (ii) Qual é o número médio de produtos na estação (estoque em processo)?

Note que  $E(X) = 60$  minutos e  $E(S) = 50$  minutos, logo  $\bar{\lambda} = 1/60$  produto por minuto e  $\mu = 1/50$  produto por minuto. Considerando a estação como um único servidor ( $m = 1$ ), sua utilização média é:

$$\frac{E(L_s)}{m} = \frac{\bar{\lambda}}{m\mu} = \frac{50}{60} \approx 0,83$$

ou seja, a estação fica ocupada cerca de 83% do tempo de operação (note que  $\bar{\lambda} = \lambda$ , logo a utilização média é igual ao fator de utilização  $\rho = \frac{\lambda}{m\mu}$ ). Como  $E(W) = 120$  minutos, utilizando a fórmula de Little, obtemos o número médio de produtos:

$$E(L) = \bar{\lambda} E(W) = \frac{1}{60} 120 = 2 \text{ produtos}$$

### 7.3 MODELO DE FILAS FUNDAMENTAL

Até agora não fizemos nenhuma suposição sobre os tipos de distribuições de probabilidade dos intervalos de tempo  $X$  entre as chegadas de usuários e seus tempos de serviço  $S$  nos servidores, exceto que as distribuições são independentes e identicamente distribuídas, isto é,  $X$  tem distribuição  $f_X(x)$  com média  $E(X)$ , e  $S$  tem distribuição  $f_S(s)$  com média  $E(S)$ . Nesta seção, apresentamos o modelo de nascimento e morte, fundamental para a teoria de filas, que supõe que essas distribuições sejam exponenciais. Antes disso, discutimos a relação das distribuições exponencial e Poisson.

#### 7.3.1 A relação das distribuições exponencial e Poisson

Em muitos sistemas de filas, a chegada de usuários ocorre de maneira totalmente aleatória, ou seja, a chegada de um usuário não é influenciada pelo instante atual ou pelo tempo decorrido desde a última chegada ou o último término de serviço. Nestes casos, esses intervalos de tempo  $X$  totalmente aleatórios são descritos pela distribuição exponencial:

$$f_X(x) = \lambda e^{-\lambda x}, \quad x > 0$$

que tem média  $E(X) = \frac{1}{\lambda}$  e variância  $V(X) = \frac{1}{\lambda^2}$ . A Figura 7.9 mostra gráficos da distribuição exponencial (função densidade de probabilidade  $f_X(x)$  e distribuição acumulada  $F_X(x) = P(x \leq x) = \int_0^x f_X(x)dx = \int_0^x \lambda e^{-\lambda x} dx = 1 - e^{-\lambda x}$ ). O fato de ela ser completamente aleatória é ilustrado pelo seguinte exemplo: considere que ônibus passam em um ponto de ônibus com

intervalos de tempo  $X$  com distribuição exponencial. Em um instante qualquer  $t$ , digamos, às 14h05, a probabilidade de o próximo ônibus levar pelo menos  $\tau = 10$  minutos para passar no ponto (isto é, passar depois das 14h15), independe do tempo decorrido desde a passagem do último ônibus, digamos, há  $\tau_0 = 5$  minutos (às 14h). Ou seja,

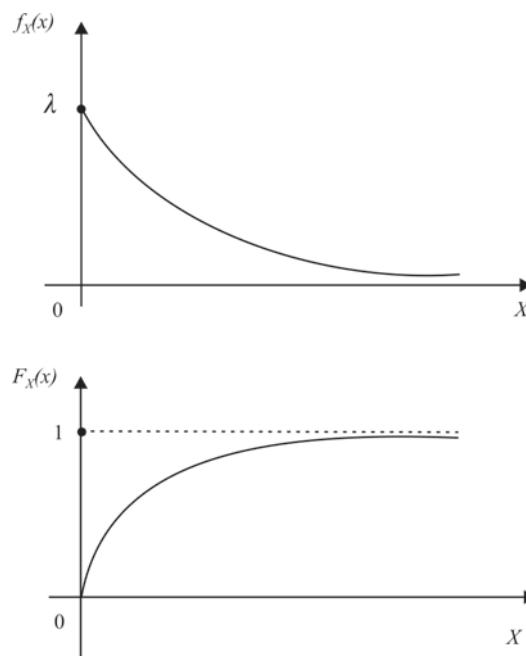
$$P(X > \tau_0 + \tau | X > \tau_0) = P(X > 15 | X > 5) = P(X > 10)$$

Isso decorre da propriedade de falta de memória da distribuição exponencial. Lembrando que  $P(X > x) = 1 - P(X \leq x) = 1 - F(x) = e^{-\lambda x}$ , temos que:

$$P(X > \tau_0 + \tau | X > \tau_0) = \frac{P(X > \tau_0 + \tau, X > \tau_0)}{P(X > \tau_0)} = \frac{P(X > \tau_0 + \tau)}{P(X > \tau_0)} = \frac{e^{-\lambda(\tau_0 + \tau)}}{e^{-\lambda\tau_0}} = e^{-\lambda\tau} = P(X > \tau)$$

Assim, a probabilidade de uma chegada nos próximos  $\tau$  minutos (a partir do instante  $t$ ) não depende dos últimos  $\tau_0$  minutos (antes do instante  $t$ ), ou seja, não importa quando ocorreu a última chegada. Em outras palavras, o futuro é independente da história passada. Por isso diz-se que as chegadas ocorrem de maneira totalmente aleatória. A distribuição exponencial é a única distribuição de probabilidade (de variável aleatória contínua) com a propriedade de falta de memória (no caso de variável aleatória discreta, a única distribuição com essa propriedade é a distribuição geométrica).<sup>3</sup>

A natureza completamente aleatória da distribuição exponencial também pode ser descrita da seguinte forma. Suponha que haja  $n$  chegadas durante um período de tempo de 0 a  $t$ . Então, pode-se mostrar que, se os intervalos de tempo entre chegadas forem exponencialmente distribuídos, os  $n$  instantes de chegada são independentes e uniformemente distribuídos ao longo do período de 0 a  $t$ .<sup>4</sup>



**Figura 7.9** Função densidade de probabilidade  $f_X(x)$  e distribuição acumulada  $F_X(x)$  da exponencial.

<sup>3</sup> Para uma demonstração disso, veja, por exemplo, Feller (1957).

<sup>4</sup> Ver, por exemplo, Wagner (1986).

Um resultado importante para a teoria de filas é que, se o intervalo de tempo entre chegadas de usuários é exponencial com média  $E(X) = 1/\lambda$ , então o número de chegadas de usuários  $N(t)$  durante um período de tempo de 0 a  $t$  tem distribuição de Poisson com média  $E(N(t)) = \lambda t$ . Isso é mostrado a seguir.

Sem perda de generalidade, admitimos que no instante  $t = 0$  o sistema está vazio, isto é,  $N(0) = 0$ . Como o número de chegadas de usuários no sistema durante o período de 0 a  $t$ ,  $N(t)$ , coincide com o número de usuários presentes no sistema no instante  $t$ , definimos  $P(N(t) = n)$  como a probabilidade de o sistema estar no estado  $n$  (isto é, com  $n$  usuários) no instante  $t$ . Dado que o intervalo de tempo entre chegadas  $X$  é exponencialmente distribuído, a probabilidade de não ocorrerem chegadas nos próximos  $\tau$  minutos é:

$$P(N(t+\tau) = n | N(t) = n) = P(X > \tau) = e^{-\lambda\tau}$$

Isso é válido para qualquer instante  $t$  e qualquer intervalo de tempo  $\tau$  por causa da falta de memória da distribuição exponencial. Para um intervalo de tempo  $\tau$  suficientemente pequeno, digamos,  $\Delta t$ , temos que (expansão em série de Taylor):

$$P(N(t+\Delta t) = n | N(t) = n) = e^{-\lambda\Delta t} = 1 - \lambda\Delta t + \frac{(-\lambda\Delta t)^2}{2!} + \frac{(-\lambda\Delta t)^3}{3!} \dots = 1 - \lambda\Delta t + o(\Delta t)$$

onde  $o(\Delta t)$  é tal que  $\lim_{\Delta t \rightarrow 0} o(\Delta t)/\Delta t = 0$ , isto é,  $o(\Delta t)$  tende a zero mais rapidamente do que  $\Delta t$  no limite  $\Delta t \rightarrow 0$ . Em outras palavras, para pequenos valores de  $\Delta t$ , o termo  $1 - \lambda\Delta t$  é grande, comparado à soma dos termos restantes  $o(\Delta t)$ . Por exemplo, suponha que  $\lambda = 4$  chegadas por hora e  $\Delta t = 0,01$  hora. Segue-se que  $1 - \lambda\Delta t = 1 - 4(0,01) = 0,96$ , enquanto  $e^{-\lambda\Delta t} = e^{-4(0,01)} \approx 0,96079$ , que são relativamente próximos. Admitindo-se que, durante um intervalo de tempo suficientemente pequeno, no máximo uma chegada pode ocorrer, segue-se que, se  $\Delta t \rightarrow 0$ , então:

$P(N(t+\Delta t) = n   N(t) = n) \approx 1 - \lambda\Delta t$	(não ocorre uma chegada)
$P(N(t+\Delta t) = n+1   N(t) = n) \approx \lambda\Delta t$	(ocorre uma chegada)
$P(N(t+\Delta t) = k   N(t) = n) \approx 0, \text{ para } k > n+1$	(ocorre mais de uma chegada)

Na primeira equação, a probabilidade (condicional) de o sistema estar no estado  $n$  no instante  $t + \Delta t$ , dado que o sistema está no estado  $n$  no instante  $t$  (isto é, não ocorre nenhuma chegada durante  $\Delta t$ ), é aproximadamente igual a  $1 - \lambda\Delta t$ . Na segunda, a probabilidade de o sistema estar no estado  $n + 1$  no instante  $t + \Delta t$ , dado que o sistema está no estado  $n$  no instante  $t$  (isto é, ocorre uma chegada durante  $\Delta t$ ), é aproximadamente igual a  $\lambda\Delta t$ . Note que a probabilidade de uma chegada durante  $\Delta t$  é proporcional ao tamanho deste intervalo. A constante de proporcionalidade é dada pela taxa de chegada  $\lambda$ . A última equação indica que a probabilidade de mais de uma chegada durante  $\Delta t$  é considerada aproximadamente nula.

A Figura 7.10 ilustra essas possíveis transições de estado a partir de um estado qualquer  $n$  ( $n > 0$ ) e um instante qualquer  $t$  do sistema, durante um intervalo de tempo suficientemente pequeno  $\Delta t$ .

Logo, a probabilidade de o sistema estar no estado  $n$  (isto é, com  $n$  usuários) no instante  $t + \Delta t$  é definida por:

$$\begin{aligned} P_n(t + \Delta t) &= P_n(t)(1 - \lambda\Delta t) + P_{n-1}(t)(\lambda\Delta t) & n = 1, 2, \dots \\ P_0(t + \Delta t) &= P_0(t)(1 - \lambda\Delta t) & n = 0 \end{aligned}$$

Na primeira equação ( $n > 0$ ), a probabilidade de o sistema estar no estado  $n$  no instante  $t + \Delta t$  é igual à soma de dois termos: (i) o produto da probabilidade de o sistema estar no estado  $n$  no instante  $t$  e a probabilidade de ocorrer 0 chegada durante  $\Delta t$ ; (ii) o produto da probabilidade de o sistema estar no estado  $n - 1$  no instante  $t$  e a probabilidade de ocorrer 1 chegada durante  $\Delta t$ . Na segunda equação ( $n = 0$ ), a probabilidade de ocorrer 0 chegada durante  $t + \Delta t$  é simples-

mente o produto da probabilidade de ocorrer 0 chegada durante o período de 0 a  $t$  e a probabilidade de ocorrer 0 chegada durante  $\Delta t$ . Os produtos se aplicam nessas expressões porque as chegadas nos períodos de 0 a  $t$  e  $\Delta t$  são eventos independentes, devido à falta de memória da distribuição exponencial.

As expressões anteriores podem ser reescritas como:

$$\frac{P_n(t + \Delta t) - P_n(t)}{\Delta t} = -\lambda P_n(t) + \lambda P_{n-1}(t) \quad n = 1, 2, \dots$$

$$\frac{P_0(t + \Delta t) - P_0(t)}{\Delta t} = -\lambda P_0(t) \quad n = 0$$

No limite de  $\Delta t \rightarrow 0$ , obtemos o seguinte sistema de equações diferenciais:

$$\lim_{\Delta t \rightarrow 0} \frac{P_n(t + \Delta t) - P_n(t)}{\Delta t} = \frac{dP_n(t)}{dt} = -\lambda P_n(t) + \lambda P_{n-1}(t) \quad n = 1, 2, \dots$$

$$\lim_{\Delta t \rightarrow 0} \frac{P_0(t + \Delta t) - P_0(t)}{\Delta t} = \frac{dP_0(t)}{dt} = -\lambda P_0(t) \quad n = 0$$

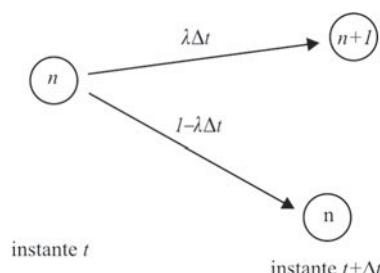
Apesar de envolver infinitas equações ( $n = 0, 1, 2, \dots$ ), este sistema pode ser resolvido admitindo-se que há  $n = 0$  usuários no instante  $t = 0$  (início), isto é,  $P_0(0) = 1$  e todos os demais  $P_n(0) = 0$ , o que resulta em  $P_0(t) = e^{-\lambda t}$ . Substituindo esta solução na primeira equação para  $n = 1$ , obtemos  $dP_1(t)/dt = -\lambda P_1(t) + \lambda e^{-\lambda t}$ , o que resulta em  $P_1(t) = (\lambda t)e^{-\lambda t}$ . E assim por diante, obtemos (por indução):<sup>5</sup>

$$P_n(t) = P(N(t) = n) = \frac{(\lambda t)^n e^{-\lambda t}}{n!} \quad n = 0, 1, 2, \dots$$

que corresponde à distribuição de Poisson, que tem média  $E(N(t)) = \lambda t$  e variância  $V(N(t)) = \lambda t$ , ou seja, igual à média. Portanto, se o intervalo de tempo  $X$  entre chegadas tem distribuição exponencial com média  $E(X) = 1/\lambda$  minuto, o número de chegadas  $N(t)$  de usuários durante um período de tempo de 0 a  $t$  tem distribuição de Poisson com média  $E(N(t)) = \lambda t$  usuários.

Note, na expressão anterior, que a probabilidade de ocorrer  $n = 0$  chegada até o instante  $t$  é  $P_0(t) = e^{-\lambda t}$ , que é exatamente igual a  $P(X > t)$ , a probabilidade da distribuição exponencial de que a primeira chegada no sistema ocorra depois do instante  $t$ .

Na prática, para verificarmos se é razoável admitir que  $X$  tem distribuição exponencial (ou, equivalentemente, que  $N(t)$  tem distribuição de Poisson), podemos coletar amostras e aplicar métodos estatísticos (testes de aderência), tais como o teste de qui-quadrado, o teste de Kolmogorov-Smirnov, o teste Lilliefors (para pequenas amostras), o teste de Anderson-Darling.<sup>6</sup>



**Figura 7.10** Possíveis transições de estado a partir do estado  $n$ .

5 Para mais detalhes desta solução, veja, por exemplo, Kleinrock (1975).

6 Ver, por exemplo, Meyer (1983), Costa Neto (1985), Conover (1999) e Law e Kelton (1991).

Convém observar que existem várias situações reais em que se pode admitir processos de chegada com distribuições exponenciais.

**Exemplo 7.2** Considere que o intervalo entre vendas de cerveja em um bar seja exponencialmente distribuído, com taxa de 20 cervejas vendidas por hora. (i) Qual a probabilidade de o bar vender 50 cervejas em 2 horas? (ii) Qual a média e o desvio-padrão do número de cervejas vendidas ao longo de 4 horas? (iii) Qual a probabilidade de o intervalo entre duas vendas consecutivas estar entre 1 e 3 minutos?

(i) Cada venda de cerveja pode ser vista como uma chegada de um usuário no sistema. Como os intervalos entre vendas seguem uma distribuição exponencial, com taxa  $\lambda = 20$  cervejas por hora, o número de cervejas vendidas durante um período  $t$  tem distribuição de Poisson, com média  $E(N(t)) = 20t$  cervejas. Logo, durante um período de  $t = 2$  horas, o número de cervejas vendidas tem distribuição de Poisson com média  $E(N(2)) = 40$  cervejas. A probabilidade de o bar vender  $n = 50$  cervejas é:

$$P_{50}(2) = P(N(2) = 50) = \frac{(40)^{50} e^{-40}}{50!} \approx 0,018$$

(ii) O número médio de cervejas vendidas durante um período de  $t = 4$  horas é  $\lambda t = 80$  cervejas. Lembrando que a distribuição de Poisson tem média igual à variância (isto é,  $V(N(t)) = E(N(t))$ ), como o desvio-padrão é a raiz quadrada da variância, segue-se que ele é igual a  $\sqrt{80} \approx 8,94$  cervejas.

(iii) Seja  $X$  o intervalo (em minutos) entre duas vendas consecutivas. Note que o intervalo médio é  $E(X) = 1/\lambda = 1/20$  hora = 3 minutos. Convertendo a taxa de vendas de horas para minutos, obtemos  $\lambda = 20$  cervejas por hora = 1/3 cerveja por minuto. Logo, a distribuição de  $X$  é  $f_X(x) = \lambda e^{-\lambda x} = \frac{1}{3} e^{-\frac{1}{3}x}$ , e a probabilidade de o intervalo entre duas vendas consecutivas estar entre 1 e 3 minutos é:

$$P(1 \leq X \leq 3) = \int_1^3 \frac{1}{3} e^{-\frac{1}{3}x} dx = e^{-\frac{1}{3}} - e^{-1} \approx 0,349 \quad (\text{ou seja, aproximadamente } 34,9\%)$$

### 7.3.2 Modelo de nascimento e morte

A distribuição de Poisson pode ser vista como um simples modelo de nascimento, isto é, descrevendo um sistema em que usuários apenas chegam (nascem) no sistema, por exemplo, um cartório que apenas emite certidões de nascimento de recém-nascidos. Modelos de nascimento mais genéricos consideram que a taxa de chegada dos usuários  $\lambda_n$  depende do estado do sistema, isto é, do número de usuários  $n$  presentes no sistema. Também podemos ter modelos de morte descrevendo um sistema em que os usuários já chegaram e ocorrem somente saídas (mortes) de usuários do sistema, por exemplo, um sistema de estoque de produtos durante o período em que não há reposição do estoque, apenas a retirada dos produtos.<sup>7</sup>

Modelos de nascimento e morte podem ser vistos como modelos de filas que consideram que usuários chegam (nascem) e são servidos (partem ou morrem) no sistema. Supõe-se que os intervalos de tempo entre as chegadas  $X$  e os tempos de serviço  $S$  são exponencialmente distribuídos, isto é, o número de usuários que chegam e o número de usuários que são servidos ao longo do tempo têm distribuições de Poisson. Obviamente, os modelos de nascimento e os modelos de morte são casos particulares dos modelos de nascimento e morte.

<sup>7</sup> Para mais detalhes e um exemplo de um modelo de morte, ver Taha (2003).

Por generalidade, consideramos que ambas as taxas de chegada e serviço dependem do estado do sistema, isto é, do número de usuários  $n$  presentes no sistema. Essas taxas são denotadas por  $\lambda_n$  e  $\mu_n$ , e indicam os números médios de usuários (que chegam ou são servidos) por unidade de tempo sempre que o sistema encontra-se com  $n$  usuários (note que supõe-se que essas taxas sejam constantes ao longo do tempo). Caso essas taxas não dependam de  $n$ , basta fazer  $\lambda_n = \lambda$  e  $\mu_n = \mu$ . Assim, admitimos que:

- Com  $n$  usuários no sistema, as chegadas (nascimentos) ocorrem de acordo com uma distribuição de Poisson com taxa de chegada  $\lambda_n$ .
- Com  $n$  usuários no sistema, os serviços (partidas ou mortes) ocorrem de acordo com uma distribuição de Poisson com taxa de serviço  $\mu_n$ .
- A disciplina de atendimento da fila é *FCFS*.

Assim como no modelo de nascimento da seção anterior, denotamos por  $N(t)$  o número de usuários presentes no sistema no instante  $t$ . No entanto, diferentemente daquele modelo,  $N(t)$  não corresponde mais ao número acumulado de chegadas de usuários no sistema até o instante  $t$ , porque agora há partidas do sistema.

Admitimos que, durante um intervalo de tempo  $(t, t + \Delta t)$  suficientemente pequeno, pode ocorrer no máximo uma chegada ou no máximo um término de serviço. Segue-se que, se  $\Delta t \rightarrow 0$ , a probabilidade de uma chegada durante o intervalo  $\Delta t$  quando o sistema encontra-se com  $n$  usuários no instante  $t$  é  $\lambda_n \Delta t$  (isto é, proporcional ao tamanho do intervalo), independentemente do tempo decorrido desde a última chegada antes de  $t$ . De modo semelhante, a probabilidade de um serviço terminar durante este intervalo  $(t, t + \Delta t)$  quando o sistema encontra-se com  $n$  usuários (com alguns ou todos em serviço) no instante  $t$  é  $\mu_n \Delta t$ , independentemente do tempo decorrido desde o início do serviço antes de  $t$ .<sup>8</sup> Como o modelo de nascimento e morte não depende da história passada do sistema, dizemos que ele é markoviano ou sem memória.

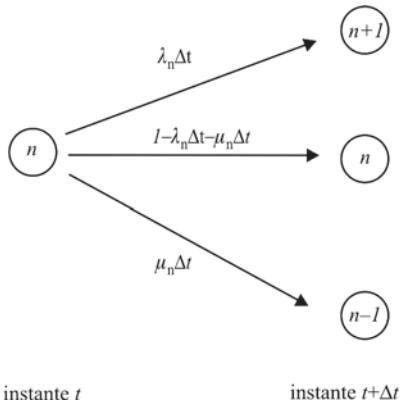
A Figura 7.11 ilustra as possíveis transições a partir de um estado  $n$  qualquer ( $n > 0$ ) e um instante  $t$  qualquer do sistema, durante um intervalo suficientemente pequeno  $\Delta t$ . Note que o sistema muda do estado  $n$  para o estado  $n + 1$  (nascimento) se ocorrer uma chegada durante  $\Delta t$ , ou seja, com probabilidade  $\lambda_n \Delta t$  (a rigor, com probabilidade  $\lambda_n \Delta t + o(\Delta t)$ , mas o termo  $o(\Delta t)$  foi omitido da figura porque tende a zero no limite  $\Delta t \rightarrow 0$ ). De modo semelhante, o sistema muda do estado  $n$  para o estado  $n - 1$  (morte) se ocorrer um término de serviço durante  $\Delta t$ , ou seja, com probabilidade  $\mu_n \Delta t$ , e permanece no estado  $n$  se não ocorrer uma chegada ou serviço durante  $\Delta t$ , ou seja, com probabilidade  $1 - \lambda_n \Delta t - \mu_n \Delta t$ .

As equações a seguir formalizam as transições da Figura 7.11:

$$\begin{aligned} P(N(t + \Delta t) = n + 1 | N(t) = n) &\approx \lambda_n \Delta t && \text{(ocorre uma chegada)} \\ P(N(t + \Delta t) = n - 1 | N(t) = n) &\approx \mu_n \Delta t && \text{(ocorre um serviço)} \\ P(N(t + \Delta t) = n | N(t) = n) &\approx 1 - \lambda_n \Delta t - \mu_n \Delta t && \text{(não ocorre chegada nem serviço)} \\ P(N(t + \Delta t) = k | N(t) = n) &\approx 0, \text{ para } |k - n| > 1 && \text{(ocorre mais de uma chegada/serviço)} \end{aligned}$$

A primeira equação indica que a probabilidade de o sistema estar no estado  $n + 1$  no instante  $t + \Delta t$ , dado que o sistema está no estado  $n$  no instante  $t$  (isto é, ocorre uma chegada durante  $\Delta t$ ), é aproximadamente igual a  $\lambda_n \Delta t$ . A segunda equação indica que a probabilidade de o sistema estar no estado  $n - 1$  no instante  $t + \Delta t$ , dado que o sistema está no estado  $n$  no instante  $t$  (isto é, ocorre um serviço durante  $\Delta t$ ), é aproximadamente igual a  $\mu_n \Delta t$ . A terceira equação indica que a probabilidade de o sistema estar no estado  $n$  no instante  $t + \Delta t$ , dado que o sistema está no estado  $n$  no instante  $t$  (isto é, não ocorre uma chegada nem um serviço durante  $\Delta t$ ), é apro-

<sup>8</sup> Isso decorre do fato de que o mínimo de diversas variáveis aleatórias independentes exponencialmente distribuídas tem uma distribuição exponencial; ver, por exemplo, Hillier e Lieberman (2004).



**Figura 7.11** Possíveis transições de estado a partir do estado  $n$ .

ximadamente igual a  $1 - \lambda_n \Delta t - \mu_n \Delta t$ . A última equação indica que a probabilidade de mais de uma chegada ou mais de um serviço durante  $\Delta t$  é considerada aproximadamente nula no modelo de nascimento e morte (também admitimos isso no modelo de nascimento da seção anterior).

Por simplicidade de notação, definimos  $P_n(t) = P(N(t) = n)$ . A Figura 7.12 ilustra o diagrama de transição de estados  $n = 0, 1, 2, \dots$ , no modelo de nascimento e morte (por simplicidade, os termos  $\Delta t$  foram omitidos). A probabilidade de o sistema estar no estado  $n$  (isto é, com  $n$  usuários) no instante  $t + \Delta t$  é definida por:

$$\begin{aligned} P_n(t + \Delta t) &= P_{n+1}(t)\mu_{n+1}\Delta t + P_n(t)[1 - (\lambda_n + \mu_n)\Delta t] + P_{n-1}(t)\lambda_{n-1}\Delta t & n = 1, 2, \dots \\ P_0(t + \Delta t) &= P_1(t)\mu_1\Delta t + P_0(t)(1 - \lambda_0\Delta t) & n = 0 \end{aligned}$$

Ou seja, a probabilidade de o sistema estar no estado  $n$  ( $n > 0$ ) no instante  $t + \Delta t$  é igual à soma de três termos: (i) a probabilidade de o sistema estar no estado  $n + 1$  no instante  $t$  e ocorrer um serviço durante  $\Delta t$ ; (ii) a probabilidade de o sistema estar no estado  $n$  no instante  $t$  e não ocorrer uma chegada ou um serviço durante  $\Delta t$ ; e (iii) a probabilidade de o sistema estar no estado  $n - 1$  no instante  $t$  e ocorrer uma chegada durante  $\Delta t$  (de modo semelhante ao estado  $n = 0$ ).

As equações anteriores podem ser reescritas como:

$$\begin{aligned} \frac{P_n(t + \Delta t) - P_n(t)}{\Delta t} &= -(\lambda_n + \mu_n)P_n(t) + \mu_{n+1}P_{n+1}(t) + \lambda_{n-1}P_{n-1}(t) & n = 1, 2, \dots \\ \frac{P_0(t + \Delta t) - P_0(t)}{\Delta t} &= -\lambda_0P_0(t) + \mu_1P_1(t) & n = 0 \end{aligned}$$

No limite  $\Delta t \rightarrow 0$ , obtemos:

$$\begin{aligned} \lim_{\Delta t \rightarrow 0} \frac{P_n(t + \Delta t) - P_n(t)}{\Delta t} &= \frac{dP_n(t)}{dt} = -(\lambda_n + \mu_n)P_n(t) + \mu_{n+1}P_{n+1}(t) + \lambda_{n-1}P_{n-1}(t) & n = 1, 2, \dots \\ \lim_{\Delta t \rightarrow 0} \frac{P_0(t + \Delta t) - P_0(t)}{\Delta t} &= \frac{dP_0(t)}{dt} = -\lambda_0P_0(t) + \mu_1P_1(t) & n = 0 \end{aligned}$$

Esse sistema de equações diferenciais descreve o sistema ao longo do tempo, ou seja, permite análises do comportamento transitório do sistema de filas.<sup>9</sup> Como em geral estamos interessados

<sup>9</sup> Para mais detalhes sobre como analisar o sistema em estado transitório, ver, por exemplo, Gross e Harris (1985) e Larson e Odoni (1981).

em analisar sistemas de filas em estado de equilíbrio, admitimos que, para valores de  $t$  suficientemente grandes, a probabilidade  $P_n(t)$  independe de  $t$ , ou seja:

$$\lim_{t \rightarrow \infty} P_n(t) = P_n \Rightarrow \frac{dP_n(t)}{dt} = 0$$

Note que  $P_n$  é a probabilidade de o sistema (em equilíbrio) estar no estado  $n$  (independentemente de  $t$ ), ou seja, a proporção de tempo que o sistema permanece no estado  $n$ . Por exemplo, se  $P_0 = 0,3$ , então no longo prazo o sistema fica vazio 30% do tempo. De modo semelhante,  $P_1 = 0,2$  implica que em 20% do tempo o sistema contém exatamente um usuário.

Como  $P_n$  não depende de  $t$ , sua taxa de variação em relação a  $t$  é nula. Com isso, o sistema de (infinitas) equações diferenciais anterior se reduz ao seguinte sistema de (infinitas) equações lineares nas variáveis  $P_n$ :

$$\begin{aligned} \lambda_0 P_0 &= \mu_1 P_1 & n = 0 \\ (\lambda_n + \mu_n) P_n &= \lambda_{n-1} P_{n-1} + \mu_{n+1} P_{n+1} & n = 1, 2, \dots \end{aligned}$$

Ou seja, em média, a taxa à qual o sistema sai do estado  $n$  (lado esquerdo das equações) deve ser igual à taxa à qual o sistema entra no estado  $n$  (lado direito das equações) (veja diagrama de transição de estados da Figura 7.12). Por exemplo, considere o estado  $n = 0$ . Note que o sistema sai do estado 0 apenas por meio de uma chegada. Dado que a taxa de chegada quando o sistema está no estado 0 é  $\lambda_0$ , e a proporção de tempo que o sistema permanece no estado 0 é  $P_0$ , segue-se que  $\lambda_0 P_0$  é a taxa à qual o sistema sai do estado 0 durante  $\Delta t \rightarrow 0$ . Por outro lado, o sistema entra no estado 0 apenas por meio de uma saída (termino de serviço) do estado  $n = 1$ . Dado que a taxa de serviço quando o sistema está no estado 1 é  $\mu_1$ , e a proporção de tempo que o sistema permanece no estado 1 é  $P_1$ , segue-se que  $\mu_1 P_1$  é a taxa à qual o sistema entra no estado 0 durante  $\Delta t \rightarrow 0$ .

Este sistema de equações de equilíbrio pode ser resolvido recursivamente, em termos de  $P_0$ . Para  $n = 0$ , temos:

$$\lambda_0 P_0 = \mu_1 P_1 \Rightarrow P_1 = \frac{\lambda_0}{\mu_1} P_0$$

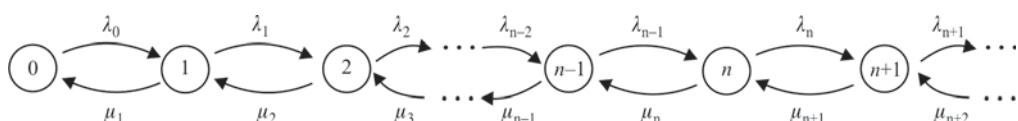
Para  $n = 1$ , temos:

$$(\lambda_1 + \mu_1) P_1 = \lambda_0 P_0 + \mu_2 P_2 \Rightarrow P_2 = \frac{(\lambda_1 + \mu_1) \left( \frac{\lambda_0}{\mu_1} P_0 \right) - \lambda_0 P_0}{\mu_2} = \frac{\lambda_0 \lambda_1}{\mu_1 \mu_2} P_0$$

e assim por diante, obtemos (por indução):

$$P_n = \frac{\lambda_0 \lambda_1 \dots \lambda_{n-2} \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_{n-1} \mu_n} P_0 = K_n P_0, \quad n = 1, 2, \dots \quad (7.5)$$

onde  $K_n = \frac{\lambda_0 \lambda_1 \dots \lambda_{n-2} \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_{n-1} \mu_n}$



**Figura 7.12** Diagrama de transição de estados do modelo de nascimento e morte.

O valor de  $P_0$  pode ser determinado lembrando que a soma das probabilidades  $P_0, P_1, P_2, \dots$ , deve ser igual a 1, ou seja:

$$\sum_{n=0}^{\infty} P_n = P_0 + \sum_{n=1}^{\infty} K_n P_0 = 1 \Rightarrow P_0 = \frac{1}{1 + \sum_{n=1}^{\infty} K_n} \quad (7.6)$$

Para que  $P_0 > 0$ , a série  $\sum_{n=1}^{\infty} K_n$  precisa convergir para um valor finito, isto é,  $\sum_{n=1}^{\infty} K_n < \infty$ . Por exemplo, se  $\lambda_n = 0$  para algum  $n$  (isso ocorre em sistemas de filas com capacidade limitada ou população finita, conforme visto na Seção 7.3), então os termos  $K_{n+1}, K_{n+2}, \dots$  são nulos e a série sempre converge. Note que, se a série não convergir, obtemos  $P_0 = P_1 = P_2 = \dots = 0$ , significando que o sistema de filas nunca estabiliza. A convergência da série é uma condição necessária para que o sistema de filas atinja um equilíbrio.<sup>10</sup> Neste caso, podemos calcular as probabilidades de equilíbrio  $P_0, P_1, P_2, \dots$  e, a partir delas, podemos calcular as medidas de desempenho (em equilíbrio) descritas na Seção 7.2, como  $E(L), E(L_q), E(L_s), E(W), E(W_q)$ . Por exemplo,

$$E(L) = \sum_{n=0}^{\infty} n P_n$$

As demais medidas podem ser obtidas a partir de  $E(L)$  usando a fórmula de Little (7.1) e as equações (7.2) – (7.4) definidas na Seção 7.2.<sup>11</sup> Conforme mencionado na Seção 7.2, no caso de a taxa de chegada  $\lambda_n$  variar com o estado  $n$  do sistema, a taxa  $\bar{\lambda}$  utilizada na fórmula de Little é a uma taxa média ponderada dessas taxas, definida por:

$$\bar{\lambda} = \sum_{n=0}^{\infty} \lambda_n P_n \quad (7.7)$$

Por exemplo, o tempo médio de permanência (em fila e serviço) no sistema é obtido por (usando a fórmula de Little):  $E(W) = \frac{E(L)}{\bar{\lambda}}$ .

**Exemplo 7.3** Uma farmácia possui três caixas. O gerente usa a seguinte estratégia de operação para abertura dos caixas: se o número de clientes na farmácia for menor ou igual a 2, a farmácia opera com apenas um caixa; se o número de clientes for igual a 3 ou 4, a farmácia opera com dois caixas; se o número de clientes for maior ou igual a 5, a farmácia opera com três caixas. Os clientes chegam na farmácia de acordo com uma distribuição de Poisson, com taxa de 10 clientes por hora. O tempo de atendimento de um cliente em um caixa segue uma distribuição exponencial, com tempo médio de 12 minutos por cliente (isto é, cada caixa atende em média cinco clientes por hora). (i) Qual a probabilidade de apenas um caixa ser aberto? (ii) Quantos caixas ficam em média fechados?

A taxa de chegada não depende do estado  $n$  do sistema, ou seja,  $\lambda_n = \lambda = 10$  clientes por hora, para qualquer  $n$ . Porém, a taxa de serviço depende do estado  $n$  do sistema:

$$\mu_n = \begin{cases} 5 \text{ clientes por hora, se } n = 1,2 \\ 10 \text{ clientes por hora, se } n = 3,4 \\ 15 \text{ clientes por hora, se } n = 5,6,\dots \end{cases}$$

Logo, a distribuição de equilíbrio  $P_0, P_1, P_2, \dots$  é dada por:

$$P_1 = \frac{\lambda_0}{\mu_1} P_0 = \frac{10}{5} P_0 = 2P_0$$

10 Para uma discussão sobre condições necessárias e suficientes para o equilíbrio do sistema, ver, por exemplo, Kleinrock (1975).

11 Em casos em que é difícil determinar uma fórmula em forma fechada para calcular  $E(L)$ , em vez do somatório  $\sum_{n=0}^{\infty} n P_n$ , podemos usar a função geratriz de momentos (transformada-z); ver, por exemplo, Kleinrock (1975) e Ross (1993).

$$\begin{aligned}
 P_2 &= \frac{\lambda_0 \lambda_1}{\mu_1 \mu_2} P_0 = \left(\frac{10}{5}\right)^2 P_0 = 4P_0 \\
 P_3 &= \frac{\lambda_0 \lambda_1 \lambda_2}{\mu_1 \mu_2 \mu_3} P_0 = \left(\frac{10}{5}\right)^2 \frac{10}{10} P_0 = 4P_0 \\
 P_4 &= \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3}{\mu_1 \mu_2 \mu_3 \mu_4} P_0 = \left(\frac{10}{5}\right)^2 \left(\frac{10}{10}\right)^2 P_0 = 4P_0 \\
 P_n &= \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 \lambda_4}{\mu_1 \mu_2 \mu_3 \mu_4 \mu_5} P_0 = \left(\frac{10}{5}\right)^2 \left(\frac{10}{10}\right)^2 \left(\frac{10}{15}\right)^{n-4} P_0 = 4 \left(\frac{10}{15}\right)^{n-4} P_0 \quad n = 5, 6, \dots
 \end{aligned}$$

onde o valor de  $P_0$  é obtido da equação (7.6) (lembrando que  $\sum_{n=0}^{\infty} z^n = \frac{1}{1-z}$  se  $|z| < 1$ ):

$$\begin{aligned}
 P_0 &= \frac{1}{1 + \sum_{n=1}^{\infty} K_n} = \frac{1}{1 + (2 + 4 + 4 + 4 + 4(2/3) + 4(2/3)^2 + 4(2/3)^3 + \dots)} \\
 &= \frac{1}{11 + 4 \left(1 + \left(\frac{2}{3}\right) + \left(\frac{2}{3}\right)^2 + \left(\frac{2}{3}\right)^3 + \dots\right)} = \frac{1}{11 + 4 \left(\frac{1}{1 - (2/3)}\right)} = \frac{1}{23}
 \end{aligned}$$

Note que a série  $\sum_{n=1}^{\infty} K_n$  converge para um valor finito, condição necessária para que o sistema atinja um equilíbrio. Substituindo o valor de  $P_0$  nas expressões anteriores, obtemos cada uma das probabilidades  $P_1, P_2, P_3, \dots$

- (i) A probabilidade de apenas um caixa ser aberto é igual à probabilidade de o número de clientes na farmácia ser menor ou igual a 2, ou seja:

$$P_0 + P_1 + P_2 = (1+2+4) \frac{1}{23} \approx 0,30$$

- (ii) Em função da estratégia de operação do gerente para abertura dos caixas (se o número de clientes na farmácia for igual a 0, 1 ou 2, a farmácia opera com apenas um caixa; se for igual a 3 ou 4, opera com dois caixas; se for maior ou igual a 5, com os três caixas), o número de caixas que ficam em média fechados é:

$$2(P_0 + P_1 + P_2) + 1(P_3 + P_4) + 0(P_5 + P_6 + \dots) = 2 \frac{1+2+4}{23} + 1 \frac{4+4}{23} = \frac{22}{23} \approx 1$$

■

## 7.4 MODELOS DE FILAS COM UM SERVIDOR E DISTRIBUIÇÕES EXPONENCIAIS

A seguir, estudamos modelos de filas de nascimento e morte do tipo  $M/M/1//\cdot\cdot$ , ou seja, com apenas um servidor ( $m = 1$ ) e processos de chegada e serviço com distribuições exponenciais. Note que estes modelos se referem a sistemas monoestágio de fila única e um servidor (tipo (i) da Figura 7.3).

