



OTIMIZAÇÃO COMBINATÓRIA E PROGRAMAÇÃO LINEAR

Modelos e Algoritmos



Preencha a **ficha de cadastro** no final deste livro e receba gratuitamente informações sobre os lançamentos e as promoções da Elsevier.

Consulte nosso catálogo completo, últimos lançamentos e serviços no site **www.elsevier.com.br**

OTIMIZAÇÃO COMBINATÓRIA E PROGRAMAÇÃO LINEAR

Modelos e Algoritmos

© 2005, Elsevier Editora Ltda.

Todos os direitos reservados e protegidos pela Lei nº 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.

Editoração Eletrônica
Estúdio Castellani
Revisão Gráfica
Roberto Mauro Facce
Singular Traduções e Serviços Editoriais

Projeto Gráfico
Elsevier Editora Ltda.
Conhecimento sem Fronteiras
Rua Sete de Setembro, 111 – 16º andar
20050-006 – Centro – Rio de Janeiro – RJ – Brasil

Rua Quintana, 753 – 8º andar
04569-011 – Brooklin – São Paulo – SP – Brasil

Serviço de Atendimento ao Cliente
0800-0265340
sac@elsevier.com.br

ISBN 978-85-352-1520-5

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 ao nosso Serviço de Atendimento ao Cliente, 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.

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

G654o

2.ed.

Goldbarg, Marco Cesar
Otimização combinatória e programação linear: modelos e
algoritmos / Marco Cesar Goldbarg, Henrique Pacca L. Luna.
-2.ed. – Rio de Janeiro: Elsevier, 2005 – 10ª reimpressão.
il.

Inclui bibliografia
ISBN 978-85-352-1520-5

1. Otimização combinatória. 2. Programação linear.
I. Luna, Henrique Pacca L. (Henrique Pacca Loureiro).
II. Título.

04-2886.

CDD – 519.4
CDU – 519.68

PREFÁCIO

Os autores são professores reconhecidos nas áreas de otimização, programação matemática e pesquisa operacional. Orientam teses de pós-graduação, lecionam disciplinas de graduação e de pós-graduação nos assuntos tratados neste livro, publicam em revistas científicas indexadas e exercem liderança acadêmica em suas universidades.

Marcos C. Goldbarg e Henrique P. L. Luna foram felizes em lançar este livro, pois escolheram um momento propício em que há muitos estudantes e profissionais no Brasil interessados nos processos científicos de apoio à decisão.

Otimização Combinatória e Programação Linear: Modelos e Algoritmos consegue um equilíbrio entre a arte de modelagem matemática em otimização e os algoritmos para solucionar esses modelos. A partir do método do simplex para solucionar problemas de programação linear os autores percorrem de maneira muito didática temas importantes como fluxo em redes, caminho mais curto em um grafo e árvore geradora mínima de um grafo, onde o leitor ou a leitora pode perceber a necessidade ou não de orientar um grafo. Está muito bem introduzida a noção de programação linear com variáveis inteiras e suas aplicações em problemas de decisão, nos quais algumas variáveis são bivalentes (0-1). A modelagem 0-1 é de grande utilidade para problemas de localização de facilidades, concepção de redes de serviço, de percursos de veículos.

Há problemas clássicos de otimização combinatória para os quais métodos específicos de programação linear com variáveis bivalentes foram desenvolvidos. Dentre esses problemas, os autores descrevem com muita clareza os seguintes: do caixeiro viajante, de particionamento, de recobrimento, de corte, de carregamento, de empacotamento e de percurso de veículos.

Dada a explosão combinatorial de alguns problemas de otimização combinatória muitas vezes não se consegue obter uma solução ótima; assim sendo, os autores consideraram também métodos heurísticos, com complexidade numérica polinomial, visando à busca de soluções boas.

O livro está bem redigido, bem ilustrado, contendo bons exercícios e uma bibliografia atualizada.

Considero um trabalho essencial para o ensino de otimização combinatória no Brasil.

NELSON MACULAN

Secretário de Ensino Superior do Ministério da Educação

Professor Titular

Universidade Federal do Rio de Janeiro

INTRODUÇÃO À SEGUNDA EDIÇÃO

A segunda edição do livro *Otimização Combinatória e Programação Linear: Modelos e Algoritmos* chega num momento oportuno, em que se aliam a necessidade de atualização da matéria e a maturidade dos autores com relação à obra. Cinco anos se passaram, quatro milheiros foram impressos, com uma média de mais de uma tiragem por ano. As melhorias agora introduzidas estão calcadas na experiência oriunda do próprio uso do livro em sala de aula e nos comentários de colegas professores que nos honraram com o uso da primeira edição.

Constituindo uma versão atualizada que preserva o conteúdo original e melhora a forma de apresentação, esta segunda edição contempla uma significativa inclusão de referências à produção científica do primeiro lustre do novo milênio. A atualização resultou em um saldo positivo de dezenas de páginas, com a supressão de algumas páginas do apêndice que não eram referenciadas no texto principal da primeira edição. A rica ilustração do livro foi reorganizada com a numeração restrita ao escopo de cada capítulo, abrangendo as referências a figuras, equações, quadros e tabelas, que por sua vez tiveram apresentação padronizada.

A presente edição buscou principalmente atualizar as citações bibliográficas, especialmente no concernente aos algoritmos de solução dos problemas de otimização combinatória abordados, e revisar as taxonomias utilizadas nos capítulos aplicados. Por exemplo, o Capítulo 9, que trata de “Problemas de Roteamento”, foi completamente refeito com um esquema mais atual de classificação da literatura pertinente. Por sua vez, os Capítulos 8 e 10, tratando, respectivamente, do Problema do Caixeiro Viajante e dos problemas de “Cobertura, Particionamento e Localização” foram também enriquecidos e atualizados. Algum material migrou de capítulo, com o fim de aglutinar matérias afins e de dar maior aderência aos nomes que rotulam os capítulos.

Esta segunda edição surge numa época em que se consolida a ligação entre otimização combinatória e programação linear, o que ratifica o acerto do nome do livro e a importância da matéria nele ensinada. Outra tendência que também já havia sido antecipada na primeira edição se refere ao interesse do uso de heurísticas clássicas e de meta-heurísticas na abordagem de problemas de otimização combinatória. Essas duas tendências mundiais são realçadas no livro e se refletem no próprio trabalho de pesquisa dos autores e de seus alunos de pós-graduação, cujos artigos científicos, teses e dissertações possuem caráter mais avançado e complementar à temática fundamentada no livro. No âmbito da otimização combinatória, busca-se um equilíbrio entre o ensino de algoritmos exatos, que em regra passam pelo uso de programação linear inteira mista, e de algoritmos aproximados, que em regra hoje usam meta-heurísticas. O estudante deve se conscientizar que a busca da otimalidade global é sempre um ideal a ser perseguido, mas que face à complexidade dos problemas do mundo real é muitas vezes necessário cunhar um método heurístico para a resolução do problema em questão. A lição é que o bom senso, a avaliação crítica e a competência devem prevalecer na abordagem de um problema, evitando-se tanto o medo da complexidade quanto a obsessão pela exatidão.

Lançada no começo do segundo lustre do novo milênio, esta segunda edição muito se beneficia de uma época em que as redes de comunicação se tornaram um agente efetivo de apoio ao ensino e à pesquisa. A falta de boas bibliotecas deixou de ser, em termos de capacidade de busca de conhecimento, um dos fatores determinantes das desigualdades regionais.

Do Nordeste brasileiro os autores obtiveram praticamente todo o acervo da produção científica, cuja referência fica acrescida nesta segunda edição. Naturalmente cabe ao leitor o aproveitamento direto dessa rica informação eletrônica. Nesse período entre a primeira e a segunda edição cresceu muito a acessibilidade ao conhecimento gerado no mundo. O material complementar que os autores disponibilizam na Web é apenas parte da informação que o leitor mais interessado deve recuperar. Nosso meio acadêmico precisa ser incentivado a tirar maior proveito da gratuidade de ricas fontes de informação pela Internet e, principalmente, do pagamento e difusão pela CAPES das edições eletrônicas dos principais periódicos científicos.

Há muitos exercícios resolvidos no livro e, embora a maioria dos exercícios propostos sejam tratáveis, há todo grau de dificuldade nos exercícios sugeridos, alguns constituindo em si problemas para dissertações de mestrado ou mesmo teses de doutorado. É normal, portanto, que haja alguns exercícios em aberto, e o leitor professor ou aluno deve entender que nesses casos a resolução pode eventualmente ser obtida depois de um esforço de pesquisa que ainda não foi empreendido nem mesmo pelos autores. Nesse sentido, o livro deve ser entendido como um trampolim para a atividade de pesquisa, e é por isso que serve também a cursos de pós-graduação.

A propósito, além do sítio da editora, o leitor interessado também pode recuperar informação em repositórios de cursos de pós-graduação e de laboratórios de pesquisa. Inclusive, a CAPES atualmente recomenda transparência e conteúdo eletrônico para todo curso de pós-graduação, disponibilizando na rede seu material de ensino e incentivando homepages dos próprios alunos, nas quais devem exibir seus exercícios resolvidos e seus trabalhos de computação científica. Essa recomendação é adotada no Programa de Pós-Graduação em Modelagem Computacional de Conhecimento da UFAL, em Maceió (<http://www.tci.ufal.br/mcc>). Em particular, nesse programa estão lançando suas homepages os onze alunos matriculados no segundo semestre de 2004 da disciplina de Otimização Contínua e Combinatória, ministrada pelo autor Henrique Pacca L. Luna, na qual nosso livro é adotado, e os alunos devem rodar em diferentes pacotes de otimização muitos dos exercícios resolvidos e propostos no livro. Devido a uma grande quantidade de informação, com o devido cuidado, o leitor pode usar diretamente esse tipo de homepage como fonte de informação complementar para o livro. Outra fonte de caráter complementar está nos sítios de laboratórios de pesquisa e do CNPq, em particular do sistema Lattes, no qual se encontram currículos e diretórios de grupos de pesquisa brasileiros. Em particular, o Núcleo de Excelência em Otimização de Sistemas Complexos, que articula um consórcio de pesquisa de instituições nordestinas, mineiras e francesas, e disponibiliza material digital em <http://nexos.tci.ufal.br>, em que se encontram artigos, teses, dissertações, palestras, figuras, modelos computacionais e exercícios resolvidos de nossos livros e notas de aula.

Reafirmamos nossos agradecimentos a todos que foram citados explícita ou implicitamente na introdução à primeira edição. A eles se somam colegas de ensino e alunos de graduação e de pós-graduação, dos quais muitos se graduaram ou viraram mestres e doutores nesse período de cinco anos entre a primeira e a segunda edição do livro. Os colegas professores da área continuam aqui representados pelo professor Nelson Maculan, hoje Secretário de Educação Superior do MEC, que muito nos honrou com a apresentação do livro. A professora Elizabeth Ferreira Gouveia, da UFRN, nos ajudou também nesta segunda edição e merece um agradecimento especial. Dessa vez também, apesar de todo o esforço de revisão, erros que eventualmente persistirem são de responsabilidade dos autores, que continuam à disposição, por meio dos endereços eletrônicos gold@dimap.ufrn.br e pacca@tci.ufal.br para receber críticas e sugestões. E finalmente agradecemos aos leitores, cujo interesse suscitou a viabilidade econômica desta segunda edição num tema que tanto nos fascina.

Marco C. Goldberg
Henrique P. L. Luna
novembro de 2004

INTRODUÇÃO À PRIMEIRA EDIÇÃO

A utilização do ferramental da “Pesquisa Operacional” na promoção da eficiência e eficácia organizacional em todos os níveis da gestão é uma realidade tornada viável pelo microcomputador e pelo avanço do estado da arte. De fato, a relevância da área de sistemas de apoio à decisão vem crescendo com o advento das estações de trabalho, que a um baixo custo, e com maior proximidade física dos executivos, oferecem capacidade de cálculo, de armazenamento e recursos gráficos antes disponíveis apenas em máquinas de grande porte.

Dentro desse cenário, torna-se cada vez mais crucial o pleno domínio desse ferramental, especialmente pelos profissionais das áreas de ciências exatas e de administração. Nesse sentido o presente trabalho foi elaborado para facilitar o acesso e o domínio das técnicas de otimização combinatória e programação linear.

O livro busca cobrir uma lacuna existente na literatura nacional no que diz respeito ao desenvolvimento de habilidades necessárias à modelagem matemática e à solução de importantes problemas de otimização. O objetivo principal desta obra é propiciar subsídios técnicos, descrever modelos matemáticos, debater algoritmos de solução e sugerir temas de aplicação, dentro da visão quantitativa de análise do fenômeno de tomada de decisão, principalmente aos alunos de graduação das engenharias, cursos de administração e de ciência da computação.

Aceito como um texto introdutório, o livro poderá ser utilizado também no contexto da pós-graduação. Objetivando tal possibilidade, é vasta e atualizada a revisão bibliográfica que acompanha cada tema específico, de modo que o estudante de pós-graduação possa, com mais facilidade, alcançar o aprofundamento necessário.

O conteúdo programático coberto pelos Capítulos de 1 a 5 engloba o que é normalmente exigido em disciplinas que são denominadas classicamente como “Pesquisa Operacional”, “Otimização de Sistemas de Engenharia”, “Programação Linear” etc. O professor dessas disciplinas de graduação poderá também optar pela substituição do Capítulo 5 pelas partes iniciais dos Capítulos 6 e 7, principalmente se o curso contar com apenas uma disciplina básica nessa área do conhecimento.

O conteúdo dos Capítulos 6 a 10 aborda temas em otimização combinatória que são objeto de disciplinas como “Otimização Combinatória”, “Algoritmos em Grafos”, “Fluxos em Redes” etc. Essa segunda parte do livro contém material um pouco mais avançado, servindo tipicamente a disciplinas optativas de nossos cursos de graduação ou a disciplinas de pós-graduação.