### 7.4.1 Modelo de fila $M/M/1/GD/\infty/\infty$ – um servidor

Conforme a Seção 7.1, o sistema de fila  $M/M/1/GD/\infty/\infty$  (ou simplesmente  $M/M/1$ ) admite que os intervalos de tempo entre as chegadas são exponencialmente distribuídos com média  $E(X) = 1/\lambda$ , e os tempos de serviço são exponencialmente distribuídos com média  $E(S) = 1/\mu$ . O sistema tem

um único servidor ( $m = 1$ ) e a disciplina de fila é qualquer.<sup>12</sup> Não há limitações para o tamanho da fila, nem para o tamanho da população de usuários. O fator de utilização do sistema é  $\rho = \frac{\lambda}{m\mu} = \frac{\lambda}{\mu}$ .

Note que este sistema pode ser representado por um modelo de nascimento e morte específico, com:

$$\begin{aligned}\lambda_n &= \lambda & n &= 0, 1, 2, \dots \\ \mu_n &= \mu & n &= 1, 2, 3, \dots\end{aligned}$$

ou seja, as taxas  $\lambda_n$  e  $\mu_n$  independem do estado  $n$  do sistema. A Figura 7.13 ilustra o diagrama de transição de estados para o sistema  $M/M/1/GD/\infty/\infty$ . Como todos os usuários que chegam também entram no sistema, segue-se que  $\bar{\lambda} = \lambda$  (isto é, a taxa de entrada é igual à taxa de chegada de usuários), portanto, o fator de utilização  $\rho$  corresponde à utilização média do sistema.

A distribuição de equilíbrio deste modelo de nascimento e morte é dada por [equações (7.5) e (7.6)]:

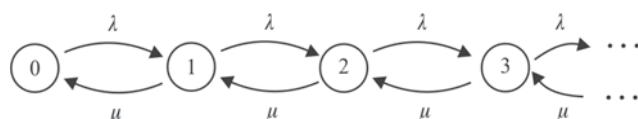
$$\begin{aligned}P_0 &= \frac{1}{1 + \sum_{n=1}^{\infty} K_n} = \frac{1}{1 + \sum_{n=1}^{\infty} \left(\frac{\lambda}{\mu}\right)^n} = \frac{1}{\sum_{n=0}^{\infty} \rho^n} & n &= 0 \\ P_n &= K_n P_0 = \rho^n P_0 & n &= 1, 2, \dots\end{aligned}$$

Admitindo que  $\rho < 1$ , obtemos (lembrando que  $\sum_{n=0}^{\infty} z^n = \frac{1}{1-z}$ , se  $|z| < 1$ ):

$$\begin{aligned}P_0 &= \frac{1}{\sum_{n=0}^{\infty} \rho^n} = \frac{1}{\left(\frac{1}{1-\rho}\right)} = 1 - \rho \\ P_n &= \rho^n P_0 = \rho^n (1 - \rho) & n &= 0, 1, 2, \dots\end{aligned}$$

Ou seja, para a série  $\sum_{n=1}^{\infty} K_n$  convergir, precisamos ter  $\rho = \frac{\lambda}{\mu} < 1$ . Esta é a condição para que o sistema  $M/M/1/GD/\infty/\infty$  atinja um equilíbrio. Se  $\lambda > \mu$ , então a taxa de chegada supera a taxa de serviço, e o sistema nunca atinge o estado de equilíbrio, pois a fila só tende a crescer com o passar do tempo.<sup>13</sup>

Na primeira expressão anterior, a probabilidade de o sistema estar no estado  $n = 0$  usuário ( $P_0$ ) é exatamente igual à probabilidade de o servidor estar desocupado ( $1 - \rho$ ). A segunda expressão mostra que a probabilidade de o sistema estar no estado  $n$  tem distribuição geométrica ( $\rho^n(1 - \rho)$ ). Dado que  $L$  tem distribuição geométrica,<sup>14</sup> segue-se que  $P(L \geq n) = \rho^n$  e o número médio de usuários no sistema são definidos por (após alguma álgebra):



**Figura 7.13** Diagrama de transição de estados do sistema  $M/M/1/GD/\infty/\infty$ .

<sup>12</sup> Aqui e nos demais modelos de filas com distribuições exponenciais, a disciplina *GD* significa qualquer disciplina que ordene os usuários de uma maneira tal que seja independente de seus tempos de serviço. Um exemplo de disciplina que depende dos tempos de serviço é a disciplina *SPT* (*shortest processing time*).

<sup>13</sup> Para uma discussão de por que o sistema não atinge um equilíbrio com  $\rho = 1$ , ver, por exemplo, Kleinrock (1975) e Gross e Harris (1985).

<sup>14</sup> Ver, por exemplo, distribuição geométrica em Feller (1957) e Magalhães e Lima (2001).

$$\begin{aligned}
 E(L) &= \sum_{n=0}^{\infty} nP_n = \sum_{n=0}^{\infty} n\rho^n(1-\rho) = (1-\rho)\rho \sum_{n=1}^{\infty} n\rho^{n-1} = (1-\rho)\rho \frac{d}{d\rho} \left( \sum_{n=0}^{\infty} \rho^n \right) \\
 &= (1-\rho)\rho \frac{d}{d\rho} \left( \frac{1}{1-\rho} \right) = \frac{(1-\rho)\rho}{(1-\rho)^2} = \frac{\rho}{1-\rho}
 \end{aligned}$$

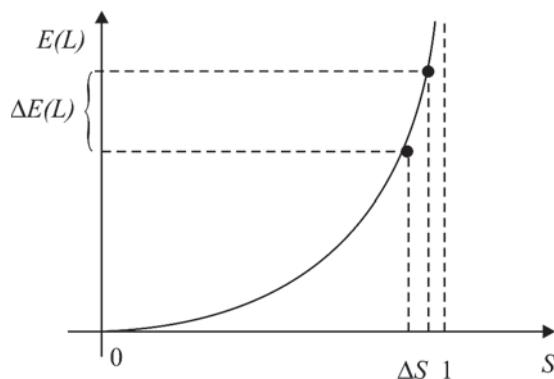
De modo semelhante, o cálculo da variância de  $L$  resulta em  $V(L) = \frac{\rho}{(1-\rho)^2}$ . As demais medidas de desempenho  $E(L_s)$ ,  $E(L_q)$ ,  $E(W)$ ,  $E(W_q)$  podem ser obtidas usando a fórmula de Little (7.1) e as equações (7.2) – (7.4) da Seção 7.2 (note que  $\bar{\lambda} = \lambda$ ). Por exemplo,

$$\begin{aligned}
 E(L_s) &= \lambda E(S) = \rho \\
 E(L_q) &= E(L) - E(L_s) = \frac{\rho}{1-\rho} - \rho = \frac{\rho^2}{1-\rho} \\
 E(W) &= \frac{E(L)}{\lambda} = \frac{\rho}{\lambda(1-\rho)} \\
 E(W_q) &= E(W) - E(S) = \frac{\rho}{\lambda(1-\rho)} - \frac{1}{\mu} = \frac{\rho}{\mu(1-\rho)}
 \end{aligned}$$

Observe que, exceto para  $E(L_s)$ , todas as demais medidas tendem ao infinito no limite  $\rho \rightarrow 1$ , como era esperado. Ou seja, quando  $\rho$  é próximo de 1, um pequeno aumento em  $\rho$  provoca um enorme aumento nessas medidas (veja Figura 7.14). Esse comportamento assintótico é comum nas medidas de desempenho dos modelos de filas.

**Exemplo 7.4** Em um *drive-in* de um restaurante de *fast-food*, chegam em média 10 carros por hora. Se o tempo médio de atendimento de cada cliente é de 4 minutos e os intervalos de tempo entre chegadas e os tempos de atendimento são exponenciais: (i) Qual a probabilidade de o sistema estar vazio? (ii) Qual o número médio de carros no *drive-in* e o número médio de carros em fila no *drive-in*? (iii) Qual o tempo médio de permanência no *drive-in* e o tempo médio de espera em fila no *drive-in*? (iv) Qual a probabilidade de haver dois ou mais carros (em fila e em serviço) no *drive-in*?

Podemos representar o sistema por um modelo  $M/M/1$ . Note que  $\lambda = 10$  carros por hora ( $E(X) = \frac{1}{\lambda} = \frac{60}{10} = 6$  minutos) e  $E(S) = 4$  minutos ( $\mu = \frac{1}{E(S)} = \frac{60}{4} = 15$  carros por hora). Logo, a utilização média do sistema é:  $\rho = \frac{\lambda}{\mu} = \frac{10}{15} \approx 0,67$ .



**Figura 7.14** Variação de  $E(L)$  em função de  $\rho$  em um sistema  $M/M/1$ .

- (i) A probabilidade de o sistema estar vazio é:  $P_0 = 1 - \rho \approx 0,33$ .
- (ii) O número médio de carros no sistema é:  $E(L) = \frac{\rho}{1 - \rho} = \frac{0,67}{1 - 0,67} = 2$  carros. O número médio de carros em serviço e na fila é:  $E(L_s) = \rho \approx 0,67$  e  $E(L_q) = \frac{\rho^2}{1 - \rho} = \frac{0,67^2}{1 - 0,67} \approx 1,33$ , respectivamente. Note que  $E(L) = E(L_s) + E(L_q)$ .
- (iii) O tempo médio de permanência no sistema é:  $E(W) = \frac{\rho}{\lambda(1 - \rho)} = \frac{0,67}{10(1 - 0,67)} = 0,2$  hora (ou 12 minutos). O tempo médio de espera em fila é:  $E(W_q) = \frac{\rho}{\mu(1 - \rho)} = \frac{0,67}{15(1 - 0,67)} \approx 0,13$  hora (ou 8 minutos). Note que  $E(W) = E(S) + E(W_q)$ .
- (iv) A probabilidade de haver dois ou mais carros no *drive-in* é:  $P(L \geq 2) = \rho^2 \approx 0,44$ . ■

**Exemplo 7.5** No Exemplo 7.4 do *drive-in* do restaurante de *fast-food*: (i) Como mudam as medidas de desempenho se o tempo médio de serviço  $E(S)$  for reduzido pela metade (isto é, a taxa de serviço  $\mu$  for duplicada)? (ii) Como mudam as medidas de desempenho se um segundo *drive-in* for aberto, igual ao primeiro, do outro lado do restaurante? Note que este último sistema pode ser visto como um sistema de múltiplas filas e múltiplos servidores em paralelo [tipo (iii) da Figura 7.3].

- (i) Se  $E(S) = 2$  minutos (antes era 4 minutos), ou seja,  $\mu = \frac{1}{E(S)} = \frac{60}{2} = 30$  carros por hora, segue-se que a utilização média do sistema é:  $\rho = \frac{\lambda}{\mu} = \frac{10}{30} \approx 0,33$  (antes era 0,67).
- (ii) Considerando que a chegada de clientes vai se dividir igualmente entre os dois *drive-in* (isto é,  $\lambda = 5$  carros por hora em cada *drive-in*), e que os carros em fila não podem trocar de fila por causa da separação física dos dois *drive-in*, os dois sistemas de fila  $M/M/1$  são independentes e temos que a utilização média de cada *drive-in* é  $\rho \approx 0,33$  (antes era 0,67).<sup>15</sup>

A Tabela 7.1 a seguir apresenta os resultados obtidos para o *drive-in* do sistema (i) com taxa de serviço duplicada (sistema com superservidor), para cada *drive-in* do sistema (ii) (sistema com 2 *drive-in*), comparados com os obtidos para o *drive-in* do sistema anterior do Exemplo 7.4 (sistema com 1 *drive-in*).

Note que, ao duplicar a taxa de serviço do servidor (superservidor), as medidas de desempenho  $E(L)$ ,  $E(L_q)$ ,  $E(W)$  e  $E(W_q)$  reduzem em bem mais da metade, em relação ao sistema anterior.

**Tabela 7.1**  
Resultados obtidos para o sistema do *drive-in*.

Sistema	$\lambda$	$\mu$	$m$	$\rho$	$P_0$	$E(L)$	$E(L_q)$	$E(W)$	$E(W_q)$	$P(L \geq 2)$
1 <i>drive-in</i>	10	15	1	0,67	0,33	2	1,33	12	8	0,44
superservidor	10	30	1	0,33	0,67	0,5	0,17	3	1	0,11
2 <i>drive-in</i>	5	15	1	0,33	0,67	0,5	0,17	6	2	0,11

<sup>15</sup> Para um modelo de filas que considera casos em que os usuários, ao chegar, escolhem entrar no sistema com menor fila, e casos em que os usuários, ao esperar em fila, escolhem mudar de fila se a fila adjacente estiver menor, veja, por exemplo, Morabito e Lima (2004).

Por outro lado, ao duplicar o sistema (dois *drive-in* em vez de um), para cada *drive-in*,  $E(L)$  e  $E(L_q)$  também reduzem como o sistema com superservidor, mas, para cada *drive-in*,  $E(W)$  e  $E(W_q)$  são o dobro do sistema com superservidor (porque o tempo médio de serviço  $E(S)$  não foi alterado). ■

Todos os resultados anteriores para o sistema  $M/M/1/GD/\infty/\infty$  são independentes da disciplina da fila. Contanto que a disciplina da fila ordene os usuários de uma maneira tal que seja independente de seus tempos de serviço, então a distribuição de probabilidade do número de usuários no sistema  $P_n$  é invariante com a disciplina da fila; consequentemente, o número médio de usuários no sistema  $E(L)$  também não varia com a disciplina da fila. Portanto, lembrando que a fórmula de Little é válida para qualquer disciplina, segue-se que  $E(L_s)$ ,  $E(L_q)$ ,  $E(W)$  e  $E(W_q)$  também não variam com a disciplina da fila.

Por exemplo, para as disciplinas *FCFS*, *SIRO* e *LCFS*, os tempos médios de permanência  $E(W)$  nos sistemas  $M/M/1/FCFS/\infty/\infty$ ,  $M/M/1/SIRO/\infty/\infty$  e  $M/M/1/LCFS/\infty/\infty$  são os mesmos, isto é:

$$E(W_{FCFS}) = E(W_{SIRO}) = E(W_{LCFS}) = \frac{\rho}{\lambda(1-\rho)}$$

No entanto, a distribuição de probabilidade  $f_W(w)$  do tempo de permanência no sistema  $W$  depende da disciplina da fila. De fato, pode ser mostrado que as variâncias de  $W$  para as disciplinas *FCFS*, *SIRO* e *LCFS* são tais que:<sup>16</sup>

$$V(W_{FCFS}) < V(W_{SIRO}) < V(W_{LCFS})$$

indicando que maiores variações no tempo de permanência no sistema são mais prováveis de ocorrer com a disciplina *LCFS* do que com a disciplina *FCFS*, o que era intuitivamente esperado. Basta lembrar que, ao chegarmos em um sistema com disciplina *LCFS* (diferentemente de um sistema com disciplina *FCFS*), o número de usuários que encontramos no sistema (em fila e serviço) é pouco informativo para estimarmos nosso tempo de permanência no sistema, uma vez que seremos o próximo usuário a ser atendido, caso nenhum usuário chegue depois de nós enquanto esperamos pelo início do atendimento.

Um resultado interessante é que, no sistema  $M/M/1/FCFS/\infty/\infty$ , a distribuição de  $W$  (tempo de permanência no sistema) é exponencial com média  $E(W) = \frac{\rho}{\lambda(1-\rho)} = \frac{1}{\mu-\lambda}$  (isto é, com taxa  $\frac{1}{E(W)} = \mu - \lambda$ ), ou seja:<sup>17</sup>

$$f_W(w) = (\mu - \lambda)e^{-(\mu - \lambda)w} \quad w > 0$$

Logo, a probabilidade de um tempo de permanência no sistema maior que  $w$  é:

$$P(W > w) = \int_w^\infty f_W(w)dw = e^{-(\mu - \lambda)w} \quad w \geq 0$$

**Exemplo 7.6** No Exemplo 7.4 do *drive-in* do restaurante de *fast-food*, considerando que os carros em fila no *drive-in* são atendidos com disciplina de fila *FCFS*, qual a probabilidade de um cliente permanecer mais de 15 minutos no sistema com 1 *drive-in*?

O tempo médio de permanência no sistema é:  $E(W) = \frac{1}{\mu - \lambda} = \frac{1}{15 - 10} = 0,2$  hora (ou 12 minutos). A probabilidade de o cliente permanecer mais de 15 minutos (0,25 hora) no sistema é:

$$P(W > 0,25) = e^{-(\mu - \lambda)0,25} = e^{-1,25} \approx 0,29$$

16 Ver, por exemplo, Kleinrock (1975) e Larson e Odoni (1981).

17 Para uma demonstração disso, ver, por exemplo, Kleinrock (1975) e Larson e Odoni (1981).

#### 7.4.2 Modelo de fila $M/M/1/GD/K/\infty$ – um servidor e capacidade limitada

O sistema de fila  $M/M/1/GD/K/\infty$  (ou simplesmente  $M/M/1/K$ ) difere do sistema  $M/M/1/GD/\infty/\infty$  analisado anteriormente apenas pela limitação  $K$  do número de usuários presentes no sistema (em fila e serviço), o que resulta que o tamanho máximo da fila é  $K - 1$ , isto é, a capacidade da sala de espera de algum sistema de serviço com apenas um servidor. Note que este sistema pode ser representado por um modelo de nascimento e morte com:

$$\lambda_n = \begin{cases} \lambda & n = 0, 1, 2, \dots, K-1 \\ 0 & n = K, K+1, \dots \end{cases}$$

$$\mu_n = \begin{cases} \mu & n = 1, 2, 3, \dots, K \\ 0 & n = K+1, K+2, \dots \end{cases}$$

(repare as limitações para a variação de  $n$ ). A Figura 7.15 ilustra o diagrama de transição de estados para o sistema  $M/M/1/GD/K/\infty$ .

Usando as expressões (7.5) e (7.6), a distribuição de equilíbrio é dada por (lembrando que  $\sum_{n=0}^N z^n = \frac{1-z^{N+1}}{1-z}$ , se  $z \neq 1$ ):

$$P_0 = \begin{cases} \frac{1}{1 + \sum_{n=1}^{\infty} K_n} = \frac{1}{1 + \sum_{n=1}^K \left(\frac{\lambda}{\mu}\right)^n} = \frac{1}{\sum_{n=0}^K \rho^n} = \frac{1-\rho}{1-\rho^{K+1}} & \rho \neq 1 \\ \frac{1}{1 + \sum_{n=1}^{\infty} K_n} = \frac{1}{\sum_{n=0}^K \rho^n} = \frac{1}{K+1} & \rho = 1 \end{cases}$$

$$P_n = \begin{cases} K_n P_0 = \rho^n P_0 = \frac{\rho^n(1-\rho)}{1-\rho^{K+1}} & \rho \neq 1 \\ K_n P_0 = \rho^n P_0 = \frac{1}{K+1} & \rho = 1 \end{cases}, \quad n = 0, 1, 2, \dots, K$$

Portanto, não precisamos mais da condição  $\rho < 1$  para que essa série converja. De fato, o sistema também atinge um equilíbrio mesmo com  $\rho$  bem maior que 1, mas o tamanho da fila tende a ser igual a  $K$  na maior parte do tempo, resultando em uma grande perda de usuários.

O número médio de usuários no sistema é definido por (após alguma álgebra):

$$E(L) = \sum_{n=0}^K n P_n = \sum_{n=0}^K n \frac{\rho^n(1-\rho)}{1-\rho^{K+1}} = \frac{1-\rho}{1-\rho^{K+1}} \sum_{n=0}^K n \rho^n = \frac{1-\rho}{1-\rho^{K+1}} \rho \frac{d}{d\rho} \left( \sum_{n=0}^K \rho^n \right)$$

$$= \frac{(1-\rho)\rho}{1-\rho^{K+1}} \frac{d}{d\rho} \left( \frac{1-\rho^{K+1}}{1-\rho} \right) = \frac{\rho[1-(K+1)\rho^K + K\rho^{K+1}]}{(1-\rho)(1-\rho^{K+1})} \quad \text{se } \rho \neq 1$$

$$E(L) = \sum_{n=0}^K n P_n = \sum_{n=0}^K n \frac{1}{K+1} = \frac{1}{K+1} \sum_{n=1}^K n = \frac{1}{K+1} \left( \frac{(1+K)K}{2} \right) = \frac{K}{2} \quad \text{se } \rho = 1$$

Note que, se  $K \rightarrow \infty$  e  $\rho < 1$ , então as expressões anteriores se reduzem a  $P_0 = 1 - \rho$ ,  $P_n = \rho^n(1 - \rho)$  e  $E(L) = \frac{\rho}{1 - \rho}$ , que correspondem aos resultados obtidos para o sistema  $M/M/1/GD/\infty/\infty$ .

As demais medidas de desempenho  $E(L_s)$ ,  $E(L_q)$ ,  $E(W)$ ,  $E(W_q)$  podem ser obtidas usando a fórmula de Little (7.1) e as equações (7.2) – (7.4) da Seção 7.2. Conforme mencionado na Seção 7.2, em sistemas em que usuários que chegam podem ser perdidos ou impedidos de entrar no sistema (Figura 7.8), como é o caso, a taxa de entrada  $\bar{\lambda}$  a ser usada na fórmula de Little deve ser a média ponderada das taxas (conforme a equação (7.7)):

$$\bar{\lambda} = \sum_{n=0}^{\infty} \lambda_n P_n = \lambda \sum_{n=0}^{K-1} P_n + 0P_K = \lambda(1 - P_K)$$

(lembre-se de que  $\lambda_K = 0$  e  $\sum_{n=0}^K P_n = 1$ ). Por exemplo, o tempo médio de permanência no sistema é obtido por (usando a fórmula de Little):  $E(W) = \frac{E(L)}{\bar{\lambda}} = \frac{E(L)}{\lambda(1 - P_K)}$ .

Note que  $(1 - P_K)$  corresponde à probabilidade de o sistema não estar cheio (isto é, com  $K$  usuários), ou seja, a probabilidade de um usuário chegar e entrar no sistema. Assim,  $\bar{\lambda} = \lambda(1 - P_K)$  é a taxa de entrada de usuários no sistema, enquanto  $\lambda P_K$  é a taxa de perda de usuários (obviamente, a taxa de chegada  $\lambda$  é a soma de ambas). Como  $\bar{\lambda} < \lambda$ , segue-se que o fator de utilização  $\rho$  não corresponde à utilização média do sistema (fração de tempo em que o servidor fica ocupado), dada por  $E(L_s) = \bar{\lambda} E(S)$ .

Do ponto de vista do projeto de um sistema de fila  $M/M/1/GD/K/\infty$ , a probabilidade de o sistema estar cheio  $P_K$  é um parâmetro importante. Note que, se  $P_K$  for pequeno, então o sistema de capacidade limitada  $K$  se comporta aproximadamente como um sistema de capacidade ilimitada  $M/M/1/GD/\infty/\infty$ .

**Exemplo 7.7** No Exemplo 7.4 do *drive-in* do restaurante de *fast-food*, considere que não haja espaço físico para mais de cinco carros (em fila e serviço), caso contrário, a fila invade a rua do restaurante e os clientes, ao chegarem, desistem de entrar na fila. (i) Qual a taxa de perda de clientes e a utilização média do sistema? (ii) Como mudam o número médio de carros na fila e o tempo médio de espera na fila?

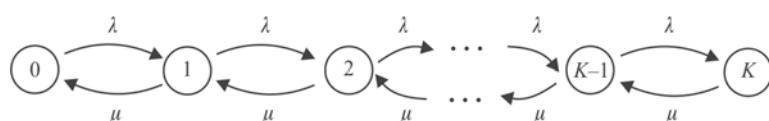
- (i) Podemos representar o sistema por um modelo  $M/M/1/GD/5/\infty$ . Lembrando que  $\rho = 0,67 \neq 1$ , a probabilidade de perda de clientes é:

$$P_5 = \frac{\rho^5(1 - \rho)}{1 - \rho^{5+1}} = \frac{0,67^5(1 - 0,67)}{1 - 0,67^6} \approx 0,05$$

Logo, a taxa de perda é:  $\lambda P_5 = 10(0,05) = 0,5$  usuário por hora. A utilização média corresponde a:

$$E(L_s) = \bar{\lambda} E(S) = \frac{\lambda(1 - P_5)}{\mu} = \frac{10(1 - 0,05)}{15} \approx 0,63$$

ou seja, o servidor fica ocupado cerca de 63% do tempo de operação.



**Figura 7.15** Diagrama de transição de estados do sistema  $M/M/1/GD/K/\infty$ .

(ii) Sem a limitação de capacidade, temos  $E(L_q) \approx 1,33$  carro e  $E(W_q) \approx 8$  minutos. Com a limitação de capacidade ( $K = 5$ ), esses valores mudam para:

$$E(L_q) = E(L) - E(L_s) = \frac{\rho[1-(5+1)\rho^5 + 5\rho^{5+1}]}{(1-\rho)(1-\rho^{5+1})} - E(L_s) \approx 0,79 \text{ carro}$$

$$E(W_q) = \frac{E(L_q)}{\bar{\lambda}} \approx 4,97 \text{ minutos}$$

A Tabela 7.2 a seguir compara os resultados dos sistemas sem (Exemplo 7.4) e com limitação de capacidade ( $K = 5$ ). ■

#### 7.4.3 Modelo de fila $M/M/1/GD/\infty/N$ – um servidor e população finita

O sistema de fila  $M/M/1/GD/\infty/N$  difere do sistema  $M/M/1/GD/\infty/\infty$  apenas pela limitação  $N$  do tamanho da população (fonte de usuários), o que resulta que o tamanho máximo do número de usuários no sistema é  $N$  (todos os usuários no sistema). Note que este sistema pode ser representado por um modelo de nascimento e morte com:

$$\lambda_n = \begin{cases} (N-n)\lambda & n = 0, 1, \dots, N \\ 0 & n = N+1, N+2, \dots \end{cases}$$

$$\mu_n = \begin{cases} \mu & n = 0, 1, \dots, N \\ 0 & n = N+1, N+2, \dots \end{cases}$$

(repare as limitações para a variação de  $n$ ), ou seja, a taxa  $\lambda_n$  varia com o estado  $n$  do sistema. Note que, quando todos os usuários estão no sistema ( $n = N$ ), a taxa de chegada  $\lambda_N$  é nula e, quando o sistema está vazio ( $n = 0$ ), a taxa de chegada é  $\lambda_0 = N\lambda$ . Seguindo os mesmos passos dos modelos anteriores, obtemos a distribuição de equilíbrio com as equações (7.5) e (7.6):

$$P_n = K_n P_0 = \frac{(N\lambda)((N-1)\lambda)((N-2)\lambda)...((N-n+1)\lambda)}{\mu^n} P_0 = \left(\frac{\lambda}{\mu}\right)^n \frac{N!}{(N-n)!} P_0, \quad n = 1, 2, \dots, N$$

em que:  $P_0 = \frac{1}{1 + \sum_{n=1}^{\infty} K_n} = \frac{1}{\sum_{n=0}^N \left(\frac{\lambda}{\mu}\right)^n \frac{N!}{(N-n)!}}$

Assim como no modelo  $M/M/1/GD/K/\infty$ , também não precisamos da condição  $\rho < 1$  para que a série converja e o modelo  $M/M/1/GD/\infty/N$  atinja um equilíbrio.

O número médio de usuários no sistema pode ser encontrado com:

$$E(L) = \sum_{n=0}^N n P_n$$

**Tabela 7.2**  
Resultados para sistemas sem e com limitação de capacidade.

Sistema	$\lambda$	$\mu$	$m$	$\rho$	Perda	$E(L_s)/m$	$E(L_q)$	$E(W_q)$
$M/M/1/GD/\infty/\infty$	10	15	1	0,67	0	0,67	1,33	8
$M/M/1/GD/5/\infty$	10	15	1	0,67	0,5	0,63	0,79	4,97

e as demais medidas de desempenho  $E(L_s)$ ,  $E(L_q)$ ,  $E(W)$ ,  $E(W_q)$  podem ser obtidas usando a fórmula de Little (7.1) e as equações (7.2) – (7.4) da Seção 7.2. Note que a taxa de entrada de usuários no sistema a ser usada na fórmula de Little deve ser [conforme (7.7)]:

$$\bar{\lambda} = \sum_{n=0}^{\infty} \lambda_n P_n = \sum_{n=0}^N (N-n)\lambda P_n = N\lambda \sum_{n=0}^N P_n - \lambda \sum_{n=0}^N nP_n = N\lambda - \lambda E(L) = \lambda(N - E(L))$$

#### 7.4.4 Modelos de decisão de filas – um servidor

Em modelos de filas com um servidor, o nível de serviço aos usuários depende fundamentalmente da taxa de serviço  $\mu$  (capacidade do sistema). Melhores níveis de serviço são obtidos com menores tempos de espera em fila ou menores perdas de usuários no sistema. Para isso, precisamos alocar mais capacidade no sistema, o que resulta em maiores custos operacionais e investimento. Conforme mencionado na Seção 7.1, modelos de decisão de sistemas de filas procuram balancear o conflito entre os custos de oferecer um nível de serviço no sistema e os custos dos atrasos sofridos pelos usuários.

No caso de sistemas com um servidor, uma decisão importante é a escolha entre diversas alternativas de capacidade  $\mu_1, \mu_2, \dots$ , para possível alocação no sistema. Para cada alternativa, podemos estimar medidas de desempenho como  $E(L)$ ,  $E(W)$  etc., usando os modelos de filas descritos nesta seção, e, em seguida, avaliar os custos envolvidos em cada alternativa associados à taxa de serviço e às medidas de desempenho desta alternativa. A alternativa de mínimo custo total é então escolhida, eventualmente com auxílio de gráficos de curvas de *trade-off* entre estas medidas e curvas de custo total.

Em alguns casos, também podemos utilizar modelos de otimização em que a taxa de serviço  $\mu$  é a variável de decisão, conforme ilustrado no exemplo a seguir.

**Exemplo 7.8**<sup>18</sup> No Exemplo 7.7 do *drive-in* do restaurante de *fast-food*, considere que o sistema tenha um custo médio de  $c\mu = 2\mu$  (unidades monetárias por hora) para operar com taxa de serviço  $\mu$ , mas lucre em média  $d = 5$  (unidades monetárias por cliente) para cada cliente que entra no sistema e é servido. Se o sistema tem limite de capacidade  $K = 5$  (clientes) devido ao espaço físico para os carros (em fila e serviço), qual é a taxa  $\mu$  que maximiza o lucro do sistema?

Dado que o custo por hora é  $c\mu$  e o número médio de clientes que entra por hora (taxa de entrada) é  $\bar{\lambda} = \lambda(1 - P_K)$  (veja Exemplo 7.7), o lucro por hora do sistema pode ser definido por:

$$\text{Lucro por hora} = \lambda(1 - P_K)d - c\mu$$

$$\text{Lembrando que } P_K = \frac{\rho^K(1-\rho)}{1-\rho^{K+1}} = \frac{(\lambda/\mu)^K(1-\lambda/\mu)}{1-(\lambda/\mu)^{K+1}}, \text{ temos que:}$$

$$\text{Lucro por hora} = \lambda \left( 1 - \frac{(\lambda/\mu)^K(1-\lambda/\mu)}{1-(\lambda/\mu)^{K+1}} \right) d - c\mu = \frac{\lambda d (1 - (\lambda/\mu)^K)}{1 - (\lambda/\mu)^{K+1}} - c\mu$$

Substituindo os valores  $\lambda = 10$  (Exemplo 7.7),  $K = 5$ ,  $c = 2$  e  $d = 5$  na expressão anterior, obtemos:

$$\text{Lucro por hora} = \frac{10(5)(1 - (10/\mu)^5)}{1 - (10/\mu)^{5+1}} - 2\mu = \frac{50(1 - (10/\mu)^5)}{1 - (10/\mu)^6} - 2\mu$$

<sup>18</sup> Este exemplo foi inspirado em um exemplo em Ross (1993).

Derivando-se essa expressão em relação à  $\mu$  e igualando-a a zero, obtemos:

$$\frac{d}{d\mu} \left( \frac{50(1-(10/\mu)^5)}{1-(10/\mu)^6} - 2\mu \right) = 0$$

A solução dessa equação (obtida por algum método para achar a raiz de um polinômio) fornece o valor de  $\mu$  que maximiza o lucro do sistema. ■

## 7.5 MODELOS DE FILAS COM MÚLTIPLOS SERVIDORES E DISTRIBUIÇÕES EXPONENCIAIS

Nesta seção, estendemos nossa análise para modelos de nascimento e morte do tipo  $M/M/m//./.$ , isto é, modelos com mais de um servidor ( $m > 1$ ) e processos de chegada e serviço com distribuições exponenciais. Note que estes modelos referem-se a sistemas monoestágios de fila única e múltiplos servidores (tipo (ii) da Figura 7.3).

### 7.5.1 Modelo de fila $M/M/m/GD/\infty/\infty$ – múltiplos servidores

O sistema de fila  $M/M/m/GD/\infty/\infty$  (ou simplesmente  $M/M/m$ ) difere do sistema  $M/M/1/GD/\infty/\infty$  anterior apenas pelo número  $m > 1$  de servidores (idênticos e em paralelo) no sistema. Os usuários esperam em fila única para atendimento nos servidores, como em diversas agências de bancos e correios. Note que este sistema pode ser representado por um modelo de nascimento e morte com:

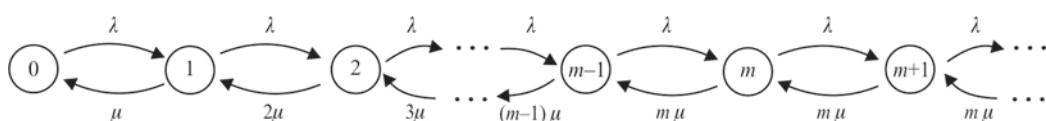
$$\lambda_n = \lambda \quad n = 0, 1, 2, \dots$$

$$\mu_n = \begin{cases} n\mu, & n = 1, 2, \dots, m-1 \\ m\mu, & n = m, m+1, \dots \end{cases}$$

ou seja, a taxa  $\mu_n$  varia com o estado  $n$  do sistema. A Figura 7.16 ilustra o diagrama de transição de estados para o sistema  $M/M/m/GD/\infty/\infty$ . Como todos os usuários que chegam também entram no sistema, segue-se que  $\bar{\lambda} = \lambda$  (isto é, a taxa de entrada é igual à taxa de chegada de usuários) e o fator de utilização  $\rho = \frac{\lambda}{m\mu}$  corresponde à utilização média do sistema (proporção de servidores ocupados).

A distribuição de equilíbrio deste modelo de nascimento e morte é dada por (equação (7.5)):

$$P_n = \begin{cases} K_n P_0 = \frac{\lambda^n}{(1\mu)(2\mu)\dots(n\mu)} P_0 = \frac{\lambda^n}{\mu^n n!} P_0 = \frac{(\rho m)^n}{n!} P_0 & n = 1, 2, \dots, m-1 \\ K_m P_0 = \frac{\lambda^n}{(1\mu)(2\mu)\dots(m\mu)(m\mu)^{n-m}} P_0 = \frac{\lambda^n}{\mu^n m! m^{n-m}} P_0 = \frac{\rho^n m^m}{m!} P_0 & n = m, m+1, \dots \end{cases}$$



**Figura 7.16** Diagrama de transição de estados do sistema  $M/M/m/GD/\infty/\infty$ .

Usando a equação (7.6), o valor de  $P_0$  é definido por (após alguma álgebra):

$$\begin{aligned} P_0 &= \frac{1}{1 + \sum_{n=1}^{\infty} K_n} = \frac{1}{\sum_{n=0}^{m-1} \frac{\lambda^n}{\mu^n n!} + \sum_{n=m}^{\infty} \frac{\lambda^n}{\mu^m m! m^{n-m}}} = \frac{1}{\sum_{n=0}^{m-1} \frac{\lambda^n}{\mu^n n!} + \frac{\lambda^m}{\mu^m m!} \sum_{n=m}^{\infty} \frac{\lambda^{n-m}}{\mu^{n-m} m^{n-m}}} \\ &= \frac{1}{\sum_{n=0}^{m-1} \frac{\lambda^n}{\mu^n n!} + \frac{\lambda^m}{\mu^m m!} \sum_{n=m}^{\infty} \rho^{n-m}} = \frac{1}{\sum_{n=0}^{m-1} \frac{\lambda^n}{\mu^n n!} + \frac{\lambda^m}{\mu^m m!} \sum_{n'=0}^{\infty} \rho^{n'}} \end{aligned}$$

Admitindo que  $\rho < 1$ , obtemos:

$$P_0 = \frac{1}{\sum_{n=0}^{m-1} \frac{\lambda^n}{\mu^n n!} + \frac{\lambda^m}{\mu^m m!} \left( \frac{1}{1-\rho} \right)} = \frac{1}{\sum_{n=0}^{m-1} \frac{(\rho m)^n}{n!} + \frac{(\rho m)^m}{m!} \left( \frac{1}{1-\rho} \right)}$$

ou seja, para a série  $\sum_{n=1}^{\infty} K_n$  convergir, precisamos ter  $\rho = \frac{\lambda}{m\mu} < 1$ . Esta é a condição para que o sistema  $M/M/m/GD/\infty/\infty$  atinja um equilíbrio. Note que, fazendo  $m = 1$ , as expressões anteriores se reduzem para o caso  $M/M/1/GD/\infty/\infty$ .

O número médio de usuários em fila no sistema é definido por (após alguma álgebra):

$$\begin{aligned} E(L_q) &= \sum_{n=m}^{\infty} (n-m) P_n = \sum_{n'=0}^{\infty} n' P_{n'+m} = \sum_{n'=0}^{\infty} n' \frac{\lambda^{n'+m}}{\mu^{n'+m} m! m^{(n'+m)-m}} P_0 \\ &= \frac{\lambda^{m+1}}{\mu^{m+1} m! m} P_0 \sum_{n'=0}^{\infty} n' \frac{\lambda^{n'-1}}{\mu^{n'-1} m^{n'-1}} = \frac{\lambda^{m+1}}{\mu^{m+1} m! m} P_0 \sum_{n'=0}^{\infty} n' \rho^{n'-1} = \frac{\lambda^{m+1}}{\mu^{m+1} m! m} P_0 \frac{d}{d\rho} \sum_{n'=0}^{\infty} \rho^{n'} \\ &= \frac{\lambda^{m+1}}{\mu^{m+1} m! m} P_0 \frac{d}{d\rho} \left( \frac{1}{1-\rho} \right) = \frac{\lambda^{m+1}}{\mu^{m+1} m! m} P_0 \frac{1}{(1-\rho)^2} = \frac{(\rho m)^m \rho}{m!(1-\rho)^2} P_0 \end{aligned} \quad (7.8)$$

As demais medidas de desempenho  $E(L_s)$ ,  $E(L)$ ,  $E(W)$ ,  $E(W_q)$  podem ser obtidas usando a fórmula de Little (7.1) e as equações (7.2) – (7.4) da Seção 7.2. Por exemplo,

$$E(L) = E(L_s) + E(L_q) = \rho m + \frac{(\rho m)^m \rho}{m!(1-\rho)^2} P_0 \quad (7.9)$$

**Exemplo 7.9** No Exemplo 7.4 do *drive-in* de um restaurante *fast-food*, como mudam as medidas de desempenho se duplicarmos o dispositivo de serviço, ou seja, os clientes da fila do *drive-in* forem atendidos por dois servidores, em vez de um? Note que este sistema pode ser visto como um sistema de fila única e múltiplos servidores em paralelo (tipo (ii) da Figura 7.3) (não confundir com o sistema composto de dois *drive-in*).

Podemos representar o sistema por um modelo  $M/M/2$ . Lembrando que  $\lambda = 10$  carros por hora,  $E(S) = 4$  minutos e  $\mu = \frac{1}{E(S)} = \frac{60}{4} = 15$  carros por hora, segue-se que a utilização média do sistema é:  $\rho = \frac{\lambda}{m\mu} = \frac{10}{2(15)} \approx 0,33$ .

(i) A probabilidade de o sistema estar vazio é:  $P_0 = \frac{1}{\sum_{n=0}^{m-1} \frac{(\rho m)^n}{n!} + \frac{(\rho m)^m}{m!} \left( \frac{1}{1-\rho} \right)} = 0,50$

(ii) O número médio de carros no sistema é:  $E(L) = \rho m + \frac{(\rho m)^m \rho}{m!(1-\rho)^2} P_0 = 0,75$ . O número médio de carros em serviço e na fila é:  $E(L_s) = \rho m \approx 0,67$  e  $E(L_q) = E(L) - E(L_s) \approx 0,08$ , respectivamente.

(iii) O tempo médio de permanência no sistema é:  $E(W) = \frac{E(L)}{\lambda} = 0,075$  hora (ou 4,5 minutos). O tempo médio de espera em fila é:  $E(W_q) = \frac{E(L_q)}{\lambda} \approx 0,008$  hora (ou 0,5 minuto). Note que  $E(W) = E(S) + E(W_q)$ .

(iv) A probabilidade de haver 2 ou mais carros no *drive-in* é:

$$P(L \geq 2) = 1 - P_0 - P_1 = 1 - P_0 - \frac{\rho^m m^m}{m!} P_0 \approx 0,39$$

A Tabela 7.3 a seguir resume os resultados obtidos para o sistema do *drive-in* com dois servidores (modelo  $M/M/2$ ), comparados com os sistemas anteriores com um *drive-in*, superservidor e dois *drive-in* (modelos  $M/M/1$ ) dos Exemplos 7.4 e 7.5.

Note que as medidas de desempenho  $E(L_q)$  e  $E(W_q)$ , relacionadas somente à fila, melhoram em relação aos sistemas com superservidor e dois *drive-in*; por outro lado, as medidas  $E(L)$  e  $E(W)$ , relacionadas à fila e ao serviço, pioram em relação ao sistema com superservidor (porque o tempo médio de serviço  $E(S)$  não foi alterado), mas  $E(W)$  ainda é melhor que o sistema com dois *drive-in*. Esta é uma das razões de diversos sistemas de serviço com vários servidores, tais como agências bancárias e de correios, adotarem fila única para todos os servidores, em vez de uma fila para cada servidor.

Esses resultados também mostram que aproximar um sistema  $M/M/m$  (no caso com  $m = 2$ ) com taxa de serviço  $\mu$ , por um sistema  $M/M/1$  com taxa de serviço equivalente  $\mu' = m\mu$  (no caso com  $\mu' = 2\mu$ ) em geral não produz bons resultados, porque as medidas da filas  $E(L_q)$  e  $E(W_q)$  são superestimadas. ■

Todos os resultados anteriores para o sistema  $M/M/m/GD/\infty/\infty$  são independentes da disciplina da fila. No entanto, assim como no caso com um único servidor, a distribuição de probabilidade

**Tabela 7.3**  
Resultados adicionais obtidos para o sistema do *drive-in*.

Sistema	$\lambda$	$\mu$	$m$	$\rho$	$P_0$	$E(L)$	$E(L_q)$	$E(W)$	$E(W_q)$	$P(L \geq 2)$
1 <i>drive-in</i>	10	15	1	0,67	0,33	2	1,33	12	8	0,44
superservidor	10	30	1	0,33	0,67	0,5	0,17	3	1	0,11
2 <i>drive-in</i>	5	15	1	0,33	0,67	0,5	0,17	6	2	0,11
2 servidores	10	15	2	0,33	0,50	0,75	0,08	4,5	0,5	0,39

dade  $f_W(w)$  do tempo de permanência no sistema  $W$  depende da disciplina da fila. Se a disciplina for *FCFS* (isto é, sistema  $M/M/m/FCFS/\infty/\infty$ ), pode-se mostrar que:<sup>19</sup>

$$P(W > w) = e^{-\mu w} \left( 1 + \frac{(\rho m)^m (1 - e^{-\mu w(m-1-\rho m)})}{m!(1-\rho)(m-1-\rho m)} P_0 \right) \quad w \geq 0$$

**Exemplo 7.10** No Exemplo 7.9 do sistema de *drive-in* com dois servidores do restaurante de *fast-food*, considerando que os carros em fila no *drive-in* são atendidos com disciplina de fila *FCFS*, qual a probabilidade de um cliente permanecer mais de 15 minutos no sistema com dois servidores?

Conforme a Tabela 7.3, o tempo médio de permanência no sistema é  $E(W) = 4,5$  minutos. A probabilidade de o cliente permanecer mais de 15 minutos (0,25 hora) no sistema é:

$$P(W > 0,25) = e^{-0,25\mu} \left( 1 + \frac{(\rho m)^m (1 - e^{-0,25\mu(m-1-\rho m)})}{m!(1-\rho)(m-1-\rho m)} P_0 \right) \approx 0,0319$$

■

### 7.5.2 Modelo de fila $M/M/\infty/GD/\infty/\infty$ – infinitos servidores (auto-serviço)

O modelo de fila  $M/M/\infty/GD/\infty/\infty$  (ou simplesmente  $M/M/\infty$ ) é um caso limite do modelo  $M/M/m/GD/\infty/\infty$ , quando  $m \rightarrow \infty$ . O número de servidores é considerado ilimitado porque o usuário também é o servidor, daí este modelo ser chamado de modelo de auto-serviço (*self-service model*). Desta forma, não há formação de filas de espera no sistema.

Por exemplo, consumidores nas gôndolas de um supermercado que retiram produtos e os colocam nos seus carrinhos de compra. O tempo de serviço é o tempo decorrido entre a chegada do consumidor nas gôndolas (com carrinho vazio) e sua partida para os caixas (com carrinho cheio). Ou seja, o consumidor não espera em fila para algum servidor carregar seu carrinho. O estado do sistema é o número de consumidores nas gôndolas. Outro exemplo é a entrada de empresas em um mercado aberto. O tempo de serviço é o período que a empresa permanece no mercado, e o estado do sistema é o número de empresas atuando no mercado. Note que as empresas não esperam em fila para entrar no mercado.

Neste modelo, temos:

$$\begin{aligned}\lambda_n &= \lambda & n &= 0, 1, 2, \dots \\ \mu_n &= n\mu & n &= 1, 2, 3, \dots\end{aligned}$$

ou seja, a taxa  $\mu_n$  varia com o estado  $n$  do sistema. A distribuição de equilíbrio deste modelo é dada por (equação (7.5)):

$$P_n = K_n P_0 = \frac{\lambda^n}{(1\mu)(2\mu)(3\mu)\dots(n\mu)} P_0 = \frac{\lambda^n}{\mu^n n!} P_0 \quad n = 0, 1, 2, \dots$$

Usando a equação (7.6), o valor de  $P_0$  é definido por (lembrando que  $\sum_{n=0}^{\infty} \frac{x^n}{n!} = e^x$ ):

$$P_0 = \frac{1}{1 + \sum_{n=1}^{\infty} K_n} = \frac{1}{\sum_{n=0}^{\infty} \frac{\lambda^n}{\mu^n n!}} = \frac{1}{e^{\frac{\lambda}{\mu}}} = e^{-\frac{\lambda}{\mu}}$$

ou seja, a série  $\sum_{n=1}^{\infty} K_n$  sempre converge. Substituindo  $P_0$  em  $P_n$ , obtemos uma distribuição de Poisson com taxa  $\frac{\lambda}{\mu}$ :

19 Ver, por exemplo, Kleinrock (1975) e Gross e Harris (1985).

$$P_n = \frac{\lambda^n}{\mu^n n!} P_0 = \frac{\left(\frac{\lambda}{\mu}\right)^n e^{-\frac{\lambda}{\mu}}}{n!} \quad n = 0, 1, 2, \dots$$

Como  $L$  tem distribuição de Poisson, segue-se que  $E(L) = \frac{\lambda}{\mu}$  (lembre-se que a média de uma Poisson é igual à sua taxa) e  $E(W) = \frac{E(L)}{\lambda} = \frac{1}{\mu}$  (fórmula de Little). Além disso, como nunca são formadas filas no sistema (com infinitos servidores), segue-se que  $E(L_q) = E(W_q) = 0$ , portanto,  $E(L_s) = E(L) = \frac{\lambda}{\mu}$  e  $E(S) = E(W) = \frac{1}{\mu}$  (conforme (7.3) e (7.4)), como era esperado. Na prática, o modelo  $M/M/\infty/GD/\infty/\infty$  pode ser uma boa aproximação para tratar situações em que o sistema tem muitos servidores em paralelo, e a probabilidade de todos os servidores estarem ocupados (simultaneamente) é pequena. Em outras palavras: (i) o número de servidores  $m$  é suficientemente grande; e (ii) a carga de trabalho (ou intensidade de tráfego)  $\lambda E(S) = \frac{\lambda}{\mu}$  é suficientemente pequena, de modo que quase nunca são formadas filas.

Uma observação importante é que os resultados deste modelo também são válidos para situações em que os tempos de serviço não são exponencialmente distribuídos, ou seja, sistemas  $M/G/\infty/GD/\infty/\infty$ , apresentados na Seção 7.7. Outra observação é que os resultados do sistema  $M/M/\infty/GD/\infty/\infty$  também são válidos para um sistema  $M/M/1/GD/\infty/\infty$  com chegadas desencorajadas com o estado  $n$  do sistema, isto é, se:

$$\begin{aligned} \lambda_n &= \frac{\lambda}{n+1} & n = 0, 1, 2, \dots \\ \mu_n &= \mu & n = 1, 2, 3, \dots \end{aligned}$$

Ou seja, um sistema de filas com servidores responsivos do tipo  $M/M/\infty/GD/\infty/\infty$  se comporta exatamente como um sistema de filas  $M/M/1/GD/\infty/\infty$  com chegadas desencorajadas conforme anteriormente.<sup>20</sup>

### 7.5.3 Modelo de fila $M/M/m/GD/K/\infty$ (com $m \leq K$ ) – múltiplos servidores com capacidade limitada

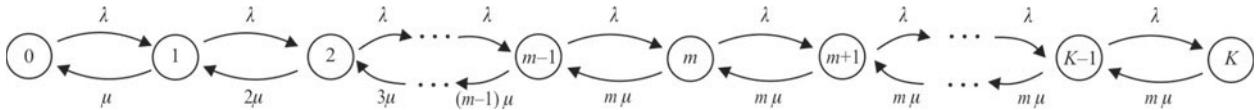
O sistema de fila  $M/M/m/GD/K/\infty$  (ou simplesmente  $M/M/m/K$ ) difere do sistema  $M/M/m/GD/\infty/\infty$  analisado anteriormente apenas pela limitação  $K$  do número de usuários presentes no sistema (em fila e serviço), o que resulta que o tamanho máximo da fila é  $K - m$  (estamos supondo  $K \geq m$ ). Por exemplo, certos sistemas urbanos de atendimento médico de urgência podem ser modelados como sistemas  $M/M/m/K$ , onde as chamadas de urgência são os usuários, as  $m$  ambulâncias são os servidores, e a disciplina de atendimento limita o tamanho da fila de espera de chamados em  $m$ , ou seja, igual ao número de ambulâncias (portanto,  $K = 2m$ ).<sup>21</sup> Note que o sistema  $M/M/m/GD/K/\infty$  pode ser representado por um modelo de nascimento e morte com:

$$\begin{aligned} \lambda_n &= \begin{cases} \lambda & n = 0, 1, 2, \dots, K-1 \\ 0 & n = K, K+1, \dots \end{cases} \\ \mu_n &= \begin{cases} n\mu & n = 1, 2, 3, \dots, m-1 \\ m\mu & n = m, m+1, \dots, K \end{cases} \end{aligned}$$

A Figura 7.17 ilustra o diagrama de transição de estados para o sistema  $M/M/m/GD/K/\infty$ .

<sup>20</sup> Para mais detalhes, ver, por exemplo, Kleinrock (1975) e Gross e Harris (1985).

<sup>21</sup> Para estudos sobre a aplicação de modelos de filas em sistemas de atendimento médico de urgência, ver, por exemplo, Galvão *et al.* (2005) e Takeda *et al.* (2006).



**Figura 7.17** Diagrama de transição de estados do sistema  $M/M/m/GD/K/\infty$ .

A distribuição de equilíbrio é dada por (equação (7.5)):

$$P_n = \begin{cases} K_n P_0 = \frac{\lambda^n}{(1\mu)(2\mu)\dots(n\mu)} P_0 = \frac{\lambda^n}{\mu^n n!} P_0 = \frac{(\rho m)^n}{n!} P_0 & n = 1, 2, \dots, m-1 \\ K_n P_0 = \frac{\lambda^n}{(1\mu)(2\mu)\dots(m\mu)(m\mu)^{n-m}} P_0 = \frac{\lambda^n}{\mu^n m! m^{n-m}} P = \frac{\rho^n m^m}{m!} P_0 & n = m, m+1, \dots, K \end{cases}$$

Seguindo os mesmos passos para o caso do sistema  $M/M/m/GD/\infty/\infty$  analisado anteriormente, o valor de  $P_0$  é definido por (após alguma álgebra):

$$P_0 = \begin{cases} \frac{1}{1 + \sum_{n=1}^{\infty} K_n} = \frac{1}{\sum_{n=0}^{m-1} \frac{\lambda^n}{\mu^n n!} + \frac{\lambda^m}{\mu^m m!} \sum_{n=m}^K \rho^{n-m}} = \frac{1}{\sum_{n=0}^{m-1} \frac{(\rho m)^n}{n!} + \frac{(\rho m)^m}{m!} \left( \frac{1 - \rho^{K-m+1}}{1 - \rho} \right)} & \rho \neq 1 \\ \frac{1}{1 + \sum_{n=1}^{\infty} K_n} = \frac{1}{\sum_{n=0}^{m-1} \frac{(\rho m)^n}{n!} + \frac{(\rho m)^m}{m!} (K-m+1)} & \rho = 1 \end{cases}$$

Portanto, não precisamos mais da condição  $\rho < 1$  para que esta série converja.

Pode-se mostrar que o número médio de usuários em fila no sistema é definido por:

$$E(L_q) = \sum_{n=m}^K (n-m) P_n = \frac{(\rho m)^m \rho}{m!(1-\rho)^2} [1 - \rho^{K-m+1} - (K-m+1)\rho^{K-m}(1-\rho)] P_0 \quad \text{se } \rho \neq 1$$

$$E(L_q) = \sum_{n=m}^K (n-m) P_n = \frac{(\rho m)^m (K-m)(K-m+1)}{2m!} P_0 \quad \text{se } \rho = 1$$

Note que, se  $K \rightarrow \infty$  e  $\rho < 1$ , então as expressões de  $P_0$ ,  $P_n$  e  $E(L_q)$  anteriores se reduzem às do modelo  $M/M/m/GD/\infty/\infty$ . Além disso, fixando  $m = 1$ , obtemos os resultados do modelo  $M/M/1/GD/K/\infty$ .

As demais medidas de desempenho  $E(L_s)$ ,  $E(L)$ ,  $E(W)$ ,  $E(W_q)$  podem ser obtidas usando a fórmula de Little (7.1) e as equações (7.2)–(7.4) da Seção 7.2. Entretanto, assim como no sistema  $M/M/1/GD/K/\infty$ , a taxa de entrada de usuários no sistema a ser usada na fórmula de Little é (equação (7.7)):

$$\bar{\lambda} = \sum_{n=0}^{\infty} \lambda_n P_n = \lambda \sum_{n=0}^{K-1} P_n + 0 P_K = \lambda (1 - P_K)$$

onde  $P_K$  é a probabilidade de o sistema estar cheio, ou seja, a probabilidade de perda de usuários do sistema. Por exemplo,

$$E(L) = E(L_s) + E(L_q) = \frac{\bar{\lambda}}{\mu} + E(L_q)$$

Como  $\bar{\lambda} < \lambda$ , segue-se que o fator de utilização  $\rho$  não corresponde à utilização média do sistema (proporção de servidores ocupados), dada por  $\frac{E(L_s)}{m} = \frac{\bar{\lambda}}{m\mu}$ .

**Exemplo 7.11** No Exemplo 7.9 do sistema de *drive-in* com dois servidores do restaurante de *fast-food*, considere que não haja espaço físico para mais de cinco carros (em fila e serviço), caso contrário a fila invade a rua do restaurante e os clientes, ao chegarem, desistem de entrar na fila. (i) Qual a taxa de perda de clientes e a utilização média do sistema? (ii) Como mudam o número médio de carros na fila e o tempo médio de espera na fila?

(i) Note que temos um sistema  $M/M/2/GD/5/\infty$ . Lembrando que  $\rho = 0,33 \neq 1$ , temos:

$$P_0 = \frac{1}{\sum_{n=0}^{m-1} \frac{(\rho m)^n}{n!} + \frac{(\rho m)^m}{m!} \left( \frac{1 - \rho^{5-m+1}}{1 - \rho} \right)} \approx 0,51$$

e a probabilidade de perda de clientes é:

$$P_5 = \frac{\rho^5 m^m}{m!} P_0 \approx 0,004$$

Lembre-se do Exemplo 7.7 que, no sistema com um *drive-in* com limitação de capacidade ( $K = 5$ ), este valor era  $P_5 = 0,05$ .

Logo, a taxa de perda é bem pequena:  $\lambda P_5 = 10(0,004) = 0,04$  usuário por hora (no sistema com um *drive-in* era de 0,5 usuário por hora). A utilização média do sistema (proporção de servidores ocupados) é:

$$\frac{E(L_s)}{m} = \frac{\bar{\lambda}}{m\mu} = \frac{\lambda(1 - P_5)}{m\mu} \approx 0,33$$

(ii) Lembre-se do Exemplo 7.9 que, para o sistema com dois servidores sem limitação de capacidade, temos  $E(L_q) \approx 0,08$  carro e  $E(W_q) \approx 0,5$  minuto. Com a limitação de capacidade ( $K = 5$ ), estes valores mudam para:

$$E(L_q) = \sum_{n=m}^5 (n-m)P_n = \frac{(\rho m)^m \rho}{m!(1-\rho)^2} [1 - \rho^{5-m+1} - (5-m+1)\rho^{5-m}(1-\rho)]P_0 = 0,07$$

$$E(W_q) = \frac{E(L_q)}{\bar{\lambda}} = 0,008 \text{ hora (ou } 0,45 \text{ minuto)}$$

A Tabela 7.4 a seguir compara os resultados dos sistemas com um e dois servidores, sem e com limitação de capacidade ( $K = 5$ ) (Exemplos 7.4, 7.7 e 7.8). ■

**Tabela 7.4**  
Resultados para sistemas com um e dois servidores, sem e com limitação de capacidade.

Sistema	$\lambda$	$\mu$	$m$	$\rho$	Perda	$E(L_s)/m$	$E(L_q)$	$E(W_q)$
$M/M/1/GD/\infty/\infty$	10	15	1	0,67	0	0,67	1,33	8
$M/M/1/GD/5/\infty$	10	15	1	0,67	0,5	0,63	0,79	4,97
$M/M/2/GD/\infty/\infty$	10	15	2	0,33	0	0,33	0,08	0,5
$M/M/2/GD/5/\infty$	10	15	2	0,33	0,04	0,332	0,07	0,45

### 7.5.4 Modelo de fila $M/M/m/GD/m/\infty$ – múltiplos servidores sem fila de espera

Um caso especial importante do modelo  $M/M/m/GD/K/\infty$  é quando  $K = m$ , isto é, o sistema não permite a formação de filas (sistema sem sala de espera). Por exemplo, em uma central de chamadas com  $m$  linhas telefônicas (servidores), se todas as linhas estiverem ocupadas, uma nova chamada (usuário) não consegue ser atendida pelo sistema. Ou em um estacionamento com  $m$  vagas (servidores), se todas as vagas estiverem ocupadas, um carro (usuário) que chega não pode estacionar. Este modelo foi inicialmente estudado pelo engenheiro dinamarquês A. Erlang (considerado o pai da teoria de filas) em 1917, para análise de sistemas telefônicos na Dinamarca.<sup>22</sup>

Substituindo  $K = m$  nas expressões de  $P_0$  e  $P_n$  do modelo  $M/M/m/GD/K/\infty$  anterior, obtemos:

$$P_0 = \frac{1}{1 + \sum_{n=1}^{\infty} K_n} = \frac{1}{\sum_{n=0}^m \frac{\lambda^n}{\mu^n n!}} = \frac{1}{\sum_{n=0}^m \frac{(\rho m)^n}{n!}}$$

$$P_n = K_n P_0 = \frac{\lambda^n}{\mu^n n!} P_0 = \frac{\frac{\lambda^n}{\mu^n n!}}{\sum_{i=0}^m \frac{\lambda^i}{\mu^i i!}} = \frac{\frac{(\rho m)^n}{n!}}{\sum_{i=0}^m \frac{(\rho m)^i}{i!}} \quad n = 0, 1, 2, \dots, m$$

Em particular,  $P_m$  é a probabilidade de o sistema estar cheio, também conhecida como fórmula de perda de Erlang. Fixando  $K = m$  na expressão de  $E(L_q)$  do modelo  $M/M/m/GD/K/\infty$ , obtemos

$$E(L_q) = 0, \text{ portanto, } E(L) = E(L_s) = \frac{\bar{\lambda}}{\mu} = \frac{\lambda(1 - P_m)}{\mu}.$$

Note que o modelo  $M/M/\infty/GD/\infty/\infty$  anterior pode ser visto como um caso especial do modelo  $M/M/m/GD/m/\infty$ , quando  $m \rightarrow \infty$ . De fato, no limite  $m \rightarrow \infty$ , as expressões de  $P_0$  e  $P_n$  anteriores se reduzem às do sistema  $M/M/\infty/GD/\infty/\infty$ , ou seja,  $P_n$  é uma distribuição de Poisson com taxa  $\frac{\lambda}{\mu}$  e, assim,  $E(L) = E(L_s) = \frac{\lambda}{\mu}$ .

Um resultado interessante é que a fórmula de perda de Erlang também é válida se a distribuição do tempo de serviço não for exponencial, ou seja, para sistemas  $M/G/m/GD/m/\infty$ , apresentados na Seção 7.7.

### 7.5.5 Modelo de fila $M/M/m/GD/K/K$ (com $m \leq K$ ) – múltiplos servidores com capacidade limitada e população finita (manutenção de máquinas)

Outro caso especial importante do modelo  $M/M/m/GD/K/\infty$  é quando  $N = K$ , isto é, o tamanho da população  $N$  é igual à capacidade do sistema  $K$ , o que resulta que o tamanho máximo da fila é  $K - m$  usuários (estamos supondo  $K \geq m$ ) e que podemos ter todos os usuários no sistema. Este modelo é útil para descrever, por exemplo, sistemas de manutenção de máquinas com  $m$  mecânicos (servidores) e  $K$  máquinas (usuários). Quando uma máquina quebra, um dos mecânicos disponíveis é chamado para consertá-la. Caso não haja mecânico disponível, a máquina quebrada entra na fila de espera. Admite-se que o intervalo de tempo entre quebras de máquinas e o tempo de reparo de uma máquina (utilizado pelo mecânico) são exponencialmente distribuídos com taxas  $\lambda$  e  $\mu$ , respectivamente. Por isso, este modelo também é chamado de modelo de manutenção de máquinas, podendo ser visto como um modelo de nascimento e morte com:

<sup>22</sup> Ver Erlang (1917).

$$\lambda_n = \begin{cases} (K-n)\lambda & n = 0, 1, 2, \dots, K-1 \\ 0 & n = K, K+1, \dots \end{cases}$$

$$\mu_n = \begin{cases} n\mu & n = 1, 2, 3, \dots, m-1 \\ m\mu & n = m, m+1, \dots, K \end{cases}$$

ou seja, ambas as taxas  $\lambda_n$  e  $\mu_n$  variam com o estado  $n$  do sistema. Note que, quando todos os usuários estão no sistema ( $n = K$ ), a taxa de chegada  $\lambda_K$  é nula e, quando o sistema está vazio ( $n = 0$ ), a taxa de chegada é  $\lambda_0 = K\lambda$ . A Figura 7.18 ilustra o diagrama de transição de estados para o sistema  $M/M/m/GD/K/K$ .

A distribuição de equilíbrio é dada por (equação (7.5)):

$$P_n = \begin{cases} K_n P_0 = \frac{(K\lambda)((K-1)\lambda)((K-2)\lambda)\dots((K-n+1)\lambda)}{(1\mu)(2\mu)\dots(n\mu)} P_0 = \frac{\lambda^n}{(K-n)!} \frac{K!}{\mu^n n!} P_0 & n = 1, 2, \dots, m-1 \\ K_n P_0 = \frac{(K\lambda)((K-1)\lambda)((K-2)\lambda)\dots((K-n+1)\lambda)}{(1\mu)(2\mu)\dots(m\mu)(m\mu)^{n-m}} P_0 = \frac{\lambda^n}{(K-n)!} \frac{K!}{\mu^n m! m^{n-m}} P_0 & n = m, m+1, \dots, K \end{cases}$$

Lembrando que  $C_n^K = \frac{K!}{n!(K-n)!}$ , podemos reescrever  $P_n$  como:

$$P_n = \begin{cases} K_n P_0 = \frac{\lambda^n K!}{\mu^n n! (K-n)!} P_0 = C_n^K \frac{\lambda^n}{\mu^n} P_0 = C_n^K (\rho m)^n P_0 & n = 1, 2, \dots, m-1 \\ K_n P_0 = \frac{\lambda^n K!}{\mu^n m! (K-n)! m^{n-m}} P_0 = C_n^K \frac{\lambda^n n!}{\mu^n m! m^{n-m}} P_0 = C_n^K \frac{\rho^n n! m^m}{m!} P_0 & n = m, m+1, \dots, K \end{cases}$$

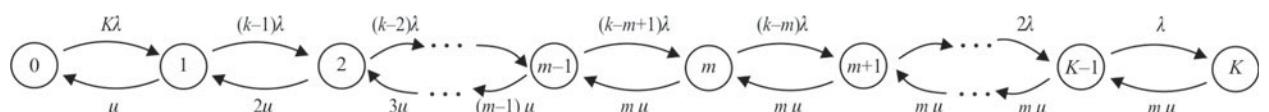
Seguindo os mesmos passos para o caso do modelo  $M/M/m/GD/K/\infty$  analisado anteriormente, o valor de  $P_0$  é definido por (após alguma álgebra) (equação (7.6)):

$$P_0 = \frac{1}{1 + \sum_{n=1}^{\infty} K_n} = \frac{1}{\sum_{n=0}^{m-1} C_n^K (\rho m)^n + \sum_{n=m}^K C_n^K \frac{\rho^n n! m^m}{m!}}$$

Assim como no modelo  $M/M/m/GD/K/\infty$ , também não precisamos da condição  $\rho < 1$  para que a série converja e o modelo  $M/M/m/GD/K/K$  atinja um equilíbrio.

O número médio de usuários no sistema pode ser encontrado com:

$$E(L) = \sum_{n=0}^K n P_n$$



**Figura 7.18** Diagrama de transição de estados do sistema  $M/M/m/GD/K/K$ .

e as demais medidas de desempenho  $E(L_s)$ ,  $E(L_q)$ ,  $E(W)$ ,  $E(W_q)$  podem ser obtidas usando a fórmula de Little (7.1) e as equações (7.2)–(7.4) da Seção 7.2. Note que a taxa de entrada de usuários no sistema a ser usada na fórmula de Little deve ser (equação (7.7)):

$$\bar{\lambda} = \sum_{n=0}^{\infty} \lambda_n P_n = \sum_{n=0}^K (K-n)\lambda P_n = K\lambda \sum_{n=0}^K P_n - \lambda \sum_{n=0}^K nP_n = K\lambda - \lambda E(L) = \lambda(K - E(L))$$

onde  $K - E(L)$  corresponde ao número médio de usuários fora do sistema. No caso de um sistema de manutenção de máquinas, esse valor corresponde ao número médio de máquinas em operação, portanto, a produtividade das máquinas pode ser estimada por:  $\frac{K - E(L)}{K}$ .

**Exemplo 7.12** Uma estação de trabalho opera com três máquinas. Cada máquina quebra em média a cada 5 horas, e o tempo médio de reparo é de 2 horas (por mecânico). Se os intervalos de tempo entre quebras de máquinas e os tempos de reparo forem exponencialmente distribuídos, analise o desempenho deste sistema de manutenção se tivermos 1 e 2 mecânicos para reparar as máquinas.

Podemos representar o sistema por um modelo  $M/M/m/GD/3/3$  com  $E(X) = 5$  horas (isto é,  $\lambda = \frac{1}{5} = 0,2$  quebra de máquinas por hora) e  $\mu = \frac{1}{2} = 0,5$  reparo por hora (isto é,  $E(S) = 2$  horas por mecânico). Temos que  $\rho = \frac{\lambda}{m\mu}$  e:

$$P_0 = \frac{1}{\sum_{n=0}^{m-1} C_n^3 (\rho m)^n + \sum_{n=m}^3 C_n^3 \frac{\rho^n n! m^m}{m!}} \quad P_n = \begin{cases} C_n^3 (\rho m)^n P_0 & n = 1, 2, \dots, m-1 \\ C_n^3 \frac{\rho^n n! m^m}{m!} P_0 & n = m, m+1, \dots, 3 \end{cases}$$

o que resulta nos valores da Tabela 7.5 para os sistemas com  $m = 1$  e  $m = 2$  mecânicos. Os demais valores da tabela são obtidos com  $E(L) = \sum_{n=0}^3 nP_n$ ,  $\bar{\lambda} = \lambda(3 - E(L))$  e  $E(W) = \frac{E(L)}{\bar{\lambda}}$ .

Note que a produtividade das máquinas é:  $\frac{3 - E(L)}{3}$ , resultando em 59,8% no sistema com  $m = 1$  mecânico, e 70,6% no sistema com  $m = 2$  mecânicos, ou seja, a produtividade aumenta em 10,8% com o emprego de um segundo mecânico. ■

#### Modelos de fila $M/M/\infty/GD/\infty/N$ , $M/M/m/GD/m/N$ e $M/M/m/GD/K/N$

Os modelos  $M/M/\infty/GD/\infty/N$ ,  $M/M/m/GD/m/N$  e  $M/M/m/GD/K/N$  podem ser analisados de maneira similar aos anteriores. Os resultados do sistema anterior  $M/M/m/GD/K/K$  com  $m \leq K$  podem ser facilmente particularizados para tratar um sistema  $M/M/m/GD/m/N$  (fazendo  $K = m$ ), ou estendidos para tratar um sistema  $M/M/m/GD/K/N$  com  $m \leq K \leq N$ , ou seja, quando a população

**Tabela 7.5**  
Resultados obtidos para sistemas com um e dois mecânicos.

Sistema	$\lambda$	$\mu$	$m$	$\rho$	$P_0$	$P_1$	$P_2$	$P_3$	$E(L)$	$\bar{\lambda}$	$E(W)$
$M/M/1/GD/3/3$	0,2	0,5	1	0,4	0,28	0,34	0,27	0,11	1,21	0,36	3,36
$M/M/2/GD/3/3$	0,2	0,5	2	0,2	0,36	0,43	0,17	0,03	0,88	0,42	2,08

ainda é finita, mas pode ser maior que a capacidade do sistema. Tal modelo também é chamado modelo de Engset.<sup>23</sup>

Por exemplo, em provedores de acesso à Internet, o sistema de acesso discado (*dial-up*) pode ser modelado como um sistema  $M/M/m/GD/m/N$ , em que as chamadas telefônicas dos clientes são os usuários; os  $m$  canais de atendimento são os servidores; e o provedor tem apenas  $N$  clientes cadastrados no sistema. Se todos os canais de atendimento estiverem ocupados, então uma nova chamada não consegue acesso ao sistema, isto é, o sistema não tem fila de espera (isto é,  $K = m$ ).<sup>24</sup>

### 7.5.6 Modelos de decisão de filas – múltiplos servidores

Em modelos de filas com múltiplos servidores, o nível de serviço aos usuários depende fundamentalmente da taxa de serviço  $\mu$  de cada servidor e do número  $m$  de servidores (capacidade do sistema). Melhores níveis de serviço em geral são obtidos com menores tempos de espera em fila ou menores perdas de usuários no sistema, que, por sua vez, exigem maiores investimentos em capacidade no sistema. Assim como no caso de um servidor (Seção 7.4), os modelos de decisão com múltiplos servidores também procuram balancear o conflito entre os custos de oferecer um nível de serviço no sistema e os custos dos atrasos sofridos pelos usuários.

Nestes modelos, uma decisão importante é a escolha entre diversas alternativas de capacidade  $(\mu_1, m_1), (\mu_2, m_2), \dots$ , para possível alocação no sistema. Ou, no caso em que  $\mu$  é constante, a escolha entre diversas alternativas  $m_1, m_2, \dots$ , para o número de servidores no sistema. Para cada alternativa, podemos estimar medidas de desempenho de interesse como  $E(L)$ ,  $E(W)$  etc. usando os modelos de filas descritos nesta seção e, em seguida, avaliar os custos envolvidos em função do número de servidores e dos valores obtidos dessas medidas. A alternativa de custo mínimo é, então, escolhida. Observe o exemplo a seguir.

**Exemplo 7.13**<sup>25</sup> Uma empresa possui diversas máquinas para produzir certos produtos. Essas máquinas eventualmente quebram e necessitam de reparo. As solicitações de reparo ocorrem conforme uma distribuição de Poisson, com taxa  $\lambda = 17,5$  solicitações por semana. Os tempos de reparo são exponenciais, com taxa  $\mu = 10$  reparos por semana, para cada mecânico da equipe de manutenção da empresa. Se cada mecânico custa à empresa  $c = 12$  unidades monetárias por semana, e o custo de oportunidade de cada máquina parada é estimado em  $d = 50$  unidades monetárias por semana, determine o número ótimo  $m$  de mecânicos.

Esta situação pode ser representada por um modelo de filas  $M/M/m$ . O custo total por semana é dado por:

$$\text{Custo por semana} = cm + dE(L_q) = 12m + 50E(L_q)$$

em que  $E(L_q)$  é o número médio de máquinas quebradas (em fila) por semana, definido pela equação (7.8). Note que esta equação é função do número de mecânicos  $m$ .

O fator de utilização resulta em:  $\rho = \frac{\lambda}{m\mu} = \frac{17,5}{10m}$ . Para que  $\rho < 1$ , temos que  $m$  precisa ser maior ou igual a dois mecânicos. A Tabela 7.6 a seguir apresenta os valores de  $E(L_q)$  e do custo total por semana para diferentes valores de  $m$ .

Note que a solução de custo semanal mínimo resulta em  $m = 4$  mecânicos.<sup>26</sup>

<sup>23</sup> Para detalhes destes modelos, ver, por exemplo, Kleinrock (1975) e Gross e Harris (1985).

<sup>24</sup> Para um estudo da aplicação de modelos de filas em provedores brasileiros de acesso à Internet, ver, por exemplo, Fontanella e Morabito (2002).

<sup>25</sup> Este exemplo foi inspirado em um exemplo em Taha (2003).

<sup>26</sup> Pode-se mostrar que a função  $E(L_q)$  é convexa em  $m$ ; ver, por exemplo, Boxma *et al.* (1990).

**Tabela 7.6**  
**Valores de  $E(L_q)$  e do custo total por semana.**

<i>m</i>	$E(L_q)$	<i>Custo por semana</i>
2	7,47	397
3	2,22	147
4	1,84	140
5	1,77	148
6	1,75	160

Em alguns casos, podemos também utilizar modelos de otimização para determinar os valores ótimos das variáveis de decisão  $\mu$  e  $m$ , de modo semelhante ao que foi feito no Exemplo 7.8 da Seção 7.4.

## 7.6 OUTROS MODELOS DE FILAS COM DISTRIBUIÇÕES EXPONENCIAIS

Os modelos de nascimento e morte ainda podem ser aplicados para tratar diversos outros sistemas de filas, descritos por:

$$\begin{aligned} \lambda_n &= c_n \lambda & n = 0, 1, 2, \dots \\ \mu_n &= d_n \mu & n = 1, 2, 3, \dots \end{aligned}$$

Em geral, temos que  $c_0 = 1$ , refletindo que  $\lambda_0 = \lambda$  quando o sistema está vazio, e  $d_1 = 1$ , refletindo que  $\mu_1 = \mu$  quando o sistema contém apenas um usuário. Os parâmetros  $c_n$  e  $d_n$  podem ser usados para modelar diferentes situações.

### 7.6.1 Modelo com desistência na chegada

Modelos com desistência na chegada representam casos em que usuários, ao chegar e observar o estado do sistema, desistem de entrar no sistema porque a fila está muito longa. Conforme mencionado, os resultados do sistema  $M/M/\infty/GD/\infty/\infty$  analisado na Seção 7.5 também são válidos para um sistema  $M/M/1/GD/\infty/\infty$  com chegadas desencorajadas com o estado  $n$  do sistema, isto é, se  $c_n = \frac{1}{n+1}$ . Em geral, a desistência é representada por:

$$c_n = \left( \frac{1}{n+1} \right)^a \quad n = 0, 1, 2, \dots$$

onde  $a$  é um parâmetro (constante não-negativa) a ser determinado em cada caso. Note que  $a = 0$  corresponde ao caso sem desistência na chegada. Alguns autores também sugerem o uso de:  $c_n = 1 - f_n$ , em que  $f_n$  é a função de desistência a ser determinada em cada caso.