Dessa forma o texto foi organizado para auxiliar os docentes no decurso de sua tarefa de incentivar e facilitar o processo de construção do conhecimento dos alunos, por meio da disponibilização de um arcabouço de consulta adequado, consistente e compacto.

Uma outra característica marcante deste trabalho são as várias aplicações práticas retiradas da realidade brasileira, o que permite aproximar a disciplina ao dia-a-dia do estudante, aumentando o incentivo à motivação. No âmbito da programação linear, o livro é caracterizado pelo uso intensivo de interpretação econômica do método simplex e das relações de dualidade e sensibilidade. A ideia é insistir no ponto de que as ferramentas de otimização indicam as atividades de maior produtividade nas empresas, e de que o uso dessas ferramentas também indicam os preços de maior competitividade.

No âmbito da otimização combinatória, como o próprio nome indica, o livro se caracteriza pela apresentação dos problemas através do uso deliberado de formulações de programação linear inteira mista. Além do interesse teórico, a formulação de problemas combinatórios através da programação in-

teira abre caminho para o uso dos eficientes pacotes comerciais de programação matemática, cujo avanço na solução exata de importantes problemas práticos tem sido marcante. Mesmo para heurísticas que dispensam a formulação de programação linear inteira, avaliações de qualidade de solução podem ser feitas muitas vezes mediante o uso de relaxação do problema de programação linear.

Lançado na virada do milênio, o livro se beneficia de uma época em que as redes de comunicação se tornaram um agente efetivo de apoio ao ensino, inserindo-se em uma proposta da editora em disponibilizar material complementar por meio da Internet, principalmente para os professores cadastrados. Mais detalhes serão fornecidos no site correspondente, em função da dinâmica da interação com professores e alunos. Para ajudá-lo a obter o máximo proveito deste livro, colocamos alguns ícones associados aos exercícios resolvidos e a resolver. Assim, os símbolos ☺, ☹, ☹, ☹, ☹ e ☹ indicam o grau de dificuldade do exercício, em ordem crescente.

O material desta primeira edição foi experimentado e revisado durante um ano, em disciplinas de graduação e de pós-graduação, tanto na UFRN quanto na UFMG. A lista seria muito longa, mas agradecemos com atenção especial a todos esses alunos que serviram de cobaias. Alguns colegas da área nos ajudaram de diferentes formas, e entendemos que todos estão aqui representados pelo Professor Nelson Maculan, que muito nos honra com a apresentação do livro. Um papel importante também coube ao pessoal de monitoria das disciplinas de Pesquisa Operacional da UFMG durante o ano de 1999. O doutorando Ricardo Poley Martins Ferreira produziu uma série de transparências que permitiram o acompanhamento do livro nas aulas. A mestrandia Renata Couto Moreira (*renatacm@lapo.dcc.ufmg*) e o graduando Adriano César Machado Pereira (*adrianoc@dcc.ufmg.br*) continuam trabalhando conosco no apoio ao ensino e na produção de material didático, principalmente na resolução de exercícios. Foi graças a esse pessoal de apoio que pudemos disponibilizar material complementar pela Internet.

À professora Elizabeth Ferreira Gouvêa nossos agradecimentos pelas importantes observações feitas na fase da revisão final.

Apesar de todo o esforço de revisão, erros que eventualmente persistirem são naturalmente de responsabilidade dos autores, que se colocam à disposição, por meio dos endereços eletrônicos *gold@dimap.ufrn.br* e *pacca@dcc.ufmg.br*, para receber observações, críticas e sugestões julgadas oportunas. Esperamos que o livro contribua para o apoio ao ensino dessa área do conhecimento que tanto nos fascina.

M. C. Goldbarg

H. P. L. Luna

novembro de 1999

1 MODELAGEM DE PROBLEMAS

1.1 – PRINCÍPIOS DO PROCESSO DE MODELAGEM

1.1.1 – Conceito Intuitivo de Modelo

O homem sempre desejou entender o seu planeta e o mundo em geral. As imposições da sobrevivência assim o determinaram. Rodeada de elementos materiais, a primeira necessidade da humanidade foi conquistar o domínio de seu meio ambiente. A segurança contra predadores e fenômenos naturais, a busca da alimentação, a organização social dos núcleos humanos etc. despertaram os primeiros questionamentos do homem, privilegiando a ação sobre o concreto. À medida que as necessidades do espírito humano foram se tornando mais complexas, cresceram as carências por aperfeiçoar o processo de compreensão do mundo. Já na Antiguidade, inúmeras e sofisticadas estruturas de abstração foram se apresentando para representar as propriedades e os diversos graus de interação entre os vários interferentes desse todo. Na impossibilidade de lidar diretamente com a complexidade do mundo, o homem tem se mostrado cada vez mais hábil na criação de metáforas para a representação e solução de sua relação com esse mesmo mundo.

Esse processo de busca de uma visão bem estruturada da *realidade* (esclarecimento) é fundamentalmente um fenômeno de modelagem. A palavra modelo pode ter diversos significados. Modelo como representação substitutiva da realidade distingue-se do verbo modelar. O verbo introduz a ideia de simulação da realidade, que é mais ampla que a simples representação.

Um modelo *é um veículo para uma visão bem estruturada da realidade*. Um modelo pode também ser visto, com os devidos cuidados, *como uma representação substitutiva da realidade*.

Todos nós já lidamos com modelos, mesmo em ocasiões nas quais não possuíamos a mínima consciência disso. Quando explicamos algo a uma pessoa usando fotografias ou gráficos, ou quando representamos planos ou sólidos através de equações matemáticas, nada mais estamos fazendo do que transmitir e interpretar a estranha realidade através de metáforas de substituição ou *modelos*.

A Geometria Euclidiana é um modelo que satisfaz a um conjunto de axiomas, ou a um modelo de contexto axiomático. De forma semelhante, nessa linha podemos identificar outros modelos que vão desde as transformadas de Laplace e sua *Mécanique Céleste* até a Teoria Quântica e o átomo de Bohr. Obviamente não existem somente modelos axiomáticos. O contexto epistemológico é um exemplo disso. Lidando com o concreto ou abordando o imaginário, a mente do homem trabalha com estruturas de “substituição” que visam a facilitar o raciocínio, estruturas essas que denominaremos genericamente de modelos.

Os modelos, para serem implementáveis, devem ser livres de pequenos detalhes onerosos. Nessa abordagem a importância do equilíbrio simplificação vezes validade é básica. Concluimos que:

*Os modelos são representações simplificadas da realidade
que preservam, para determinadas situações e enfoques,
uma equivalência adequada*

O poder de representatividade é a característica do modelo que o torna desejável. A capacidade de simplificação lhe confere factibilidade operacional. Existem vários critérios de medida da adequação ou aderência do modelo à realidade representada. Em várias ocasiões a representatividade do modelo pode ser aperfeiçoada de forma iterativa. O processo de verificação da representatividade é denominado de *validação do modelo*, sendo uma etapa indispensável em qualquer procedimento científico.

1.1.2 – Desdobramento do Conceito – Tratamento da Eficiência

O conceito de modelo como uma representação substituta da realidade possui um alcance limitado. O aspecto da eficiência é fundamental. Para alcançarmos modelos eficientes, são necessárias pelo menos três habilidades:

✓ Foco Holístico

Quando estamos procurando solucionar um problema, na maioria das ocasiões a preocupação com a concatenação e o manejo dos vários impactos de nossa solução sobre outros contextos é significativa. Se nossa solução puder criar outros problemas que possam, posteriormente, anular a contribuição de nosso esforço, o foco holístico é indispensável.

✓ Tratamento Eclético da Dimensão da Análise

Os métodos de solução a serem utilizados devem ser o mais livremente dispostos. Epistemologia e Axiologia não devem ser consideradas bases de modelagem dicotômicas, mas sim complementares. A construção de modelos é um processo que possui, como Jano, duas faces: o rosto que articula a teoria (hipotético-dedutivo) e a face que representa a validade da dedução (indutiva) na prática.

✓ Tradução Adequada

Um bom modelo exige uma conveniente tradução contextual. Uma boa tradução contextual pode ser expressa através de um correto isomorfismo entre o fenômeno e seu modelo. A Figura 1.1 esclarece o processo de tradução, ressaltando seu aspecto simplificador e estruturador.

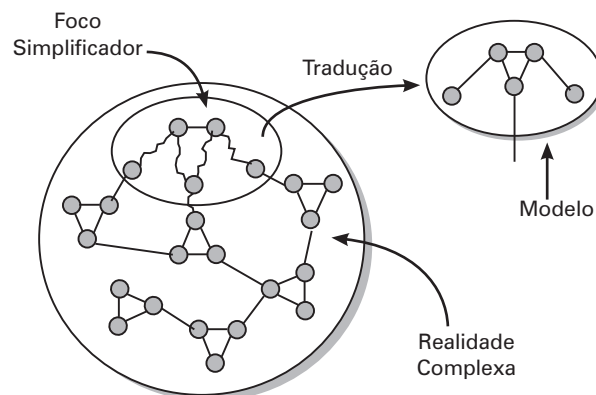


FIGURA 1.1 O processo de tradução.

Obviamente nem todos os problemas são iguais. O processo de tradução contextual deve ser capaz de identificar os elementos fundamentais da questão e transportá-los para uma representação capaz de ser manipulada por artifícios ou métodos de solução. As dificuldades dos processos de tradução e solução são de naturezas diferentes, contudo profundamente interferentes. A tradução visa claramente a

cooperar com uma certa abordagem de solução. Na medida em que a tradução produz uma representação mais ou menos tratável pelos métodos existentes, a utilizabilidade do modelo é definida. O conceito que representa esse fenômeno de interferência da tradução na possibilidade de solução é denominado de *complexidade*.

Modelos “menos complexos” ou “simples” são mais facilmente abordados por métodos empíricos, ao passo que modelos complexos dificilmente deixarão de necessitar do apoio das técnicas hipotético-dedutivas. Observamos que a facilidade de compreensão do modelo, ou sua inteligibilidade, não pode ser confundida com a sua complexidade de solução. Alguns dos mais difíceis problemas combinatórios possuem modelos de fácil entendimento e representação, mas de extraordinária dificuldade de solução. Nesse sentido, é indispensável esclarecer melhor o fenômeno da *complexidade*.

1.1.3 – Desdobramento do Conceito – Tratamento da Complexidade

Para definirmos “complexidade” escolheremos identificar as propriedades que ocasionam e interferem com o fenômeno. Essa opção permite uma visão bastante operacional sem prejudicar um possível aprofundamento teórico. O primeiro aspecto a destacar na constituição da complexidade de um modelo é sua “permeabilidade” ao meio ambiente circunvizinho. Um modelo simples possui um perímetro de interferência simples e bem definido. Um segundo ponto diz respeito à sua estrutura interna. Um modelo simples possui uma estrutura homogênea, uma morfologia uniforme e um número reduzido de variáveis. A última dimensão engloba a dinâmica, correspondendo à consideração de como a estrutura interna se altera ao longo do tempo.

Analisando essas três dimensões principais, a saber (ver Figura 2):

- Meio Ambiente.
- Domínio.
- Dinâmica.

Diremos que um modelo é simples quando:

- É pouco influenciado pelas variações em seu meio ambiente.
- É estruturalmente estável, homogêneo e possui poucas variáveis.
- Possui comportamento facilmente previsível.

1.2 – O METAMODELO SISTÊMICO

1.2.1 – Conceito de Sistema

Coube à Engenharia de Sistemas uma significativa parcela de contribuição no processo de estruturar e sistematizar os esforços de modelagem. Como esclarecemos anteriormente, o fenômeno de produção de tradução para as entidades do mundo real é uma atividade extremamente complexa e dificilmente passível de realização em uma só etapa. O Metamodelo Sistêmico¹ surge como uma proposta interessante no sentido de auxiliar, no mais alto nível, o processo de tradução.

O conceito de sistema permite a configuração de um extremamente útil pré-mapeamento entre a realidade e o modelo de representação. Dentro da abordagem sistêmica, modelar significa representar a realidade ou os *sistemas originais* através de outros *sistemas de substituição*, estruturados e comparáveis, denominados modelos.

¹ Usamos o termo metamodelo porque esse modelo é uma grande abordagem que suporta a elaboração de outros modelos mais específicos.

Definimos sistema como qualquer unidade conceitual ou física, composta de partes inter-relacionadas, interatuantes e interdependentes

Sistemas são *componentes não atômicos* (sob o ponto de vista da observação) de um determinado universo. Os sistemas, de uma forma geral, possuem ainda as seguintes propriedades:

- ✓ **Simbiose interna** ou propriedade de compartilhar funções que torna cada parte indispensável dentro da constituição do sistema.
- ✓ **Simbiose externa** ou propriedade de ser um componente participante e indispensável (pelo menos em tese) de um ecossistema social.
- ✓ **Sinergia** ou o efeito multiplicador que permite ao sistema alcançar patamares de desempenho superior ao obtido pela soma do desempenho de cada parte isolada.
- ✓ **Homeostase** ou a capacidade de conservar seu estado de equilíbrio.
- ✓ **Entropia negativa** ou a capacidade de importar a energia necessária de seu ecossistema social para compensar a natural degradação entrópica e realizar autoajustes em direção ao equilíbrio organizacional \times meio ambiente.

Outro elemento fundamental no conceito é a *concepção da propriedade recursiva*. As partes de um sistema podem ser igualmente consideradas subsistemas gozando da mesma natureza de propriedades previstas para o todo. Essa propriedade permite uma enorme flexibilidade de raciocínio. Com essa poderosa ferramenta, muitas situações complexas da realidade, antes de difícil abordagem, puderam ser “organizadas” conceitualmente com facilidade. Na verdade a abordagem sistêmica facilitou muito o processo de representação da realidade e, em consequência, sua melhor compreensão.