### 7.6.2 Modelo com abandono na espera ou com aceleração do servidor

Modelos com abandono na espera representam casos em que usuários, após esperar muito tempo na fila, abandonam o sistema. Estes casos podem ser modelados aumentando-se a taxa de ser-

viço do sistema em função do estado do sistema (por exemplo, número de usuários na fila). Por exemplo,  $\mu_n = d_n \mu = \mu + g_n$  (isto é,  $d_n = 1 + \frac{g_n}{\mu}$ ), em que  $g_n$  é a função de abandono a ser determinada em cada caso.

Modelos com aceleração (*speed up*) do servidor representam casos em que o servidor acelera a velocidade do serviço em função do estado do sistema. Em geral, a aceleração é representada por:

$$d_n = n^b \quad n = 1, 2, 3, \dots$$

onde  $b$  é um parâmetro (constante não-negativa) a ser determinado em cada caso. Note que  $b = 0$  corresponde ao caso sem aceleração do servidor (por exemplo, sistema  $M/M/1$ ). Alguns autores também sugerem generalizar ainda mais este modelo para representar melhor sistemas com múltiplos servidores ( $m > 1$ ), de modo que  $\mu_n$  varie com o número de usuários por servidor ( $n/m$ ), isto é:

$$d_n = \begin{cases} n, & n=1, 2, \dots, m \\ \left(\frac{n}{m}\right)^b m, & n=m+1, m+2, \dots \end{cases}$$

Note também que  $b = 0$  corresponde ao caso sem aceleração dos servidores (por exemplo, sistema  $M/M/m$ ). Embora seja praticamente impossível obter expressões em forma fechada para  $P_n$ ,  $E(L)$ ,  $E(W)$  etc., para estes sistemas, em geral é relativamente fácil resolver as equações de equilíbrio numericamente.

**Exemplo 7.14**<sup>27</sup> Durante um período de pico em uma banca de jornais, clientes chegam em média a cada 20 segundos (isto é, três clientes por minuto), de acordo com uma distribuição exponencial. A probabilidade de o cliente desistir de comprar jornal é  $n/3$ , onde  $n$  é o número de clientes presentes na banca. Se o tempo de atendimento for exponencialmente distribuído com média de 30 segundos (isto é, dois clientes por minuto), qual a taxa esperada de desistência de clientes?

Note que  $\lambda = 3$  clientes por minuto,  $\mu = 2$  clientes por minuto (portanto,  $\rho > 1$ ), e a função de desistência é  $f_n = \frac{n}{3}$ . Logo, considerando  $c_n = 1 - f_n$ , quando o sistema está vazio ( $n = 0$ ), chegam  $\lambda$  clientes por minuto no sistema, e a partir de  $n = 3$  clientes presentes no sistema, não há mais chegadas (porque  $c_n = 0$ ). Ou seja,  $\lambda_0 = 3$ ,  $\lambda_1 = 2$ ,  $\lambda_2 = 1$ ,  $\lambda_n = 0$ ,  $n = 3, 4, 5, \dots$

A distribuição de probabilidade é dada por (equação (7.5)):

$$P_1 = \frac{\lambda_0}{\mu_1} P_0 = \frac{3}{2} P_0$$

$$P_2 = \frac{\lambda_0 \lambda_1}{\mu_1 \mu_2} P_0 = \frac{3}{2} \frac{2}{2} P_0 = \frac{3}{2} P_0$$

$$P_3 = \frac{\lambda_0 \lambda_1 \lambda_2}{\mu_1 \mu_2 \mu_3} P_0 = \frac{3}{2} \frac{2}{2} \frac{1}{2} P_0 = \frac{3}{4} P_0$$

e  $P_4 = P_5 = \dots = 0$ . O valor de  $P_0$  é obtido por (equação (7.6)):

$$P_0 = \frac{1}{1 + \sum_{n=1}^{\infty} K_n} = \frac{1}{1 + (\frac{3}{2} + \frac{3}{2} + \frac{3}{4})} = \frac{4}{19}$$

<sup>27</sup> Este exemplo foi inspirado em um exemplo em Bronson e Naadimuthu (1997).

portanto,  $P_1 = \frac{6}{19}$ ,  $P_2 = \frac{6}{19}$  e  $P_3 = \frac{3}{19}$  (note que a soma é igual a 1). A taxa esperada de desistência é:

$$\begin{aligned}\sum_{n=0}^{\infty} (\lambda - \lambda_n) P_n &= (3-3)P_0 + (3-2)P_1 + (3-1)P_2 + (3-0)P_3 + \sum_{n=4}^{\infty} (3-n)P_n \\ &= 0 + 1\frac{6}{19} + 2\frac{6}{19} + 3\frac{3}{19} + 0 \approx 1,42 \text{ cliente por minuto}\end{aligned}$$

■

### *Modelos com chegadas e serviços em lote*

Os modelos de nascimento e morte ainda podem ser utilizados para tratar sistemas com chegadas de usuários em lote e/ou serviços de usuários em lote (por exemplo, chegadas e partidas de ônibus ou elevadores com diversos passageiros).<sup>28</sup>

## 7.7 MODELOS DE FILAS COM DISTRIBUIÇÕES GENÉRICAS

Sistemas de filas em que os processos de chegada e/ou serviço dos usuários não podem ser descritos por distribuições exponenciais (processos de Poisson) em geral são bem mais complexos e difíceis de analisar. Alguns casos mais simples são estudados a seguir. Casos mais complicados podem ser tratados por meio de modelos de filas mais elaborados que estão além do escopo deste livro, ou por meio de simulação discreta.<sup>29</sup>

### 7.7.1 Modelo de fila $M/G/1/GD/\infty/\infty$ – um servidor

Em diversas situações, não é razoável admitir que o tempo de serviço tem distribuição exponencial. Por exemplo, em um sistema de manufatura, a variância dos tempos de processamento de tarefas em uma máquina em geral é bem menor do que a de uma distribuição exponencial. Ou em um sistema logístico, os tempos de coleta e/ou entrega de mercadorias por um veículo (por exemplo, um entregador de pizza) em geral não são bem representados por uma distribuição exponencial, por causa dos tempos de viagem que em geral não são exponencialmente distribuídos.<sup>30</sup> Nestes casos, o modelo  $M/G/1/GD/\infty/\infty$  (ou simplesmente  $M/G/1$ ) é mais adequado.

Lembre-se que, nos modelos de nascimento e morte a partir da Seção 7.3, a probabilidade de um serviço terminar durante um intervalo  $(t, t + \Delta t)$  quando o sistema encontra-se com  $n$  usuários (com alguns ou todos em serviço) no instante  $t$  é simplesmente  $\mu_n \Delta t$ , independentemente do tempo decorrido desde o início do serviço antes de  $t$ . Ou seja, nos modelos de nascimento e morte, essa probabilidade não depende da história passada do sistema, isto é, esses modelos são markovianos, sem memória. No entanto, em um sistema  $M/G/1$ , essa probabilidade depende do tempo decorrido desde o início do serviço antes de  $t$ , porque a distribuição do tempo de serviço tem memória. Logo, o número de usuários no sistema  $n$  não é mais uma descrição completa de estado do sistema. Precisaríamos também saber o tempo decorrido desde o início do serviço, que é uma variável contínua. Essas complicações tornam a análise do sistema  $M/G/1$  bem mais complicada que a do sistema  $M/M/1$  e suas variações.

Pode-se mostrar que, em um sistema  $M/G/1$ , dados a taxa de chegada  $\lambda$ , a taxa de serviço  $\mu = \frac{1}{E(S)}$  (em que  $E(S)$  é o tempo médio de serviço), e a variância do tempo de serviço  $V(S)$ , se

28 Ver, por exemplo, o método dos estágios em Kleinrock (1975), Gross e Harris (1985) e Ross (1993).

29 Para detalhes de simulação discreta, ver, por exemplo, Law e Kelton (1991), Banks (1998) e Saliby e Funchal (1986).

30 Ver, por exemplo, Larson e Odoni (1981).

o fator de utilização  $\rho = \frac{\lambda}{m\mu} = \frac{\lambda}{\mu} < 1$ , então podemos obter o número médio de usuários do sistema com:<sup>31</sup>

$$E(L) = \rho + \frac{\rho^2 + \lambda^2 V(S)}{2(1-\rho)} \quad (7.10)$$

A expressão (7.10) é conhecida como fórmula de Pollaczek e Khintchine. O primeiro termo do lado direito dessa expressão corresponde a  $E(L_s)$ , e o segundo termo corresponde a  $E(L_q)$ . Se a variância do tempo de serviço  $V(S)$  cresce, então  $E(L_q)$  também cresce, como era esperado. Note que não precisamos saber qual a distribuição de probabilidade  $f_S(s)$  do tempo de serviço, mas apenas sua média  $E(S)$  e sua variância  $V(S)$ . Note também que, se  $f_S(s)$  for exponencial, então  $V(S) = \frac{1}{\mu^2}$  e a expressão anterior se reduz a  $E(L) = \frac{\rho}{1-\rho}$ , que corresponde ao número médio de usuários em um sistema  $M/M/1$  (Seção 7.3).

Lembre-se, da Seção 7.2, de que, para qualquer sistema de fila com um servidor, temos que  $\lambda E(S) = 1 - P_0$ , ou seja, a utilização média do sistema corresponde à proporção de tempo que o servidor fica ocupado. Como no sistema  $M/G/1$  temos que a taxa de chegada é igual à taxa de entrada (isto é,  $\lambda = \lambda$ ), segue-se que:

$$P_0 = 1 - \rho$$

As demais medidas de desempenho  $E(L_s)$ ,  $E(L_q)$ ,  $E(W)$ ,  $E(W_q)$  podem ser obtidas a partir de  $E(L)$  usando a fórmula de Little (7.1) e as equações definidas (7.2)–(7.4) na Seção 7.2. Por exemplo, o tempo de permanência no sistema é:  $E(W) = \frac{E(L)}{\lambda}$ .

**Exemplo 7.15** No Exemplo 7.4 do sistema de *drive-in* com um servidor do restaurante de *fast-food*, considere que o dispositivo de serviço anterior foi substituído por novo dispositivo totalmente automatizado, cuja distribuição do tempo de atendimento não é mais exponencial, mas determinística. Como mudam as medidas de desempenho analisadas?

Podemos representar o sistema por um modelo  $M/D/1$ . Note que  $\lambda = 10$  carros por hora,  $E(S) = 4$  minutos,  $V(S) = 0$  minuto<sup>2</sup> (porque a distribuição de serviço agora é determinística) e  $\mu = \frac{1}{E(S)} = \frac{60}{4} = 15$  carros por hora. A utilização média do sistema é:  $\rho = \frac{\lambda}{\mu} = \frac{10}{15} \approx 0,67$ . Logo, o número médio de carros no sistema é (equação (7.10)):

$$E(L) = \rho + \frac{\rho^2 + \lambda V(S)}{2(1-\rho)} = 0,67 + \frac{0,67^2}{2(1-0,67)} \approx 1,33 \text{ carro}$$

O número médio de carros em serviço e na fila é:  $E(L_s) = \rho \approx 0,67$  e  $E(L_q) = E(L) - \rho \approx 0,67$  carro. O tempo médio de permanência no sistema é:  $E(W) = \frac{E(L)}{\lambda} \approx 0,133$  hora (ou 8 minutos). O tempo médio de espera em fila é:  $E(W_q) = E(W) - E(S) \approx 0,067$  hora (ou 4 minutos). A Tabela 7.7 a seguir compara os resultados para os sistemas  $M/M/1$  (Exemplo 7.4) e  $M/D/1$ .

Note que os números de carros no sistema e em fila e os tempos de permanência e espera em fila reduziram em relação ao sistema anterior, o que era esperado dado que o tempo de serviço agora é uma constante (sem incertezas).

<sup>31</sup> Ver, por exemplo, Gross e Harris (1974) e Larson e Odoni (1981).

**Tabela 7.7**  
**Resultados para os sistemas  $M/M/1$  e  $M/D/1$ .**

Sistema	$\lambda$	$\mu$	$m$	$\rho$	$P_0$	$E(L)$	$E(L_q)$	$E(W)$	$E(W_q)$
$M/M/1/GD/\infty/\infty$	10	15	1	0,67	0,33	2	1,33	12	8
$M/D/1/GD/\infty/\infty$	10	15	1	0,67	0,33	1,33	0,67	8	4

■

### 7.7.2 Modelo de fila $M/G/1/GD/\infty/\infty$ – um servidor e chegadas em lote

No modelo  $M/G/1/GD/\infty/\infty$  com chegadas em lote (ou simplesmente  $M/G/1$  com chegadas em lote),  $\lambda$  agora corresponde à taxa de chegada de lotes (lotes por hora), enquanto  $E(S) = \frac{1}{\mu}$  e  $V(S)$  são a média e variância do tempo de serviço por usuário, respectivamente. Sejam  $E(N)$  e  $V(N)$  a média e variância do tamanho do lote; logo, a taxa de chegada de usuários é  $\lambda E(N)$  (usuários por hora).

Definindo-se o fator de utilização por  $\rho = \frac{\lambda E(N)}{\mu}$ , pode-se mostrar que, para  $\rho < 1$ , o número médio de usuários é.<sup>32</sup>

$$E(L) = \rho + \frac{\lambda E(S)(V(N) + E(N)^2 - E(N)) + \rho^2 + \lambda^2 E(N)^2 V(S)}{2(1-\rho)}$$

Note que, se  $E(N) = 1$  e  $V(N) = 0$ , a expressão anterior corretamente se reduz a (7.10). As demais medidas de desempenho são obtidas conforme antes. Por exemplo, o tempo médio de permanência no sistema é:

$$E(W) = \frac{E(L)}{\lambda E(N)}$$

### 7.7.3 Modelo de fila $M/G/m/GD/m/\infty$ – múltiplos servidores sem fila de espera

É possível mostrar que o sistema  $M/G/m/GD/m/\infty$  (ou simplesmente  $M/G/m/m$ ) se comporta exatamente como o sistema  $M/M/m/m$ , ou seja, ambos têm a mesma distribuição de probabilidade do número de usuários no sistema  $n$ :

$$P_n = K_n P_0 = \frac{\lambda^n}{\mu^n n!} P_0 = \frac{\frac{\lambda^n}{\mu^n n!}}{\sum_{i=0}^m \frac{\lambda^i}{\mu^i i!}} = \frac{\frac{(\rho m)^n}{n!}}{\sum_{i=0}^m \frac{(\rho m)^i}{i!}} \quad n = 0, 1, 2, \dots, m$$

(note que essa expressão é idêntica à apresentada para o sistema  $M/M/m/m$ ). Apesar de a demonstração desse resultado não ser simples,<sup>33</sup> ele era, de certa forma, intuitivamente esperado, dado que nenhum dos dois sistemas permite filas de espera. Conforme mencionado na Seção 7.5,  $P_m$  é a probabilidade de o sistema estar cheio, também conhecida como fórmula de perda de Erlang.

**Exemplo 7.16** Uma farmácia possui um serviço de entrega rápida de remédios em domicílio, e para isso dispõe de funcionários com motocicletas que entregam o pedido logo após ele ser so-

<sup>32</sup> Ver, por exemplo, Ross (1993).

<sup>33</sup> Ver, por exemplo, Gross e Harris (1985).

licitado por telefone. Se ocorrer um pedido quando todos os funcionários estiverem ocupados (fazendo alguma entrega), a farmácia recorre a terceiros para fazer a entrega, mas a qualidade do serviço ao cliente cai e os custos de entrega aumentam. Durante um certo período, os intervalos entre chegadas de pedidos são exponencialmente distribuídos com média de 15 minutos, mas os tempos de entrega (incluindo os tempos de viagem até o domicílio do cliente e de retorno à loja) têm distribuição genérica com média de 30 minutos por pedido. Qual o número mínimo de funcionários necessário para que não mais do que 10% das entregas sejam realizadas por terceiros?

Note que o sistema pode ser representado por um modelo  $M/G/m/m$  com  $\lambda = 1/15$  pedido por minuto e  $\mu = 1/30$  entrega por minuto, e o número de funcionários  $m$  deve ser tal que a fórmula de perda de Erlang seja menor ou igual a 10%, ou seja:

$$P_m = \frac{\frac{\lambda^m}{\mu^m m!}}{\sum_{i=0}^m \frac{\lambda^i}{\mu^i i!}} = \frac{\frac{(1/15)^m}{(1/30)^m m!}}{\sum_{i=0}^m \frac{(1/15)^i}{(1/30)^i i!}} \leq 0,10$$

Substituindo-se  $m = 1, 2, 3, \dots$  nessa fórmula, segue-se que o menor valor de  $m$  que satisfaz a desigualdade é  $m = 4$  funcionários (resultando em  $P_4 \approx 0,095$ ). ■

#### 7.7.4 Modelo de fila $M/G/\infty/GD/\infty/\infty$ – infinitos servidores (auto-serviço)

O sistema  $M/G/\infty/GD/\infty/\infty$  (ou simplesmente  $M/G/\infty$ ) pode ser visto como um caso especial do sistema anterior  $M/G/m/GD/m/\infty$  no limite  $m \rightarrow \infty$ , ou seja,

$$P_n = \frac{\frac{\lambda^n}{\mu^n n!}}{\sum_{i=0}^{\infty} \frac{\lambda^i}{\mu^i i!}} = \frac{\frac{\lambda^n}{\mu^n n!}}{e^{\frac{\lambda}{\mu}}} = \frac{\left(\frac{\lambda}{\mu}\right)^n e^{-\frac{\lambda}{\mu}}}{n!} \quad n = 0, 1, 2, \dots$$

Logo, temos que  $E(L) = \frac{\lambda}{\mu} = E(L_s)$  e  $E(W) = \frac{1}{\mu} = E(S)$  (como seria esperado, dado que temos infinitos servidores) e  $E(L_q) = E(W_q) = 0$ .

Note que o sistema  $M/G/\infty$  se comporta exatamente como o sistema  $M/M/\infty$  da Seção 7.5, uma vez que ambos têm a mesma distribuição de probabilidade do número de usuários no sistema  $n$  (note que as expressões anteriores são idênticas às apresentadas para o sistema  $M/M/\infty$ ). Esse resultado era intuitivamente esperado, dado que nenhum dos dois sistemas permite filas de espera.

**Exemplo 7.17** Em uma certa região são abertos em média três novos restaurantes por ano. O tempo médio em que um restaurante se mantém no mercado é de dez anos. Se o número de restaurantes que entram no mercado segue uma distribuição de Poisson: (i) Qual o número médio de restaurantes funcionando em um ano? (ii) Qual a probabilidade de haver 25 restaurantes funcionando em um ano?

Temos que  $\lambda = 3$  novos restaurantes abertos por ano e  $\mu = \frac{1}{10} = 0,1$  restaurante fechado por ano. Admitindo que essas taxas se mantêm constantes ao longo dos anos, podemos representar esta situação por um modelo de fila  $M/G/\infty$  (note que não sabemos a distribuição de probabilidade do tempo em que um restaurante se mantém no mercado).

(i) O número médio de restaurantes funcionando em um ano é:  $E(L) = \frac{\lambda}{\mu} = \frac{3}{1/10} = 30$ .

(ii) A probabilidade de haver 25 restaurantes funcionando em um ano é:

$$P_{25} = \frac{\left(\frac{\lambda}{\mu}\right)^{25} e^{-\frac{\lambda}{\mu}}}{25!} = \frac{30^{25} e^{-30}}{25!} \approx 0,05$$

■

### 7.7.5 Modelos de fila $G/G/1/GD/\infty/\infty$ e $G/G/m/GD/\infty/\infty$

Ao contrário dos sistemas com distribuições não-exponenciais  $M/G/1$ ,  $M/G/m/m$  e  $M/G/\infty$ , os sistemas  $G/M/1$ ,  $G/G/1$ ,  $M/G/m$ ,  $G/M/m$  e  $G/G/m$  com distribuições não-exponenciais são de difícil análise e, na prática, em geral são tratados por meio de aproximações.<sup>34</sup> Assim como nos sistemas anteriores, uma condição necessária para que estes sistemas atinjam um equilíbrio é  $\rho < 1$ .

Um resultado de certa forma surpreendente é que, no limite  $\rho \rightarrow 1$  (sistema tendendo à saturação), o tempo de espera em fila em um sistema  $G/G/m$  tende a uma distribuição exponencial com média:

$$\lim_{\rho \rightarrow 1} E(W_q) = \frac{\lambda \left( V(X) + \frac{V(S)}{m} \right)}{2(1-\rho)}$$

em que  $V(X)$  e  $V(S)$  são as variâncias do intervalo de tempo entre as chegadas  $X$  e do tempo de serviço  $S$ , respectivamente. De fato, é possível mostrar que essa média é um limitante superior para o tempo médio de espera em fila em um sistema  $G/G/m$  (para qualquer  $\rho < 1$ ).<sup>35</sup> Note que esse resultado também é válido para um sistema  $G/G/1$  simplesmente substituindo  $m = 1$ .

#### Modelo $G/G/1$

Lembrando que os coeficientes quadráticos de variação das variáveis aleatórias  $X$  e  $S$  são definidos a partir de  $E(X)$ ,  $E(S)$ ,  $V(X)$  e  $V(S)$  por:

$$C_x^2 = \frac{V(X)}{E(X)^2}$$

$$C_s^2 = \frac{V(S)}{E(S)^2}$$

uma aproximação para o número médio de usuários  $E(L)$  em um sistema  $G/G/1$  é a fórmula de Kraemer e Lagenbach-Belz:<sup>36</sup>

$$E(L) \approx \begin{cases} \rho + \frac{\rho^2(C_x^2 + C_s^2)}{2(1-\rho)} & \text{se } C_x^2 \geq 1 \\ \rho + \frac{\rho^2(C_x^2 + C_s^2)}{2(1-\rho)} e^{-\frac{2(1-\rho)(1-C_x^2)^2}{3\rho(C_x^2 + C_s^2)}} & \text{se } C_x^2 < 1 \end{cases} \quad (7.11)$$

Em cada caso da expressão (7.11), o primeiro termo do lado direito corresponde a  $E(L_s)$  e o segundo termo a  $E(L_q)$ . Note que, para um sistema  $M/G/1$ , temos que  $C_x^2 = 1$  (lembre-se que

<sup>34</sup> Para fórmulas exatas em forma fechada (mas envolvendo cálculos integrais) para  $E(W_q)$  em sistemas  $G/M/1$  e  $G/M/m$ , ver, por exemplo, Ross (1993).

<sup>35</sup> Para mais detalhes, ver, por exemplo, Kleinrock (1975) e Larson e Odoni (1981).

<sup>36</sup> A rigor, a fórmula (7.11) corresponde a uma pequena modificação da fórmula original, sugerida por Whitt (1983).

$E(X) = \frac{1}{\lambda}$  e  $V(X) = \frac{1}{\lambda^2}$  para a distribuição exponencial), e a expressão anterior é exata porque se reduz à fórmula de Pollaczek e Khintchine (7.10):

$$E(L) = \rho + \frac{\rho^2(1+C_S^2)}{2(1-\rho)} = \rho + \frac{\rho^2(1+\mu^2V(S))}{2(1-\rho)} = \rho + \frac{\rho^2 + \lambda^2V(S)}{2(1-\rho)}$$

Além disso, essa fórmula também é exata no limite  $\rho \rightarrow 1$ , porque se reduz corretamente ao limitante superior do tempo médio de espera em fila  $E(W_q)$  definido anteriormente.

Outras aproximações para estimar  $E(L)$  foram propostas na literatura:<sup>37</sup>

$$E(L) \approx \rho + \frac{\rho^2(1+C_S^2)}{2(1-\rho)} \frac{(C_X^2 + \rho^2 C_S^2)}{(1 + \rho^2 C_S^2)} \quad (7.12)$$

$$E(L) \approx \rho + \frac{\rho^2(1+C_S^2)}{2(1-\rho)} \frac{(2-\rho)C_X^2 + \rho^2 C_S^2}{(2-\rho + \rho C_S^2)} \quad (7.13)$$

Note que essas aproximações também são exatas para um sistema  $M/G/1$  (substituindo  $C_X^2 = 1$ ).

Diversos experimentos computacionais têm mostrado que essas três aproximações (7.11), (7.12) e (7.13) resultam em boas aproximações quando  $C_X^2 \leq 2$ . Para ilustrar isso, a Tabela 7.8 a seguir compara os resultados obtidos para  $E(L)$  usando essas três aproximações, em sistemas de filas  $E_p/E_q/1$ .

**Tabela 7.8**  
Resultados obtidos para  $E(L)$  com as três aproximações.

		$p$	4	3	2	1
$q$	$C_S^2$	$C_X^2$	0,25	0,333	0,5	1
		Aproximação				
4	0,25	(7.11)	2,81	3,16	3,86	5,96
		(7.12)	2,81	3,16	3,86	5,96
		(7.13)	2,76	3,13	3,86	5,96
		Simulação	2,77	3,12	3,83	
3	0,333	(7.11)	3,11	3,47	4,17	6,30
		(7.12)	3,12	3,47	4,18	6,30
		(7.13)	3,10	3,47	4,20	6,30
		Simulação	3,10	3,45	4,16	6,30
2	0,5	(7.11)	3,73	4,09	4,81	6,98
		(7.12)	3,74	4,10	4,82	6,98
		(7.13)	3,77	4,14	4,88	6,98
		Simulação	3,75	4,11	4,82	
1	1	(7.11)	5,64	6,02	6,76	9,00
		(7.12)	5,66	6,03	6,77	9,00
		(7.13)	5,80	6,17	6,90	9,00
		Simulação	5,74	6,11	6,83	

<sup>37</sup> Para mais detalhes destas aproximações, ver, por exemplo, Buzacott e Shanthikumar (1993).

(isto é, com distribuição de Erlang de ordem  $p$  para o intervalo de tempo entre chegadas  $X$  e distribuição de Erlang de ordem  $q$  para o tempo de serviço  $S$ ) e com  $\rho = 0,9$ . Para  $p = 1$ , o sistema  $G/G/1$  se reduz a um sistema  $M/G/1$ , e todas as aproximações são exatas. Para  $q = 1$ , o sistema  $G/G/1$  se reduz a um sistema  $G/M/1$ . Para comparar a qualidade das aproximações para  $p > 1$ , a tabela inclui os resultados obtidos com um modelo de simulação discreta.

Note que as três aproximações ficam muito próximas dos resultados obtidos com simulação (o maior desvio em relação à simulação foi de apenas 1,74%).

As demais medidas de desempenho  $E(L_s)$ ,  $E(L_q)$ ,  $E(W)$ ,  $E(W_q)$  podem ser obtidas a partir de  $E(L)$  usando a fórmula de Little (7.1) e as equações (7.2)–(7.4) definidas na Seção 7.2. Por exemplo,

o tempo de permanência no sistema é:  $E(W) = \frac{E(L)}{\lambda}$ . Outras aproximações de modelos de filas  $G/G/1$  e variações também podem ser encontradas na literatura.<sup>38</sup>

### Modelo $G/G/m$

Aproximar um sistema  $G/G/m$  (com taxa de serviço  $\mu$ ) por um sistema  $G/G/1$  com um superservidor (com taxa de serviço equivalente  $\mu' = m\mu$ ) em geral não produz bons resultados, porque as medidas de fila  $E(L_q)$  e  $E(W_q)$  do sistema  $G/G/m$  são menores que as do sistema  $G/G/1$  com superservidor (de modo semelhante ao que foi visto com os sistemas  $M/M/m$  e  $M/M/1$  com superservidor na Seção 7.5).

Uma simples aproximação para o número médio de usuários  $E(L)$  em um sistema  $G/G/m$ , baseada na fórmula de Kraemer e Lagenbach-Belz (7.11), é dada por:<sup>39</sup>

$$E(L) \approx \rho m + \frac{(C_x^2 + C_s^2)}{2} E(L_{q_{M/M/m}})$$

em que  $E(L_{q_{M/M/m}})$  é o número médio de usuários em fila em um sistema  $M/M/m$  [veja equação (7.8) da Seção 7.5]. O primeiro termo do lado direito da expressão anterior corresponde a  $E(L_s)$  e o segundo termo aproxima  $E(L_q)$  de um sistema  $G/G/m$ . Note que, se  $C_x^2 = 1$  e  $C_s^2 = 1$ , então a expressão anterior é exata porque se reduz corretamente à expressão (7.9) do número médio de usuários em um sistema  $M/M/m$ . Além disso, essa expressão também é exata no limite  $\rho \rightarrow 1$ , porque se reduz corretamente ao limitante superior do tempo médio de espera em fila  $E(W_q)$  definido anteriormente.

Outra aproximação para  $E(L)$  em um sistema  $G/G/m$ , também baseada na fórmula de Kraemer e Lagenbach-Belz, é dada por:<sup>40</sup>

$$E(L) \approx \rho m + \frac{E(L_{q_{M/M/m}})}{E(L_{q_{M/M/1}})} E(L_{q_{G/G/1}})$$

onde  $E(L_{q_{M/M/1}})$  e  $E(L_{q_{M/M/m}})$  são os números médios de usuários em fila nos sistemas  $M/M/1$  e  $M/M/m$  (veja Seções 7.4 e 7.5), e  $E(L_{q_{G/G/1}})$  é o número médio de usuários em fila em um sistema  $G/G/1$  obtido a partir da fórmula de Kraemer e Lagenbach-Belz [segundo termo do lado direito da equação (7.11)].

As demais medidas de desempenho  $E(L_s)$ ,  $E(L_q)$ ,  $E(W)$ ,  $E(W_q)$  podem ser obtidas a partir de  $E(L)$  usando a fórmula de Little (7.1) e as equações (7.2)–(7.4) definidas na Seção 7.2.

<sup>38</sup> Ver, por exemplo, Rodrigues e Carvalho (2001) para aproximações baseadas em distribuições do tipo fase (*Ph*) para modelos de filas  $G/G/1/K$ .

<sup>39</sup> Para mais detalhes, ver Whitt (1983) e Bitran e Morabito (1996).

<sup>40</sup> Para mais detalhes, ver Buzacott e Shanthikumar (1993). Outras aproximações mais elaboradas ainda podem ser obtidas em Whitt (1993).

**Exemplo 7.18** Nos Exemplos 7.4, 7.9 e 7.15 do *drive-in* de um restaurante *fast-food* (com  $m = 1$  e  $m = 2$  servidores), considere que a distribuição do tempo de atendimento dos clientes não é mais exponencial (como nos Exemplos 7.4 e 7.9) nem determinística (como no Exemplo 7.15), mas uma distribuição genérica desconhecida, com média  $E(S) = 4$  minutos e  $V(S) = 4$  minutos<sup>2</sup>, ou seja, com  $C_s^2 = \frac{V(S)}{E(S)^2} = 0,25$ .

Considere, também, que a distribuição dos intervalos de tempos entre chegadas dos clientes não é mais exponencial (como nos Exemplos 7.4, 7.8 e 7.13), mas uma distribuição genérica desconhecida, com média  $E(X) = 6$  minutos e  $V(X) = 18$  minutos<sup>2</sup>, ou seja, com  $C_x^2 = \frac{V(X)}{E(X)^2} = 0,5$ .

Como mudam os números médios de carros em serviço e em fila em cada caso?

- (i) Supondo que o sistema tenha apenas um servidor (como nos Exemplos 7.4 e 7.13), utilizamos um modelo  $G/G/1$  com fator de utilização  $\rho = \frac{\lambda}{\mu} \approx 0,67$ . Aplicando, por exemplo, a fórmula (7.11) de Kraemer e Lagenbach-Belz, obtemos ( dado que  $C_x^2 < 1$ ):

$$E(L) \approx \rho + \frac{\rho^2(C_x^2 + C_s^2)}{2(1-\rho)} e^{-\frac{2(1-\rho)(1-C_x^2)^2}{3\rho(C_x^2 + C_s^2)}} \approx 1,11 \text{ carro}$$

Lembre-se de que, nos Exemplos 7.4 ( $M/M/1$ ) e 7.15 ( $M/D/1$ ), os valores de  $E(L)$  são 2 e 1,33 carros, respectivamente.

- (ii) Supondo que o sistema tenha dois servidores (como no Exemplo 7.9), temos um modelo  $G/G/2$  com fator de utilização  $\rho = \frac{\lambda}{m\mu} \approx 0,33$ . Aplicando a aproximação baseada na fórmula de Kraemer e Lagenbach-Belz, obtemos:

$$E(L) \approx \rho m + \frac{(C_x^2 + C_s^2)}{2} E(L_q)_{M/M/m} \approx 0,70$$

em que  $E(L_q)_{M/M/m} \approx 0,08$  (ver Exemplo 7.9). Lembre-se que, no Exemplo 7.9 ( $M/M/2$ ), o valor de  $E(L)$  é 0,75 carro. Usando a outra aproximação, também baseada na fórmula de Kraemer e Lagenbach-Belz, obtemos:

$$E(L) \approx \rho m + \frac{E(L_q)_{M/M/m}}{E(L_q)_{G/G/1}} E(L_q)_{G/G/1} \approx 0,71$$

onde  $E(L_q)_{M/M/1} \approx 1,33$  (ver Exemplo 7.4) e  $E(L_q)_{G/G/1} = E(L)_{G/G/1} - \rho = 1,11 - 0,33 \approx 0,78$

## 7.8\* MODELOS DE FILAS COM MÚLTIPLO CLASSES DE USUÁRIOS E PRIORIDADE

Em todos os modelos analisados nas seções anteriores, não há distinção entre classes de usuários. Suponha agora que existam  $1, 2, \dots, r$  classes de usuários, cada classe  $i$  ( $i = 1, \dots, r$ ) tem seu processo de chegada  $A_i$  e seu processo de serviço  $B_i$ . A classe 1 tem prioridade sobre a classe 2, que, por sua vez, tem prioridade sobre a classe 3, e assim por diante, até a classe  $r$ . Desta maneira, assim que um servidor terminar um serviço, o próximo usuário da fila a ser servido é aquele da classe de maior prioridade. Se houver mais de um usuário da mesma classe de prioridade, a escolha do usuário dentro da classe utiliza as disciplinas *FCFS*, *SIRO* ou *LCFS*.

### 7.8.1 Modelo de fila $M_i/G_i/1/NRP/\infty/\infty$ – um servidor e prioridade sem interrupção

Considere que o processo de chegada da classe  $i$  tem distribuição exponencial (isto é,  $A_i = M_i$ ) com taxa  $\lambda_i$ , e o processo de serviço da classe  $i$  tem distribuição genérica ( $B_i = G_i$ ) com taxa  $\mu_i = \frac{1}{E(S_i)}$  (em que  $E(S_i)$  é o tempo médio de serviço) e variância do tempo de serviço  $V(S_i)$ . Como o sistema

tem apenas um servidor ( $m = 1$ ), o fator de utilização da classe  $i$  é  $\rho_i = \frac{\lambda_i}{\mu_i}$ . Note que este sistema de fila pode ser visto como um sistema com múltiplas filas em paralelo em um único servidor (Figura 7.19).

Se  $\sum_{i=1}^r \rho_i < 1$ , é possível mostrar que o tempo médio de espera em fila da classe  $k$  para o sistema com disciplina de prioridade sem interrupção ( $C = NRPR$ ) é dado por:<sup>41</sup>

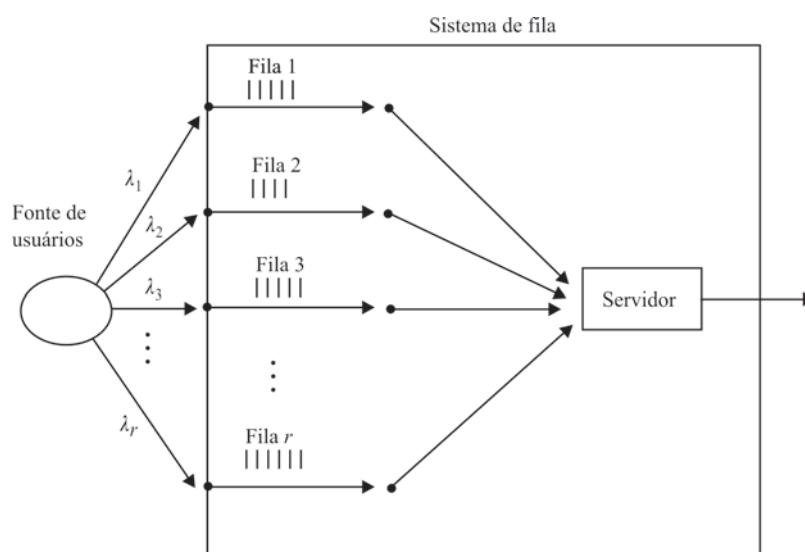
$$E(W_{qk}) = \frac{\sum_{i=1}^r \lambda_i (V(S_i) + E(S_i)^2)}{2(1 - \sum_{i=1}^{k-1} \rho_i)(1 - \sum_{i=1}^k \rho_i)} \quad k = 1, 2, \dots, r$$

em que  $\sum_{i=1}^{k-1} \rho_i = 0$  se  $k = 1$ . Note que, se  $r = 1$ , a expressão anterior se reduz à fórmula de Pollaczek e Khintchine (7.10).

Caso  $\sum_{i=1}^p \rho_i \geq 1$  para algum  $p \leq r$ , então o tempo médio de espera é infinito para as classes  $p, p+1, \dots, r$ , porque o sistema não atinge equilíbrio para essas classes.

As demais medidas de desempenho para cada classe  $k$ ,  $E(L_{sk})$ ,  $E(L_{qk})$ ,  $E(L_k)$  e  $E(W_k)$  podem ser obtidas a partir de  $E(W_{qk})$  usando a fórmula de Little (7.1) e as equações (7.2)–(7.4), definidas na Seção 7.2. Por exemplo, o número médio de usuários da classe  $k$  na fila é:  $E(L_{qk}) = \lambda_k E(W_{qk})$ .

**Exemplo 7.19** Uma máquina processa dois tipos de produtos 1 e 2. Para processar uma unidade do produto 1, a máquina consome em média 3 minutos (isto é, 1/20 hora por produto), com variância de 2 minutos<sup>2</sup> (isto é, 1/1,800 hora<sup>2</sup>), e para processar uma unidade do produto 2, ela leva em média 4 minutos (1/15 hora por produto), também com variância de 2 minutos<sup>2</sup> (1/1,800 hora<sup>2</sup>). Se os processos de chegada dos produtos 1 e 2 têm distribuição de Poisson, com taxas de dez unidades do produto 1 por hora e cinco unidades do produto 2 por hora, respectivamente, e a disciplina de fila for SPT, determine o tempo médio de permanência de cada produto no sistema.



**Figura 7.19** Sistema de filas com múltiplas classes de usuários com prioridade e um servidor.

41 Ver, por exemplo, Larson e Odoni (1981).

Note que  $\lambda_1 = 10$  unidades por hora,  $\lambda_2 = 5$  unidades por hora,  $\mu_1 = 20$  unidades por hora ( $E(S_1) = 1/20$  hora),  $\mu_2 = 15$  unidades por hora ( $E(S_2) = 1/15$  hora),  $V(S_1) = V(S_2) = 1/1,800$  hora<sup>2</sup>. Logo,  $\rho_1 = 10/20 = 0,5$  e  $\rho_2 = 5/15 \approx 0,33$ . Como a disciplina de fila é *SPT*, o produto 1 tem prioridade de processamento sobre o produto 2.

Dado que  $\rho_1 + \rho_2 < 1$ , o sistema atinge um equilíbrio com ambos os produtos. Segue-se que o tempo médio de espera em fila de cada produto é:

$$E(W_{q1}) = \frac{\sum_{i=1}^2 \lambda_i (V(S_i) + E(S_i)^2)}{2(1 - \sum_{i=1}^1 \rho_i)(1 - \sum_{i=1}^1 \rho_i)} = \frac{10\left(\frac{1}{1800} + \frac{1}{20^2}\right) + 5\left(\frac{1}{1800} + \frac{1}{15^2}\right)}{2(1-0)(1-0,5)} \approx 0,06 \text{ hora}$$

$$E(W_{q2}) = \frac{\sum_{i=1}^2 \lambda_i (V(S_i) + E(S_i)^2)}{2(1 - \sum_{i=1}^2 \rho_i)(1 - \sum_{i=1}^2 \rho_i)} = \frac{10\left(\frac{1}{1800} + \frac{1}{20^2}\right) + 5\left(\frac{1}{1800} + \frac{1}{15^2}\right)}{2(1-0,5)(1-0,88)} \approx 0,33 \text{ hora}$$

Logo, o tempo médio de permanência de cada produto no sistema é:

$$E(W_1) = E(S_1) + E(W_{q1}) = \frac{1}{20} + 0,06 \approx 0,11 \text{ hora}$$

$$E(W_2) = E(S_2) + E(W_{q2}) = \frac{1}{15} + 0,33 = 0,4 \text{ hora}$$

Uma observação importante é que, ao contrário das disciplinas *FCFS*, *SIRO* e *LCFS*, a disciplina *SPT* modifica a distribuição de probabilidade do número de usuários no sistema  $P_\eta$ ; consequentemente, modifica o número médio de usuários  $E(L)$  e, por causa da fórmula de Little, também modifica  $E(L_q)$ ,  $E(W)$  e  $E(W_q)$ . Um argumento para mostrar isso é notar que a área  $l(\tau)$  entre  $a(t)$  e  $c(t)$  ao longo de 0 e  $\tau$  na Figura 7.7 da Seção 7.2 se modifica ao se aplicar a disciplina *SPT*.

De fato, não é difícil mostrar que, usando a disciplina *SPT*, isto é, reordenando as classes de usuários e atribuindo prioridades a elas de tal maneira que:  $E(S_1) \leq E(S_2) \leq \dots \leq E(S_r)$ , estamos minimizando o número médio e o tempo médio dos usuários no sistema.<sup>42</sup> Note, no entanto, que, apesar de essas medidas melhorarem em média para os usuários do sistema, elas pioram para os usuários das classes menos prioritárias  $r, r-1, \dots$

### 7.8.2 Modelo de fila $M_i/M/1/PRP/\infty/\infty$ – um servidor e prioridade com interrupção

Neste modelo, o processo de chegada de cada classe  $i$  tem distribuição exponencial com taxa  $\lambda_i$  (isto é,  $A_i = M_i$ ), mas o processo de serviço tem a mesma distribuição exponencial (com taxa  $\mu$ ) para todas as classes (isto é,  $B_i = M$ ). Como temos apenas um servidor ( $m = 1$ ), o fator de utilização da classe  $i$  é  $\rho_i = \frac{\lambda_i}{\mu}$  e o fator de utilização do sistema é  $\rho = \frac{\sum_{i=1}^r \lambda_i}{\mu} = \sum_{i=1}^r \rho_i$ .

Se  $\rho < 1$ , é possível mostrar que o tempo médio de permanência no sistema da classe  $k$  para o sistema com disciplina de prioridade com interrupção ( $C = PRP$ ) é dado por:

$$E(W_k) = \frac{1/\mu}{(1 - \sum_{i=1}^{k-1} \rho_i)(1 - \sum_{i=1}^k \rho_i)} \quad k = 1, 2, \dots, r$$

<sup>42</sup> Ver, por exemplo, Kleinrock (1976) e Larson e Odoni (1981). Este resultado é uma extensão do resultado discutido na Seção 3.6.3 do Capítulo 3, para o caso dos tempos de processamento serem considerados determinísticos.

em que  $\sum_{i=1}^{k-1} \rho_i = 0$  se  $k = 1$ . Note que, se  $r = 1$ , a expressão anterior se reduz a  $\frac{1}{\mu(1-\rho)}$ , que corresponde ao tempo médio de permanência em um sistema  $M/M/1$  (Seção 7.4).

É interessante notar que, por causa da falta de memória da distribuição exponencial, esse resultado é válido para os casos em que: (i) o serviço que foi interrompido continua a partir do ponto em que parou (interrupção sem perda); e (ii) o serviço que foi interrompido precisa ser recomeçado desde o início (interrupção com perda total).

As demais medidas de desempenho para cada classe  $k$ ,  $E(L_{sh})$ ,  $E(L_{qk})$ ,  $E(L_k)$  e  $E(W_{qk})$ , podem ser obtidas a partir de  $E(W_k)$  usando a fórmula de Little (7.1) e as equações (7.2)–(7.4) definidas na Seção 7.2. Por exemplo, o número médio de usuários da classe  $k$  no sistema é:  $E(L_k) = \lambda_k E(W_k)$ .

**Exemplo 7.20** Considere um sistema de fila  $M/M/1/PRP/\infty/\infty$  com  $r = 2$  classes de usuários, com taxas de chegada  $\lambda_1$  e  $\lambda_2$ , respectivamente, e taxa de serviço  $\mu$ . Digamos, um posto de atendimento médico (com apenas um médico) que separa os pacientes que chegam em urgentes e não urgentes. Os pacientes urgentes têm prioridade de atendimento sobre os não urgentes. (i) Determine os tempos médios de permanência no sistema de cada classe. (ii) Relacione estes tempos com o tempo médio de permanência em um sistema de fila  $M/M/1/FCFS/\infty/\infty$  com taxa de chegada  $\lambda = \lambda_1 + \lambda_2$  e taxa de serviço  $\mu$ .

Admitindo que  $\lambda = \lambda_1 + \lambda_2 < \mu$ , então ambas as classes de usuários atingem um equilíbrio no sistema.

(i) Temos que os tempos médios de permanência de cada classe no sistema são:

$$E(W_1) = \frac{1/\mu}{(1 - \sum_{i=1}^{1-1} \rho_i)(1 - \sum_{i=1}^1 \rho_i)} = \frac{1/\mu}{1 - \rho_1} = \frac{1}{\mu - \lambda_1}$$

$$E(W_2) = \frac{1/\mu}{(1 - \sum_{i=1}^{2-1} \rho_i)(1 - \sum_{i=1}^2 \rho_i)} = \frac{1/\mu}{(1 - \rho_1)[1 - (\rho_1 + \rho_2)]} = \frac{\mu}{(\mu - \lambda_1)[\mu - (\lambda_1 + \lambda_2)]}$$

(ii) Note que o tempo médio de permanência de um usuário qualquer no sistema, independentemente de sua classe, é dado por:

$$\begin{aligned} E(W) &= \frac{\lambda_1}{\lambda_1 + \lambda_2} E(W_1) + \frac{\lambda_2}{\lambda_1 + \lambda_2} E(W_2) = \frac{\lambda_1}{\lambda_1 + \lambda_2} \frac{1}{\mu - \lambda_1} + \frac{\lambda_2}{\lambda_1 + \lambda_2} \frac{\mu}{(\mu - \lambda_1)[\mu - (\lambda_1 + \lambda_2)]} \\ &= \frac{\lambda_1}{\lambda(\mu - \lambda_1)} + \frac{\lambda_2 \mu}{\lambda(\mu - \lambda_1)(\mu - \lambda)} = \frac{\lambda_1(\mu - \lambda) + \lambda_2 \mu}{\lambda(\mu - \lambda_1)(\mu - \lambda)} = \frac{\lambda(\mu - \lambda_1)}{(\mu - \lambda)} = \frac{1}{\mu - \lambda} \end{aligned}$$

que é exatamente igual ao tempo médio de permanência de um usuário em um sistema  $M/M/1/FCFS/\infty/\infty$  com taxa  $\lambda$  (Seção 7.4). Note que:  $E(W_1) < E(W) < E(W_2)$ .

Logo, se não houver prioridade entre as duas classes e a disciplina da fila for  $FCFS$ , então os tempos médios de permanência das duas classes são iguais a  $E(W)$ , ou seja, o tempo de permanência da classe 1 aumenta (dado que  $E(W) > E(W_1)$ ) e o tempo médio de permanência da classe 2 diminui (dado que  $E(W) < E(W_2)$ ). ■

### 7.8.3 Modelo de fila $M_i/M/m/NRPR/\infty/\infty$ – múltiplos servidores e prioridade sem interrupção

No caso de termos múltiplos servidores ( $m > 1$ ), o fator de utilização da classe  $i$  é  $\rho_i = \frac{\lambda_i}{m\mu}$  e o fator de utilização do sistema é  $\rho = \frac{\sum_{i=1}^r \lambda_i}{m\mu} = \sum_{i=1}^r \rho_i$ . Note que, assim como no modelo anterior, estamos admitindo que o processo de serviço tem a mesma distribuição exponencial (com taxa  $\mu$ ) para todas as classes (isto é,  $B_i = M$ ).

Se  $\rho < 1$ , é possível mostrar que o tempo médio de espera em fila da classe  $k$  para o sistema com disciplina de prioridade sem interrupção ( $C = NRPR$ ) é dado por:

$$E(W_{qk}) = \frac{\frac{(\rho m)^m}{m\mu(1-\rho)m!} P_0}{(1 - \sum_{i=1}^{k-1} \rho_i)(1 - \sum_{i=1}^k \rho_i)} \quad k = 1, 2, \dots, r$$

em que  $\sum_{i=1}^{k-1} \rho_i = 0$ , se  $k = 1$ , e  $P_0$  é a probabilidade de um sistema  $M/M/m$  (com fator de utilização  $\rho$ ) estar vazio (ver Seção 7.5). Note que, se  $r = 1$ , a expressão anterior se reduz ao tempo médio de espera em fila em um sistema  $M/M/m$  (Seção 7.5).

Caso  $\sum_{i=1}^p \rho_i \geq 1$  para algum  $p \leq r$ , então o tempo médio de espera é infinito para as classes  $p, p+1, \dots, r$ , porque o sistema não atinge um equilíbrio para estas classes.

As demais medidas de desempenho para cada classe  $k$ ,  $E(L_{sk})$ ,  $E(L_{qk})$ ,  $E(L_k)$  e  $E(W_k)$ , podem ser obtidas a partir de  $E(W_{qk})$  usando a fórmula de Little (7.1) e as equações (7.2)–(7.4) definidas na Seção 7.2. Por exemplo, o número médio de usuários da classe  $k$  na fila é:  $E(L_{qk}) = \lambda_k E(W_{qk})$ .

**Exemplo 7.21**<sup>43</sup> Uma delegacia policial classifica as chamadas em dois tipos: emergenciais (tipo 1) e não emergenciais (tipo 2). As chamadas do tipo 1 têm prioridade (sem interrupção) sobre as chamadas do tipo 2. Os intervalos de tempo entre chegadas das chamadas de cada tipo são exponenciais, com taxas  $\lambda_1 = 10$  e  $\lambda_2 = 20$  chamadas por hora, respectivamente. A delegacia tem  $m = 5$  viaturas para atender as chamadas e os tempos de serviço de cada viatura (tempo de viagem e tempo de atendimento no local da chamada) também são exponenciais, com taxas  $\mu = 7,5$  chamadas por hora. Determine os tempos médios de espera em fila das chamadas de cada tipo.

Os fatores de utilização de cada tipo de chamada e do sistema são:

$$\rho_1 = \frac{\lambda_1}{m\mu} = \frac{10}{5(7,5)} \approx 0,27$$

$$\rho_2 = \frac{\lambda_2}{m\mu} = \frac{20}{5(7,5)} \approx 0,53$$

$$\rho = \rho_1 + \rho_2 = 0,8$$

A probabilidade de um sistema  $M/M/5$  (com fator de utilização  $\rho = 0,8$ ) estar vazio é:

$$P_0 = \frac{1}{\sum_{n=0}^{m-1} \frac{(\rho m)^n}{n!} + \frac{(\rho m)^m}{m!} \left( \frac{1}{1-\rho} \right)} = \frac{1}{\sum_{n=0}^4 \frac{4^n}{n!} + \frac{4^5}{5!} \left( \frac{1}{1-0,8} \right)} \approx 0,013$$

<sup>43</sup> Este exemplo foi inspirado em um exemplo em Winston (2004).

Logo, os tempos médios de espera de cada tipo de chamada são:

$$E(W_{q1}) = \frac{\frac{(\rho m)^m}{m\mu(1-\rho)m!} P_0}{(1 - \sum_{i=1}^{1-1} \rho_i)(1 - \sum_{i=1}^1 \rho_i)} = \frac{\frac{4^5}{5(7,5)(1-0,8)5!} P_0}{1-0,27} \approx 0,02 \text{ hora (ou } 1,2 \text{ minuto)}$$

$$E(W_{q2}) = \frac{\frac{(\rho m)^m}{m\mu(1-\rho)m!} P_0}{(1 - \sum_{i=1}^{2-1} \rho_i)(1 - \sum_{i=1}^2 \rho_i)} = \frac{\frac{4^5}{5(7,5)(1-0,8)5!} P_0}{(1-0,27)(1-0,27-0,53)} \approx 0,10 \text{ hora (ou } 6 \text{ minutos)}$$

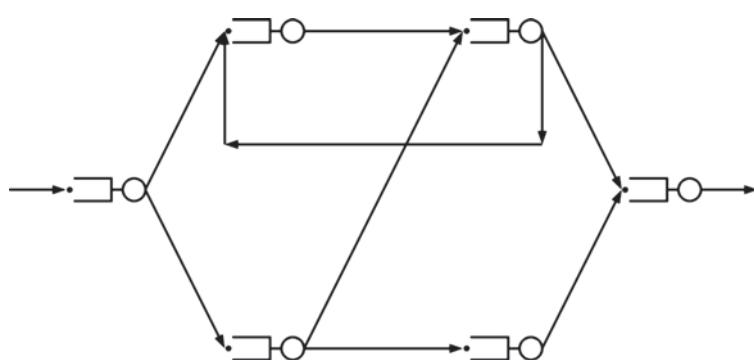
Note que esses tempos correspondem aos tempos médios em fila para cada tipo de chamada. Se quisermos estimar os tempos médios de espera de cada chamada até a chegada da viatura, ainda temos de adicionar o tempo médio de viagem da viatura. ■

### 7.9\* MODELOS DE REDES DE FILAS

As seções anteriores apresentaram modelos para analisar sistemas de filas contendo apenas um dispositivo de serviço (com um ou mais servidores), como os sistemas dos tipos (i) e (ii) ilustrados na Figura 7.3. A seguir, analisamos sistemas de redes de filas compostos de vários sistemas dos tipos (i) e (ii) interconectados, isto é, arranjados em paralelo e/ou em série. Dois exemplos simples de redes de filas são os sistemas do tipo (iii) e (iv) ilustrados na Figura 7.3, em que os dispositivos de serviço estão arranjados apenas em paralelo e apenas em série, respectivamente. Uma rede de filas com arranjo mais geral aparece ilustrada na Figura 7.20.

O conjunto de nós e arcos compõe a rede de filas com as seguintes características: (a) número de nós; (b) seqüência de serviços (roteiros); e (c) tipo de rede de filas: aberta, fechada e mista. O número de nós corresponde ao número de dispositivos de serviço na rede. A seqüência de serviços (ou roteiro) ao longo dos nós, determinística ou probabilística, pode ser seqüencial, seqüencial com realimentação (*feedback*), arborescente, acíclica e cíclica. Arcos com realimentação podem ser utilizados em redes de manufatura para representar, por exemplo, retrabalho, e roteiros probabilísticos podem ser utilizados para modelar, por exemplo, falhas de máquinas.

Uma rede de filas pode ser aberta, fechada ou mista. Em uma rede de filas aberta (*open queueing network*), os usuários assim que chegam entram na rede, recebem serviço em um ou mais



**Figura 7.20** Rede de filas aberta.

nós, e eventualmente saem da rede. Note que a rede da Figura 7.20 é aberta: os usuários entram na rede pelo nó do lado esquerdo da figura, percorrem roteiros seqüenciais com realimentação ao longo dos nós (conforme indicado pelos arcos da figura), eventualmente aguardam em fila em frente aos dispositivos de serviço dos nós, e deixam a rede pelo nó do lado direito da figura. O número de usuários circulando na rede é uma variável aleatória.

Em uma rede de filas fechada (*closed queueing network*), ao contrário, não há chegadas ou partidas externas, o número de usuários circulando na rede é mantido fixo, enquanto a taxa de chegada de usuários em cada nó é uma variável aleatória. Se a rede tiver múltiplas classes de usuários, é possível redefinir sub-redes abertas para algumas classes e sub-redes fechadas para outras. Neste caso, a rede de filas é chamada mista.

Os modelos de redes de filas comumente são separados em modelos de redes com distribuições exponenciais (também chamados de redes de Jackson em homenagem aos trabalhos de J. Jackson em 1957 e 1963)<sup>44</sup> e modelos de redes com distribuições genéricas. As redes de Jackson são aquelas em que os processos de chegadas externas de usuários na rede e os processos de serviço de usuários em cada nó da rede são representados por distribuições exponenciais, ou seja, são markovianos ou sem memória ( $M$ ). As redes genéricas são aquelas em que esses processos são descritos por distribuições genéricas ( $G$ ).

### 7.9.1 Modelos de redes de filas com distribuições exponenciais

Um resultado importante para analisar redes de Jackson é o seguinte: considere o sistema de fila  $M/M/m/GD/\infty/\infty$  estudado na Seção 7.5, isto é, com intervalos de tempo entre chegadas com distribuição exponencial com taxa  $\lambda$ , tempos de serviço com distribuição exponencial com taxa  $\mu$  (para cada um dos  $m$  servidores), e sem limites de capacidade da fila ou tamanho da população.

Se o fator de utilização  $\rho = \frac{\lambda}{m\mu}$  for menor que 1, então, em equilíbrio, é possível mostrar que o intervalo de tempo entre partidas de usuários deste sistema também tem distribuição exponencial com taxa  $\lambda$ . Isto é, os usuários saem do sistema de acordo com a mesma distribuição de probabilidade com que entraram no sistema. Esse resultado não é válido se algum nó da rede tiver limite de capacidade ou distribuição de serviço não exponencial (exceto para o sistema  $M/G/\infty$ ).<sup>45</sup>

O resultado anterior implica que se tivermos: (a) uma rede de filas do tipo (iv) da Figura 7.3, ou seja, com 1, 2, ...,  $J$  dispositivos de serviço arranjados em série (por exemplo, como em uma linha de montagem); (b) o processo de chegada no primeiro dispositivo ( $j = 1$ ) for exponencial com taxa  $\lambda$  (isto é,  $A_j = M$ ); (c) cada dispositivo  $j$  ( $j = 1, 2, \dots, J$ ) contiver  $m_j$  ( $m_j \geq 1$ ) servidores idênticos em paralelo e sala de espera infinita; e (d) o processo de serviço em cada dispositivo  $j$  for exponencial com taxa  $\mu_j$  (para cada um dos  $m_j$  servidores) (isto é,  $B_j = M_j$ ), então, em equilíbrio, a rede de filas pode ser analisada como  $J$  sistemas de filas  $M/M/m/GD/\infty/\infty$  independentes (cada um com parâmetros  $\lambda, \mu_j, m_j$ ).

Ou seja, cada nó da rede pode ser analisado separadamente pelo modelo  $M/M/m/GD/\infty/\infty$  da Seção 7.5. A condição de equilíbrio é que, em cada nó  $j$  da rede, tenhamos o fator de utilização  $\rho_j = \frac{\lambda}{m_j \mu_j} < 1$ .

**Exemplo 7.22** No Exemplo 7.4 do *drive-in* de um restaurante de *fast-food* das Seções 7.4 e 7.5, suponha agora que os clientes tenham de passar por duas janelas de atendimento: na primeira eles fazem seus pedidos e pagam, e na segunda eles recebem seus pedidos (Figura 7.21). Se chegam

<sup>44</sup> Ver Jackson (1957, 1963).

<sup>45</sup> Ver, por exemplo, Disney *et al.* (1973) e Gross e Harris (1985).

em média 10 carros por hora no *drive-in*, o tempo médio de atendimento de cada cliente na primeira janela é de 3 minutos, o tempo médio de atendimento na segunda janela é de 2 minutos (total de 5 minutos), não há limites para as capacidades das filas de cada janela, e os intervalos de tempo entre chegadas no *drive-in* e os tempos de atendimento em cada janela são exponenciais: (i) Qual o número médio de carros em cada fila do *drive-in*? (ii) Qual o tempo médio de espera nas duas filas do *drive-in*?

Conforme o resultado anterior, este sistema de rede de filas em série pode ser analisado como  $J = 2$  simples sistemas de filas  $M/M/1$  independentes (ver Seção 7.4).

Note que  $\lambda = 10$  carros por hora ( $E(X) = \frac{1}{\lambda} = \frac{60}{10} = 6$  minutos),  $E(S_1) = 3$  minutos ( $\mu_1 = \frac{1}{E(S_1)} = \frac{1}{3} = 20$  carros por hora) e  $E(S_2) = 2$  minutos ( $\mu_2 = \frac{1}{E(S_2)} = \frac{1}{2} = 30$  carros por hora).

Logo, a utilização média em cada janela do sistema é:  $\rho_1 = \frac{\lambda}{\mu_1} = \frac{10}{20} = 0,5$  e  $\rho_2 = \frac{\lambda}{\mu_2} = \frac{10}{30} \approx 0,33$ .

Como ambas são menores que 1, o sistema atinge um equilíbrio.

(i) O número médio de carros em cada fila do *drive-in* é:

$$E(L_{q1}) = \frac{\rho_1^2}{1 - \rho_1} = \frac{0,5^2}{1 - 0,5} = 0,5 \text{ carro}$$

$$E(L_{q2}) = \frac{\rho_2^2}{1 - \rho_2} = \frac{0,33^2}{1 - 0,33} \approx 0,17 \text{ carro}$$

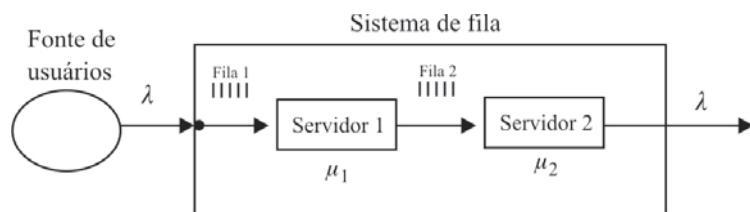
(ii) O tempo médio de espera nas duas filas do *drive-in* é a soma dos tempos médios de espera na fila de cada janela:

$$E(W_q) = E(W_{q1}) + E(W_{q2}) = \frac{E(L_{q1})}{\lambda} + \frac{E(L_{q2})}{\lambda} = \frac{0,5}{10} + \frac{0,17}{10} \approx 0,067 \text{ hora (ou 4 minutos).}$$

Comparando esses resultados com os obtidos no exemplo da Seção 7.4 (contendo apenas uma janela com tempo médio de atendimento de 4 minutos), note que, ao utilizar duas janelas com maior tempo médio total de atendimento (de 4 minutos para 5 minutos), os tempos médios de espera em fila se reduzem (de 8 minutos para 4 minutos). Além disso, os tempos médios de permanência no sistema também se reduzem (de  $4 + 8 = 12$  minutos para  $5 + 4 = 9$  minutos). ■

Em 1957, J. Jackson mostrou que a decomposição discutida anteriormente também pode ser aplicada a redes de filas mais gerais, com dispositivos de serviço arranjados em série e/ou em paralelo, como a rede do tipo (iv) ilustrada na Figura 7.20.

Admite-se que os processos de chegadas externas e os processos de serviços em cada nó ( $j = 1, 2, \dots, J$ ) da rede são exponenciais (isto é,  $A_j = M_j$  e  $B_j = M_j$ ). Usuários podem chegar em



**Figura 7.21** *Drive-in* de um restaurante *fast-food* como uma rede de filas em série.

cada nó  $j$  tanto de fora da rede, com taxa de chegada externa  $r_j$ , como de outros nós da rede, com taxa de chegada interna  $\sum_{i=1}^J p_{ij}\lambda_i$ , em que  $p_{ij}$  é a fração de usuários do nó  $i$  para o nó  $j$ , e  $\lambda_i$  é a taxa de chegadas (externas mais internas) no nó  $i$ . Assim, a taxa total de chegada em cada nó  $j$  é obtida resolvendo-se o sistema linear:

$$\lambda_j = r_j + \sum_{i=1}^J p_{ij}\lambda_i \quad j = 1, 2, \dots, J$$

Admite-se, também, que cada nó  $j$  da rede não tem limite de capacidade e contém  $m_j$  servidores (idênticos em paralelo), cada um com taxa de serviço  $\mu_j$ . Se o fator de utilização em cada nó  $j$  for menor que 1 (isto é,  $\rho_j = \frac{\lambda_j}{m_j\mu_j} < 1$ ), então a rede pode ser decomposta em  $J$  sistemas de filas  $M/M/m/GD/\infty/\infty$  independentes, cada um analisado em condição de equilíbrio com parâmetros  $\lambda_j$ ,  $\mu_j$  e  $m_j$ , conforme a Seção 7.5.<sup>46</sup>

Para encontrar  $E(L)$ , o número médio de usuários na rede, basta somar o número médio de usuários em todos os  $J$  nós, ou seja,

$$E(L) = \sum_{j=1}^J E(L_j)$$

As demais medidas de desempenho da rede podem ser obtidas aplicando-se a fórmula de Little. Por exemplo, o tempo médio de permanência de um usuário na rede é:  $E(W) = \frac{E(L)}{\lambda}$ , em que  $\lambda = \sum_{j=1}^J r_j$  é a taxa de chegada externa da rede.

**Exemplo 7.23** Considere uma rede de filas com dois nós 1 e 2, cada um com um único servidor. Usuários chegam de fora da rede na estação 1 de acordo com um processo de Poisson com taxa  $r_1 = 4$  usuários por hora, e na estação 2 de acordo com um processo de Poisson com taxa  $r_2 = 5$  usuários por hora. As taxas de serviço dos nós 1 e 2 são  $\mu_1 = 8$  e  $\mu_2 = 10$  usuários por hora, respectivamente. Após terminarem o serviço no nó 1, os usuários seguem para o nó 2 com probabilidade  $p_{12} = 0,5$ , ou saem da rede com probabilidade  $1 - 0,5$  (isto é,  $p_{11} = 0$ ). Após terminarem o serviço no nó 2, os usuários seguem para o nó 1 com probabilidade  $p_{21} = 0,25$ , ou saem da rede com probabilidade  $1 - 0,75$  ( $p_{22} = 0$ ). Determine: (i) o número médio de usuários na rede; e (ii) o tempo médio de permanência na rede.

As taxas de chegada de usuários em cada nó são obtidas por:

$$\lambda_1 = r_1 + \sum_{i=1}^2 p_{i1}\lambda_i = 4 + p_{11}\lambda_1 + p_{21}\lambda_2 = 4 + 0,25\lambda_2$$

$$\lambda_2 = r_2 + \sum_{i=1}^2 p_{i2}\lambda_i = 5 + p_{12}\lambda_1 + p_{22}\lambda_2 = 5 + 0,5\lambda_1$$

o que resulta em:  $\lambda_1 = 6$  e  $\lambda_2 = 8$  usuários por hora (verifique).

(i) O número médio de usuários na rede é (usando o modelo  $M/M/1$  da Seção 7.4):

$$E(L) = E(L_1) + E(L_2) = \frac{\lambda_1}{\mu_1 - \lambda_1} + \frac{\lambda_2}{\mu_2 - \lambda_2} = \frac{6}{8 - 6} + \frac{8}{10 - 8} = 3 + 4 = 7 \text{ usuários}$$

(ii) Como a taxa de chegada na rede é:  $\lambda = r_1 + r_2 = 4 + 5 = 9$  usuários por hora, temos que o tempo médio de permanência na rede é:

$$E(W) = \frac{E(L)}{\lambda} = \frac{7}{9} \text{ horas}$$

<sup>46</sup> Ver Jackson (1957). Baskett *et al.* (1975) estenderam estes resultados de decomposição para redes de filas markovianas ainda mais genéricas; veja também Humes e Kumal (1997).

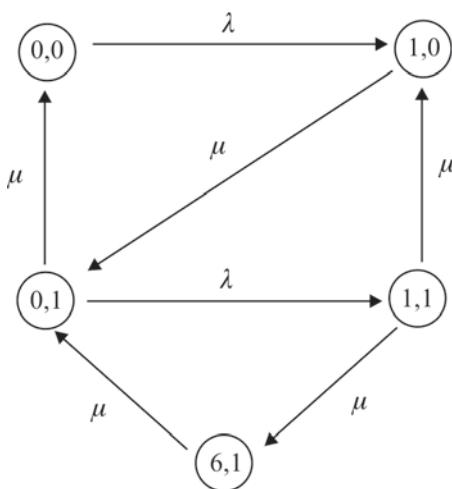
Conforme mencionado, as análises anteriores são válidas somente se os nós da rede não têm limitação de capacidade. Redes de filas com distribuições exponenciais e capacidade limitada podem ser analisadas definindo o estado do sistema como  $(n_1, n_2, \dots, n_j)$ , em que cada  $n_j$  representa o número de usuários no nó  $j$  da rede, e construindo equações de transição de estados, de modo semelhante ao que foi feito para o modelo de nascimento e morte na Seção 7.3. O exemplo a seguir ilustra isso.

**Exemplo 7.24** Uma linha de produção tem duas estações de trabalho operando em série e fabrica produtos com roteiro de fabricação passando primeiro pela estação 1 e depois pela estação 2. Considere que os produtos chegam na estação 1 de acordo com uma distribuição exponencial com taxa de  $\lambda$  unidades por hora, e que cada estação é modelada como uma única máquina com tempo de processamento exponencial com taxa de  $\mu$  unidades por hora. Considere, também, que não há espaço para estoque intermediário de produtos na frente de cada estação, por isso, a estação 1 fica bloqueada toda vez que termina de processar um produto e a estação 2 está ocupada. Assim, as chegadas dos produtos são impedidas de entrar na rede toda vez que a estação 1 estiver ocupada ou bloqueada. Determine: (i) o número médio de produtos na rede; e (ii) o número médio de estações ocupadas na rede.

A Figura 7.22 ilustra o diagrama de transição entre os possíveis estados  $(n_1, n_2)$  da rede, definidos por  $(0,0)$ ,  $(1,0)$ ,  $(0,1)$ ,  $(1,1)$  e  $(b,1)$ , em que, em particular,  $(b,1)$  representa a estação 1 bloqueada ( $b$ ) e a estação 2 ocupada (1). Note que o estado  $(b,0)$  não pode existir.

De modo semelhante ao que foi feito para o modelo de nascimento e morte na Seção 7.3, podemos construir as equações de equilíbrio para cada estado  $(n_1, n_2)$ , lembrando que a taxa em que o sistema entra no estado deve ser igual à taxa em que ele sai do estado. Seja  $P_{n_1 n_2}$  a probabilidade de equilíbrio do estado  $(n_1, n_2)$ . Temos que (veja Figura 7.22):

$$\begin{aligned}
 \mu P_{01} &= \lambda P_{00} && \text{estado } (0,0) \\
 \mu P_{10} + \mu P_{b1} &= (\lambda + \mu) P_{01} && \text{estado } (0,1) \\
 \lambda P_{00} + \mu P_{11} &= \mu P_{10} && \text{estado } (1,0) \\
 \lambda P_{01} &= 2\mu P_{11} && \text{estado } (1,1) \\
 \mu P_{11} &= \mu P_{b1} && \text{estado } (b,1)
 \end{aligned}$$



**Figura 7.22** Diagrama de transição entre estados da rede.

Esse sistema de equações lineares nas variáveis  $P_{n_1 n_2}$  pode ser resolvido lembrando que a soma das probabilidades deve ser igual a 1, ou seja (conforme equação (7.6)):

$$P_{00} + P_{01} + P_{10} + P_{11} + P_{b1} = 1$$

Seja  $\rho = \frac{\lambda}{\mu}$  o fator de utilização de cada estação. A solução do sistema anterior resulta em (note que a soma é igual a 1):

$$\begin{aligned} P_{00} &= \frac{2}{3\rho^2 + 4\rho + 2} \\ P_{01} &= \frac{2\rho}{3\rho^2 + 4\rho + 2} \\ P_{10} &= \frac{\rho^2 + 2\rho}{3\rho^2 + 4\rho + 2} \\ P_{11} &= \frac{\rho^2}{3\rho^2 + 4\rho + 2} \\ P_{b1} &= \frac{\rho^2}{3\rho^2 + 4\rho + 2} \end{aligned}$$

- (i) Com a distribuição de equilíbrio, podemos calcular diferentes medidas de desempenho para a rede. O número médio de produtos na rede é obtido com:

$$E(L) = 0P_{00} + 1(P_{01} + P_{10}) + 2(P_{b1} + P_{11}) = \frac{5\rho^2 + 4\rho}{3\rho^2 + 4\rho + 2}$$

- (ii) O número médio de estações ocupadas na rede é:

$$0P_{00} + 1(P_{01} + P_{10} + P_{b1}) + 2P_{11} = \frac{4\rho^2 + 4\rho}{3\rho^2 + 4\rho + 2}$$

#### *Modelos de redes de filas com distribuições genéricas*

A análise de redes de filas com distribuições genéricas em geral é difícil. Alguns autores têm proposto métodos aproximados baseados na decomposição da rede em  $1, 2, \dots, J$  sistemas de filas  $G_j/G/m_j$  independentes, um para cada nó da rede, inspirados na análise das redes de Jackson. Os processos de chegadas externas de usuários na rede e serviços dos usuários em cada nó são descritos apenas pelas médias e variâncias das distribuições. Para estimar medidas de desempenho de cada nó e da rede, os métodos envolvem a solução de alguns sistemas lineares.<sup>47</sup>

#### **7.9.2 Modelos de decisão em rede de filas**

Nos sistemas de redes de filas, o nível de serviço aos usuários depende fundamentalmente das taxas de serviço ( $\mu_j$ ) e do número de servidores ( $m_j$ ) em cada nó da rede (capacidade do sistema). Por exemplo, em uma rede de manufatura, melhores níveis de serviço em geral são obtidos com menores quantidades de produtos no sistema (estoque em processo) e menores tempos de permanência destes produtos na rede (*leadtimes* de produção), e essas medidas dependem da capacidade da rede.

<sup>47</sup> Ver, por exemplo, Buzacott e Shanthikumar (1993) e Bitran e Morabito (1996, 1999).

Assim como no caso de sistemas de filas com apenas um dispositivo de serviço (Seções 7.4 e 7.5), também podemos definir modelos de decisão para escolher a melhor entre diversas alternativas de capacidade para a rede. Para cada alternativa, podemos estimar as medidas de desempenho e os custos envolvidos, para depois compará-los e escolher a melhor configuração de capacidade para a rede. Em alguns casos, podemos também utilizar modelos de otimização para determinar os melhores valores de  $\mu_j$  e  $m_j$  em cada nó  $j$  da rede, de maneira a minimizar os tamanhos das filas ou os tempos de permanência dos usuários na rede.<sup>48</sup> Em geral, os modelos de otimização de rede de filas são complexos; uma exceção é o exemplo a seguir.

**Exemplo 7.25** Examine uma rede de filas do tipo da analisada no Exemplo 7.23, com processos de chegadas externas e processos de serviços em cada nó  $j$  ( $j = 1, 2, \dots, J$ ) da rede com distribuições exponenciais. Usuários podem chegar em cada nó  $j$  tanto de fora da rede, com taxa de chegada externa  $r_j$ , como de outros nós da rede, com taxa de chegada interna  $\sum_{i=1}^J p_{ij}\lambda_i$ , em que  $p_{ij}$  é a fração de usuários do nó  $i$  para o nó  $j$ , e  $\lambda_i$  é a taxa de chegadas (externas mais internas) no nó  $i$ . Assim, a taxa total de chegada em cada nó  $j$  é obtida resolvendo-se o sistema linear:

$$\lambda_j = r_j + \sum_{i=1}^J p_{ij}\lambda_i \quad j = 1, 2, \dots, J$$

Considere, ainda, que cada nó  $j$  tem apenas um servidor (isto é, um sistema  $M/M/1$ ), que o custo de capacidade no nó  $j$  é  $c_j\mu_j$ , e que a soma dos custos de capacidade de todos os nós da rede não pode superar o limite de orçamento  $C$ , ou seja,  $\sum_{j=1}^J c_j\mu_j \leq C$ .

Deseja-se determinar os valores ótimos das taxas de serviço  $\mu_j$ ,  $j = 1, \dots, J$ , de maneira a minimizar o número total de usuários na rede  $E(L) = \sum_{j=1}^J E(L_j)$ , ou seja, resolver o seguinte modelo de programação não-linear nas variáveis  $\mu_j$ :

$$\begin{aligned} \text{Min } & \sum_{j=1}^J E(L_j) = \sum_{j=1}^J \frac{\lambda_j}{\mu_j - \lambda_j} \\ & \sum_{j=1}^J c_j\mu_j \leq C \\ & \mu_j \geq 0 \quad j = 1, 2, \dots, J \end{aligned}$$

É possível mostrar que o valor ótimo de cada taxa de serviço  $\mu_j$  é facilmente obtido pela fórmula:<sup>49</sup>

$$\mu_j = \lambda_j + \frac{\sqrt{c_j\lambda_j}}{\sum_{i=1}^J \sqrt{c_i\lambda_i}} \frac{C - \sum_{i=1}^n c_i\lambda_i}{c_j} \quad j = 1, 2, \dots, J$$

■

## 7.10 EXERCÍCIOS<sup>50</sup>

**Exercício 7.1** Em uma cabine telefônica (com apenas um telefone), usuários chegam conforme um processo de Poisson com taxa de 10 usuários por hora. O tempo de conversação é exponencialmente distribuído com média de 5 minutos por usuário. Considerando que, quando a cabine telefônica está ocupada, os usuários que chegam esperam em uma fila única com disciplina de atendimento *FCFS*, determine: (a) qual o número médio de usuários em fila e no sistema (fila e cabine); (b) qual o tempo médio de permanência dos usuários em fila e no sistema; (c) qual a por-

<sup>48</sup> Para exemplos de modelos de otimização em redes de filas aplicados em sistemas de manufatura, ver, por exemplo, Askin e Standridge (1993), Gershwin (1994) e Bitran e Morabito (1996, 1999).

<sup>49</sup> Ver, por exemplo, Kleinrock (1976).