Podemos denominar de modelo cada interpretação de um sistema formal criado axiomáticamente

No processo de representação substitutiva, muitos são os fatores a considerar. Na verdade, é praticamente impossível levar em conta todos os elementos intervenientes em um problema real. Como o objetivo básico do processo é alcançar uma compreensão aceitável da realidade, os modelos devem ser formulados de modo a “captarem” apenas os “elementos fundamentais” no processo de tradução, simplificando ao máximo o método de solução a ser utilizado.

Os modelos possuem diversas vantagens, além do fato de simplificarem a representação de determinado sistema. Os modelos podem revelar relacionamentos não aparentes, bem como facilitar a experimentação (ou o aprendizado por tentativa e erro controlado), o que não é, normalmente, viável em sistemas reais. Como a estrutura do modelo independe dos dados de operação ou instância, a análise é altamente auxiliada. Procurando satisfazer aos requisitos de *qualidade*, modelos quantitativos de otimização buscam alternativas de máxima produtividade e, alguns deles, determinam automaticamente preços de máxima competitividade.

1.3 – MODELOS DE OTIMIZAÇÃO

1.3.1 – O Contexto dos Modelos de Otimização

Um modelo não é igual à realidade, mas suficientemente similar para que as conclusões obtidas através de sua análise e/ou operação possam ser estendidas à realidade.

Em consequência, para a formalização desse modelo é indispensável definir:

- ✓ A *estrutura relacional* do sistema representado.
- ✓ O *comportamento funcional* de cada subsistema ou componente atômico.
- ✓ Os *fluxos de inter-relacionamento*.

Como nosso objetivo é perseguir uma proposta factível de modelagem, vamos examinar as condições para executar uma boa tradução à luz da complexidade. A Figura 1.2 resume as três dimensões debatidas no item número 1 e que representam as possibilidades de aplicação prática da abordagem sistêmica (Suntherland [1975] e Bertalanffy [1968]).

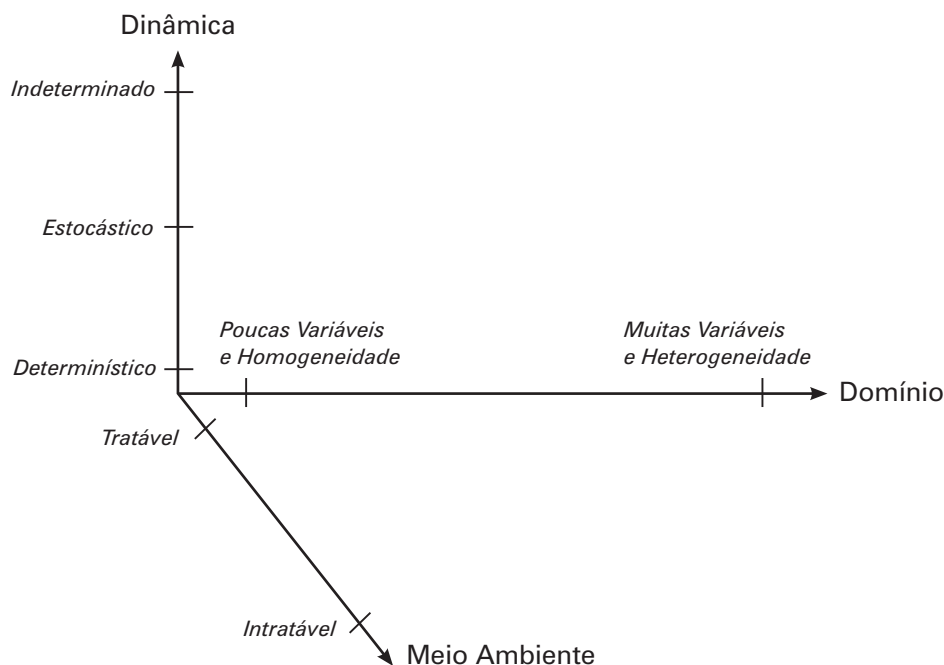


FIGURA 1.2 Dimensões da complexidade de modelos.

É claro que o iconógrafo da figura pode ser aplicado para a análise de toda a sorte de sistemas, inclusive os socioeconômicos (que podem representar uma sociedade ou um país). Concentrando-nos no objetivo do presente trabalho, os modelos de otimização, podemos identificar sua área de restrição face à complexidade de tradução entre os dois planos representados pela Figura 1.3. O plano interior, que define modelos determinísticos, tratáveis e de pequeno porte, é denominado plano dos mecanismos.

A estrutura relacional dos sistemas modelados pode ser representada por desenhos ou símbolos.

O comportamento funcional pode ser representado por funções de desempenho em que as possíveis entradas nos subsistemas são associadas às saídas geradas pelo mesmo.

Da própria definição de modelo e de seus objetivos, derivam as principais características dos modelos de otimização:

- ✓ O primeiro tema se refere à obtenção das *propriedades analíticas* do modelo.
- ✓ O segundo tema está na ênfase em uma *melhoria mensurável* no processo. Nesse ponto são envolvidos conceitos de otimização face às possibilidades de o problema possuir mais de uma solução possível. Os modelos de otimização são normalmente amparados em variáveis quantitativas bem definidas como o iconógrafo evidencia.
- ✓ O terceiro tema é o *reconhecimento explícito* das interações no modelo e sobre o modelo.

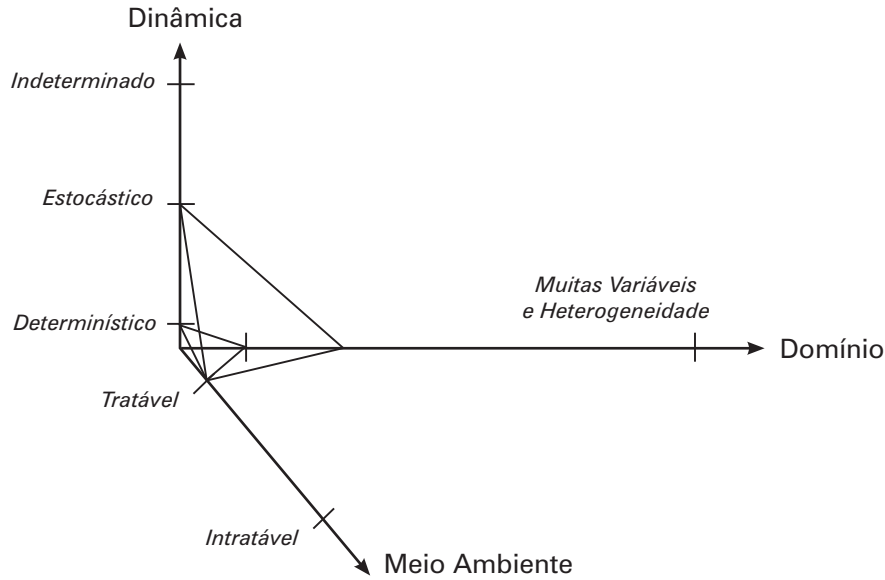


FIGURA 1.3 Espaço viável para a atuação dos modelos de otimização.

Um *problema de otimização contínua* pode ser formalizado matematicamente da seguinte forma (ver Luenberger [1984]):

Minimizar $f(x)$
sujeito a:

$$\begin{aligned} h_i(x) &= 0, & i &= 1, \dots, m_h \\ g_j(x) &\leq 0, & j &= 1, \dots, m_g \\ x &\in \mathfrak{R}^n \end{aligned}$$

Em que $f: \mathfrak{R}^n \rightarrow \mathfrak{R}$, $h: \mathfrak{R}^n \rightarrow \mathfrak{R}$ e $g: \mathfrak{R}^n \rightarrow \mathfrak{R}$ são funções contínuas, geralmente diferenciáveis em problemas tratáveis de grande porte. Dado um conjunto finito $E = \{1, 2, 3, \dots, n\}$ e uma coleção de subconjuntos F de E expresso como:

$$\emptyset \neq F \subseteq 2^{|E|}$$

e uma função objetivo:

$$C: F \rightarrow \mathfrak{R}$$

um *problema de otimização discreta* pode ser entendido como o desejo de obter um conjunto $S^* \in F$ satisfazendo:

$$\begin{aligned} C(S^*) &\geq C(S), \forall S \in F && \text{(problema de maximização)} \\ C(S^*) &\leq C(S), \forall S \in F && \text{(problema de minimização)} \end{aligned}$$

e sujeito a uma série de restrições.

S é também denominada uma configuração do problema. S^* é chamada a melhor dentre todas as configurações segundo o critério de otimização $C(S)$. F é o espaço das configurações. C é a função objetivo. Essa formalização é bastante geral, obviamente voltaremos a esse modelo adotando as nomenclaturas clássicas para as configurações S (denominadas também soluções viáveis) e F (espaço das soluções viáveis).

1.4 – CLASSIFICAÇÃO DE MODELOS

Os modelos podem ser construídos sob diversas arquiteturas. Como muitos são os enfoques pelos quais podemos abordar o processo de construção de modelos, podemos distinguir várias classificações possíveis:

Classificação Quanto à Natureza do Modelo

A Figura 1.4 apresenta uma classificação quanto à natureza dos modelos:

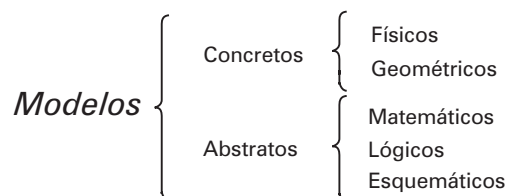


FIGURA 1.4 Uma classificação geral dos modelos.

1.4.1 – Classificação Quanto às Propriedades dos Modelos

Pode-se abordar a classificação dos modelos pelas propriedades que os mesmos são capazes de representar. Sob esse aspecto, segundo Ackoff (1971), destacam-se os seguintes grupos:

- Modelos icônicos.
- Modelos analógicos.
- Modelos simbólicos.

Nos *modelos icônicos* as propriedades relevantes dos objetos reais são representadas como tais. Nesse tipo de modelo, uma das mais significativas diferenças entre a realidade e o modelo está na escala. Os modelos icônicos são *imagens* da realidade. Alguns exemplos desse tipo de modelo são as fotografias e os mapas. Os modelos icônicos são, geralmente, concretos e de manipulação experimental difícil.

Os *modelos analógicos* usam um conjunto de propriedades inerentes ao modelo para representar o conjunto de propriedades da realidade. Um exemplo clássico é a substituição de sistemas hidráulicos por elétricos. Os grafos são modelos analógicos que utilizam grandezas geométricas e posições no plano para representar diversas variáveis e suas relações, representando, em muitas situações, *problemas de decisão*.

Os *modelos simbólicos* usam letras, números e outros símbolos para representar as variáveis e suas relações. Em grande parte dos casos, tomam a forma de relações lógicas ou matemáticas (equações). De uma forma geral, todos os modelos e, em especial, os modelos simbólicos são desenvolvidos de uma forma iterativa por aproximações. Os fluxogramas e os DFD (Diagrama de Fluxo de Dados) são típicos modelos simbólicos desenvolvidos, normalmente, em etapas de aproximação. Os modelos elaborados nas primeiras fases da modelagem são, muitas vezes, denominados de *modelos conceituais*. Nesses modelos conceituais, não raro as relações entre os elementos do sistema são descritas de forma qualitativa. Os DFD e as Redes de Petri são exemplos disso. Em otimização objetiva-se, sempre que possível, a construção de modelos simbólicos porque permitem o tratamento quantitativo do sistema e de suas propriedades.

1.4.2 – Classificação Quanto às Variáveis Controladas

Uma outra importante distinção entre os modelos pode ser feita através da natureza das variáveis envolvidas. Alguns modelos possuem variáveis controláveis, outros não. Em geral, os modelos com variáveis controláveis são *explanatórios*, e os demais, *descritivos*.

1.4.3 – Classificação de Emshoff (1970)

Segundo Emshoff, os modelos podem ser classificados como:

- ✓ Descritivos.
- ✓ Físicos.
- ✓ Simbólicos.
- ✓ Procedimentais.

Os modelos *descritivos* são expressos em linguagem corrente. São muito limitados, uma vez que seu método de predição é interno.

Os modelos *físicos* variam desde as representações em miniatura para organização de *layout* até os túneis de vento. Possuem alto custo e são muito específicos.

Os modelos *procedimentais* podem ser também denominados de simulação.

1.5 – O PROCESSO DE MODELAGEM

1.5.1 – Passos para a Modelagem

É possível, de uma forma bastante geral, resumir o *processo de modelagem* ou de construção de modelos na ótica operacional, pelos passos sugeridos pelo fluxograma da Figura 1.5.

A definição do problema é uma das fases mais importantes do processo e compreende a clara percepção do desafio colocado. O problema deve ser traduzido em elementos palpáveis englobando:

- ✓ Objetivos.
- ✓ Variáveis de decisão ou controle.
- ✓ Níveis de detalhe.