<sup>50</sup> Alguns exercícios desta lista foram inspirados em exercícios em Larson e Odoni (1981) e Ross (1993).

centagem de tempo em que o sistema fica vazio; (d) qual a probabilidade de o sistema estar com dois ou mais usuários; e (e) qual a probabilidade de um usuário permanecer mais de 10 minutos no sistema.

**Exercício 7.2** No Exercício 7.1, se os usuários reduzirem o tempo médio de conversação pela metade (2,5 minutos), determine como mudam os valores dos itens (a), (b), (c), (d) e (e). Compare os resultados com os obtidos no Exercício 7.1.

**Exercício 7.3** No Exercício 7.1, se um segundo telefone for instalado na cabine telefônica, considerando que, quando os dois telefones estão em uso, os usuários que chegam esperam em uma fila única com disciplina de atendimento FCFS, determine como mudam os valores dos itens (a), (b), (c), (d) e (e). Compare os resultados com os obtidos nos Exercícios 7.1 e 7.2.

**Exercício 7.4** Em um sistema  $M/M/m/GD/\infty/\infty$ , mostre que a probabilidade de um usuário ter de esperar em fila ao chegar é:

$$P_s = \sum_{n=m}^{\infty} P_n = \frac{\frac{(m\rho)^m}{m!} \left( \frac{1}{1-\rho} \right)}{\sum_{n=0}^{m-1} \frac{(m\rho)^n}{n!} + \frac{(m\rho)^m}{m!} \left( \frac{1}{1-\rho} \right)}$$

Esta expressão é conhecida por fórmula C de Erlang. Determine o valor desta probabilidade de saturação do sistema  $P_s$  para o caso da cabine telefônica do Exercício 7.3 (com  $m = 2$  telefones).

**Exercício 7.5** Derive o modelo  $M/M/\infty/GD/\infty/N$  (população finita) e mostre que a distribuição de equilíbrio  $P_n$  e o valor esperado  $E(L)$  do número de usuários no sistema é:

$$P_n = \frac{C_n^N (\lambda / \mu)^n}{(1 + \lambda / \mu)^N} \quad n = 0, 1, 2, \dots, N \quad \text{e} \quad E(L) = \frac{N\lambda / \mu}{1 + \lambda / \mu}$$

Compare os resultados com os do sistema  $M/M/\infty/GD/\infty/\infty$  descrito na Seção 7.5.

**Exercício 7.6** Derive o modelo  $M/M/m/GD/m/N$  com  $m \leq N$  e mostre que a distribuição de equilíbrio  $P_n$  (conhecida como distribuição de Engset) é:

$$P_n = \frac{C_n^N (\lambda / \mu)^n}{\sum_{k=0}^m C_k^N (\lambda / \mu)^k} \quad n = 0, 1, 2, \dots, m$$

Compare os resultados com os dos sistemas  $M/M/m/GD/m/\infty$  e  $M/M/m/GD/K/K$  descritos na Seção 7.5.

**Exercício 7.7** Derive o modelo  $M/M/m/GD/K/N$  com  $m \leq K \leq N$  e mostre que:

$$P_n = C_n^N \left( \frac{\lambda}{\mu} \right)^n \frac{n!}{m!} m^{m-n} P_0 \quad n = m, m+1, \dots, K$$

Compare os resultados com os do sistema  $M/M/m/GD/K/K$  da Seção 7.5.

**Exercício 7.8** Uma empresa de ônibus urbano está montando uma equipe de mecânicos para manutenção de seus ônibus. A quebra dos ônibus tem distribuição de Poisson com média de 1 ônibus por hora. O tempo de reparo de um ônibus para uma equipe de apenas um mecânico tem distribuição exponencial com média de 2 horas por ônibus. Cada mecânico custa \$10 por hora, e cada ônibus fora de atividade (isto é, parado esperando por manutenção) custa \$40 por hora para a empresa. Ambos os ônibus e os mecânicos trabalham 8 horas por dia. Se a taxa de reparo de ônibus é proporcional ao tamanho da equipe de mecânicos, qual o número de mecânicos que minimiza o custo total esperado por hora?

**Exercício 7.9** Um aeroporto dispõe apenas de uma pista de aterrissagem. Nos períodos de pico, a chegada de aviões no aeroporto segue uma distribuição de Poisson com média de 55 aviões por hora, dos quais 40 aviões são comerciais e os outros 15 são particulares. O tempo de aterrissagem de um avião na pista tem distribuição uniforme no intervalo [48, 72] segundos. Períodos de pico neste aeroporto correspondem a 1.000 horas por ano, e o custo de um minuto de espera em fila para aterrissagem é de \$12 para um avião comercial (devido ao consumo extra de combustível, hora extra de tripulação etc.). Qual o custo anual estimado do tempo de espera em fila de um avião comercial?

**Exercício 7.10** No Exercício 7.9, suponha que o administrador do aeroporto está pensando em aumentar a tarifa de aterrissagem de um avião, para reduzir a congestão do tráfego aéreo do aeroporto nos períodos de pico. Ele acredita que apenas os aviões particulares são sensíveis a (pequenos) aumentos na tarifa, e estima que a taxa de chegada de aviões particulares deve reduzir para  $15 - 0,25x$  aviões por hora,  $0 \leq x \leq 60$ , em que  $x$  é o aumento na tarifa. Qual é o melhor aumento na tarifa do ponto de vista de um avião comercial?

**Exercício 7.11** No Exercício 7.1, imagine que o processo de chegada de usuários na cabine telefônica tem distribuição genérica (e não de Poisson) com mesmo intervalo médio entre chegadas ( $E(X) = 6$  minutos por usuário), mas com coeficiente quadrático de variação menor:  $C_x^2 = 0,5$ . De modo semelhante, considere que o processo de serviço também tem distribuição genérica (e não exponencial) com mesmo tempo médio de serviço ( $E(S) = 5$  minutos por usuário), mas com coeficiente quadrático de variação menor:  $C_s^2 = 0,5$ . Utilize as três aproximações para sistemas  $G/G/1$  descritas na Seção 7.7 para determinar como mudam os valores dos itens (a) e (b). Compare os resultados com os obtidos no Exercício 7.1.

**Exercício 7.12** No Exercício 7.3, considere que o processo de chegada de usuários na cabine telefônica tem distribuição genérica com intervalo médio entre chegadas  $E(X) = 6$  minutos por usuário e coeficiente quadrático de variação  $C_x^2 = 0,5$ . De modo semelhante, considere que o processo de serviço em cada telefone também tem distribuição genérica com tempo médio de serviço  $E(S) = 5$  minutos por usuário e com coeficiente quadrático de variação  $C_s^2 = 0,5$ . Utilize as duas aproximações para sistemas  $G/G/m$  descritas na Seção 7.7 para determinar como mudam os valores dos itens (a) e (b). Compare os resultados com os obtidos no Exercício 7.3.

**Exercício 7.13** Em um ponto de táxi, os táxis chegam (e estacionam na fila de táxis) de acordo com um processo de Poisson com taxa de um táxi por hora, e os passageiros chegam de acordo com um processo de Poisson com taxa de 1,25 passageiro por hora. Os passageiros que chegam e não encontram um táxi no ponto, esperam na fila de passageiros somente se a fila tiver no máximo dois passageiros. Determine: (a) qual o número médio de táxis em fila esperando por passageiros; (b) qual o número médio de passageiros esperando em fila por táxis; e (c) qual o número médio de passageiros por hora que chegam mas não entram na fila porque ela tem três passageiros.

**Exercício 7.14** Um salão de cabeleireiro tem apenas duas cadeiras: uma para lavar o cabelo do cliente (cadeira 1) e outra para cortar o cabelo do cliente (cadeira 2). O cliente, ao chegar no salão, senta na cadeira 1 se ela estiver vazia e, depois de ter seu cabelo lavado, senta na cadeira 2 se ela estiver vazia; caso contrário, espera na cadeira 1 até que a cadeira 2 esteja vazia. Se o cliente, ao chegar, encontrar a cadeira 1 ocupada, ele desiste de entrar no salão. Considere que os clientes chegam no salão de acordo com um processo de Poisson com taxa de um cliente por hora, e que os tempos de atendimento nas cadeiras 1 e 2 sejam exponencialmente distribuídos com taxas de um cliente por hora e dois clientes por hora, respectivamente. Determine: (a) proporção de clientes que chegam e entram no salão; (b) número médio de clientes no salão; e (c) tempo médio de permanência de um cliente que entrou no salão.

## 7.11 RESUMO DOS RESULTADOS DO CAPÍTULO

Modelo  $M/M/1/GD/\infty/\infty$ :

$$P_n = \rho^n(1-\rho) \quad n = 0, 1, 2, \dots$$

$$E(L) = \frac{\rho}{1-\rho}$$

Modelo  $M/M/1/GD/K/\infty$ :

$$P_n = \begin{cases} \frac{\rho^n(1-\rho)}{1-\rho^{K+1}} & \rho \neq 1 \\ \frac{1}{K+1} & \rho = 1 \end{cases} \quad n = 0, 1, \dots, K$$

$$E(L) = \begin{cases} \frac{\rho[1-(K+1)\rho^K + K\rho^{K+1}]}{(1-\rho)(1-\rho^{K+1})} & \rho \neq 1 \\ \frac{K}{2} & \rho = 1 \end{cases}$$

Modelo  $M/M/1/GD/\infty/N$ :

$$P_n = \frac{\left(\frac{\lambda}{\mu}\right)^n \frac{N!}{(N-n)!}}{\sum_{i=0}^N \left(\frac{\lambda}{\mu}\right)^i \frac{N!}{(N-i)!}} \quad n = 0, 1, \dots, N$$

$$E(L) = \sum_{n=0}^N nP_n$$

Modelo  $M/M/m/GD/\infty/\infty$ :

$$P_0 = \frac{1}{\sum_{n=0}^{m-1} \frac{(\rho m)^n}{n!} + \frac{(\rho m)^m}{m!} \left(\frac{1}{1-\rho}\right)}$$

$$P_n = \begin{cases} \frac{(\rho m)^n}{n!} P_0 & n = 1, 2, \dots, m-1 \\ \frac{\rho^n m^m}{m!} P_0 & n = m, m+1, \dots \end{cases}$$

$$E(L) = \rho m + \frac{(\rho m)^m \rho}{m!(1-\rho)^2} P_0$$

Modelo  $M/M/\infty/GD/\infty/\infty$ :

$$P_n = \frac{\left(\frac{\lambda}{\mu}\right)^n e^{-\frac{\lambda}{\mu}}}{n!} \quad n = 0, 1, 2, \dots$$

$$E(L) = \frac{\lambda}{\mu}$$

Modelo  $M/M/m/GD/K/\infty$ :

$$P_0 = \begin{cases} \frac{1}{\sum_{n=0}^{m-1} \frac{(\rho m)^n}{n!} + \frac{(\rho m)^m}{m!} \left( \frac{1 - \rho^{K-m+1}}{1 - \rho} \right)} & \rho \neq 1 \\ \frac{1}{\sum_{n=0}^{m-1} \frac{(\rho m)^n}{n!} + \frac{(\rho m)^m}{m!} (K-m+1)} & \rho = 1 \end{cases}$$

$$P_n = \begin{cases} \frac{(\rho m)^n}{n!} P_0, & n = 1, 2, \dots, m-1 \\ \frac{\rho^n m^m}{m!} P_0, & n = m, m+1, \dots, K \end{cases}$$

$$E(L) = \begin{cases} \frac{\lambda(1-P_K)}{\mu} + \frac{(\rho m)^m \rho}{m!(1-\rho)^2} [1 - \rho^{K-m+1} - (K-m+1)\rho^{K-m}(1-\rho)] P_0 & \rho \neq 1 \\ \frac{\lambda(1-P_K)}{\mu} + \frac{(\rho m)^m (K-m)(K-m+1)}{2m!} P_0 & \rho = 1 \end{cases}$$

Modelo  $M/M/m/GD/m/\infty$ :

$$P_n = \frac{\frac{(\rho m)^n}{n!}}{\sum_{i=0}^m \frac{(\rho m)^i}{i!}} \quad n = 0, 1, 2, \dots, m$$

$$E(L) = \frac{\lambda(1-P_m)}{\mu}$$

Modelo  $M/M/m/GD/K/K$ :

$$P_0 = \frac{1}{\sum_{n=0}^{m-1} C_n^K (\rho m)^n + \sum_{n=m}^K C_n^K \frac{\rho^n n! m^m}{m!}}$$

$$P_n = \begin{cases} C_n^K (\rho m)^n P_0 & n = 1, 2, \dots, m-1 \\ C_n^K \frac{\rho^n n! m^m}{m!} P_0 & n = m, m+1, \dots, K \end{cases}$$

$$E(L) = \sum_{n=0}^K n P_n$$

Modelo  $M/G/1/GD/\infty/\infty$ :

$$E(L) = \rho + \frac{\rho^2 + \lambda^2 V(S)}{2(1-\rho)}$$

Modelo  $M/G/1/GD/\infty/\infty$  com chegada em lote:

$$E(L) = \rho + \frac{\lambda E(S)(V(N) + E(N)^2 - E(N)) + \rho^2 + \lambda^2 E(N)^2 V(S)}{2(1-\rho)}, \text{ em que } \rho = \frac{\lambda E(N)}{\mu}$$

Modelo  $M/G/m/GD/m/\infty$ :

$$P_n = \frac{(\rho m)^n}{\sum_{i=0}^m \frac{n!}{i!} (\rho m)^i} \quad n = 0, 1, 2, \dots, m$$

$$E(L) = \frac{\lambda(1-P_m)}{\mu}$$

Modelo  $M/G/\infty/GD/\infty/\infty$ :

$$P_n = \frac{\left(\frac{\lambda}{\mu}\right)^n e^{-\frac{\lambda}{\mu}}}{n!} \quad n = 0, 1, 2, \dots$$

$$E(L) = \frac{\lambda}{\mu}$$

Modelo  $G/G/1/GD/\infty/\infty$ :

$$E(L) \approx \begin{cases} \rho + \frac{\rho^2(C_x^2 + C_s^2)}{2(1-\rho)} & \text{se } C_x^2 \geq 1 \\ \rho + \frac{\rho^2(C_x^2 + C_s^2)}{2(1-\rho)} e^{-\frac{2(1-\rho)(1-C_x^2)^2}{3\rho(C_x^2 + C_s^2)}} & \text{se } C_x^2 < 1 \end{cases}$$

$$E(L) \approx \rho + \frac{\rho^2(1+C_s^2)}{2(1-\rho)} \frac{(C_x^2 + \rho^2 C_s^2)}{(1+\rho^2 C_s^2)}$$

$$E(L) \approx \rho + \frac{\rho^2(1+C_s^2)}{2(1-\rho)} \frac{(2-\rho)C_x^2 + \rho^2 C_s^2}{(2-\rho + \rho C_s^2)}$$

Modelo  $G/G/m/GD/\infty/\infty$ :

$$E(L) \approx \rho m + \frac{\lambda(C_x^2 + C_s^2)}{2} E(L_q)_{M/M/m}$$

$$E(L) \approx \rho m + \frac{E(L_q)_{M/M/m}}{E(L_q)_{M/M/1}} E(L_q)_{G/G/1}$$

Modelo  $M_i/G_i/1/NPRP/\infty/\infty$

$$E(W_{qk}) = \frac{\sum_{i=1}^r \lambda_i (V(S_i) + E(S_i)^2)}{2(1 - \sum_{i=1}^{k-1} \rho_i)(1 - \sum_{i=1}^k \rho_i)} \quad k = 1, 2, \dots, r$$

Modelo  $M_i/M/1/PRP/\infty/\infty$

$$E(W_k) = \frac{1/\mu}{(1 - \sum_{i=1}^{k-1} \rho_i)(1 - \sum_{i=1}^k \rho_i)} \quad k = 1, 2, \dots, r$$

Modelo  $M_i/M/m/NPRP/\infty/\infty$

$$E(W_{qk}) = \frac{\frac{(\rho m)^m}{m\mu(1-\rho)m!} P_0}{(1 - \sum_{i=1}^{k-1} \rho_i)(1 - \sum_{i=1}^k \rho_i)} \quad k = 1, 2, \dots, r$$

# Apêndice

## Elementos de sistemas de equações lineares

Um sistema de equações lineares  $m \times n$ , isto é, com  $m$  equações e  $n$  variáveis, é dado por:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= b_m \end{aligned}$$

o qual é bem definido pela *matriz aumentada* de seus coeficientes:

$$\left[ \begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & & & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{array} \right]$$

Sistemas de equações lineares podem ter: (a) uma única solução; (b) nenhuma solução; ou (c) infinitas soluções. Revisamos este tópico da álgebra linear por meio de alguns exemplos simples. Antes, porém, lembramos as *operações elementares* que alteram os coeficientes do sistema, sem que seu conjunto de soluções seja alterado.

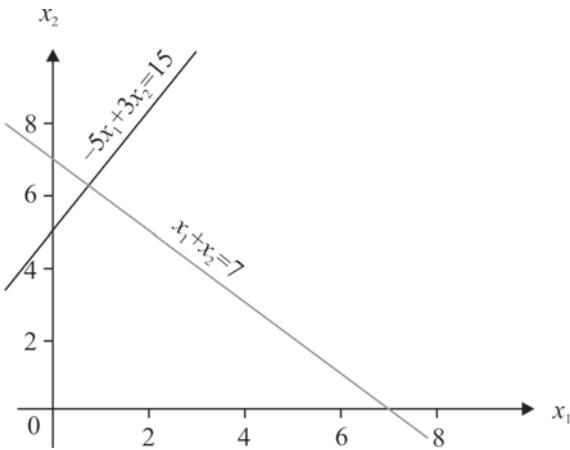
**Definição A.1** (*operações elementares*): As operações elementares são de três tipos (podem ser aplicadas a uma matriz ou a um sistema de equações lineares): (i) multiplique uma linha (ou equação) por um número não-nulo; (ii) adicione uma linha (ou equação) à outra, (iii) troque duas linhas (ou equações) de posição. ■

É intuitivo e pode-se mostrar com rigor matemático que estas operações elementares, aplicadas a um sistema, não alteram sua solução. É usual utilizar uma combinação das duas primeiras operações elementares: adicione um múltiplo de uma linha à outra.

**Exemplo A.1** Considere o seguinte sistema de equações lineares:

$$\begin{cases} x_1 + x_2 = 7 \\ -5x_1 + 3x_2 = 15 \end{cases}$$

Este sistema, com apenas duas variáveis, pode ser facilmente representado em um plano cartesiano ( $x_1, x_2$ ), pois cada uma das equações representa uma reta (Figura A.1), e a solução do sistema é o ponto que satisfaz as duas equações simultaneamente, isto é, a interseção das retas.



**Figura A.1** Representação gráfica do sistema linear com duas variáveis.

O sistema pode ser convenientemente representado pela matriz aumentada:

$$\left[ \begin{array}{cc|c} 1 & 1 & 7 \\ -5 & 3 & 15 \end{array} \right]$$

Resolvemos o sistema pelo método de eliminação de Gauss (escalonamento), bastando multiplicar a primeira linha da matriz aumentada (que corresponde a multiplicar a primeira equação do sistema) por 5 e adicioná-la à segunda linha, para obtermos um sistema equivalente, porém triangular:

$$\left[ \begin{array}{cc|c} 1 & 1 & 7 \\ 0 & 8 & 50 \end{array} \right] \text{ que corresponde ao sistema: } \begin{cases} x_1 + x_2 = 7 \\ 8x_2 = 50 \end{cases}$$

Este sistema triangular superior (elementos abaixo da diagonal são nulos) pode ser resolvido por substituição. Pela segunda equação, temos  $8x_2 = 50$  e, portanto,  $x_2 = 50/8 = 25/4$ . Substituindo o valor de  $x_2$  na primeira equação, temos  $x_1 + 25/4 = 7$  e, então,  $x_1 = 7 - 25/4 = 3/4$ . Assim,

a solução do sistema de equações lineares é:  $\begin{cases} x_1 = 3/4 = 0,75 \\ x_2 = 25/4 = 6,25 \end{cases}$

■

**Exemplo A.2** Considere o seguinte sistema de equações lineares:

$$\begin{cases} x_1 + x_2 + x_3 = 6 \\ 3x_1 + 6x_2 + x_3 = 2 \\ x_1 + 4x_2 + 3x_3 = 10 \end{cases}$$

A representação gráfica deste sistema é possível assim como no Exemplo 3.5, porém, agora são necessários três eixos e o conjunto de pontos que satisfaz cada equação é um plano no  $R^3$ . Para sistemas de dimensões maiores, geralmente é impraticável a representação gráfica, porém são resolvidos analogamente ao exemplo anterior. A matriz aumentada deste sistema é:

$$\left[ \begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 3 & 6 & 1 & 2 \\ 1 & 4 & 3 & 10 \end{array} \right] \xleftarrow{L_1} \left[ \begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 3 & -2 & -16 \\ 1 & 4 & 3 & 10 \end{array} \right] \xleftarrow{L_2} \left[ \begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 3 & -2 & -16 \\ 0 & 1 & 1 & 4 \end{array} \right] \xleftarrow{L_3} \left[ \begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 3 & -2 & -16 \\ 0 & 0 & 1 & 4 \end{array} \right]$$

o qual é resolvido pelo método de eliminação de Gauss em dois passos:

*Passo 1:* Pivô  $a_{11} = 1$ , os multiplicadores são:  $m_{21} = a_{21}/a_{11} = 3$ ,  $m_{31} = a_{31}/a_{11} = 1/1 = 1$ . As novas linhas  $L_2^{(1)}$  e  $L_3^{(1)}$  são dadas por:

$$L_2^{(1)} \leftarrow L_2 - m_{21}L_1$$

$$L_3^{(1)} \leftarrow L_3 - m_{31}L_1$$

$$\left[ \begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 3 & -2 & -16 \\ 0 & 1 & 1 & 4 \end{array} \right] \xleftarrow{L_1^{(1)}} \left[ \begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 3 & -2 & -16 \\ 0 & 0 & 1 & 4 \end{array} \right] \xleftarrow{L_2^{(1)}} \left[ \begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 3 & -2 & -16 \\ 0 & 0 & 1 & 4 \end{array} \right] \xleftarrow{L_3^{(1)}}$$

*Passo 2:* Pivô  $a_{22} = 3$ , o multiplicador é  $m_{32} = a_{32}/a_{22} = 1$ . A nova linha  $L_3^{(2)}$  é dada por:

$$L_3^{(2)} \leftarrow L_3^{(1)} - m_{32}L_2^{(1)}.$$

$$\left[ \begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 3 & -2 & -16 \\ 0 & 0 & 1 & 4 \end{array} \right] \xleftarrow{L_1^{(2)}} \left[ \begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 3 & -2 & -16 \\ 0 & 0 & 1 & 4 \end{array} \right] \xleftarrow{L_2^{(2)}} \left[ \begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 1 & -2/3 & -16/3 \\ 0 & 0 & 1 & 4 \end{array} \right] \xleftarrow{L_3^{(2)}}$$

O sistema triangular superior no segundo passo é resolvido facilmente por substituição: pela última linha, podemos determinar  $x_3$  ( $x_3 = 5$ ) e substituí-lo na segunda equação, que determina  $x_2$  ( $x_2 = -2$ ). Substituindo  $x_3$  e  $x_2$  na primeira equação, temos o valor de  $x_1$  ( $x_1 = 3$ ). Assim, a solução do sistema de equações lineares é:  $(3, -2, 5)$ . ■

Nos Exemplos A.1 e A.2, as matrizes dos coeficientes são invertíveis. A inversa de uma matriz quadrada  $\mathbf{A}$   $m \times m$ , quando existe, é única, é denotada por  $\mathbf{A}^{-1}$  e satisfaz  $\mathbf{AA}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ , em que  $\mathbf{I}$  é a matriz identidade  $m \times m$ . Assim, a solução única de um sistema de equações lineares  $m \times m$ ,  $\mathbf{Ax} = \mathbf{b}$ , cuja matriz dos coeficientes é invertível, pode ser representada por:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

(decorre de  $\mathbf{Ax} = \mathbf{b} \Rightarrow \mathbf{A}^{-1}\mathbf{Ax} = \mathbf{A}^{-1}\mathbf{b} \Rightarrow \mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ )

O leitor deve ter em mente que essa representação da solução única, embora sugira um método numérico de solução (que seria dado por: (i) inverta a matriz  $\mathbf{A}$ ; (ii) multiplique a inversa de  $\mathbf{A}$  pelo vetor  $\mathbf{b}$ ), em geral não é um método numérico eficiente para a resolução de sistemas de equações lineares. Quando a matriz dos coeficientes não tem propriedades especiais, o método mais utilizado é o método de eliminação de Gauss, como feito nos Exemplos A1 e A2.

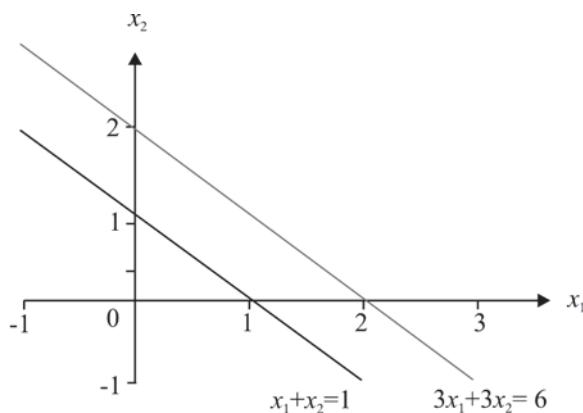
**Exemplo A.3** Considere o seguinte sistema de equações lineares:

$$\begin{cases} x_1 + x_2 = 1 \\ 3x_1 + 3x_2 = 6 \end{cases}$$

ilustrado na Figura A.2, que mostra que não existe solução para este sistema.

Considere a matriz aumentada e o método de eliminação de Gauss (apenas um passo, pois o sistema é  $2 \times 2$ ):

$$\left[ \begin{array}{cc|c} 1 & 1 & 1 \\ 3 & 3 & 6 \end{array} \right] \rightarrow \left[ \begin{array}{cc|c} 1 & 1 & 1 \\ 0 & 0 & 3 \end{array} \right]$$



**Figura A.2** Representação gráfica do sistema de equações lineares sem solução.

O sistema triangular superior equivalente é:

$$\begin{cases} x_1 + x_2 = 1 \\ 0 = 3, \end{cases}$$

o qual não é satisfeito, quaisquer que sejam os valores de  $x_1$  e  $x_2$ . ■

**Exemplo A.4** Considere o seguinte sistema de equações lineares:

$$\begin{cases} x_1 + x_2 + x_3 = 6 \\ 3x_1 + 6x_2 + x_3 = 2 \\ 2x_1 + 2x_2 + 2x_3 = 7 \end{cases}$$

Aplicamos o método de eliminação de Gauss (note que, após apenas um passo, o sistema já está na forma triangular):

$$\left[ \begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 3 & 6 & 1 & 2 \\ 2 & 2 & 2 & 7 \end{array} \right] \rightarrow \left[ \begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 3 & -2 & -16 \\ 0 & 0 & 0 & -5 \end{array} \right]$$

A última linha corresponde à equação  $0x_1 + 0x_2 + 0x_3 = -5$ , a qual não é satisfeita, quaisquer que sejam  $x_1$ ,  $x_2$  e  $x_3$ . Portanto, o sistema original, que tem as mesmas soluções que o sistema escalonado, também não tem solução. ■

Nos Exemplos A.3 e A.4, o conjunto das linhas da matriz dos coeficientes consiste em um conjunto linearmente dependente. Neste caso, dizemos que o posto da matriz não é completo por linhas.

**Definição A.2** (*posto completo por linhas*): Uma matriz  $\mathbf{A} m \times n$  tem posto completo por linhas se o conjunto das  $m$  linhas de  $\mathbf{A}$  for linearmente independente. Denotamos  $\text{posto}(\mathbf{A}) = m$ . ■

A dependência linear das linhas de uma matriz pode ser identificada quando aplicamos o método de eliminação de Gauss (escalonamento por linhas) e obtemos, na forma escalonada, uma ou mais linhas nulas. Assim, para uma matriz  $\mathbf{A} m \times n$  ( $m$  linhas e  $n$  colunas), com  $\text{posto}(\mathbf{A}) = m$  necessariamente  $m \leq n$ , pois se  $n > m$  então surgiriam linhas nulas na forma escalonada e, portanto, o conjunto de linhas seria linearmente dependente.

**Exemplo A.5** Considere o seguinte sistema linear:

$$\begin{cases} x_1 + x_2 = 2 \\ 2x_1 + 2x_2 = 4 \end{cases}$$

cujas soluções são representadas na Figura A3.

Aplicamos o método de eliminação de Gauss na matriz aumentada:

$$\left[ \begin{array}{cc|c} 1 & 1 & 2 \\ 2 & 2 & 4 \end{array} \right] \rightarrow \left[ \begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 0 & 0 \end{array} \right]$$

O sistema triangular equivalente ao original é:

$$\begin{cases} x_1 + x_2 = 2 \\ 0 = 0 \end{cases}$$

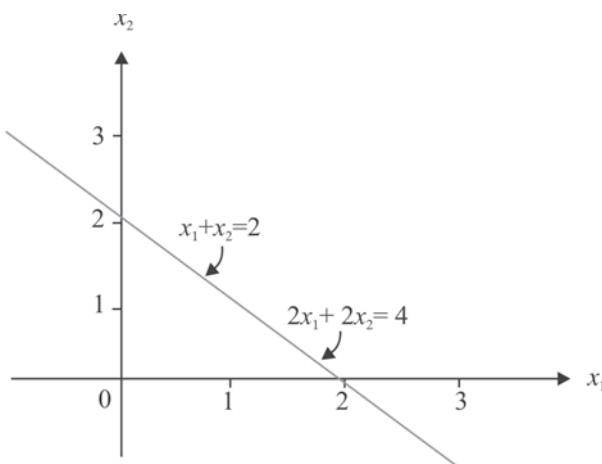
A segunda equação  $0x_1 + 0x_2 = 0$  é satisfeita para quaisquer  $x_1$  e  $x_2$  e, portanto, é redundante, isto é, pode ser retirada sem que afete o conjunto de soluções. Assim, podemos atribuir qualquer valor a  $x_2$  e determinarmos a variável  $x_1$ . Por exemplo, se  $x_2 = 0$  então  $x_1 = 2$ , ou se  $x_2 = 1$  então  $x_1 = 1$  etc. A variável  $x_2$  é chamada *variável independente*. ■

**Exemplo A.6** Considere o seguinte sistema linear:

$$\begin{cases} x_1 + x_2 + x_3 = 6 \\ 3x_1 + 6x_2 + x_3 = 2 \\ 2x_1 + 2x_2 + 2x_3 = 12 \end{cases}$$

Aplicamos o método de eliminação de Gauss:

$$\left[ \begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 3 & 6 & 1 & 2 \\ 2 & 2 & 2 & 12 \end{array} \right] \rightarrow \left[ \begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 3 & -2 & -16 \\ 0 & 0 & 0 & 0 \end{array} \right]$$



**Figura A.3** Representação gráfica do sistema linear com infinitas soluções.

A terceira equação é redundante, pois  $0x_1 + 0x_2 + 0x_3 = 0$  é satisfeita para quaisquer que sejam  $x_1$ ,  $x_2$  e  $x_3$ . O sistema linear tem infinitas soluções, uma vez que podemos atribuir qualquer valor à variável  $x_3$  (variável independente) e, assim, determinamos unicamente as variáveis  $x_2$  e  $x_1$ . Por exemplo, se  $x_3 = 0$ , então,  $x_2 = -\frac{1}{3}$  e  $x_1 = 6 + \frac{1}{3} = \frac{19}{3}$  etc. ■

Nos Exemplos A.5 e A.6, uma equação (poderia ter sido mais de uma) é redundante, isto é, irrelevante para a definição do conjunto das possíveis soluções. Portanto, se eliminarmos tal equação redundante, obtemos um sistema de equações lineares equivalente, com um número menor de equações. Quando o método de eliminação de Gauss é aplicado para a resolução do sistema de equações lineares, essa redundância é detectada na forma escalonada, pois surge uma equação trivial  $0 = 0$ .

**Exemplo A.7** Considere o sistema com duas equações e quatro variáveis:

$$\begin{aligned} x_1 + 2x_2 - x_3 &= 3 \\ -2x_1 + x_2 + x_3 + 2x_4 &= -1. \end{aligned}$$

Podemos escalar este sistema, transformando as colunas pivôs em colunas da matriz identidade:

$$\left[ \begin{array}{ccccc} 1 & 2 & -1 & 0 & 3 \\ -2 & 1 & 1 & 2 & -1 \end{array} \right] \rightarrow \left[ \begin{array}{ccccc} 1 & 2 & -1 & 0 & 3 \\ 0 & 5 & -1 & 2 & 5 \end{array} \right] \rightarrow \left[ \begin{array}{ccccc} 1 & 0 & -\frac{3}{5} & -\frac{4}{5} & 1 \\ 0 & 1 & -\frac{1}{5} & \frac{2}{5} & 1 \end{array} \right]$$

Isto é, o sistema escalonado é dado por:

$$\begin{aligned} x_1 - \frac{3}{5}x_3 - \frac{4}{5}x_4 &= 1 \\ x_2 - \frac{1}{5}x_3 + \frac{2}{5}x_4 &= 1 \end{aligned}$$

A forma escalonada anterior é equivalente a multiplicar o sistema original pela inversa da matriz  $\mathbf{B} = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}$  dos coeficientes das variáveis  $x_1$  e  $x_2$ , algo que faz surgir a matriz identidade como matriz coeficiente das variáveis  $x_1$  e  $x_2$ . Como exercício, o leitor pode calcular  $\mathbf{B}^{-1}$  e fazer a multiplicação. Também pode estender o raciocínio para um sistema qualquer  $m \times n$ . Na forma escalonada, vemos que as duas equações são não-nulas e, portanto,  $\text{posto}(\mathbf{A})=2$ , isto é, o conjunto formado pelas duas linhas de  $\mathbf{A}$  é linearmente independente. Além disso, o sistema original tem o mesmo conjunto de soluções que o sistema escalonado. Assim, podemos determinar soluções para o sistema original pela atribuição de valores às variáveis  $x_3$  e  $x_4$  (variáveis independentes) e determinar prontamente os valores das variáveis  $x_1$  e  $x_2$ .

Por exemplo, se fixarmos  $x_3 = x_4 = 0$ , então  $x_1 = 1$  e  $x_2 = 1$ , ou se  $x_3 = \delta$ ,  $x_4 = 0$ , então  $x_1 = 1 + \frac{3}{5}\delta$  e  $x_2 = 1 + \frac{1}{5}\delta$  etc. O número de variáveis independentes define o grau de liberdade do sistema, neste caso, o grau de liberdade é igual a dois, pois podemos atribuir valores independentes às duas variáveis independentes. Se considerarmos sistemas com posto completo por linhas (sem equações redundantes), como é o caso deste exemplo, o número de variáveis independentes é  $n - m$ , pois, na forma escalonada,  $m$  variáveis têm suas colunas da matriz identidade (variáveis  $x_1$  e  $x_2$ ) e restarão exatamente  $n - m$  variáveis (variáveis  $x_3$  e  $x_4$ ). ■

O escalonamento de matrizes utilizado nos exemplos anteriores (usando operações elementares), que nos permitiram tirar algumas conclusões, agora é colocado em uma forma mais geral, embora sem formalidade matemática. Considere o sistema  $\mathbf{Ax} = \mathbf{b}$ , em que  $\mathbf{A}$  é uma matriz  $m \times n$ . As operações elementares (Definição A.1) aplicadas no sistema (ou, equivalentemente, na matriz aumentada  $(\mathbf{A} \mid \mathbf{b})$ ) permitem escrever o sistema na forma escalonada, o qual tem a mesma solução do sistema original. Temos duas possibilidades:

(i) As linhas nulas na forma escalonada da matriz  $\mathbf{A}$  são as mesmas linhas nulas na forma escalonada da matriz aumentada  $(\mathbf{A} \mid \mathbf{b})$  (Figura A.4).

Linhas nulas indicam equações do tipo  $0x_1 + 0x_2 + \dots + 0x_n = 0$ , as quais podem ser descartadas. O conjunto de linhas não-nulas na forma escalonada da matriz  $\mathbf{A}$  é linearmente independente, como também o conjunto de colunas que contém os pivôs (note que esses conjuntos têm necessariamente o mesmo número de elementos). O número de linhas não-nulas (ou o número de colunas dos pivôs) na forma escalonada da matriz  $\mathbf{A}$  define o  $\text{posto}(\mathbf{A})$ , que é o maior conjunto de linhas de  $\mathbf{A}$  linearmente independentes. A possibilidade (i) anterior pode ser enunciada por  $\text{posto}(\mathbf{A}) = \text{posto}(\mathbf{A} \mid \mathbf{b})$ .

Se fixarmos as variáveis associadas às colunas não-pivôs, temos um sistema resultante (nas variáveis associadas às colunas pivôs) bem determinado (veja Exemplo A.7). As variáveis associadas às colunas não-pivôs são chamadas variáveis independentes e podem assumir quaisquer valores, e as variáveis associadas às colunas pivôs são chamadas variáveis dependentes. Como o número de pivôs é igual ao  $\text{posto}(\mathbf{A})$ , então  $(n - \text{posto}(\mathbf{A}))$  é o número de variáveis independentes.

Se  $\text{posto}(\mathbf{A}) = m$ , isto é, o posto é igual ao número de linhas da matriz  $\mathbf{A}$  (portanto, o máximo possível), dizemos que a matriz  $\mathbf{A}$  tem posto completo por linhas (Definição A.2). Neste caso, o número de variáveis independentes é igual a  $n - m$ . A outra possibilidade é a seguinte:

(ii) O número de linhas não-nulas na forma escalonada da matriz aumentada  $(\mathbf{A} \mid \mathbf{b})$  é maior que o número de linhas não-nulas na forma escalonada da matriz  $\mathbf{A}$  (Figura A.5).

A possibilidade (ii) significa que temos uma equação do tipo  $0x_1 + 0x_2 + \dots + 0x_n = \beta \neq 0$ , que não pode ser satisfeita para quaisquer valores de  $x_1, x_2, \dots, x_n$ . Portanto, o sistema original não tem solução, pois tem as mesmas soluções do sistema na forma escalonada.

Quando consideramos um problema de otimização linear, podemos trabalhar com a seguinte hipótese:

**Hipótese básica:**  $\text{posto}(\mathbf{A}) = m$  (*posto completo por linhas*)

$$\begin{array}{cccccc|c} * & x & x & x & x & \dots & x & x \\ * & x & x & x & \dots & x & x \\ * & x & \dots & x & x \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{array}$$

Posto ( $\mathbf{A}$ )

**Figura A.4** Forma escalonada da matriz aumentada  $(\mathbf{A} \mid \mathbf{b})$ . \* denota um pivô não-nulo e x um elemento qualquer.

$$\begin{array}{cccccc|c|c} * & x & x & x & x & \dots & x & x & \\ * & x & x & x & \dots & x & x & \text{Posto } (\mathbf{A}) & \text{Posto } (\mathbf{A}/\mathbf{b}) \\ * & x & \dots & x & x & & x \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & * & \end{array}$$

**Figura A.5** Forma escalonada da matriz aumentada em que  $\text{posto}(\mathbf{A}) < \text{posto}(\mathbf{A} \mid \mathbf{b})$ .