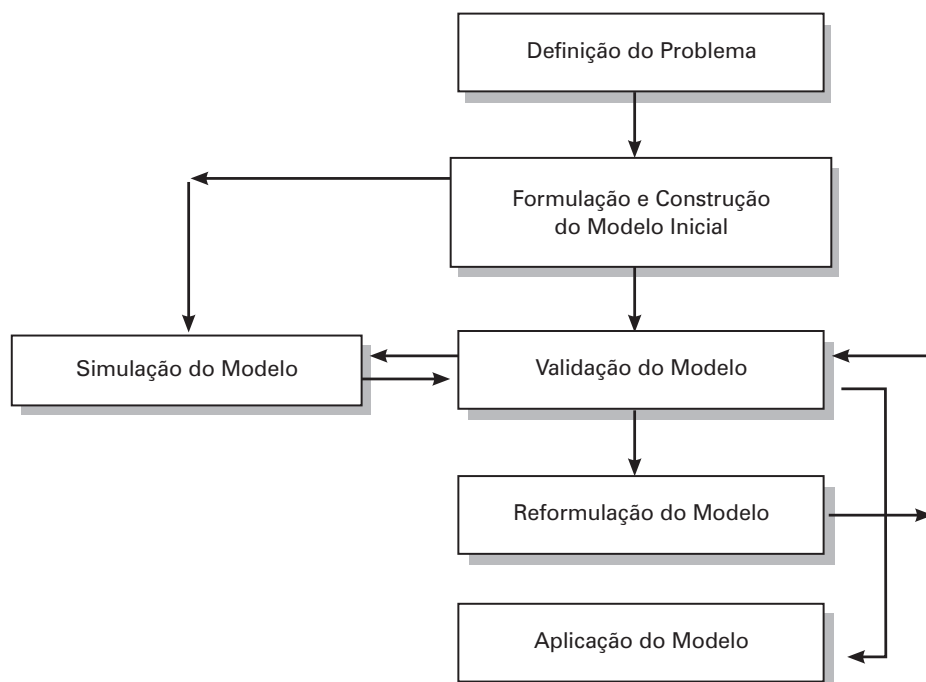


FIGURA 1.5 O processo de construção de modelos.

O segredo do sucesso do modelo de otimização depende da adequação de sua tradução, também denominada “formulação”. O próprio termo “formular”, largamente empregado para exprimir o processo de construção de modelos de otimização, traz consigo uma enorme carga *quantitativa e matemática*. Por outro lado, a adequação pretendida depende também de elementos que escapam ao conteúdo estritamente técnico, envolvendo a percepção do elaborador do modelo (ou equipe de elaboração), uma faculdade cognitiva de alto nível. As *fórmulas* ou *equações* do modelo não existem prontas e acabadas na natureza, elas têm de ser identificadas ou criadas. Estranhamente, o rigor da tradução é obtido através de processos pouco rigorosos ou conhecidos, envolvendo:

- ✓ Intuição.
- ✓ Experiência.
- ✓ Criatividade.
- ✓ Poder de síntese etc.

Temos daí duas consequências imediatas para o desenvolvimento de modelos:

- ✓ Existe uma enorme dificuldade de modelar o *processo de formulação*.
- ✓ Existe uma forte tendência a considerar a atividade de formulação de um modelo como uma *arte*.

A abordagem artística do fenômeno de formulação tem suas justificativas, mas ela poderá trazer em si um elemento perverso: deslocar o foco do desenvolvimento das técnicas de modelagem para um contexto pouco conhecido e controlável. Se, por um lado, a construção de um modelo é *inegavelmente* uma atividade subjetiva, podendo exigir características inatas do modelador, por outro, na maioria das ocasiões, conjugar o verbo modelar implicará um esforço absolutamente técnico. Apesar do lado genial e quase místico, na maioria dos casos da vida real, os fatores predominantes da elaboração serão conhecimentos e habilidades paroquiais, cuja aprendizagem e desenvolvimento estarão perfeitamente ao alcance do indivíduo mediano.

Na fase de formulação do modelo de otimização são definidos *os tipos de variáveis* a utilizar na apresentação, bem como o *nível apropriado de agregação* dessas variáveis. Ainda na formulação devem ser *representadas as restrições do problema*, tanto as quantitativas como as de natureza lógica. O modelo deverá ser adequado à natureza dos dados de entrada e de saída, bem como ser capaz de expressar as funções de desempenho que possivelmente serão exigidas no processo de otimização. As funções de desempenho, via de regra, serão denominadas de *funções objetivo*. A formulação será completada com o estabelecimento das *hipóteses de representação* que irão orientar a escolha e a possível utilização de modelos já existentes e de técnicas de solução (exatas, heurísticas etc.) para o caso.

A construção de modelos determina a inclusão de *parâmetros e constantes* que serão responsáveis pela definição e dimensionamento das relações entre as variáveis do modelo (*constantes de similaridade*). Na fase de validação do modelo, cumpre comparar seu comportamento com a realidade e, se necessário, atuar sobre esses elementos de forma a aproximar ao máximo o comportamento do sistema modelo ao do sistema real.

1.5.2 – Padrões para a Construção de Modelos de Otimização

Apesar de não considerarmos a técnica de construção de modelos como verdadeiramente uma arte, dificilmente seria possível reunir em um algoritmo específico e autocontido todos os passos indispensáveis para modelarmos um sistema genérico. Buscando o equilíbrio entre a arte e a técnica, podemos propor uma sistematização, se não completa, pelo menos parcial desse processo. Segundo Ackoff (1971), poderão ser considerados cinco padrões de construção de modelos:

Padrão 1: quando a *estrutura do sistema é suficientemente simples e evidente* para ser compreendida por inspeção. Nesse caso, o modelo pode ser construído com facilidade, o que não significa que não possa

ser muito difícil ou até mesmo impossível avaliar as variáveis não controladas e diversos outros parâmetros. O número de variáveis controladas pode também tornar impossível a solução prática do problema. O padrão 1 de modelagem aplica-se claramente aos sistemas pertencentes ao plano do mecanismo do iconógrafo da Figura 1.3.

Padrão 2: quando a *estrutura do sistema é relativamente aparente*, mas a *representação simbólica não é tão aparente*. Nessa situação, a busca de um sistema análogo com estrutura já conhecida é uma boa opção. O sistema análogo poderá auxiliar na descoberta das propriedades do sistema em estudo.

Exemplificamos a aplicação do padrão 2 no caso do modelo de *simulated annealing*, em que os máximos de uma complexa função são pesquisados através de uma analogia com o processo de recozimento de certos materiais.

Padrão 3: quando a *estrutura do sistema não é aparente*, contudo, uma *análise estatística* do mesmo pode atender ao desejado. Nesse caso, o sistema é considerado uma *caixa preta*, em que conhecemos, com segurança, as respostas para determinados estímulos.

Padrão 4: quando a *estrutura do sistema não é aparente e nem é possível isolar os efeitos das diversas variáveis através de uma análise estatística*. Nesse caso, uma boa política será o projeto de experimentos, de forma a determinar variáveis e correlações relevantes e reduzir o caso ao padrão 3.

Padrão 5: quando verificamos as situações do padrão 4, *porém as experimentações possíveis sobre o modelo são limitadas* para o fim desejado. Será o fim da linha? Nesse caso, existem ainda os modelos de conflitos e jogos de operações (ver Ackoff [1971]). Se isso ainda não for suficiente, então a dimensão criativa da modelagem deve ser ativada.

1.6 – MODELAGEM MATEMÁTICA

1.6.1 – Fundamentos da Modelagem Matemática

Conceitos Básicos

As técnicas e algoritmos que abordaremos no presente trabalho destinam-se a estruturar e a solucionar os modelos quantitativos que podem ser expressos matematicamente. Nesse ramo do conhecimento humano, destaca-se a *Pesquisa Operacional* (PO), uma tradicional disciplina que congrega diversas das mais consagradas técnicas da modelagem matemática. Os modelos de PO são estruturados de forma lógica e amparados no ferramental matemático de representação, objetivando claramente a determinação das melhores condições de funcionamento para os sistemas representados. Os principais modelos de PO são denominados de Programação Matemática e constituem uma das mais importantes variedades dos modelos quantitativos. Inicialmente, cabe desfazer uma pequena dúvida que pode ocorrer aos iniciantes, especialmente devido ao uso da palavra “programação”. Programação aqui é entendida no sentido de planejamento. Apesar do termo ter se consagrado ultimamente como associado ao processo de comando computacional ou “programação” computacional, ele é igualmente adequado para expressar as atividades genéricas de programação de atividades. Inevitavelmente, contudo, a Programação Matemática irá implicar programação computacional, uma vez que o número de variáveis de decisão e restrições é enorme na prática. Obviamente, isso será levado em conta pela modelagem e pelas técnicas de solução empregadas. No momento oportuno ressaltaremos esse fato.

O campo da Programação Matemática é enorme, e suas técnicas consagraram-se em face à sua grande utilidade na solução de problemas de otimização. Em virtude das várias peculiaridades inerentes aos diversos contextos de programação (planejamento), os métodos de solução sofreram especializações e particularizações. O processo de modelagem matemática, em si, pouco varia; contudo, as técnicas de solução acabaram agrupadas em várias subáreas como:

Programação Linear

Um caso particular dos modelos de programação em que as **variáveis são contínuas** e apresentam comportamento **linear**, tanto em **relação às restrições como à função objetivo**. Caso extremamente importante devido à eficiência dos algoritmos de solução existentes e à possibilidade da transformação dos modelos de Programação Não-linear em modelos de Programação Linear. A literatura nessa classe de problemas é particularmente rica, cabendo citar, entre os autores brasileiros, Barroso e Ellenrieder (1971), Puccini (1975), Maculan e Pereira (1980), Humes e Humes (1987) e Gonzaga (1989).

Programação Não linear

Um modelo de otimização constitui um problema de Programação Não linear **se exibir qualquer tipo de não linearidade, seja na função objetivo ou em qualquer de suas restrições**. A introdução de não linearidade é bem tratável nos casos de convexidade, situação em que se preservam propriedades importantes tanto sob a ótica da Programação Matemática como da teoria econômica. Em tais casos, algoritmos eficientes podem resolver grandes problemas reais, e o comportamento descentralizado de maximização de lucros fica sintonizado com a maximização do benefício social líquido no setor econômico em questão.

Programação Inteira

Um modelo de otimização constitui um problema de Programação Inteira **se qualquer variável não puder assumir valores contínuos, ficando condicionada a assumir valores discretos**. O requisito de que variáveis tenham de ser inteiras normalmente implica maior complexidade computacional do que a oriunda de situações de não linearidade de funções. O livro de Maculan (1978) é uma tradicional referência da literatura brasileira nessa área.

Cada tópico relacionado desenvolve técnicas e modelos específicos para a solução de uma família de problemas do mundo real. Muitas dessas técnicas e modelos encontram abordagens similares em outros campos do conhecimento. Um exemplo disso são algumas técnicas de busca da Programação Inteira que utilizam metáforas como a da evolução genética ou do resfriamento de metais.

1.6.2 – As Situações de Tomadas de Decisão e o Contexto da Programação Matemática

Sem dúvida, o estudo por diletantismo é um atrativo para algumas pessoas, mas não para todas. De fato a maioria das pessoas sente-se mais motivada ao estudo quando é capaz de vislumbrar que o conhecimento adquirido será útil, fará diferença em sua vida. Imaginamos ser esse o caso do leitor. Antes de nos debruçarmos sobre as elegantes e, eventualmente, trabalhosas técnicas de Programação Matemática, convém estabelecer o convencimento de sua utilidade. Nesse ponto, é interessante responder à seguinte pergunta:

*Em que situações da vida real poderemos utilizar
a Programação Matemática?*

A Programação Matemática, na prática, é fortemente direcionada ao apoio da tomada de decisão no gerenciamento de sistemas de grande porte, especialmente no que diz respeito ao tratamento de variáveis quantificadas. A técnica permite a modelagem de inter-relações entre variáveis que dificilmente seriam visíveis de forma intuitiva. Com a utilização dos meios de processamento automático de dados, os chamados programas de Programação Matemática podem examinar inúmeras configurações viáveis do problema proposto pelo tomador de decisão e selecionar, dentro de certos critérios, as “melhores”.

A tomada de decisão é um tema de grande importância no mundo atual. Decisões e ações são o produto final do trabalho de gerentes, executivos, engenheiros e políticos. **A tomada de decisão é o ato**

de selecionar, dentre várias decisões possíveis, a mais adequada para o alcance de certo objetivo. Precede a essa escolha, normalmente, um processo elaborado de:

- ✓ Representação adequada das variáveis e restrições do problema.
- ✓ Levantamento das alternativas viáveis. Possíveis heurísticas e meta
- ✓ Estabelecimento de critérios de avaliação dessas alternativas.
- ✓ Comparação das alternativas.
- ✓ Análise de impacto da tomada de decisão.

Os passos citados não são facilmente executados, uma vez que existem muitas razões e empecilhos que obstruem o processo. Talvez o primeiro obstáculo seja a própria deficiência de informações a respeito do ambiente e as incertezas em relação ao futuro. Um outro pacote de problemas reside nos conflitos de interesse e dificuldade de padronização de valores de avaliação. Para superar esses desafios, surgiu a *Teoria da Decisão*, uma disciplina orientada para a tomada de decisão, através de uma abordagem sistemática, quantitativa e normativa. Assim sendo, a teoria da decisão procura explicar o comportamento do homem racional, e visa ao desenvolvimento de métodos e técnicas capazes de auxiliar os decisores a realizarem escolhas de uma forma eficiente e eficaz.

Podemos contextualizar a Teoria da Decisão como uma disciplina que incorpora a *Teoria de Utilidade*, a *Teoria de Probabilidade* e a *Pesquisa Operacional*. A Teoria da Utilidade visa ao estudo de uma base comum para a comparação de valores e riscos diferentes, propiciando o conhecimento de preferências de agentes econômicos. A Teoria da Probabilidade fornece um mecanismo para lidar com incerteza ou com informações incompletas. Finalmente, a Pesquisa Operacional visa o desenvolvimento de métodos e técnicas para resolução de problemas e tomada de decisão. Os seguintes elementos são comuns em tomada de decisões, independentemente da natureza ou situação:

- ✓ **Decisor:** refere-se ao indivíduo ou grupo de indivíduos responsável pela escolha da estratégia a seguir.
- ✓ **Objetivo:** é a especificação do que se pretende alcançar.
- ✓ **Escala de valor ou preferência:** refere-se ao critério adotado pelo decisor para selecionar alguma alternativa, medindo a situação ou valor de utilidade que o decisor associa a cada resultado.
- ✓ **Soluções ou estratégias alternativas:** correspondem às diferentes alternativas de ação que o decisor pode escolher.
- ✓ **Estado da natureza ou ambiente:** refere-se à condição do ambiente em que as decisões estão, geralmente, fora do controle do decisor.
- ✓ **Resultado ou consequência:** refere-se ao resultado que pode ocorrer na implementação de uma determinada estratégia em um determinado estado da natureza ou ambiente.

O objetivo primordial da tomada de decisão empresarial é a maximização da utilidade do decisor, na prática traduzida pela *maximização do lucro* ou pela *minimização do custo*. Esse objetivo ocorre em qualquer das seguintes situações em que se pode conceber o uso de um sistema de apoio à tomada de decisão:

Situação de certeza: em que se tem informações completas, conhecendo-se *a priori* o resultado associado a cada ação. Essa situação é tipicamente indicada para o uso das técnicas de Programação Matemática.

Situação de risco ou incerteza: em que se tem informações parciais, sabendo-se que para cada ação podem resultar duas ou mais consequências, cada uma associada a um estado da natureza cuja probabilidade seja conhecida. A complexidade computacional imposta por esses modelos pode nos induzir ao uso de métodos heurísticos, em detrimento do ideal de otimização.

Situação de conflito: é aquela em que o estado da natureza é substituído por um oponente racional que visa, ao mesmo tempo, a maximizar a sua utilidade e a minimizar a utilidade do adversário. Tomada de decisão em situação de conflito é estudada em *Teoria dos Jogos*.

A Programação Matemática tipicamente ampara a Pesquisa Operacional em seu contexto de atuação. De uma forma geral, podemos dizer que a contribuição da Programação Matemática dentro da modelagem e solução de problemas de decisão direciona-se preferencialmente às situações de certeza permitindo, em casos reais:

- ✓ Estabelecer melhorias mensuráveis na operação do sistema.
- ✓ Automatizar processos e identificar gargalos operacionais.
- ✓ Fornecer pontos de referência para a análise e a avaliação da adequação de planos diretores e operacionais elaborados.
- ✓ Ajudar nas tomadas de decisões sobre as necessidades de expansão de instalações ou implantação de novas infraestruturas.
- ✓ Desenvolver análises comparativas de desempenho operacional.
- ✓ Determinar valores de referência para os diversos produtos em diferentes estágios da cadeia de produção, processamento, estocagem e transporte.

Como veremos no decorrer deste trabalho, a Programação Matemática poderá ser útil em quase todas as áreas das atividades produtivas. Nesse ponto, estamos aptos a completar a apresentação do grande quadro que emoldura o processo de modelagem, abordando a dimensão do fluxo das informações que permeia a atividade. A Figura 1.6 será útil nesse sentido. O esquema complementa o fluxograma da Figura 1.5, na medida em que detalha a inter-relação entre a elaboração do modelo e o processo cognitivo do modelador.

A fase de aquisição de conhecimento sobre o sistema normalmente é iniciada através de uma abordagem analítica. O sistema é examinado e, via de regra, decomposto em uma rede de relacionamentos causa x efeito para que possa ter sua arquitetura e desempenho entendidos e justificados. Após a análise, cabe ao modelador identificar as partes mais importantes e significativas do sistema, elaborando, para o caso, uma representação simplificada e eficiente. O processo de síntese caminha no sentido inverso ao da análise, reorganizando as partes já analisadas anteriormente só que agora sobre uma nova estrutura, a do modelo. A síntese é elaborada sobre uma representação tentativa que vai se adaptando na própria medida de sua construção. Trata-se de uma etapa pouco precisa, por alguns considerada

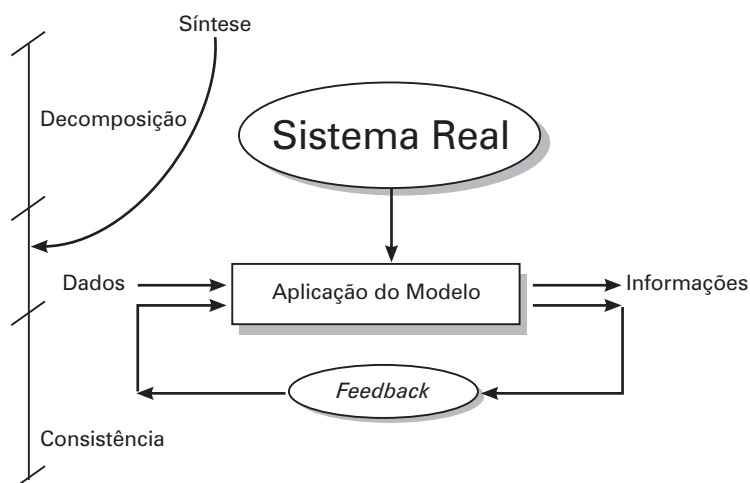


FIGURA 1.6 Modelagem matemática.

como uma *arte*, que resulta em um modelo protótipo. Concluído o desenho do modelo, cumpre testá-lo junto ao problema real. A fase de confrontação entre o modelo e o sistema representado é também denominada de *validação*, e sua aplicação prospectiva, de *simulação* (ver Figura 1.5). Como regra geral, a validação exige um procedimento iterativo que vai refinando paulatinamente o ajuste entre o modelo e a realidade. A base da validação é o funcionamento do modelo quando alimentado com dados reais e sua comparação com o desempenho do sistema original. Na maioria das ocasiões, isso é possível de ser feito, mas nem sempre. A operação de *feedback* ou realimentação da Figura 1.6 representa esse processo iterativo de ajuste (ver também etapa de *reformulação do modelo*), observado sob a face da interferência da consolidação das informações e sua transformação em conhecimento.

1.6.3 – Características dos Modelos de Programação Matemática

Como o processo de modelagem depende do espírito criativo do homem, talvez não possamos tão cedo definir claramente os limites dos modelos de Programação Matemática e de suas aplicações. Apesar disso, eles são indiscutivelmente mais habilitados a abordar o contexto tecnológico (sugerindo-se o uso de variáveis numéricas) dentro das decisões em situações de certeza. Genericamente, poderíamos dizer que seu emprego clássico seria:

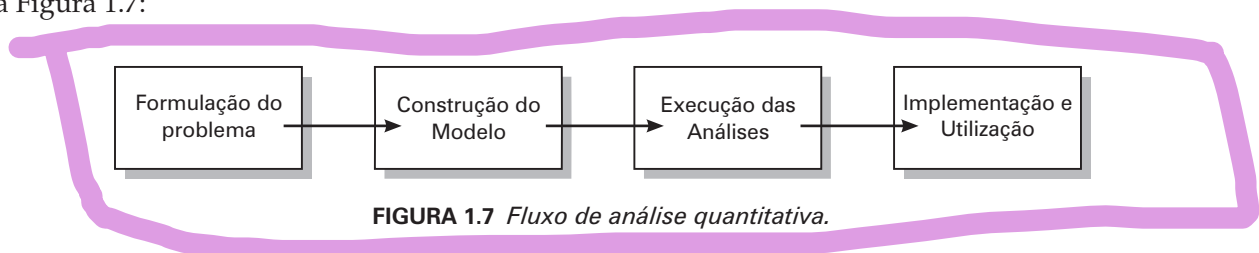
*Alocar de forma eficiente recursos limitados e que podem
ser disputados por atividades alternativas*

Nos modelos matemáticos, a representação de determinado sistema é geralmente realizada por um conjunto de equações ou expressões matemáticas. Se existem n decisões quantificáveis a serem tomadas, então pode-se associar a cada decisão uma variável do modelo denominada de forma mnemônica de *variável de decisão*, cujos valores o próprio modelo deverá determinar. Simbolicamente, as variáveis de decisão são representadas por letras minúsculas com índices como: x_i , $i = 1, 2, \dots, n$.

Para expressar a medida da eficácia procurada, normalmente uma *função numérica* das variáveis de decisão é estruturada. Essa função $z = f(x_1, \dots, x_n)$ é normalmente denominada *função objetivo*. O comportamento matemático dessa função pode ser o mais variado possível. Em programação linear essa função e todas as restrições impostas às variáveis são expressões lineares.

A limitação dos recursos pode ser traduzida para o modelo através de *restrições* aos valores das variáveis, que podem ser expressas matematicamente por meio de *equações* e *inequações*.

Podemos resumir de forma sucinta os passos do processo de *análise quantitativa* conforme expresso na Figura 1.7:



Como um modelo complexo, via de regra, não pode ser elaborado em uma só etapa, as características das fases descritas na Figura 1.7 são:

A etapa da *formulação* ou *tradução* das condições do problema para o contexto de modelagem:

- ✓ A definição de variáveis controláveis (de decisão ou controle) e não controláveis (externa ou de estado).
- ✓ A elaboração da função objetivo e do critério de otimização.
- ✓ A formalização das restrições do modelo.

A etapa da construção do modelo engloba:

- ✓ A elaboração da estrutura de entrada e saída de informações.
- ✓ As fórmulas de inter-relação.
- ✓ Os horizontes de tempo.

A etapa de execução das análises compreende:

- ✓ Análise da sensibilidade da solução.
- ✓ Levantamento da precisão dos dados.
- ✓ Estudo da estabilidade computacional.
- ✓ Levantamento das demais especificações do modelo.

A etapa de implementação dos resultados e atualização do modelo compreende:

- ✓ Um grande processo de *feedback* repassando as etapas anteriores, vivenciando o uso do modelo no sistema de produção ou prestação de serviços.

Um modelo construído da forma anteriormente descrita é também denominado *modelo de otimização*. É interessante observar ainda que os *modelos de otimização*, por objetivarem a redução do contexto em busca da simplificação, podem ser caracterizados pelas seguintes dicotomias:

Determinístico	⇔	Probabilístico
Restrito	⇔	Irrestrito
Monocritério	⇔	Multicritério
Contínuo	⇔	Discreto
Unidecisor	⇔	Diversos decisores
Univariável	⇔	Multivariável
Linear	⇔	Não linear
Uniobjetivo	⇔	Multiobjetivo

1.6.4 – Dificuldades de Representação na Modelagem Matemática

A dificuldade de *formular* um problema é semelhante à de *representar adequadamente* um sistema do mundo real. O primeiro nível de dificuldade incide sobre o modelador (ou a equipe) e está associado à sua capacidade de perceber os relacionamentos entre *causa e efeito* e encontrar as causas fundamentais. Um segundo nível engloba o domínio das técnicas de representação do fenômeno em um contexto diferente do original, o contexto do modelo. Apenas para introduzir o tema, levantaremos alguns obstáculos à modelagem:

Dificuldade face aos instrumentos de representação

Nem todos os fenômenos podem ser expressos em termos da lógica convencional, ou podem ter o seu comportamento representado por meio de variáveis de decisão. Eventualmente será necessária a identificação ou a criação de variáveis “artificiais” ou lógicas para a compatibilização e consistência da representação.

Dificuldade face ao grau de incerteza do comportamento das variáveis

Como vimos anteriormente, existem situações de risco e de conflito que dificultam a previsão do comportamento das variáveis no futuro, mesmo quando estas estão bem definidas. Alguns fenômenos, contudo, podem ultrapassar os limites do risco e ter um comportamento imprevisível por natureza. Tais casos são atualmente estudados dentro da *Teoria do Caos*. Em alguns casos, apenas parte do problema possui essa característica, o que representa um desafio ainda maior para as hipóteses simplificadoras.

Dificuldade com a ferramenta de solução

Os modelos devem ser implementáveis. Na atualidade a ferramenta de consenso para a análise quantitativa é o computador. Apesar de seus extraordinários recursos, em algumas situações, face à natureza das variáveis de decisão e de seu inter-relacionamento, mesmo ele pode ser inadequado à representação necessária. Infelizmente, esse instrumento pode impor severas restrições às conclusões teóricas de desempenho em virtude de, por exemplo, degenerescência numérica. A Figura 1.8 resume os hiatos acumulados nas etapas de tradução e construção de um modelo de otimização.

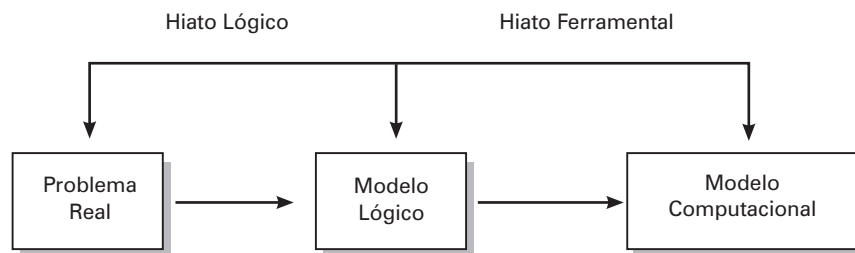


FIGURA 1.8 *Etapas de tradução de um modelo matemático.*

As dificuldades anteriormente elencadas implicam um número significativo de situações, que o modelo acaba se afastando do ideal pretendido para a otimização, ou por tornar-se muito complexo, ou pelo estado da arte não dispor das técnicas exatas necessárias. Nesse caso, os modelos poderão ser solucionados por técnicas não necessariamente exatas denominadas de heurísticas. Dessa forma, uma técnica heurística admite sacrificar o ideal de perfeição sugerido pela otimização em troca de melhorar a eficiência de um processo de busca. Ressaltamos que essa alternativa é razoável apenas quando não existe outra alternativa computacionalmente aceitável, ou quando o próprio modelo já abriu mão do ideal de perfeição.

Murphy (1997) apresenta uma interessante revisão e taxonomia da evolução dos modelos de Programação Linear. Trata-se de um texto de referência para quem deseja iniciar o aprofundamento no tema.

1.6.5 – Outras Técnicas para a Solução de Sistemas de Tomada de Decisão

Nas condições em que o ideal da otimização se torna praticamente pouco razoável, outras abordagens se apresentam bastante competitivas. Dentre elas destacaremos: a *Simulação* (SM) e a *Inteligência Artificial* (IA).