Esta hipótese não perde generalidade, já que  $\text{posto}(\mathbf{A}) < m$  indica equações redundantes, que podem ser eliminadas (ou o sistema é impossível, indicando que não há solução factível e, portanto, o problema está resolvido). Se a matriz for quadrada, isto é,  $m = n$ , então o sistema  $\mathbf{Ax} = \mathbf{b}$  tem solução única e, portanto, não há o problema de otimização a ser resolvido, já que a solução é única. Além disso, com a hipótese de que  $\text{posto}(\mathbf{A}) = m$ , o número de equações deve ser menor ou igual ao número de variáveis, ou seja,  $m \leq n$ , pois, caso contrário, na forma escalonada apareceriam necessariamente linhas nulas e, então,  $\text{posto}(\mathbf{A}) < m$  (considere uma matriz cujo desenho é um retângulo em pé e reflita sobre sua forma escalonada).

Em resumo:

- Podemos supor, sem perda de generalidade, que a matriz dos coeficientes de um sistema de equações lineares, que define as restrições de um problema de otimização, tem posto completo por linhas, ou seja,  $\text{posto}(\mathbf{A}) = m$ .
- A hipótese de que  $\text{posto}(\mathbf{A}) = m$  implica  $m \leq n$ , isto é, o número de equações é menor ou igual ao número de variáveis. Para o caso trivial  $m = n$ , o sistema tem solução única e, portanto, a resolução do problema de otimização é trivial.
- Quando consideramos um problema de otimização linear na forma padrão (Seção 2.4.1 do Capítulo 2), admitimos, sem perda de generalidade, que  $m < n$  (é comum que  $m \ll n$ ; lese  $m$  muito menor que  $n$ ). Com isso, o sistema linear  $\mathbf{Ax} = \mathbf{b}$  tem sempre infinitas soluções e o grau de liberdade é  $n - m$  (isto é, o número de variáveis independentes).

# Referências Bibliográficas

---

1. Aarts, E.H.L.; Lenstra, J.K. "Local Search in Combinatorial Optimization". Chichester: Wiley, 1997.
2. Aarts, E.H.L.; Korst, J.H.M.; van Laarhoven, P.J.M. "Simulated annealing". In: Aarts, E.H.L.; Lenstra, J.K. (Ed.), *Local Search in Combinatorial Optimization*. Chichester: Wiley, 1997, p. 91-120.
3. Achterberg, T.; Koch, T.; Martin, A. "Branching rules revisited". *Operations Research Letters*, n. 33, p. 42-54, 2005.
4. Adler, I.; Karmarkar, N.; Resende, M.G.C.; Veiga, G. "An implementation of Karmarkar's algorithm for linear programming". *Mathematical Programming*, n. 44, p. 297-335, 1989.
5. Ahuja, R.K.; Magnanti, T.L.; Orlin, J.B. *Network Flows*. Nova Jersey: Prentice Hall, 1993.
6. Ahuja, R.K.; Cunha, C.B. "Very large-scale neighborhood search for the K-constraint multiple knapsack problem". *Journal of Heuristics*, n. 11, p. 465-481, 2005.
7. Alcântara, A.A.M.; Santanna, A.P.; Lins, M.P.E. "Restringindo flexibilidade de pesos em DEA utilizando análise de regressão MSE". *Pesquisa Operacional*, n. 23, p. 347-357, 2003.
8. Almeida, A.T. "Multicriteria decision making on maintenance: spares and contracts planning". *European Journal of Operational Research*, n. 129, p. 235-241, 2001.
9. Almeida, M.R.; Hamacher, S.; Pacheco, M.A.C.; Velasco, M.B.R. "Applying genetic algorithms to the production scheduling of a petroleum refinery". *International Journal of Industrial Engineering*, n. 10, p. 35-44, 2003.
10. Alvarenga, A. G.; Gomes, F. J. N.; Mestria, M. "Metaheuristic methods for a class of the facility layout problem". *Journal of Intelligent Manufacturing*, n. 11, p. 421-430, 2000.
11. Alvarez-Valdes, R.; Parreño, F.; Tamarit, J. M. "A Tabu Search Algorithm for the Pallet Loading Problem". *OR Spectrum*, n. 27, 1, p. 43-61, 2005.
12. Alvim, A.C.F.; Glover, F.; Ribeiro, C.C.; Aloiso, D. J. "A hybrid improvement heuristic for the bin packing problem". *Journal of Heuristics*, n. 10, p. 205-229, 2004.
13. Anderson, E.J.; Glass, C.A.; Potts, C.N. *Machine scheduling*. In: Aarts, E.H.L.; Lenstra, J.K. (Ed.). *Local Search in Combinatorial Optimization*. Chichester: Wiley, 1997, p. 361-414.
14. Arabeyre, J.P.; Fearnley, J.; Steiger, F.C.; Teather, W. "The Airline Crew Scheduling Problem: A survey". *Transportation Science*, n. 3, p. 140-163, 1969.
15. Araújo, S.A.; Arenales, M. "Planejamento e programação da produção numa fundição cativa automatizada de grande porte". *Investigação Operacional*, n. 24, 2004, p. 197-210.
16. Araújo, S.A.; Arenales, M.N.; Clark, A.R. "Dimensionamento de lotes e programação do forno numa fundição de pequeno porte". *Gestão & Produção*, n. 11, v. 2, p. 65-176, 2004.
17. Arce, A.; Ohishi, T.; Soares, S. "Optimal dispatch of generating units of the Itaipú hydroelectric plant". *IEEE Transactions on Power Systems*, n. 17, p. 154-158, 2002.
18. Armentano, V.A.; Arroyo, J.E.C. "An application of a multi-objective tabu search algorithm to a bicriteria flowshop problem". *Journal of Heuristics*, n. 10, p. 463-481, 2004.
19. Armentano, V.A.; França, P.M.; Toledo F.M.B. "A network flow model for the capacitated lot-sizing problem". *Omega*, n. 27, p. 275-284, 1999.
20. Armentano, V.A.; Scrich, C.R. "Tabu search for minimizing total tardiness in a job shop". *International Journal of Production Economics*, n. 63, p. 131-140, 2000.

21. Athamtürk, A. e Sanelsberg, M.W.P. Integer - programming software systems. *Annals of Operations Research*, n. 150, p. 67-124, 2005.
22. Avellar, J.V.G.; Milioni, A.Z.; Rabello, T.N. "Modelos DEA com variáveis limitadas ou soma constante". *Pesquisa Operacional*, n. 25, p. 135-150, 2005.
23. Baker, K.R. *Introduction to Sequencing and Scheduling*. Nova York: Wiley, 1974.
24. Balbo, A.R.; Baptista, E.C.; Arenales, M. "An Adaptation of the dual-affine interior point method for the surface flatness problem". *European Journal of Operational Research*, 2006, a ser publicado.
25. Balestieri, J.A.P.; Correia, P.B. "Multiobjective linear model for pre-feasibility design of cogeneration systems". *Energy – The International Journal*, v. 22, n. 5, p. 537-549, 1997.
26. Banks, J. *Handbook of Simulation*. Atlanta, GA: John Wiley & Sons, 1998.
27. Barcia, R.M.; Cândido, M.A.; Khator, S. "A genetic algorithm based procedure for more realistic job shop scheduling problems". *International Journal of Production Research*, v. 36, n. 12, p. 3.427-3.457, 1998.
28. Baskett, F.; Chandy, K.M.; Muntz, R.R.; Palacios, F.G. "Open, closed, and mixed networks of queues with different classes of customers". *Journal of the Association for Computing Machinery*, v. 22, n. 2, p. 248-260, 1975.
29. Bazaraa, M.S.; Jarvis, J.J.; Sherali, H.D. *Linear Programming and Network Flows*. 2. ed. Nova York: Wiley, 1990.
30. Beasley, J. "An Exact Two-Dimensional Non Guillotine Cutting Tree Search Procedure". *Operations Research*, n. 33, p. 49-64, 1985.
31. Becker, C.; Scholl, A. "A survey on problems and methods in generalized assembly line balancing". *European Journal of Operational Research*, n. 168, p. 694-715, 2006.
32. Bellman, R.; Dreyfus, S. *Applied Dynamic Programming*. Princeton: Princeton University Press, 1962.
33. Benders, J. "Partitioning procedures for solving mixed-integer programming problems". *Numerische Mathematik*, n. 4, p. 238-252, 1962.
34. Bertsekas, D.P. *Dynamic Programming and Optimal Control*. 3. ed. Belmont, Massachusetts: Athena Scientific, 2005, v. 1.
35. Bertsekas, D.P. *Dynamic Programming and Optimal Control*. 2. ed. Belmont, Massachusetts: Athena Scientific, 2001, v. 2.
36. Birgin, E. G.; Morabito, R; Nishihara, F. "A note on an L-approach for solving the manufacturer's pallet loading problem". *Journal of the Operational Research Society* n. 56, p. 1.448-1.451, 2005.
37. Birgin, E. G.; Ronconi, D. P.; Martínez, J. M. "Optimizing the Packing of Cylinders into a Rectangular Container: A Nonlinear Approach". *European Journal of Operational Research*, n. 160, p. 19-33, 2005.
38. Bertsimas, D.; Tsitsiklis, J. N. *Introduction to Linear Optimization*. Belmont, Massachusetts: Athena Scientific, 1997.
39. Bitran, G.; Morabito, R. "Open queueing networks: Optimization and performance evaluation models for discrete manufacturing systems". *Production and Operations Management*, n. 5, p. 163-193, 1996.
40. Bitran, G.; Morabito, R. "An overview of trade-off curves in manufacturing system design" *Production and Operations Management*, n. 8, p. 56-75, 1999.
41. Bitran, G.R.; Yanasse, H.H. "Computational complexity of the capacitated lot size problem" *Management Science*, n. 28, p. 1174-1186, 1982.
42. Bitran, G.R.; Yanasse, H.H. "Deterministic approximations to stochastic production problems". *Operations Research*, n. 32, p. 999-1018, 1984.
43. Blazewicz, J.; Dror, M.; Weglarz, J. "Mathematical programming formulations for machine scheduling: A survey". *European Journal of Operational Research*, n. 51, p. 283-300, 1991.
44. Blazewicz, J.; Ecker, K.H.; Pesch, E.; Schmidt, G.; Weglarz, J. *Scheduling Computer and Manufacturing Processes*, Heidelberg: Springer, 1996.
45. Blazewicz, J.; Kubiak, W.; Szwarcfiter, J. L. "Scheduling unit time tasks on flow shops under resource constraints". *Annals of Operations Research*, n. 16, p. 255-266, 1988.
46. Blum, C.; Roli, A. "Metaheuristics in combinatorial optimization: Overview and conceptual comparison" *ACM Computing Surveys*, n. 35, p. 268-308, 2003.
47. Boaventura Netto, P.O. *Grafos - Teoria, Modelos, Algoritmos*. 3. ed. São Paulo: Edgard Blücher Ltda, 2003.
48. Borenstein, D.; Betencourt, P.R. "A Multi-criteria model for the justification of IT investments", *Infor*, v. 43, n. 1, p. 1-21, 2005.
49. Bornstein, C.T.; Namen, A.A. "Uma ferramenta para avaliação de resultados de diversos modelos de otimização de dietas". *Pesquisa Operacional*, n. 24, p. 445-465, 2004.
50. Bornstein, C.T.; Yanasse, H.H.; Abreu, N.M.M. "Roberto Galvão – 60 anos – reflexões, retrospectiva e reconhecimento". *Pesquisa Operacional*, n. 24, p. 3-6, 2004.
51. Boxma, O. J.; Rinnooy Kan, A; Van Vliet, M. "Machine allocation problems in manufacturing networks". *European Journal of Operational Research*, n. 45, p. 47-54, 1990.

52. Bradley, S.P.; Hax, A.C.; Magnanti, T.L. *Applied Mathematical Programming*. Reading: Addison Wesley, 1977.
53. Brahimi, N.; Dauzere-Peres, S.; Nagid, N.M.; Nordli, A. "Single item lot sizing problems". *European Journal of Operational Research*, n. 168, p. 1-16, 2006.
54. Brearley, A.L.; Mitra, G.; Williams, H.P. "An analysis of mathematical programs prior to applying the simplex method". *Mathematical Programming*, n. 7, p. 263-282, 1973.
55. Bregalda, P.F.; Oliveira, A.A.F.; Bornstein, C.T. Introdução à programação linear. 3. ed. Rio de Janeiro: Editora Campus, 1988.
56. Bronson, R.; Naadimuthu, G. *Operations Research*. Nova York: McGraw-Hill, 1997.
57. Buzacott, J.; Shanthikumar, J. *Stochastic Models of Manufacturing Systems*. Nova Jersey: Prentice-Hall, 1993.
58. Caixeta-Filho, J.V.; van Swaay-Neto, J.M.; Wagemaker, A.P. "Optimization of the production planning and trade of lily flowers at Jan de Wit Company". *Interfaces*, n. 32, p. 35-46, 2002.
59. Campello, R.E.; Maculan Filho, N. "Lagrangean relaxation for a lower bound to a set partitioning problem with side constraints". *Discrete Applied Mathematics*, n. 18, p. 119-136, 1987.
60. Campello, R.E.; Ribeiro, C.C.; Pereira, M.V.F.; Abreu, N.M.M. "Nelson Maculan Filho: cientista e educador". *Pesquisa Operacional*, n. 23, p. 5-28, 2003.
61. Carnieri, C.; Guillermo, A.; Gavinho, L. "Solution procedures for cutting lumber into furniture parts". *European Journal of Operational Research*, n. 73, p. 495-501, 1994.
62. Carvalho, M.F.; Silva Filho, O.S.; Fernandes, C.A. "O planejamento da manufatura - práticas industriais e métodos de otimização". *Gestão & Produção*, n. 5, p. 34-59, 1998.
63. Cavichia, M.C.; Arenales, M.N. "Piecewise linear programming via interior points". *Computers and Operations Research*, n. 27, p. 1.303-1.324, 2000.
64. Cheung, W.; Zhou, W. "Using genetic algorithms for job shop scheduling with sequence-dependent setup times". *Annals of Operations Research*, n. 107, p. 65-81, 2001.
65. Chiyoshi, F.; Galvão, R. D. "A statistical analysis of simulated annealing applied to the p-median problem". *Annals of Operations Research*, n. 96, p. 61-74, 2000.
66. Christofides, N.; Hadjiconstantinou, E. "An exact algorithm for orthogonal 2-D cutting problems using guillotine cuts". *European Journal of Operational Research*, n. 83, p. 21-38, 1995.
67. Chvátal, V. "Edmonds polytopes and a hierarchy of combinatorial problems". *Discrete Mathematics*, n. 4, p. 305-337, 1973.
68. Conover, W.J. *Practical Nonparametric Statistics*. Nova York: Wiley, 1999.
69. Cook, S.A. "The complexity of theorem-proving procedures". *Proceedings of the 3rd Annals ACM Symposium on Theory of Computing*, Nova York: Association for Computing Machinery, p. 151-158, 1971.
70. Cordier, C., Marchand, H., Landy, R. e Wolsey, L.A. *bc-opt*: a branch-and-cut code for mixed integer programs. *Mathematical Programming*, Ser. A 86, 335-353, 1999.
71. Correa, E.S.; Steiner, M.T.A.; Freitas, A.A.; Carnieri, C. "A genetic algorithm for solving a capacitated p-median problem". *Numerical Algorithms*, n. 35, p. 373-388, 2004.
72. Costa, A.M. "A survey on benders decomposition applied to fixed-charge network design problems". *Computers and Operations Research*, n. 32, p. 1429-1450, 2005.
73. Costa Neto, P.L., *Estatística*. São Paulo: Edgard Blucher, 1985.
74. Croes, G.A. "A method for solving travelling salesman problems". *Operations Research*, n. 6, p. 791-812, 1958.
75. Dakin, R. "A tree search algorithm for mixed integer programming". *Computer Journal*, n. 8, p. 250-255, 1965.
76. Dantzig, G.B.; Wolfe, P. "Decomposition principle for linear programs". *Operations Research*, n. 8, p. 101-111, 1960.
77. Dantzig, G.B. *Linear Programming and Extensions*. Nova Jersey: Princeton University Press, 1963.
78. Denardo, E.V. *Dynamic Programming, Models and Application*. Nova Jersey: Prentice-Hall, 1982.
79. Desaulniers, G.; Desrosiers, J.; Solomon, M.M. *Column Generation*. Nova York: Springer, 2005.
80. Desrosiers, J.; Soumis, F.; Desrochers, M. "Routing with time window by column generation". *Networks*, n. 14, p. 545-565, 1984.
81. Dias, T.M.; Ferber, D.F.; Souza, C.C.; Moura, A.V. "Constructing nurse schedules at large hospitals". *International Transactions in Operational Research*, n. 10, p. 245-265, 2003.
82. Diestel, R. *Graph Theory*. 3. ed., Heidelberg: Springer, 2005.
83. Disney, R.L.; Farrell, R.L.; Morais, P.R. "A characterization of M/G/1 queues with renewal departure processes". *Management Science*, v. 19, n. 11, p. 1.222-1.228, 1973.
84. Drexel, A.; Kimms, A. "Lot sizing and scheduling – Survey and extensions". *European Journal of Operational Research*, n. 99, p. 221-235, 1997.
85. Dyckhoff, H.; Finke, U. *Cutting and Packing in Production and Distribution: Typology and Bibliography*. Springer Verlag, 1992.

86. Dyckhoff, H.; Scheithauer, G; Terno, J. *Cutting and Packing*. In: Amico, M.; Maffioli, F.; Martello, S. (Eds.), *Annotated Bibliographies in Combinatorial Optimization*. Nova York: John Wiley & Sons, 1997, p. 393-414.
87. Easton, K.; Nemhauser, G.L.; Trick, M.A. *The traveling tournament problem: description and benchmarks*. In: Walsh T. (Ed.), "Principles and Practice of Constraint Programming" *Lecture Notes in Computer Science*, v. 2.239, p. 580-584, Nova York: Springer, 2001.
88. Eilon, S.; Christofides, N. "The loading problem". *Management Science*, n. 17, p. 259-268, 1971.
89. Ensslin, L.; Corrêa, E.C.; Vansnick, J.C.; Bana e Costa, C. A. "Decision support systems in action: integrated application in a multicriteria decision aid process". *European Journal of Operational Research*, n. 113, p. 585-594, 1999.
90. Erlang, A. "Solution of some problems in the theory of probabilities of some significance in automatic telephone exchanges". *Post Office Electrical Engineer's Journal*, n. 10, p. 189-197, 1917.
91. Feller, W. *An Introduction to Probability Theory and its Applications*. Nova York: Wiley, 1957.
92. Ferreira, C.E.; Martin, A.; Weismantel, R. "Solving Multiple Knapsack Problems". *SIAM Journal on Optimization*, n. 6, p. 858-877, 1996.
93. Ferreira, C.E.; Souza, C.C.; Wakabayashi, Y. "Rearrangement of DNA fragments: a branch-and-cut algorithm". *Discrete Applied Mathematics*, n. 116, p. 161-177, 2002.
94. Ferreira, P.A.V. "Solving multiple objective problems in the objective space". *Journal of Optimization Theory and Applications*, v. 89, n. 3, p. 659-680, 1996.
95. Ferreira Filho, V.J.M.; Galvão, R.D. "A tabu search heuristic for the concentrator location problem". *Location Science*, n. 6, p. 189-210, 1999.
96. Fisher, M.; Jaikumar, R. "A generalized assignment heuristic for vehicle routing". *Networks*, n. 11, p. 109-124, 1981.
97. Fontanella, G.; Morabito, R. "Analyzing the tradeoff between investing in service channels and satisfying the targeted user service for Brazilian internet service providers". *International Transactions in Operational Research*, n. 9, p. 247-260, 2002.
98. Ford, R.L.; Fulkerson, D.R. *Flows in Networks*. Princeton: Princeton University Press, 1962.
99. Friedlander, A.; Lyra, C.; Tavares, H.; Medina, E.L. "Optimization with staircase structure – an application to generation to scheduling". *Computers and Operations Research* v. 17, n. 2, p. 143-152, 1990.
100. França, P.M.; Gendreau, M.; Laporte, G.; Muller, F.M. "A tabu search for the multiprocessor scheduling problem with sequence dependent setup times". *International Journal of Production Economics*, n. 43, p. 79-89, 1996.
101. França, P.M.; Sosa, N.G.; Pureza, V. "An Adaptive tabu search algorithm for the capacitated clustering problem". *International Transactions in Operational Research*, n. 6, p. 665-678, 1999.
102. French, S. *Sequencing and Scheduling: An Introduction to the Mathematics of the Job Shop*. Chichester: Wiley, 1982.
103. Fukasawa, R.; Longo, H.; Lysgaard, J.; Aragão, M.P.; Reis, M.; Uchoa, E.; Werneck, R.F. "Robust branch-and-cut-and-price for the capacitated vehicle routing problem". *Mathematical Programming*, a ser publicado, 2005.
104. Galvão, R.D. "Uncapacitated facility location problems: contributions". *Pesquisa Operacional*, 24, p. 7-38, 2004.
105. Galvão, R.D.; Chiyoshi, F; Morabito, R. "Towards unified formulations and extensions of two classical probabilistic location problems" *Computers and Operations Research*, v. 32, n. 1, p. 15-33, 2005.
106. Garey, M.R.; Johnson, D.S. *Computers and Intractability*. Nova York: Freeman, 1979.
107. Garfinkel, R.S.; Nemhauser, G.L. *Integer Programming*. Nova York: Wiley, 1972.
108. Gass, S.I. *Linear Programming: Methods and Applications*, New York, 1958.
109. Gass, S.I. *An Illustrated Guide to Linear Programming*. Nova York: McGraw-Hill, 1970.
110. Geoffrion, A.M e Graves, G.W. "Multicommodity distribution system design by Benders decomposition". *Management Science*, n. 20, p. 822-844, 1974.
111. Geoffrion, A.M. "Lagrangian relaxation for integer programming algorithms: A framework and state-of-the art survey". *Mathematical Programming Study*, n. 2, p. 82-114, 1974.
112. Gheorghe, A.V. *Decision Processes in Dynamic Probabilistic Systems*. Massachusetts: Kluwer, 1990.
113. Ghiani, G.; Laporte, G.; Musmanno, R. *Introduction to Logistics Systems Planning and Control*. Inglaterra: John Wiley & Sons, 2004.
114. Gilmore, P.C.; Gomory, R.E. "A linear programming approach to the cutting stock problem". *Operations Research*, n. 9, p. 849-859, 1961.
115. Gilmore, P.C.; Gomory, R.E. "A linear programming approach to the cutting stock problem – Part II". *Operations Research*, n. 11, p. 863-888, 1963.
116. Glover, F.; Hultz, J.; Klingman, D. "Improved computer-based planning techniques – Part II". *Interfaces*, n. 9, p. 12-20, 1979.
117. Glover, F.; Laguna, M. *Tabu Search*. Massachusetts: Kluwer, 1997.
118. Goldbarg, M.C.; Luna, H.P.L. *Otimização Combinatória e Programação Linear*. Rio de Janeiro: Editora Campus, 2000.

119. Golden, B; Liberatore, M.; Lieberman, C. "Models and solutions techniques for cash flow management". *Computers and Operations Research*, n. 6, p. 13-20, 1979.
120. Golveia, L.; Voß, S. "A classification of formulations for the (time-dependent) traveling salesman problem". *European Journal of Operational Research*, n. 83, p. 69-82, 1995.
121. Gomes, C.F.S. "Using MCDA methods THOR in an application for outranking the ballast water management options". *Pesquisa Operacional*, n. 25, p. 11-28, 2005.
122. Gomes, M.J.N.; Palhanob, A. "The capacitated centred clustering problem". *Computers and Operations Research*, n. 33, p. 1639-1663, 2006.
123. Gomory, R.E. "Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, n. 64, p. 275-278, 1958.
124. Gomory, R.E. *An algorithm for integer solutions to linear programs*. In: Graves, R.; Wolfe, P. (Ed.) *Recent Advances in Mathematical Programming*. Nova York: McGraw-Hill, 1963, p. 269-302.
125. Gonçalves Filho, E.V.; Marçola, J.A. "Annualized hours as a capacity planning tool in make-to-order or assembly-to-order environment: an agricultural implements company case". *Production Planning & Control*, v. 12, n. 4, p. 388-398, 2001.
126. Gonzaga, C.C. "Search Directions for Interior Linear Programming Methods". *Algorithmica*, n. 6, p. 153-181, 1991.
127. Gramani, M.C.N.; França, P.M. "The combined cutting stock and lot-sizing problem in industrial processes". *European Journal of Operational Research*, a ser publicado, 2006.
128. Gross, D.; Harris, C. *Fundamentals of Queueing Theory*. Nova York: Wiley, 1985.
129. Guan, M. "Grafic programming using odd and even points", *Chinese Mathematics*, n. 1, p., 273-277, 1962.
130. Hadley, G. *Nonlinear and Dynamic Programming*. Reading, Massachusetts: Addison Wesley, 1964.
131. Haghani, A.; Jung, S. "A dynamic vehicle problem with time-dependent travel times". *Computers and Operations Research*, n. 32, p. 2.959-2.986, 2005.
132. Hall, N.G.; Sriskandarajah, C. *Operations Research*, n. 44, p. 510-525, 1996.
133. Hakimi, S.L. "Optimum locations of switching centers and the absolute centers and medians of a graph". *Operations Research*, n. 12, p. 450-459, 1964.
134. Hakimi, S.L. "Optimum distribution of switching centers in a communication network and some related graph theoretic problems". *Operations Research*, n. 13, p. 462-475, 1965.
135. Hax, A.; Candea, D. *Production and Inventory Management*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
136. Hillier, F.S e Lieberman, G.J. *Introduction to Operations Research*. 8. ed. Nova York: Mac-Graw Hill, 2004.
137. Hoto, R.; Arenales, M; Maculan, N. "The compartmentalised knapsack problem: a case study". *European Journal of Operational Research*, a ser publicado no Special Issue on Cutting and Packing, 2006.
138. Humes, C.; Kumar, P.R. "The delay of open markovian queueing networks: Uniform functional bounds, heavy traffic pole multiplicities, and stability". *Mathematics of Operations Research*, v. 22, n. 4, p. 921-954, 1997.
139. Jackson, J. "Networks of waiting lines". *Operations Research*, n. 5, p. 518-521, 1957.
140. Jackson, J. "Job shop-like queueing systems". *Management Science*, n. 10, p. 131-142, 1963.
141. Johnson, L.A.; Montgomery, D.G. *Operations Research in Production Planning, Scheduling and Inventory Control*. Nova York: Wiley, 1974.
142. Jungnickel, D. *Graphs, Networks and Algorithms*. New York: Springer, 1999.
143. Kantorovich, L. "Mathematical methods of organizing and planning production" (translated from a report in Russian, dated 1939). *Management Science*, n. 6, p. 366-422, 1960.
144. Karimi, B.; Fatemi Ghomi, S.M.T.; Wilson, J.M. "The capacitated lot sizing problem: a review of models and algorithms". *Omega*, n. 31, p. 365-378, 2003.
145. Karp, R.M. *Reducibility among combinatorial problems*. In: Miller, R.E.; Thatcher, J.W. (Ed.) *Complexity of Computer Computations*. Nova York: Plenum Press, 1972, p. 85-193.
146. Kirby, M.W. *Operational Research in War and Peace*. Londres: Imperial College Press, 2003.
147. Kiyuzato, L.; Ronconi, D.P.; Salamoni, R.; Tsai, C.K.; Yoshizaki, H. "Minimizing total tardiness: a case study in autoparts factory". *International Transactions in Operational Research*, n. 9, p. 371-379, 2002.
148. Kleinrock, L. *Queueing Systems – Theory*. Nova York: Wiley, 1975. v. 1.
149. Kleinrock, L. *Queueing Systems: Computer Applications*. Nova York: Wiley, 1976. v. 2.
150. Klose, A.; Drexl, A. "Facility location models for distribution system design". *European Journal of Operational Research*, n. 162, p. 4-29, 2005.
151. Kohayakawa, Y.; Miyazawa, F.K.; Raghavan, P.; Wakabayashi, Y. "Multidimensional cube packing". *Algorithmica*, v. 40, n. 3, p. 173-187, 2004.
152. Kolisch, R.; Padman, R. "An integrated survey of deterministic project scheduling". *Omega*, n. 29, p. 249-272, 2001.

153. Krarup, J. e Bilde, O. Plant location, set cone-ring and economic lot sizes: An  $O(mn)$  algorithm for structured problems, em L. Collatz et al. (eds.). *Optimierung bei graphentheoretischen und ganzzahligen Problemen*, Birkhauser Verlag, Basel, p. 155-180, 1977.
154. Kwok, Y.; Ahmad, I. "On multiprocessor task scheduling using efficient state space search approaches" *Journal of Parallel and Distributed Computing*, n. 65, p. 1.515-1.532, 2005.
155. Lachtermacher, G. *Pesquisa Operacional na Tomada de Decisões*. 2. ed. Rio de Janeiro: Editora Campus, 2004.
156. Land, A.; Doig, A. "An automatic method of solving discrete programming problems". *Econometrica*, n. 28, p. 497-520, 1960.
157. Larson, R.; Odoni, A. *Urban Operations Research*. Nova Jersey: Prentice-Hall, 1981.
158. Law, A.M.; Kelton, W.D. *Simulation Modeling and Analysis*. 2. ed. Nova York: McGraw-Hill, 1991.
159. Lawler, E.L.; Lenstra, J.K.; Rinnoy Kan, A.H.G.; Shmoys, D.B. "Sequencing and scheduling: algorithms and complexity". In: Graves, S.C. Rinnoy Kan, A.H.G. Zipkin, P.H. (Ed.) "Logistics of Production and Inventory". *Handbooks in Operations Research and Management Science*, Amsterdã: North-Holland, v. 4, p. 445-522, 1993.
160. LeBlanc, L.J.; Shtub, A; Anandalingam, G. "Formulating and solving production planning problems". *European Journal of Operational Research*, n. 112, p. 54-80, 1999.
161. Ledyard, J. O.; Olson, M.; Porter, D.; Swanson, J.A; Torma, D.P. "The first use of a combined value auction for transportation services". *Social Science Working Paper*, n. 1.093, Division of the Humanities and Social Sciences, California Institute of Technology, Pasadena, CA, 2000.
162. Lin, S.; Kernighan, B.W. "An effective heuristic algorithm for the traveling-salesman problem". *Operations Research*, n. 21, p. 498-516, 1971.
163. Linneroth, J.T e Savelsbergh, M.W.P. "A computational study of search strategies for mixed integer programming". *INFORMS Journal on Computing*, n. 11, p. 173-185, 1999.
164. Linhares, A.; Yanasse, H.H.; Torreão, J.R.A. "Linear gate assignment: a fast statistical mechanics approach". *IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems*, n. 18, p. 1.750-1.758, 1999.
165. Lins, M.P.E.; Silva, A.C.M.; Rosa, L.P. "Regional variation in energy consumption of appliances: conditional demand analysis applied to Brazilian households". *Annals of Operations Research*, n. 117, p. 235-246, 2003.
166. Lins, L.; Lins, S.; Morabito, R. "An  $n$ -tet graph approach for non-guillotine packings of  $n$ -dimensional boxes into an  $n$ -container". *European Journal of Operation Research*, n. 141, p. 421-439, 2002.
167. Little, J.D. "A proof for the queueing formula:  $L = \lambda W$ ". *Operations Research*, n. 9, p. 383-387, 1961.
168. Little, J.D.C.; Murty, K.G.; Sweeney, D.W.; Karel "An algorithm for the traveling salesman problem". *Operations Research*, n. 11, p. 972-989, 1963.
169. Loiola, E.; Abreu, N.M.M.; Boaventura Netto, P.O. "Uma revisão comentada das abordagens do problema quadrático de alocação". *Pesquisa Operacional*, n. 21, p. 73-110, 2004.
170. Lucena, A. "The time-dependent traveling salesman problem – The deliveryman case". *Networks*, n. 20, p. 753-763, 1990.
171. Magalhães, M.N.; Lima, AC. *Noções de Probabilidade e Estatística*. 3. ed. São Paulo: IME-USP, 2001.
172. Maros, I. *Computational Techniques of the simplex method*. Kluwer Academic Publishers, 2003.
173. Marins F.A.S.; Senne, E.L.F.; Machado, A.; Perin, C.; Darbydown, K. "Algorithms for network piecewise-linear programs: A comparative study". *European Journal of Operational Research Society*, n. 97, p. 183-199, 1997.
174. Marques, F.; Arenales, M. "O problema da mochila compartimentada e aplicações". *Pesquisa Operacional*, v. 22, n. 3, p. 285-304, 2002.
175. Martí, R.; Laguna, M.; Glover "Principles of scatter search". *European Journal of Operational Research*, n. 169, p. 359-372, 2006.
176. Martinez, J.M.; Santos, L.T.; Santos, S.A. "Problemas minimax e aplicações". *Matemática Universitária*, n. 16, p. 46-59, 1994.
177. Mateus, G.R.; Luna, H.P.L.; Sirihal, B. "Heuristics for distribution network design in telecommunication". *Journal of Heuristics*, n. 6, p. 131-148, 2000.
178. Meier, H. "Simultaneous lot sizing and scheduling by combining local search with dual reoptimization". *European Journal of Operational Research*, n. 120, p. 311-326, 2000.
179. Meyer, P. L. *Probabilidade: Aplicações à Estatística*. Rio de Janeiro: Livros Técnicos e Científicos Ed., 1983.
180. Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. 3. ed. Nova York: Springer, 1996.
181. Minieka, E. *Otimization Algorithms for Networks and Graphs*. New York: Marcel Dekker, 1978.
182. Miyazawa, F.K.; Wakabayashi, Y. "Parametric online algorithms for packing rectangles and boxes". *European Journal of Operational Research*, n. 150, p. 281-293, 2003.

183. Moccellin, J.V.; Santos, M.O. "An adaptive metaheuristic for permutation flowshop scheduling." *Control and Cybernetics*, n. 29, p. 761-771, 2000.
184. Moder, J.J.; Philips, C.R.; Davis E.W. *Project Management with CPM, PERT and Precedence Diagramming*. 3. ed. Nova York: Van Nostrand Reinhold, 1983.
185. Möhring, R.H.; Stork, F. "Linear preselective policies for stochastic project scheduling". *Mathematical Methods of Operations Research*, n. 52, p. 501-515, 2000.
186. Morabito, R.; Arenales, M. "Optimizing the cutting of stock plates in a furniture company". *International Journal of Production Research*, n. 38, 12, p. 2725-2742, 2000.
187. Morabito, R.; Belluzzo, L. "Optimizing the cutting of wood fibre plates in the hardboard industry". *European Journal of Operational Research*, 2006.
188. Morabito, R.; Lima, F. "A Markovian queueing model for the analysis of user waiting times in supermarket checkouts". *International Journal of Operations and Quantitative Management*, n. 10, p. 165-177, 2004.
189. Morton, T.E.; Pentico, D. *Heuristic Scheduling Systems*. Nova York: Wiley, 1993.
190. Munhoz, J.R.; Morabito, R. "A goal programming model for frozen concentrated orange juice production and distribution system". *OPSEARCH*, v. 38, n. 6, p. 630-646, 2001.
191. Murty, K.G. *Operations Research*. Nova Jersey: Prentice Hall, 1995.
192. Nagano, M. S.; Moccellin, J. V. "A high quality solution constructive heuristic for flow shop sequencing". *Journal of the Operational Research Society*, n. 53, p. 1.374-1.379, 2002.
193. Nahmias, S. *Production and Operations Analysis*. Homewood, IL: Irwin, 1995.
194. Nemhauser, G.L. *Introduction to Dynamic Programming*. Nova York: Wiley, 1966.
195. Nemhauser, G.L.; Wolsey, L.A. *Integer and Combinatorial Optimization*. Nova York: Wiley, 1999.
196. Nicholson, T.A.J. *Optimization in Industry: Optimization Techniques*. London: Longman Press, v. 1, Chapter 10, 1971.
197. Novaes, A. G. *Sistemas Logísticos: Transporte, Armazenagem e Distribuição Física de Produtos*. São Paulo: Edgard Blucher, 1989.
198. Novaes, A. G. "Rapid-transit efficiency analysis with the assurance-region DEA method". *Pesquisa Operacional*, v. 21, n. 2, p. 179-198, 2001.
199. Nowicki, E.. "The permutation flow shop with buffers: A tabu search approach". *European Journal of Operational Research*, n. 116, p. 205-219, 1999.
200. Oliveira, J. F.; Ferreira, J. S. "An improved version of Wang's algorithm for two-dimensional cutting problems". *European Journal of Operational Research*, n. 44, p. 256-266, 1990.
201. Oliveira, A.R.L.; Lyra, C. "Interior point methods for the polynomial L\_infinite fitting problem". *International Transactions in Operational Research*, v. 11, n. 3, p. 309-322, 2004.
202. Owen, S.H.; Daskin, M.S. "Strategic facility location: A review". *European Journal of Operational Research*, n. 111, p. 423-447, 1998.
203. Padberg M.; Rinaldi G "Optimization of 532-city in symmetric TSP". *Operations Research Letters*, n. 6, p. 1-7, 1987.
204. Parada, V.; Palma, R.; Sales, D; Alvarenga, A.G. "A comparative numerical analysis for the guillotine two-dimensional cutting problem". *Annals of Operations Research*, n. 96, p. 245-254, 2000.
205. Pearn, W.L.; Chung, S.H.; Chen, A.Y; Yang, M.H. "A case study on the multistage IC final testing scheduling problem with reentry". *International Journal of Production Economics*, n. 88, p. 257-267, 2004.
206. Perin, C.; Podestá-Gomes, V.; Moretti, A.C. "Uma experiência com geração de colunas em problemas de corte de estoque". *TEMA – Tendências em Matemática Aplicada e Computacional*, v. 4, n. 2, p. 217-226, 2003.
207. Pinedo, M.L. *Scheduling – Theory, Algorithms and Systems*. Nova Jersey: Prentice Hall 1995.
208. Pinedo, M.L. *Planning and Scheduling in Manufacturing and Services*. Nova York Springer, 2005.
209. Pirkul, H. "An integer programming model for the allocation of databases in a distributed computer system". *European Journal of Operational Research*, n. 26, p. 401-411, 1986.
210. Pizzolato, N.D.; Barcelos, F.B; Lorena, L.A.N. "School location methodology in urban area of developing countries". *International Transactions in Operational Research*, n. 11, p. 667-681, 2004.
211. Puccini, A. L.; Pizzolato, N. D. *Programação linear*. Livros Técnicos e Científicos Editora, 1987.
212. Pureza, V.; Morabito, R. "Some experiments with a simple tabu search algorithm for the manufacturer's pallet loading problem". *Computers and Operations Research*, n. 33, p. 804-819, 2006.
213. Ralphs, T.K. On the mixed Chinese postman problem, *Operations Research Letters*, n. 14, p. 123-127, 1993.
214. Rangel, L.A.D.; Gomes, L.F.A.M.; Lins, M.P.E.; Fuks, S. "Avaliação dos programas de pós-graduação em engenharia da UFRJ empregando uma variante desenvolvida do método UTA". *Pesquisa Operacional*, n. 23, p. 285-299, 2003.