Simulação

A abordagem de simulação aplica-se quando situações de incerteza ou a própria complexidade do sistema dificulta o esforço de compreensão para o exato equacionamento do sistema, ou quando a magnitude do modelo de otimização o torna computacionalmente inviável. Os modelos de simulação contornam então essas dificuldades com um uso mais intensivo de dados estatísticos e com um maior esforço

de validação do modelo. A desvantagem dessa abordagem é a necessidade de um esforço maior dos especialistas na identificação de soluções não examinadas pelo modelo.

Inteligência Artificial

A Inteligência Artificial (IA) é o ramo da informática que concerne à automação do comportamento inteligente. No seu surgimento as expectativas da disciplina englobavam um ramo do conhecimento que procuraria retratar a mente com o uso de simulações no computador (ver LeDoux [1996]). Os primeiros problemas em IA envolviam jogos (o xadrez sobressaiu-se inicialmente) e provas automáticas de teoremas, pois tais problemas envolvem claramente funções cognitivas superiores. Essa tradição construiu uma ponte entre IA e PO, tanto pelo caráter lógico-matemático desses primeiros problemas, como pelas relações entre a lógica e a Programação Inteira (que serão exploradas no Capítulo 5 deste livro). Outro ponto comum dizia respeito ao inevitável uso das então emergentes técnicas de busca em grafos. Desde então, aprendeu-se que inteligência requer também a manipulação de uma conveniente base de conhecimento e que nenhum computador seria rápido o suficiente para dominar a explosão combinatória decorrente desses problemas. Essa realidade novamente uniu IA e PO na busca de algoritmos eficientes para o processamento do conhecimento. Além de diversas questões filosóficas envolvendo a própria definição de inteligência, consciência etc., a área de IA é rica em formalismos lógico-matemáticos envolvendo a representação e o processamento do conhecimento. O fundamento filosófico da disciplina desenvolveu-se principalmente associado ao funcionalismo (ver Neisser ([1976])). As técnicas de IA e PO se complementam em vários contextos, sendo marcante o uso, por ambas, de técnicas heurísticas.

Talvez um dos maiores obstáculos à integração entre as técnicas de PO e IA resida nas diferenças conceituais entre os paradigmas utilizados para a representação do conhecimento em cada uma das abordagens. O iconógrafo da Figura 1.3, proposto por Suntherland (1975), pode auxiliar nessa compreensão. Os modelos de otimização são menos sensíveis dos que os de IA em relação a variações no eixo do domínio e mais sensíveis em relação ao eixo da dinâmica. Por outro lado, para ser possível retratar os processos mentais de tomada de decisão, mesmo na linha do funcionalismo, o aspecto da redundância da arquitetura neuronal do cérebro é de difícil modelagem em ambas as técnicas. Modernos trabalhos na linha de inteligência computacional indicam novos paradigmas de representação do conhecimento como, por exemplo, o holográfico. Também acena-se com a inclusão de abordagens filosóficas mais amplas do que a proposta pelo funcionalismo (ver Emery [1969], Pribram [1976] e Teuber [1972], [1974]).



1.7 – O PAPEL DOS MODELOS QUANTITATIVOS DENTRO DA GESTÃO MODERNA

Apesar do conteúdo predominantemente especializado e técnico deste livro, cremos ser oportuno abordar, mesmo que rapidamente, o contexto maior em que essa tecnologia sofisticada será empregada. Aliás, essa é uma tendência cada vez mais visível na ciência, seu questionamento em favor do fim maior que é o homem, e sua qualidade de vida. Cremos ser indispensável despertar a consciência para a importância crescente do uso intensivo das ferramentas quantitativas dentro dos sistemas produtivos da atualidade, bem como alertar para seus limites e futuros desafios.

1.7.1 – O Problema da Gestão em um Mundo em Crise

O debate da busca da qualidade e produtividade como estratégia de aumento de competitividade é, atualmente, um ponto central para a gestão. O mundo globalizado está sofrendo uma extraordinária mutação no sentido de expandir a consciência dos clientes. As mudanças atuais são tão substanciais que algumas empresas questionam até mesmo a natureza de seus problemas. O comportamento humano, as pressões sociais, a ingerência do estado, a ecologia etc. acabaram transformando o ambiente organizacional em uma sopa de letras. É fundamental para quem pretende investir em um ferramental mais sofisticado para o apoio à tomada de decisão ser capaz de responder bem às seguintes perguntas:

- ✓ Como podemos estar certos de que de fato estaremos tratando “o problema principal” quando usarmos os modelos quantitativos para solucionar nossos casos?
- ✓ Não seriam os modelos quantitativos muito “simplificados” e “afastados” da complexa e incerta realidade organizacional?

Esse ponto é tão sério que utilizaremos as palavras do filósofo Karl R. Popper para o resumir:

“Para entender um problema, temos de tentar ao menos algumas das soluções mais óbvias e descobrir que elas falham; então, redescobrimos que existe uma dificuldade – um problema”

Karl R. Popper

Para podermos responder com sinceridade às perguntas anteriormente concretizadas e perceber que *na prática a teoria é diferente* especialmente no caso de *modelos incompletos* ou *mal formulados*, é indispensável desenvolver a visão estratégica do papel dos modelos quantitativos dentro do processo de suporte da tomada de decisão. Na realidade prática, uma empresa pretende:

-
- ✓ Proporcionar emprego eficiente para os fatores de produção.
 - ✓ Proporcionar um justo retorno dos fatores de entrada.
 - ✓ Satisfazer às necessidades e expectativas dos clientes em bens e serviços.
 - ✓ Proporcionar condições para a satisfação das necessidades dos funcionários.
 - ✓ Aumentar o bem-estar da sociedade.
-

Diante de um conjunto tão grande de objetivos é natural que se imponha o estabelecimento de prioridades. A busca da produtividade, a ênfase na otimização de tarefas e o princípio da *maximização*² não são atratores aleatórios. As organizações são fenômenos em que o esforço por reduzir a variabilidade de respostas é uma tarefa crítica e, por isso, demanda substancial atenção e esforço. A prioridade por estabilização de processos e previsibilidade acaba resultando em uma natural polarização focal na otimização de processos. É claro que devemos reconhecer que existem outros importantes aspectos a considerar para se desenvolver uma boa estratégia de gestão. Não é apenas no ideal quantitativo que se esgotam os problemas, o homem é o foco. É exatamente reconhecendo isso que poderemos melhor entender a importância do amparo das técnicas de PO na tomada de decisão.

Um bom gestor deve ser capaz de perceber que nem tudo poderá ser quantificado e que sua primeira prioridade é, de fato, o ser humano. Um bom gestor deve saber que é o material humano que fará a diferença; contudo, deverá também saber que para atender à dimensão humana, sua forma de administrar deverá equilibrar o emocional com o racional. Dominar a dimensão da ação racional não é uma opção política ou uma boa *peculiaridade pessoal*, é atualmente algo absolutamente indispensável, uma obrigação.

É certo que não é na otimização de recursos que a sobrevivência a longo prazo da empresa estará sendo garantida, mas sem ela a organização não tem qualquer chance no futuro. O problema da gestão

² Esse princípio afirma que as organizações procurarão sempre adquirir mais dos recursos que lhe fornecerão energia investida em produção. Isso resulta em fusões e crescimento de funções na tentativa de controlar o ambiente externo.

em um mundo globalizado tem deslocado seu eixo competitivo para a criatividade e a motivação humana, isso é indiscutível, mas cabe sempre alertar que *sem fazer certo as coisas, não há como fazer as coisas certas* de modo economicamente aceitável. Sem buscarmos o ideal da otimização não poderemos criar uma plataforma firme na qual a criatividade e a visão empreendedora possam se sustentar.

1.7.2 – O Dilema Competitividade × Vivacidade

As regras de sucesso da década de 1980 apontavam para um binômio que se convencionou denominar: qualidade e produtividade. Muitos de nós hoje falamos automaticamente essas palavras quando pensamos em competitividade. Na época em que esse binômio se difundiu, a qualidade era vista como um elemento adjetivo do produto e um fator de atração e valor. A produtividade, por outro lado, abordaria o aspecto da higidez organizacional. Resumindo, poderíamos dizer que *produtividade* consistiria em *fazer certo a coisa*, enquanto *qualidade* em *fazer a coisa certa*.

Nessa visão, os modelos quantitativos estariam mais fortemente associados ao aspecto da produtividade do que ao da qualidade. Pensando assim, alguns até seriam tentados a arriscar a velha e errada máxima de que “qualidade custa mais”. Edward Deming, com sua extraordinária capacidade de visão, já na década de 1970 sugeria a inadequação da dicotomia qualidade × produtividade. O problema dos modelos mentais é que eles normalmente acabam interferindo sobre a realidade que tentavam entender. Uma visão simplista da organização contribui para a criação de uma realidade igualmente simplista. Uma empresa pensada apenas sob um ponto de vista acaba realmente reduzindo-se a isso. Nem máquina nem instrumento de caridade, uma empresa é quase um organismo vivo que deverá ser capaz de equilibrar de forma saudável a substância técnica e a excelência na ação, com seres humanos motivados por terem suas necessidades e expectativas respeitadas na parceria capital × trabalho. Há de encontrar o equilíbrio.

O problema da atualidade é que as empresas precisam ser cada vez mais adaptativas para sobreviver em um mundo em mudanças. Ser adaptativo não é uma tarefa fácil.

“A adaptação é um conceito especial e oneroso, que deve ser empregado somente quando necessário.”

G. C. Willians

Na atualidade é importante saber qual a melhor estratégia a adotar para compatibilizar as respostas das empresas às constantes mudanças no meio ambiente organizacional. Nessa busca por respostas pelo menos quatro são os fatores a serem levados em conta:

Previsibilidade das mudanças

Mudanças imprevisíveis normalmente não possuem respostas prontas nos *esquemas* organizacionais. Nesse caso, não existe o que *otimizar*, há o que *criar*.

Complexidade dos esquemas disponíveis

Quanto menor e mais simples forem os *esquemas* da organização no momento da crise, menor chance terá ela de ter sucesso com a utilização de apenas uma estratégia de otimização dos recursos já existentes (estratégia ofensiva).

Tempo máximo para a resposta

As estratégias demandam tempo para sua implementação. A estratégia defensiva é mais rápida do que profundas reestruturações no tecido organizacional (estratégia evolutiva). A otimização disponibiliza rapidamente o melhor desempenho organizacional.

Natureza das mudanças ambientais ocorridas

Quanto mais profundas forem as mudanças, menos adequadas serão as respostas baseadas em pequenas reestruturações.

A Estratégia da Vivacidade para sobrevivência

Se falamos na sobrevivência contra a concorrência via *competitividade* estamos considerando que as regras que definirão o que seja a competitividade serão fixas, o que significa que estamos falando em concorrência leal, mercado livre etc. É dentro desse tipo de *competição* que o modelo biológico sugere preferencialmente as estratégias defensivas de adaptação. Na realidade atual do mundo empresarial, a *competição biológica* tradicional está ficando cada vez mais rara. Como a ocorrência de mudanças ambientais na atualidade não é uma hipótese possível mas um fato esperado, uma importante conclusão pode ser alcançada em relação às conhecidas estratégias de aumento de competitividade:

As estratégias que focalizam a competitividade como solução estão apostando que a abordagem adaptativa defensiva é o caminho para a redução das ameaças organizacionais.

Como nem todas as situações de mudança são iguais, cabe pensar na possibilidade de outros padrões de resposta. Usar a mesma solução para problemas diferentes pode não ser uma boa ideia. Se o meio ambiente muda, parece sábio o uso de uma estratégia voltada para fora da organização, ou seja, uma estratégia ofensiva. Isso implicaria pensar na “segurança” organizacional levando em conta conceitos como:

- ✓ Promover o maior número possível de esquemas diferentes em competição dentro da organização.
- ✓ Flexibilidade para a adoção dos novos esquemas.
- ✓ Boa saúde organizacional para suportar o constante estresse das reestruturações.

A sobrevivência é um intrincado resultado em uma teia de relacionamento. Na ótica da sobrevivência às mudanças ambientais, o desejável não é simplesmente uma capacidade competitiva relativa mas, em última análise, uma habilidade definitiva de adaptação mais profunda ou de segunda ordem: a *vivacidade*.

Vivacidade organizacional pode ser compreendida como o resultado final obtido por um corpo saudável, treinado e adaptado ao seu meio ambiente.

O poder de vivacidade é um resultado complexo obtido da interação dos inúmeros atuadores do ecossistema organizacional. Um detalhe interessantíssimo da vivacidade é que ela representa, em última análise, uma estratégia ampla para a mudança e, conseqüentemente, generalista. Com ela existe uma boa capacidade de resistir à concorrência desde que o cenário da mudança seja amplo ou intenso, incluindo a inovação (Porter [1990]). Buscando a vivacidade, o corpo organizacional se prepara para resistir às fortes mudanças ambientais e ser *competitivo* pela inovação. Através das estratégias defensivas a organização prepara-se para ser *competitiva* através da otimização do antigo arcabouço.

Um aspecto importante é perceber que as necessidades e expectativas de uma organização razoavelmente complexa devem estar voltadas para ambas as ameaças: concorrência e mudanças ambientais. Nesse sentido a preservação da vivacidade parece ser uma necessidade mais premente do que o foco único na concorrência por três razões:

- ✓ A *vivacidade* automaticamente traz consigo um razoável grau de *competitividade*.
- ✓ Independe do sentido da mudança. Trata-se de uma prontidão generalista para a sobrevivência pelo investimento no próprio organismo.
- ✓ Por estar fortemente associada aos elementos internos, sobre eles repercute diretamente. A palavra *competitividade* pode soar como uma abstração para o tecido organizacional, todavia a saúde da entidade e suas trocas com o meio ambiente repercutem diretamente sobre as pessoas.

Um outro modo de entender a importância da vivacidade como habilidade fundamental para a sobrevivência é perceber que a essência dos movimentos rumo à autoadaptação está muito mais associada à perseguição de objetivos de compensação de desequilíbrios e harmonização com o meio ambiente, do que do exercício do poder em benefício próprio. Edward Deming (Deming [1990]) percebeu essa verdade e por isso alertou que, de uma forma geral, a competição exacerbada é, ao contrário do que muita gente pensa, altamente danosa tanto como cultura (Judd [1995] e Winder [1994]) como para o aperfeiçoamento de um contexto. A lógica da “razão ao vencedor” cria perversos desequilíbrios que se encarregarão de, em última análise, reduzir a vivacidade de todos os participantes do contexto. A lógica da harmonia e autoajuste ressalta que *não é suficiente simplesmente sobreviver*. A sobrevivência é uma habilidade para ser exercida de forma contínua e sustentável. A lógica da luta por um lugar ao sol ou da conquista dos primeiros postos pode não incluir os principais elementos para a sustentabilidade dessa mesma posição. Otimizando o corpo organizacional para uma luta feroz no presente, a gestão poderá estar eliminando a possibilidade da continuidade desse organismo altamente especializado quando as regras mudarem. O objetivo da harmonização é mais amplo e poderoso. A vivacidade como habilidade mestre está voltada para garantir o desempenho futuro. De nada adianta vencer um combate e perder a guerra.

*A capacidade de vencer a concorrência não mantém a
vivacidade ao longo do tempo.*

Vivacidade inclui a preservação da habilidade de resposta de uma organização frente às incertezas da fluida realidade e representa um estágio além do domínio do mercado, lucro líquido ou aumento de produtividade. O truque está associado à identificação de atratores mais adequados e ao desenvolvimento das habilidades de ruptura e reestruturação dos “esquemas” de resposta. *É na interação com o meio ambiente e suas peculiaridades que está a resposta*. Mas, falando em segurança, seguro até quando? *“Seguro até a próxima mudança...”*

Voltando ao ponto de debate: no centro do problema encontramos a necessidade de conduzir uma entidade quase viva pelos caminhos ou descaminhos da nua e crua realidade. Na periferia encontramos a necessidade de perceber o que é fundamental e o que é acessório nesse processo.

Quando alguma coisa mudou na Terra e fez com que os dinossauros desaparecessem, o que aconteceu não foi um problema de concorrência. Grandes ou pequenos, carnívoros ou herbívoros, depois de milhões de anos de continuado sucesso... todos morreram.

*O que ocasionou a extinção dos dinossauros não foi uma
concorrência fraticida. Possivelmente foi uma surpreendente
mudança ambiental que os varreu da face da Terra.*

O que acontece é que sempre que houver uma mudança, *alguém* saberá tirar partido do fato e isso não é um problema de haver ou não haver uma guerra em curso. A capacidade de aproveitar a situação depende da capacidade de se adequar ou até mesmo da sorte. Ter sucesso em um novo contexto é completamente diferente de vencer o concorrente dentro do antigo contexto. No caso organizacional *a especialização desenfreada torna o corpo organizacional pronto para o momento atual, mas acaba criando a sua maior*

fragilidade futura: a falta de esquemas redundantes que possam competir para auxiliar na sua adaptação futura... e, em algumas áreas, atualmente isso parece ser tão ou mais importante do que a própria ameaça da concorrência. Um bom modelo de concepção de gestão deve procurar uma posição que otimize todas as variáveis do risco e não somente as associadas à sobrevivência imediatista. É necessário encontrar uma posição de equilíbrio entre a realização dos potenciais adaptativos pela especialização e sua preservação em formato generalista para os tempos futuros.

Vivacidade representa uma posição de Especialização Flexível.

Basicamente, o tema que estamos discutindo está relacionado a duas estratégias de sobrevivência com modelos mentais sutilmente diferentes. A estratégia da competitividade concentra atenção nas ameaças da concorrência e predadores. Seu modelo mental é o da confrontação e conquista. O meio ambiente é pouco notado nos aspectos oportunistas e mais percebido nos pontos ameaçadores. As técnicas desse modelo de concepção dão prioridade à produtividade e aos impactos de curto e médio prazo, pois cada novo momento é uma nova batalha. *A ideia básica é que o futuro pertence a quem for um vencedor no presente.* Sua filosofia poderia ser resumida em dois pensamentos falaciosos:

- ✓ Se não sobrevivermos ao hoje não chegaremos ao futuro.
- ✓ Se formos sempre os primeiros, sempre estaremos vivos.

A estratégia da vivacidade concentra atenção no ser humano e na sua adequação ao meio ambiente. O impacto das ações organizacionais sobre o meio ambiente é um dos fatores fundamentais na medida dessa adequação. É evidente que a ameaça da concorrência é percebida como um dos componentes desse mesmo meio ambiente, contudo o modelo de resposta não é obrigatoriamente a confrontação direta e nem exige a vitória. *A estratégia objetiva a harmonização com o meio ambiente* e é dentro desse contexto que os concorrentes serão considerados. Na mentalidade da vivacidade o objetivo não é ser o primeiro sempre e sim estar bem o tempo inteiro. O objetivo não é ser o maior ou o melhor e sim estar muito bem. O alvo não é conquistar todos os clientes do mundo mas *os clientes certos*. Não é necessário ser o campeão em todas as competições para se alcançar qualidade de vida; pelo contrário, o estresse da luta paranoica esvazia os louros da vitória e erradica a saúde.

Esse é um dos mais danosos paradigmas que a mentalidade da competitividade impõe: *O sucesso só pode ser alcançado após a vitória na guerra da concorrência.* Talvez a vitória sobre a concorrência só possa ser obtida com uma bela guerra, contudo os levantamentos estatísticos de desempenho não permitem afirmar que para se alcançar uma longa, saudável e profícua vida, obrigatoriamente alguém tenha de se envolver e vencer guerras. A história dessa mentalidade parece estar cheia de *vencedores*, mas possui poucos *sobreviventes*. Um trivial estudo nos mostra que os vencedores do passado simplesmente *sumiram*. Foram tão hábeis em conquistar o mercado... e daí? Seu único legado é a mentalidade que se perpetua, modelo mental que capitaliza as glórias e repassa para outros *culpados* os insucessos. Felizmente nem todos pensam que os vencedores têm razão.

1.7.3 – Os Modelos Quantitativos e a Vivacidade

Como ferramentas sofisticadas e poderosas, os modelos quantitativos são capazes de permitir ao tomador de decisão a necessária ampliação em seu campo de visão. As técnicas de otimização permitem no mínimo:

- ✓ O aumento do conhecimento sobre o comportamento do meio ambiente organizacional.
- ✓ O aumento do conhecimento sobre a própria organização.
- ✓ A antecipação de impactos e a simulação de políticas de ação.
- ✓ A economia de meios e recursos.

Os modelos quantitativos não são capazes de, sozinhos, fazerem uma empresa vencedora na tarefa de estar bem, mas evitam que as energias sejam gastas de forma irracional. Permitem que o tomador de decisão possa tirar o melhor proveito possível daquilo que ele dispõe. Os modelos quantitativos não definem uma estratégia para o futuro, mas podem ser capazes de examinar inúmeros cenários e ajudar o gestor a informar seus “sentimentos” das possibilidades mais prováveis. É impossível ao tecido organizacional alcançar um patamar razoável de vivacidade trabalhando longe da melhor realização de seus potenciais.

O papel dos modelos quantitativos no momento atual é oferecer ferramentas poderosas para assistir ao tomador de decisão e protegê-lo de erros evitáveis. Talvez uma das piores decisões que alguém possa tomar atualmente no mundo empresarial seja subutilizar ou desconhecer o uso do ferramental quantitativo, porque talvez não seja essa a decisão de seus concorrentes ou parceiros.

1.7.4 – Conclusão

No contexto descrito, em que muitas são as dúvidas e poucas as certezas, os modelos quantitativos permitem a ampliação da capacidade de abordar racionalmente aquilo que assim possa ser feito. Os modelos quantitativos são poderosos instrumentos para que o gestor possa ampliar seu sentido e sua percepção, economizando energias e aproveitando oportunidades. Finalmente:

*Os modelos quantitativos não tomam as decisões,
mas podem torná-las muito mais claras e fáceis.*

2

MODELOS DE PROGRAMAÇÃO LINEAR

2.1 – MODELOS DE PROGRAMAÇÃO LINEAR

2.1.1 – Características do Modelo de Programação Linear

Dentre os diversos modelos já citados, concentraremos nossa atenção inicialmente no *modelo de Programação Linear* (PL). Esse modelo é básico para a compreensão de todos os outros modelos da Programação Matemática. Os conceitos nele firmados serão estendidos aos demais, concedendo suporte a estudos mais avançados. Uma outra vantagem desse modelo está na extraordinária eficiência dos algoritmos de solução hoje existentes, disponibilizando alta capacidade de cálculo e podendo ser facilmente implementado até mesmo através de planilhas e com o auxílio de microcomputadores pessoais.

Os modelos de Programação Linear constituem um tipo especial de modelos de otimização. Para que um determinado sistema possa ser representado por meio de um modelo de PL, ele deve possuir as seguintes características:

Proporcionalidade: a quantidade de recurso consumido por uma dada atividade deve ser proporcional ao nível dessa atividade na solução final do problema. Além disso, o custo de cada atividade é proporcional ao nível de operação da atividade.

Não Negatividade: deve ser sempre possível desenvolver dada atividade em qualquer nível não negativo, e qualquer proporção de um dado recurso deve sempre poder ser utilizado.

Aditividade: o custo total é a soma das parcelas associadas a cada atividade.

Separabilidade: pode-se identificar de forma separada o custo (ou consumo de recursos) específico das operações de cada atividade.

Um modelo de Programação Linear é um modelo matemático de otimização no qual todas as funções são lineares da variável contínua x .

Formulação algébrica geral

Podemos formular de uma forma geral o Problema de Programação Linear (PPL) como se segue:

$$\text{Otimizar } z = \sum_{j=1}^n c_j x_j$$

sujeito a:

$$\begin{aligned} \sum_{j=1}^n a_{ij} x_j &\geq b_i \quad i = 1, 2, \dots, p \\ \sum_{j=1}^n a_{ij} x_j &= b_i \quad i = p+1, p+2, \dots, m \\ x_j &\geq 0, j = 1, 2, \dots, q \\ x_j &\in \mathfrak{R}, j = q+1, q+2, \dots, n \end{aligned}$$

Adotaremos as seguintes notações:

$M = \{1, 2, \dots, m\}$, o conjunto dos índices das restrições do problema;

$N = \{1, 2, \dots, n\}$ o conjunto dos índices das variáveis.

$M_1 \subset M$ e $N_1 \subset N$;

$A = \{a_{ij}\} \equiv$ matriz de restrições;

$a_j \equiv j$ -ésima coluna de A ;

$x = (x_j), j \in N$ / vetor coluna de n componentes;

$c = (c_j), j \in N$ / vetor linha de n componentes;

$d = (d_i), i \in M$ / vetor coluna de m componentes.

O termo *otimizar* é utilizado aqui para, genericamente, representar as possibilidades de maximizar ou minimizar a função objetivo. O problema consiste em, dados a matriz A e os vetores b e c , achar o vetor de variáveis contínuas x que satisfaça ao conjunto de restrições e que otimize o valor do critério z .

2.1.2 – Formulações Equivalentes

Além da *forma mista*, existem mais duas formas distintas de se desenvolver um PPL:

Forma Canônica

$$\text{Otimizar } z = cx$$

sujeito a:

$$\begin{aligned} Ax &\leq b \text{ ou } Ax \geq b \\ x &\geq 0 \end{aligned}$$

Forma Padrão

$$\text{Otimizar } z = cx$$

sujeito a:

$$\begin{aligned} Ax &= b \\ x &\geq 0 \\ \text{sendo } b &\geq 0 \text{ dado.} \end{aligned}$$

As formulações são absolutamente equivalentes pois através de operações elementares podemos transformá-las conforme a Figura 2.1.

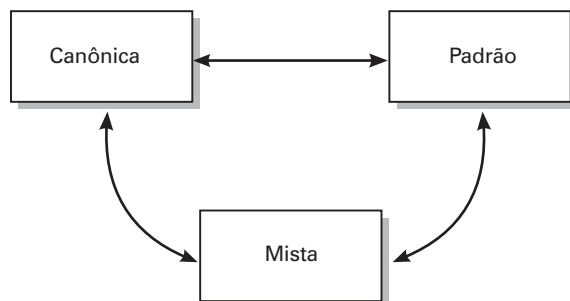


FIGURA 2.1 Transformações de formulações.

Operações elementares

Um mesmo modelo de PL, composto pelo conjunto de equações anteriormente apresentadas, pode, sem qualquer perda para suas propriedades matemáticas, ser reescrito em cada uma das formas básicas. Esse processo de tradução é realizado através das seguintes operações elementares:

Operação 1: mudança no critério de otimização, ou seja, transformação de maximização para minimização e vice-versa. Essa mudança pode ser realizada através da seguinte propriedade:

Maximizar $(f(x))$ corresponde a Minimizar $(-f(x))$ e
Minimizar $(f(x))$ corresponde a Maximizar $(-f(x))$.