215. Rangel, M.S.; Ferreira, D. "Um modelo de dimensionamento de lotes para uma fábrica de refrigerantes". *TEMA: Tendências em Matemática Aplicada e Computacional*, n. 4, p. 237-246, 2003.
216. Rardin, R.L. *Optimization in Operations Research*. Nova Jersey: Prentice Hall, 1998
217. Ravagnani, M.A.S.S.; Silva, A.P.; Arroyo, P.A.; Constantino, A.A. "Heat exchanger network synthesis and optimisation using genetic algorithm". *Applied Thermal Engineering*, n. 25, p. 1003-1017, 2005.
218. Ree, S.; Yoon, B.S. "A two-stage heuristic approach for the newspaper delivery problem. *Computers and Industrial Engineering*, n. 30, p. 501-509, 1996.
219. Reeves, C.R. "Genetic algorithms for the operations researcher". *INFORMS Journal on Computing*, n. 9, p. 231-250, 1997.
220. Resende, M.G.C.; Ribeiro, C.C. Greedy randomized adaptive search procedures. In: *Handbook of Metaheuristics*, F. Glover; G. Kochenberger (Ed.). Massachusetts: Kluwer, 2003, p. 219-249.
221. Resende, M.G.C.; Werneck, R.F. "A hybrid heuristic for the p-median problem". *Journal of Heuristics*, n. 10, p. 59-88, 2004.
222. Ribeiro, C.C.; Urrutia S. *Heuristics for the mirrored traveling tournament problem*, a ser publicado, 2005.
223. Ribeiro, C.C.; Souza, M.C. "Tabu search for the Steiner problem in graphs". *Networks*, n. 36, p. 138-146, 2000.
224. Ribeiro, J.L.D.; Fogliatto, F.S.; Caten, C.S. "Minimizing manufacturing and quality costs in multiresponse optimization". *Quality Engineering*, v. 13, n. 2, 191-201, 2001.
225. Rodrigues, R.C.; Carvalho, S.V. "Optimization of a PH/PH/1/N queue with controllable service time". *International Transactions in Operational Research*, n. 8, p. 235-248, 2001.
226. Rodrigues, M.M; Souza, C.C.; Moura, A.V. "Vehicle and crew scheduling for urban bus lines". *European Journal of Operational Research*, n. 170, p. 844-862, 2006.
227. Ronconi, D.P.; Armentano, V.A. "Lower bounding schemes for flowshops with blocking-in process". *Journal of the Operational Research Society*, n. 52, p. 1.289-1.297, 2001.
228. Ross, S. *Introduction to Probability Models*. San Diego, CA: Academic Press, 1993.
229. Ruggiero-Gomes, M.A.; Martinez, J. M.; Moretti, A.C. "Comparing Algorithm for Solving Non-linear Systems of Equations". *SIAM Journal of Scientific and Statistical Computing*, v. 13, n. 2, p. 459-483, 1992.
230. Saliby, E.; Funchal, G. "Estudo da eficiência da amostragem descritiva na simulação de um problema de filas". *Pesquisa Operacional*, v. 6, n. 2, p. 65-78, 1986.
231. Santoro, M. C.; Pacheco, R. F. "Proposta de classificação hierarquizada dos modelos de solução para o problema de job shop scheduling". *Gestão & Produção*, São Paulo, v. 6, n. 1, p. 1-15, 1999.
232. Santos, H.G.; Ochi, L.S.; Souza, M.J.F. "An efficient tabu Search heuristic for the school timetabling problems". *Lecture Notes in Computer Science*, n. 3.059, p. 468-482, 2004.
233. Santos-Meza, E.; Santos, M. O.; Arenales, M. "A lot-sizing problem in an automated foundry". *European Journal of Operational Research*, v. 139, n. 3, p. 490-500, 2002.
234. Savelsbegh, M.P.W. "Preprocessing and probing techniques for mixed integer programming problems". *ORSA Journal on Computing*, n. 6, p. 445-454, 1994.
235. Schrage, L.E. *Optimization Modeling with LINDO*. 5. ed. Pacific Grove, CA: Duxbury Press, 1997.
236. Shapiro, J. *Modeling the supply chain*. Pacific Grove CA: Duxbury Press, 2001.
237. Scholl, A.; Becker, C. "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing". *European Journal of Operational Research*, n. 116, p. 666-693, 2006.
238. Senne, L.F., Lorena, L.A.N.; Pereira, M.A. "A branch-and-price approach to p-median location problems". *Computers and Operations Research*, n 32, p. 1.655-1.664, 2005.
239. Shimizu, T. *Decisões nas Organizações*. São Paulo: Atlas, 2001.
240. Shtub, A.; Kogan, K. "Capacity planning by the dynamic multi-resource generalized assignment problem" *European Journal of Operational Research*, n. 105, p. 91-99, 1998.
241. Silva, J.L.C.; Soma N.Y.; Maculan, N. "A greedy search for the three-dimensional bin packing problem: the packing static stability case". *International Transactions in Operational Research*, n. 10, p. 141-153, 2003.
242. Silver, E.A. "An overview of heuristic solution methods". *Journal of the Operational Research Society*, n. 55, p. 936-956, 2004.
243. Simuany-Stern, Z.; Winer, I. "The one dimensional cutting stock problem with two objectives". *Journal of the Operational Research Society*, n. 45, p. 231-236, 1994.
244. Soares de Mello, J.C.C.B.; Gomes, E.G.; Leta, F.R.; Soares de Mello, M.H.C. "Algoritmo de alocação de recursos discretos com análise de envoltória de dados". *Pesquisa Operacional*, v. 26, n. 1, 2006.
245. SOBRAPO (1993). *25 anos da pesquisa operacional no Brasil*. Edição comemorativa do Jubileu de Praia da Sobrapo. Rio de Janeiro: Sobrapo, 1993.

246. Soma, N.Y.; Toth, P. "An exact algorithm for the subset sum problem". *European Journal of Operational Research*, n. 136, p. 57-66, 2002.
247. Souza, K.X.S.; Armentano, V.A. "Multi item capacitated lot sizing by a cross decomposition based algorithm". *Annals of Operations Research*, n. 50, p. 557-574, 1994.
248. Sousa, R.S.; Silva, C.T.L.; Arenales, M. "Métodos do tipo dual simplex para problema de otimização linear canalizados". *Pesquisa Operacional*, v. 25, n. 3, p. 349-382, 2005.
249. Stafford Jr, E.F.; Tseng F.T. "Two models for a family of flowshop sequencing problems" *European Journal of Operational Research*, n. 142, p. 282-293, 2002.
250. Stafford Jr, E.F.; Tseng F.T.; Gupta, J.N.D. "Comparative evaluation of MILP flowshop model". *Journal of the Operational Research Society*, n. 56, p. 88-101, 2005.
251. Stecke, K.E. "Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems". *Management Science*, n. 29, p. 273-288, 1983.
252. Stern, J. M. "Simulated Annealing with a Temperature Dependent Penalty Function". *ORSA Journal On Computing*, v. 4, n. 3, p. 311-319, 1992.
253. Strusevich, V.A. "Shop scheduling problems under precedence constraints". *Annals of Operations Research*, n. 69, p. 351-357, 1997.
254. Sueri, C.; Stadtler, H. "The capacitated lot-sizing problem with linked lot sizes". *Management Science*, n. 49, p. 1.039-1.054, 2003.
255. Szwarcfiter, J.L. "Job shop scheduling with unit time operations under resource constraints and release dates". *Discrete Applied Mathematics*, n. 18, p. 227-233, 1987.
256. Taha, H.A. *Operations Research*. 7. ed. Nova Jersey: Prentice Hall, 2003.
257. Takeda, R.; Widmer, J.; Morabito, R. "Analysis of ambulance decentralization in urban emergency medical service using the hypercube queueing model". *Computers and Operations Research*, v. 34(3), p. 727-741, 2006.
258. Taube Netto, M. "Integrated Planning for Poultry Production at Sadia". *Interfaces*, v. 26, p. 38-53, 1996.
259. Tijms, H. *Stochastic Modeling and Analysis: A computational approach*. Nova York: John Wiley & Sons, 1986.
260. Tijms, H.C. *Stochastic Models – An Algorithmic Approach*, Nova York: Wiley, 1995.
261. Toregas, C.; Swain, R.; Revele, C.; Bergman, L. "The location of emergency service facilities". *Operations Research*, n. 19, p. 1.363-1.373, 1971.
262. Toso, E.; Morabito, R. "Otimização do dimensionamento e sequenciamento de lotes de produção: Estudo de caso numa fábrica de rações". *Gestão & Produção*, v. 12, n. 2, p. 203-217, 2005.
263. Toth, P.; Vigo, D. *The Vehicle Routing Problem*. Philadelphia: SIAM, 2002.
264. Vanderbeck, F.; Savelsbergh, M.P.W. "A generic view of Dantzig-Wolfe decomposition in mixed integer programming". *Operations Research Letters*, a ser publicado, 2006.
265. Vanderbei, R.J. *Linear Programming – Foundations and Extensions*. 2. ed. Kluwer Academic Publishers, 2001.
266. Vries, S.; Vohra, R.V. "Combinatorial auctions: A survey". *INFORMS Journal on Computing*, n. 15, p. 284-309, 2003.
267. Waescher, G.; Haussner, H.; Schumann, H. "An improved typology of cutting and packing problems". *European Journal of Operational Research*, a ser publicado no Special Issue on Cutting and Packing, 2006.
268. Wagner, H.M. *Pesquisa Operacional*. 2. ed. Rio de Janeiro: Prentice-Hall do Brasil, 1986.
269. Wagner, H.M.; Whitin, T.M. "Dynamic version of the economic lot size model". *Management Science*, n. 5, p. 89-96, 1958.
270. White, D.J. *Dynamic Programming*. San Francisco: Holden-Day, 1969.
271. Whitt, W. "The queueing network analyzer". *The Bell System Technical Journal*, n. 62, p. 2.779-2.815, 1983.
272. Whitt, W. "Approximations for the GI/G/m queue". *Production and Operations Management*, n. 2, p. 114-161, 1993.
273. Williams, H.P. *Model Building in Mathematical Programming*. 4. ed. Chichester: Wiley, 1999.
274. Winston, W.L. *Operations Research*. 4. ed. Toronto: Thomson, 2004.
275. Wolsey, L.A. *Integer Programming*. Nova York: Wiley, 1998.
276. Xavier, A.E.; Oliveira, A.A. "Optimal covering of plane domains by circles via hyperbolic smoothing". *Journal of Global Optimization*, n. 31, p. 493-504, 2004.
277. Yamashita, D.S.; Armentano, V.A.; Laguna, M. "Scatter search for project scheduling with resource availability cost" *European Journal of Operational Research*, n. 169, p. 623-637, 2006.
278. Yanasse, H.H. "A search strategy for the one-size assortment problem". *European Journal of Operational Research*, n. 74, p. 135-142, 1994.
279. Yanasse, H.H.; Soma, N.Y. "A new enumeration scheme for the knapsack problem". *Discrete Applied Mathematics*, n. 18, p. 235-245, 1987.

280. Yanasse, H.H.; Soma, N.Y.; Maculan, N. "An algorithm for determining the K-best solutions of the one-dimensional knapsack problem". *Pesquisa Operacional*, n. 20, p. 117-134, 2000.
281. Yanasse, H.H.; Zinober, A.S.I.; Harris, R.G. "Two dimensional cutting stock with multiple stock sizes". *Journal of the Operational Research Society*, n. 42, p. 673-683, 1991.
282. Yanasse, H.H.; Zinober, A.S.I.; Harris, R.G. "Cutting stock problems: position of free cut". *Investigación Operativa*, n. 4, p. 69-84, 1994.
283. Yoshizaki, H.T.Y., Muscat, A.R.N.; Biazzi, J.L. "Decentralizing ethanol distribution in south eastern Brazil". *Interfaces*, Estados Unidos, v. 26, n. 6, p. 24-34, 1996.
284. Zambon, K.L.; Carneiro, A.A.F.M.; Silva, A.N.R.; Negri, J.C. "Análise de decisão multicritério na localização de usinas termoelétricas utilizando SIG". *Pesquisa Operacional*, v. 25, n. 2, p. 183-199, 2005.
285. Zenios, S. A. *Financial Optimization*. Cambridge University Press, 1993.

# Índice Remissivo

---

- Ajuste de curvas, 44, 47, 48, 156  
Algoritmo, 84, 85, 97, 137, 247, 251, 254, 259, 307, 313, 322, 333, 340-344  
Algoritmo aproximado, 164, 178  
Algoritmo eficiente, 233  
    Não eficiente, 233  
Algoritmos de planos de corte, 251, 274  
Algoritmo de Dijkstra, 304, 307-309, 312-314, 322  
Algoritmo de Floyd, 322, 323, 369  
Algoritmo de Ford, 313, 314, 317, 340, 342, 344  
Algoritmo de Ford e Fulkerson, 342, 344, 347, 348  
Algoritmo genético, 271-273  
Algoritmo de Kruskal, 333  
Análise, 124, 441  
    em equilíbrio, 441  
    transiente, 451  
Árvore, 240, 246, 248, 250, 257, 260, 292, 331  
    Definição, 292  
    geradora, 292, 293  
Árvore *branch-and-bound*, 246, 248, 274  
Árvore geradora mínima, 331-336  
    formulação matemática, 16, 21, 26, 29, 33, 40, 304, 331  
    algoritmo de Kruskal, 333  
Atraso de tarefa, 228  
Atraso máximo, 216, 217, 219  
Avanço de tarefa, 217, 219  
  
Balanceamento de linha de montagem, 227  
Base, 356, 357  
*Branch-and-bound*, 239, 246, 274, 248, 257, 260  
    limítante superior, 251, 257, 260  
    variáveis de ramificação, 244, 260  
    nó filho, 248, 250  
    nó ativo, 259  
*Branch-and-cut*, 256, 259,  
*Branch-and-price*, 516  
Busca tabu, 272  
  
Cadeia, 291  
  
Caixeiro viajante, 186, 188, 189, 191  
    grafo não-orientado-simétrico, 186  
    grafo orientado, 400  
    m-caixeiro-viajante, 188  
    aquisição, 189  
    lucro, 191  
Caminho, 291, 298, 299, 302, 303, 307, 309-314, 322, 328, 329, 341  
    Definição, 291  
    mínimo, 297, 298, 338  
    máximo, 327  
    mais longo, 328, 329  
Cardinalidade, 186, 187, 289  
Carteiro chinês, 192-194  
    grafo não-orientado, 186, 189, 191-193, 290  
    grafo orientado, 187, 188, 194, 195, 204, 208, 228, 290, 294, 399, 400  
Ciclo, 186, 193, 291, 355, 362  
    Definição, 291  
Círculo, 291, 304, 314, 317, 318, 322  
    Definição, 291  
Cláusula, 233  
Cobertura, 181, 182  
Coeficiente quadrático de variação, 496  
Colônia de formigas, 272  
Complexidade computacional, 232  
Condição de otimalidade, 79, 82, 83, 97, 121, 122, 134, 356, 357, 361, 364-366  
Condição de otimalidade dual, 135  
Conjunto convexo, 235  
Conservação de estoque, 30, 32  
Controle ótimo, 46  
Corte, 174-178, 239  
    Unidimensional, 175  
    Bidimensional, 176  
    de um grafo, 348  
Corte de Gomory, 251, 253, 256  
CPM, 328, 329, 331  
Custo fixo, 7, 10, 33, 167, 301  
Custo fixo em fluxos em redes, 204

- Custos relativos, 79, 355, 356, 359-361, 363  
 Curva de *trade-off*, 433, 434
- Decisões, 1, 3-5, 7, 26, 30, 31, 35, 50, 146, 181, 387  
 Decomposição de Benders, 274  
 Decomposição de Dantzig-Wolfe, 274  
 Degeneração, 65-67, 77-79, 363  
 Derivadas parciais, 44, 125  
 Designação, 21, 178, 180, 181  
   Generalizada, 179, 180  
   múltiplos recursos, 180  
   múltiplos níveis, 180  
   quadrática, 181  
 Desigualdade válida, 253  
 Desvio padrão, 28, 449  
 Diagrama de transição, 451, 452, 455, 459, 460, 463, 467, 468, 471, 492  
 Dimensionamento de lotes, 12, 29, 30, 33, 47, 206, 210, 213, 238, 264, 266, 267, 269, 273, 376  
   um item, sem restrição de capacidade, 206  
   múltiplos itens e restrição de capacidade, 208  
   múltiplos níveis, 180, 208, 209, 211  
   múltiplos itens e preservação de preparação, 209  
 Dimensionamento e programação de lotes, 29, 33  
   problema discreto, 213  
   problema geral, 214  
 Direção, 58, 80, 132  
   Simplex, 80  
   dual simplex, 135, 138-140, 145  
 Distribuição, 408, 429, 437-439, 442, 445  
   Acumulada, 445, 446  
   de Engset, 473, 495  
   de Erlang, 495,  
   exponencial, 446-449, 454, 475  
   genérica, 439, 476, 479, 483, 496  
   geométrica, 446, 455  
   normal, 28, 331  
   de Poisson, 429, 447, 448-450, 453, 466, 467, 470, 473, 479  
   uniforme, 496  
 Dual, 110, 115, 118, 120, 126, 128, 130  
 Dualidade forte, 123  
 Disciplina, 437  
   da fila, 437, 438  
   de prioridade sem interrupção, 483, 487  
   de prioridade com interrupção, 485
- EDD  
 Eliminação de nós, 246  
 Empacotamento, 39, 40 174, 178, 180, 181, 182, 184, 185, 375, 402  
 Envoltória convexa, 235, 236, 238, 251  
 Equação recursiva, 388, 389, 392, 394-396, 400, 407, 408, 423, 425  
   Progressiva, 388, 389, 392, 394, 395  
   regressiva, 388, 395, 396, 407, 408
- Escalonamento de matriz, 506  
 Estados, 386, 387  
 Estado de equilíbrio, 441, 452, 455  
   Transiente, 441, 451  
 Estágios, 299, 300, 301, 386-389  
 Estoque em processo, 215, 438, 445, 493  
 Estratégia simplex, 79, 80-82, 87  
 Estratégia dual simplex, 132, 134-137
- Fator de utilização, 441, 444, 445, 460, 463, 469, 473, 477, 478, 483, 484, 487, 489, 491, 493  
 FCFS, 494, 495  
 Fluxo de caixa, 35  
 Flutuação livre, 330  
 Fluxo máximo, 339, 340, 342, 347-349, 369  
 Fluxo máximo corte mínimo, 369  
 Folga, 53, 68, 69, 85, 91, 94, 98-101, 114, 116, 172  
 Folgas complementares, 122, 123, 161, 162  
 Forma padrão, 50-54, 66, 71, 76, 84, 85, 91, 93, 100, 110, 115  
 Fórmula de, 442  
   Kraemer e Lagenbach-Belz, 481, 482, 483  
   Little, 442-445, 453, 456, 458, 460, 462, 467, 487, 491  
   perda de Erlang, 470, 478, 479  
   Pollaczek e Khintchine, 477, 481, 484  
 Formulações alternativas, 234  
   mochila 0-1, 270, 271, 286, 302  
   localização de facilidades com capacidade ilimitada, 201, 237  
   dimensionamento de lotes de um item sem restrição de capacidade, 267, 269  
 Função, 111, 170-179, 180-186, 190-194, 196, 201, 207, 211, 214, 219, 222, 233, 238, 249, 262  
   Dual, 111-114, 135, 136  
   Lagrangiana, 111  
   linear por partes, 170
- Geração de colunas, 263, 274  
 Gestão financeira, 35, 375  
 Grade de horários, 231, 232  
 Grafo, 192, 194, 227, 229, 289, 290, 293  
   Definição, 289  
   Orientado, 290  
   não-orientado, 290  
   fricamente conectado, 292  
   fortemente conectado, 292  
   representação matricial, 293
- GRASP, 272
- Heurística  
   busca local, 271-272  
   construtiva, 270-271  
   vizinhança, 271-272
- Hiperplano, 59, 252  
 Horizonte de planejamento, 6-8, 20, 29, 31, 33, 35-37, 152-152, 197, 205-206, 228, 232, 280, 376-381

- Índices básicos, 71, 84-85, 87-88, 134, 137, 144-145, 157  
 Índices não-básicos, 84-85, 87-88, 103, 134, 137, 157  
 Indução  
   Progressiva, 381-386  
   Regressiva, 381-382, 385-387  
 Instância, 232-234  
 Instante mais cedo, 34, 329  
 Instante mais tarde, 329-330
- Jogadores, 146-147, 407, 414-415  
 Jogo-da-velha, 414-415, 428  
 Jogo de soma zero, 414
- Lateness*, 215-219  
*Lateness* máximo, 218-219  
 LCFS, 438-439, 458, 483-485  
*Leadtimes* de produção, 493  
 Lista de adjacência, 294-295  
 Linguagens de modelagem, 172  
 Localização de facilidades, ix, xi, 12, 182, 195, 200-202, 204, 237, 273, 287  
 p-medianas, 12, 200-201  
 p-centros, 201  
 capacidade ilimitada, 201, 237  
 capacidade limitada, 201-204, 287  
 fonte única, 202-204, 287  
 Logística, ix, xi, 2, 15, 184, 195
- Makespan*, 215, 220-221, 225, 228, 281  
 Manutenção de equipamentos, 13, 426  
 Matriz, 40, 47, 50-51, 54, 60, 70-71, 77, 80-102, 110-111, 114-115, 118, 123, 129-139, 145-146, 154, 157-160, 163, 182-185, 222-223, 239, 252, 263, 293-296, 324-326, 337-338, 350-354, 369, 501-508  
 de adjacências, 293  
 básica, 71, 84-85, 92-93, 123, 138-139, 145-146, 158-160  
 de incidência nó-arco, 294, 337-338, 350-354, 369  
 de incidência nó-aresta, 293  
 Medidas de desempenho, 13, 215-218, 434-435, 439, 453-457, 460, 462, 464-465, 468, 472-473, 477-478, 482-487, 491-494  
 Meta-heurísticas, 270-274  
 Método de Benders, 265-266, 274  
   problema mestre, 266  
   subproblema, 265  
 Método do caminho crítico – COM, 329  
 Método de Dantzig-Wolfe, 266, 269, 277  
   problema mestre, 268-269, 274  
   subproblema, 266-269  
 Método dual simplex, 66, 110, 129, 135, 137-145, 155-156, 158, 162, 244, 254  
 Método  
   das duas fases, 100-103, 155
- eliminação de Gauss, 69, 77, 83, 90, 92, 96, 502-506  
 heurístico, 9-10, 270  
 hierárquico, 106-108  
 m-grande, 358  
 ótimo, 8-9, 200, 270  
 dos pesos, 106, 109  
 de pontos interiores, 15  
 Métodos de enumeração implícita, 239  
 Método simplex, 1, 12, 15, 66, 71, 74, 79, 82, 84, 86, 89-95, 99-103, 109, 124, 128-129, 134, 149, 154-156, 158-161, 269, 307, 333, 350, 354-358, 368, 371-372, 402, 428  
   para variáveis canalizadas, 93, 356  
   para redes, 350, 354, 368, 371  
   revisado, 92, 160  
   para o problema de transporte, 350  
*Mix* de produção, 26-28, 128, 154  
 Mochila, 12, 172-175, 179, 232-233, 237, 248, 251, 267, 269-271, 287, 302-304, 307, 327, 392, 394-395  
 Inteira, 173  
 Múltiplas, 173  
 Empacotamento, 174-175  
 Modelo markoviano, 13, 423, 425-426, 439, 450, 476, 489  
 Modelo  
   *M/M/1/GD//∞*, 454-455, 458-461, 463-464, 467, 469, 474, 478, 497  
   *M/M/1/GD/K/∞*, 459-461, 468, 497  
   *M/M/1/GD//∞/N*, 461, 497  
   *M/M/m/GD//∞/∞*, 463-468, 489, 495, 497  
   *M/M/∞/GD//∞/∞*, 466-467, 470, 474, 495, 498, 499  
   *M/M/m/GD/K/∞*, 467-468, 470-471, 498  
   *M/M/m/GD/m/∞*, 470, 495, 498  
   *M/M/m/GD/K/K*, 470-472, 495, 499  
   *M/M/∞/GD//∞/N*, 472, 495  
   *M/M/m/GD/m/N*, 472-473, 495  
   *M/M/m/GD/K/N*, 472, 495  
   *M/G/1/GD//∞/∞*, 476, 478, 497, 499  
   *M/G/m/GD/m/∞*, 470, 478-479, 499  
   *M/G/∞/GD//∞/∞*, 467, 479, 499  
   *G/G/1/GD//∞/∞*, 480, 499  
   *G/G/m/GD//∞/∞*, 480, 500  
   *Mi/Gi/1/NPRP//∞/∞*, 483, 500  
   *Mi/M/1/PRP//∞/∞*, 485-486, 500  
   *Mi/M/m/NPRP//∞/∞*, 487, 500  
 Modelo com  
   abandono na espera, 474  
   aceleração do servidor, 474  
   capacidade limitada, 459, 467, 470, 478  
   chegadas em lote, 478  
   desistência na chegada, 474  
   infinitos servidores, 466, 479  
   múltiplas classes de usuários, 13, 483-484  
   múltiplos servidores, 13, 463-464, 467, 470, 473, 478, 487  
   população finita, 461, 470, 495

- prioridade, 483, 485, 487
- serviços em lote, 476
- troca de fila, 438
- Modelo de
  - decisão de filas, 462, 473
  - nascimento e morte, 445, 449-452, 454-455, 459, 461, 463, 467, 470, 474, 476, 492
- Modelo sem fila de espera, 470, 478
- Multicritério, 109,
- Múltiplas soluções ótimas, 8, 64, 89-90, 126, 157-158, 161
- Multiplicadores simplex, 355, 359, 361, 363-367
- Multiobjetivo, ix, 13, 104-106
- Nó
  - cabeça, 290-291, 295, 302-304, 327-328, 348, 351, 354
  - cauda, 290-291, 295-296, 302-303, 327-328, 330, 348, 351, 354
- Notação de Kendall-Lee, 438,
- NPRP, 439
- Número de tarefas atrasadas, 215, 217-218
- Objetivos conflitantes, 104, 106, 146
- Otimização combinatória, 164, 173, 181, 233, 274,
- Otimização recursiva, 388
- Pacotes de otimização, 172, 224
- Partição
  - Básica, 71-74, 76, 78, 80, 82-86, 91, 99, 101-102, 127, 131, 134-135, 137-140, 142-146, 158, 354
- Passo, 80, 81-83, 86, 93, 102-103, 134, 136-140, 145,
- PERT, 228, 331
- Pivotamento, 92, 98, 245-246
- Planejamento estático, 26, 28
- Planejamento estratégico, 200
- Planejamento dinâmico, 29
- Planejamento tático
  - planejamento agregado, 30, 205
  - programa mestre de produção, 205
  - planejamento de recursos de produção, 205
- Planejamento operacional
  - programação da produção, 12, 26, 153, 163, 180, 186, 192, 205, 215, 226, 273, 289, 407, 438
- Planejamento da produção, 1, 12, 13, 15, 26, 29, 31, 110, 153, 206, 228, 270, 298, 375, 376, 378, 381, 383
- Plano ótimo, 377, 380, 381, 382, 383, 384, 385, 387, 389, 397
- Poliedro, 234, 239, 263, 264
- Política, 27, 105, 110, 147, 183, 282, 387, 390, 396, 396, 398, 425, 426, 427, 428, 429
- Definição, 3, 4
- Ótima, 387, 388, 394, 397, 415, 423, 424, 425, 427, 428
- Ponto extremo, 61, 251, 263, 266
- Pré-processamento
  - limitação de variáveis, 260, 261, 262
  - redundância, 262
  - fixação de variáveis, 261, 262
  - infactibilidade, 69, 137, 247, 250, 262
- Princípio da otimalidade de Bellman, 387
- Probabilidade de transição, 425
- Problema artificial, 100, 101, 102, 103, 359
- Problema dual, 110, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 129, 130, 134, 135, 136, 141, 142, 161, 162, 261, 265, 266, 269, 274
- Problema de caminho mínimo
  - entre dois nós, 297
  - entre todos os pares de nós, 322, 323
  - algoritmo de Dijkstra, 304, 307, 308, 309, 312, 313, 314, 368, 369
  - algoritmo de Ford, 313, 314, 317, 340, 342, 344, 347, 348
  - algoritmo de Floyd, 322, 323, 369
- Problema de caminho mais longo, 303, 304, 327, 328, 329, 369
- COM, 228, 329, 331
- PERT, 228, 331
- Problema de designação, 26, 179, 187, 201, 203, 280, 368, 372
- Problema da dieta ou mistura, 16, 17, 18, 20, 150, 151, 152
- Problema da dimensionalidade em programação dinâmica, 402
- Problemas de fluxo
  - de custo mínimo, 9, 12, 181, 189, 191, 192, 193, 195, 205, 266, 298, 331, 336, 367, 376, 378, 388, 399, 400, 473
  - caminho mínimo, 12, 297, 298, 304, 307, 308, 309, 310, 311, 312, 314, 317, 322, 323, 326, 327, 338, 370, 400, 403
  - caminho máximo, 12, 303, 304, 327, 328, 329, 338
  - fluxo máximo, 339, 340, 342, 347, 348, 349, 369
  - Ford e Fulkerson, 2, 12, 194, 340, 342, 344, 347, 348
- Problema de lote, 181, 394-395
- Problema da mochila, 172, 173, 232, 233, 237, 248, 251, 267, 270, 271, 287, 302, 303, 304, 307, 327, 392, 394, 395, 401
- Problema de planejamento da produção, 110, 270, 298
- Problema primal, 110, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 124, 125, 126, 127, 129, 130, 131, 134, 136, 138, 139, 142, 143, 144, 161, 162, 244, 266
- Problema de satisfaibilidade, 233
- Problema de transporte, 21, 22, 24, 202, 331, 349, 350
- Procedimento de Chvátal-Gomory, 252, 274

- Processo de  
chegada, 436, 437, 439, 444, 483, 485, 489, 496  
serviço, 436, 437, 438, 439, 444, 483, 485, 487,  
489, 496
- Processos markovianos de decisão, 423, 425
- Produção, 15, 16, 17, 20, 21, 22, 24, 26, 27, 28, 29,  
30, 32, 33, 37, 38, 39, 40, 42, 50, 104, 110, 126,  
128, 129, 151, 153, 154, 156, 163, 167, 168,  
175, 180, 186, 192, 205, 206, 207, 208, 209,  
210, 211, 213, 214, 215, 216, 220, 222, 224,  
226, 227, 228, 238, 270, 273, 285, 286
- Programação de força de trabalho, 232
- Programação dinâmica determinística, ix, 12, 375,  
388, 408  
conceitos básicos, 386  
equação recursiva progressiva, 388, 389, 392,  
394, 395  
equação recursiva regressiva, 388, 395, 396, 407,  
408  
indução da decomposição, 391  
problema da dimensionalidade, 402
- Programação dinâmica estocástica, ix, 13, 407, 408,  
412, 414, 425  
equação recursiva regressiva, 388, 395, 396, 407,  
408
- Programação inteira, v, 163, 164, 166, 170, 171,  
172, 235, 236, 239, 247, 249, 251, 252, 253,  
273, 274, 287  
mista, 165, 234, 236, 239, 256, 265, 266, 269  
binária, 165, 166
- Programação linear, ix, 1, 2, 4, 12, 149, 165, 166,  
171, 172, 176, 179, 202, 236, 237, 239, 244,  
248, 249, 253, 254, 260, 262, 265, 266, 274,  
405, 426  
forma padrão, 50, 51, 52, 53, 54, 66, 71, 76, 84,  
85, 91, 93, 97, 100, 110, 115, 116, 117, 118,  
124, 128, 137, 138, 143, 151, 154, 508
- Programação da produção, 12, 26, 153, 163, 180,  
186, 192, 205, 215, 226, 273, 289, 407, 438  
uma máquina, 438  
tempo de preparação, 192  
máquinas paralelas, 12, 220, 221, 226, 279  
*job shop*, 12, 180, 215, 222, 223, 224, 225, 226,  
426  
variantes, 225-226  
*flow shop*, 12, 224, 225, 226
- Programação de força de trabalho, 232
- Programação de projetos, 12, 34, 228, 289, 327
- Programação de serviços, 230
- Programação em torneios de esporte, 231
- Programação em transporte, 231
- PRP, 439, 485, 486, 500
- Raios, 91, 263
- Raio extremo, 263, 266
- Rede  
Definição, 289, 290  
Orientada, 290, 291
- Rede de filas, 436, 437, 488, 489, 490, 491, 493,  
494  
Aberta, 488  
Fechada, 489
- Região factível, 51, 55, 56, 57, 60, 61, 62, 63, 64,  
66, 67, 68, 69, 72, 73, 74, 91, 93, 94, 114, 132,  
134, 154, 156, 157, 162, 240, 246, 247, 251,  
254, 260, 355  
Ilimitada, 63, 64
- Regra de Dantzig, 79, 81, 360
- Relaxação, 112, 142, 266, 268, 333  
Lagrangiana, 12, 110, 263, 264, 269, 274  
Linear, 165, 166, 239, 240, 244, 245, 246, 249,  
253, 254, 259, 268, 269, 274, 333
- Relações lógicas, 170
- Restrição  
Adicional, 143, 144, 145, 231, 244, 265, 406  
Ativa, 69, 126, 132, 136, 137  
ativada ou desativada, 169  
básica, 136  
de chance, 27, 28  
disjuntivas, 169, 170, 223  
inativa, 132, 135, 136  
não-básica, 136
- Roteamento em nós, 186
- Roteamento em arcos, 192, 195
- Roteamento de veículos, 12, 178, 195, 197, 199,  
274, 289  
janelas de tempo, 196, 197, 274  
periódico, 197  
múltiplos depósitos, 199
- Scatter search*, 272
- Seleção de nós, 248, 249
- Simplex  
direção, 80, 81, 82, 83, 86, 87, 91, 93, 102, 103  
iteração, 84, 355  
método, 1, 12, 71, 74, 79, 84, 89, 91, 92, 93,  
95, 99, 101, 102, 103, 109, 124, 128, 129, 134,  
149, 154, 156, 269, 307, 333, 350, 354, 355,  
356, 357, 358, 368, 371, 372, 402, 428  
método dual, 12, 66, 110, 129, 132, 134, 135,  
136, 137, 138, 139, 140, 141, 142, 143, 145,  
155, 156, 158, 159, 160, 161, 162, 244, 248,  
254, 255, 258
- Simulação discreta, ix, 4, 13, 438, 476, 482
- Simulated annealing*, 272
- Sistema de fila, 434, 435, 436, 438, 439, 441, 442,  
443, 444, 451, 453, 454, 459, 460, 461, 463,  
464, 467, 477, 484, 486, 489
- Sistemas de reserva, 230
- Sistemas flexíveis de manufatura, 215
- SIRO, 438, 439, 458, 483, 485
- Solução básica  
degenerada, 363  
dual, 125, 134, 135, 137, 138, 139, 140, 142,  
143, 145, 146, 162  
factível, 72, 73, 74, 78, 79, 82, 84, 85, 94, 95,

- 99, 137, 354, 355, 356, 359  
 inicial, 357  
 inicial para problemas de fluxo, 357, 358, 359  
 ótima, 90, 111, 124, 127, 244, 254, 367  
 Solução geral do sistema, 72  
 Solução gráfica, 76, 92, 130  
 Solução ilimitada, 91, 98  
 Solução ótima, 8, 9, 18, 26, 37, 38, 45, 51, 52,  
   54-114, 120-271, 322, 358, 367, 372, 400-406,  
   426  
 Soma dos atrasos, 217, 218, 226  
 Soma dos atrasos e avanços, 217, 218  
 SPT, 216, 219, 226, 438, 455, 484, 485
- Tabela simplex, 94, 95, 96, 97, 98  
 Tamanho de lote, 181, 395  
 Tamanho do passo, 80, 81, 83, 87, 93, 102, 103,  
   136, 145  
 Tamanho do passo dual, 137  
 Taxa de  
   chegada, 440, 441, 443, 444, 447, 449, 450,  
   452, 453, 455, 460, 461, 463, 471, 476, 477,  
   478, 486, 489, 491, 494, 496  
   entrada, 443, 444, 455, 460, 462, 463, 468, 472,  
   477  
   perda, 460, 469  
   serviço, 438, 440, 441, 450, 452, 453, 455, 457,  
   462, 465, 473, 476, 482, 486, 491, 494  
 Técnica de avaliação de programas e revisão –  
   PERT, 228, 331  
 Tempo de fluxo total, 215, 216, 219, 223  
 Tempo de  
   espera em fila, 13, 434, 436, 440, 456, 457, 462,  
   465, 466, 473, 477, 480, 481, 482, 484, 485,  
   487, 490, 496  
   permanência, 440, 442, 458, 466, 477, 482, 486  
   serviço, 195, 198, 437, 440, 442, 466, 470, 476,  
   477, 478, 480, 482, 483  
 Teoria de filas, ix, 2, 4, 13, 433, 435, 445, 447, 470  
 Teste de  
   Anderson-Darling, 448  
   Kolmogorov-Smirnov, 448  
   Lilliefors, 448  
   qui-quadrado, 448  
 Trajetória central, 66  
 Valor esperado, 150, 408, 412, 419, 421, 422, 423,  
   430, 442, 495  
 Variância, 28, 331, 445, 448, 449, 456, 458, 476,  
   477, 478, 480, 483, 484, 493  
 Variáveis  
   artificiais, 100, 101, 103, 106, 129  
   básicas, 72, 75, 80, 81, 84, 85, 86, 87, 93, 94,  
   95, 96, 97, 99, 101, 123, 124, 139, 145, 160,  
   253, 355, 356, 359, 361, 363, 364, 365, 366, 367  
   de decisão, 8, 20, 26, 28, 29, 33, 36, 51, 336,  
   474  
   dependentes, 507  
   duais, 75, 113, 115, 116, 117, 118, 125, 129,  
   261, 266, 268, 269  
   folga, 53, 68, 85, 91, 94, 98, 99, 100, 114, 115,  
   122, 139, 144, 154, 162, 244  
   independentes, 44, 69, 70, 157, 158, 506, 507,  
   508  
   livres, 53, 54  
   não-básicas, 72, 75, 77, 79, 80, 81, 86, 87, 89,  
   90, 91, 93, 94, 95, 96, 97, 99, 101, 134, 144,  
   244, 245, 246, 253, 355, 356, 359, 360, 361,  
   363, 364, 365, 366, 367, 372  
 Vértice, 61, 65, 66, 67, 69, 73, 74, 80, 82, 93, 94,  
   130, 131, 132, 134, 137, 157, 177, 186, 193,  
   263, 289, 291, 355, 369