Operação 2: transformação de uma variável livre $(x_j \in \mathbb{R})$, em variável não negativa. Nesse caso, a mudança exigirá a substituição da variável em transformação por duas variáveis auxiliares, ambas maiores ou iguais a zero, mas cuja soma é igual à variável original, ou seja:

$$x_n = x_n^1 - x_n^2 \text{ e } x_n^1 \geq 0, x_n^2 \geq 0$$

Operação 3: transformação de desigualdades em igualdades e vice-versa. Nessa situação, temos dois casos a examinar:

✓ Caso de transformação de restrições de menor ou igual em restrições de igualdade

Supondo a restrição que se segue:

$$x_1 + x_2 + \dots + x_n \leq b$$

Para transformá-la em uma restrição de igualdade, podemos introduzir uma **variável de folga** x_{n+1} capaz de “completar” a desigualdade, o que permite representar a restrição da seguinte forma:

$$x_1 + x_2 + \dots + x_n + x_{n+1} = b, \quad x_{n+1} \geq 0$$

✓ Caso de transformação de restrições de maior ou igual em restrições de igualdade

Supondo a restrição que se segue:

$$x_1 + x_2 + \dots + x_n \geq b$$

Para transformá-la em uma restrição de igualdade, podemos introduzir uma *variável de folga* x_{n+1} com coeficiente negativo capaz de “completar” a desigualdade, passando a representar a restrição da seguinte forma:

$$x_1 + x_2 + \dots + x_n - x_{n+1} = b, \quad x_{n+1} \geq 0$$

2.2 – MODELANDO PROBLEMAS ATRAVÉS DA PROGRAMAÇÃO LINEAR

2.2.1 – Passos para a Formulação de um PPL

Podemos decompor o processo de organização de um modelo de Programação Linear nas seguintes etapas:

✓ Definição das atividades

Após a análise do problema, as atividades que o compõem são definidas. Normalmente, associada a cada atividade uma unidade de medida deve ser adotada.

✓ Definição dos recursos

Considerando os *insumos disponíveis dentro de cada atividade*, determinam-se os recursos que estão sendo usados e produzidos em cada uma.

✓ Cálculo dos coeficientes de insumo/produção

É indispensável estabelecer claramente *como as atividades e os recursos estão relacionados* em termos de recursos necessários por unidade de atividade produzida.

✓ Determinação das condições externas

Considerando que os recursos são limitados, cumpre *determinar a quantidade de cada recurso disponível para o processo modelado*. Essas são as denominadas condições externas do modelo.

✓ Formalização do Modelo

Consiste em associar quantidades não negativas x_1, x_2, \dots, x_n a cada uma das atividades, escrever as equações de balanceamento e indicar o uso de cada recurso.

Para exemplificar o processo descrito anteriormente, apresentaremos agora uma série de exemplos simples de formulação de modelos de Programação Linear (PL). Esses problemas de formulação apresentam pequena dificuldade de tradução e objetivam exemplificar o processo de identificação de variáveis e constituição da estrutura do programa. Através do exame das formulações propostas, você poderá perceber as nuances básicas da modelagem matemática. Ao lado esquerdo de cada problema de formulação vamos sugerir um ícone conforme a seguinte escala:

- ☺ – fácil
- ☺ – pequeno grau de dificuldade
- ☹ – razoável grau de dificuldade
- ☹ – difícil
- ☹ – desafio

que estará associado a sua dificuldade de elaboração. Trata-se de uma graduação subjetiva que visa apenas a permitir uma orientação ao leitor. Na Seção 2.2.2, exemplificamos modelos de PL em que todas as variáveis são contínuas, o que facilita o cálculo do ótimo. Na Seção 2.2.3, apresentamos exemplos em que variáveis inteiras podem ser aproximadas por valores contínuos. Finalmente, na Seção 2.2.4, mostramos exemplos em que essa aproximação não é válida, resultando nos modelos de programação linear inteira, de muito maior dificuldade computacional.

2.2.2 – Exemplos de Modelagem de Problemas de PL Contínua

1 – O Problema das Ligas Metálicas



Uma metalúrgica deseja maximizar sua *receita bruta*. A Tabela 2.1 ilustra a proporção de cada material na mistura para a obtenção das ligas passíveis de fabricação. O preço está cotado em Reais por tonelada da liga fabricada. Também em toneladas estão expressas as restrições de disponibilidade de matéria-prima. Formular o modelo de Programação Matemática.

TABELA 2.1 RESTRIÇÕES/CUSTOS DO EXEMPLO 1

	<i>Liga Especial de Baixa Resistência (*)</i>	<i>Liga Especial de Alta Resistência (*)</i>	<i>Disponibilidade de Matéria-prima</i>
Cobre	0,5	0,2	16 Ton
Zinco	0,25	0,3	11 Ton
Chumbo	0,25	0,5	15 Ton
Preço de Venda (R\$ por Ton)	R\$3.000	R\$5.000	(*) $\frac{\text{Ton de minério}}{\text{Ton de liga}}$

Solução

O primeiro passo para a constituição de um modelo de Programação Matemática normalmente diz respeito à definição das variáveis de decisão. Essa etapa é simples quando a função objetivo e as restrições estão diretamente associadas a essas variáveis. No exemplo, o objetivo é maximizar a receita de uma produção cujos quantitativos relativos são passíveis de *programação*. O problema está colocado exatamente no planejamento desses quantitativos; portanto, nada mais natural que as variáveis do programa sejam os quantitativos das ligas produzidas.

1. Escolha da variável de decisão

$x_i \equiv$ quantidade em toneladas produzidas da liga especial de baixa resistência ($i = 1$) e especial de alta resistência ($i = 2$).

Com as variáveis de decisão escolhidas devemos expressar a função objetivo como uma função dessas variáveis:

2. Elaboração da função objetivo

$z = \text{Maximizar } \{f(x) = 3.000x_1 + 5.000x_2\}$

Receita bruta em Reais em função da quantidade produzida em toneladas de ligas especiais de baixa e alta resistência.

Evidentemente, nos programas lineares as variáveis de decisão deverão estar sujeitas às limitações tecnológicas e/ou econômicas. No presente exemplo, as restrições à produção das ligas estão associadas à disponibilidade dos metais de composição. Como para cada metal existe um nível de estoque máximo, teremos três restrições de disponibilidade de material. Eventualmente uma restrição pode englobar mais de uma exigência do enunciado, outras vezes uma exigência poderá necessitar de várias restrições tecnológicas para ser modelada.

3. Formulação das restrições tecnológicas

a) Restrição associada à disponibilidade do cobre:

$$0,5x_1 + 0,2x_2 \leq 16$$

b) Restrição associada à disponibilidade do zinco:

$$0,25x_1 + 0,3x_2 \leq 11$$

c) Restrição associada à disponibilidade do chumbo:

$$0,25x_1 + 0,5x_2 \leq 15$$

Na maioria dos problemas reais as variáveis só podem assumir valores nulos ou não negativos. São exemplos típicos variáveis que englobam peso, espaço, número de itens, configurações, pessoas etc. Eventualmente podemos dar sentido a valores negativos de tempo, unidades monetárias etc. Em última análise, contudo, os valores negativos das variáveis podem ser eliminados por convenientes transformações de variáveis dentro dos modelos de Programação Matemática (ver Figura 2.1 e operação elementar 2). Assim sendo, trabalharemos com variáveis não negativas.

4. Restrições de não negatividade

$$x_1 \geq 0, x_2 \geq 0$$

Chegando finalmente ao problema completo:

$$\text{Maximizar } z = 3.000x_1 + 5.000x_2$$

sujeito a:

$$0,5x_1 + 0,2x_2 \leq 16$$

$$0,25x_1 + 0,3x_2 \leq 11$$

$$0,25x_1 + 0,5x_2 \leq 15$$

$$x_1 \geq 0, x_2 \geq 0$$

2 – O Problema da Dieta



O objetivo do presente programa é determinar, em uma dieta para a redução calórica, as quantidades de certos alimentos que deverão ser ingeridos diariamente, de modo que determinados requisitos nutricionais sejam satisfeitos a custo mínimo. Existem vários problemas abordando esse tema, o presente exemplo é um dos mais simples possíveis.

Suponha que, por motivos justificáveis, uma certa dieta alimentar esteja restrita a leite desnatado, carne magra de boi, carne de peixe e uma salada de composição bem conhecida. Sabendo-se ainda que os requisitos nutricionais serão expressos em termos de vitaminas A, C e D e controlados por suas quantidades mínimas (em miligramas), uma vez que são indispensáveis à preservação da saúde da pessoa que estará se submetendo à dieta. A Tabela 2.2 resume a quantidade de cada vitamina em disponibilidade nos alimentos e a sua necessidade diária para a boa saúde de uma pessoa.

TABELA 2.2 RESTRIÇÕES DE NUTRIENTES NA DIETA ALIMENTAR DO EXEMPLO 6

<i>Vitamina</i>	<i>Leite (litro)</i>	<i>Carne (kg)</i>	<i>Peixe (kg)</i>	<i>Salada (100g)</i>	<i>Requisito Nutricional Mínimo</i>
A	2 mg	2 mg	10 mg	20 mg	11 mg
C	50 mg	20 mg	10 mg	30 mg	70 mg
D	80 mg	70 mg	10 mg	80 mg	250 mg
Custo	2 reais	4 reais	1,5 real	1 real	

Formular o programa para a otimização dos recursos envolvidos:

Solução

1. Escolha da variável de decisão

$x_i \equiv$ quantidade do alimento do tipo $i =$ (l-leite, c-carne, p-peixe, s-salada) a ser utilizada na dieta escolhida.

2. Elaboração da função objetivo

$$z = \text{Minimizar } \{f(x) = 2x_l + 4x_c + 1,5x_p + x_s\}$$

Número total de unidades monetárias gastas com a dieta.

3. Formulação das restrições tecnológicas

a) Restrição associada à demanda de vitamina A:

$$2x_l + 2x_c + 10x_p + 20x_s \geq 11$$

b) Restrição associada à demanda de vitamina C:

$$50x_l + 20x_c + 10x_p + 30x_s \geq 70$$

c) Restrição associada à demanda de vitamina D:

$$80x_l + 70x_c + 10x_p + 80x_s \geq 250$$

4. Restrições de não negatividade

$$x_l \geq 0, x_c \geq 0, x_p \geq 0, x_s \geq 0.$$

3 – O Problema do Sítio



Um sitiante está planejando sua estratégia de plantio para o próximo ano. Por informações obtidas nos órgãos governamentais, sabe que as culturas de trigo, arroz e milho serão as mais rentáveis na próxima safra. Por experiência, sabe que a produtividade de sua terra para as culturas desejadas é a constante na Tabela 2.3:

TABELA 2.3 RESTRIÇÕES DO PROBLEMA DO PLANTIO

Cultura	Produtividade em kg por m ² (experiência)	Lucro por kg de Produção (Informações do Governo)
Trigo	0,2	10,8 centavos
Arroz	0,3	4,2 centavos
Milho	0,4	2,03 centavos

Por falta de um local de armazenamento próprio, a produção máxima, em toneladas, está limitada a 60. A área cultivável do sítio é de 200.000 m². Para atender às demandas de seu próprio sítio, é imperativo que se plante 400 m² de trigo, 800 m² de arroz e 10.000 m² de milho.

Solução

1. Escolha da variável de decisão

Neste exemplo poderíamos ficar em dúvida sobre qual variável de decisão a utilizar: quantidade de quilos a ser produzida em cada cultura ou área a ser plantada. Optaremos pela área a ser plantada calculando-se, em consequência, as variáveis em área.

$$x_i \equiv \text{área em m}^2 \text{ a ser plantada da cultura do tipo } i = (T\text{-Trigo, } A\text{-Arroz, } M\text{-Milho}).$$

2. Elaboração da função objetivo

Os coeficientes da função objetivo deverão ser calculados multiplicando-se a produtividade por quilo pelo lucro previsto para cada quilo. O resultado do coeficiente será uma *unidade monetária*, no caso, o centavo.

$$z = \text{Maximizar } \{f(x) = 2,16x_T + 1,26x_A + 0,812x_M\}$$

Lucro em centavos.

3. Formulação das restrições tecnológicas

a) Restrições associadas à demanda do sítio (em unidade de área – m²):

$$x_T \geq 400$$

$$x_A \geq 800$$

$$x_M \geq 10.000$$

b) Restrição associada a área total disponível:

$$x_T + x_A + x_M \leq 200.000$$

c) Restrição associada ao armazenamento (em quilos):

Nesse caso teremos de utilizar os coeficientes da produtividade por unidade de área para obter um valor final em quilos:

$$0,2x_T + 0,3x_A + 0,4x_M \leq 60.000$$

4. Restrições de não negatividade

$$x_T \geq 0, x_A \geq 0, x_M \geq 0$$

4 – O Problema da Cooperativa Agrícola



Uma cooperativa agrícola opera três fazendas que possuem produtividades aproximadamente iguais entre si. A produção total por fazenda depende fundamentalmente da área disponível para o plantio e da água de irrigação. A cooperativa procura diversificar sua produção de modo que vai plantar este ano três tipos de cultura em cada fazenda, a saber: milho, arroz e feijão. Cada tipo de cultura demanda por uma certa quantidade de água. Para reduzir o conflito no uso das colheiteiras, que são alugadas pela cooperativa, estabeleceram-se limites de área de produção dentro de cada tipo de cultura. Para evitar a concorrência entre os cooperados, acordou-se que a proporção de área cultivada seja a mesma para cada uma das fazendas. As Tabelas 2.4 e 2.5 resumem os dados tecnológicos. Pede-se a elaboração de um programa de produção que defina a área de cada cultura que será plantada em cada fazenda, de modo a otimizar o lucro total da produção da cooperativa.

TABELA 2.4 ÁGUA DISPONÍVEL E ÁREA DE CULTIVO POR FAZENDA

<i>Fazenda</i>	<i>Área Total para Cultivo (Acre)</i>	<i>Água Disponível (Litros)</i>
1	400	1.800
2	650	2.200
3	350	950

Solução

1. Escolha da variável de decisão

A variável de decisão está imposta no próprio enunciado.

$x_{ij} \equiv$ área em acres que, na fazenda i ($i = 1, 2, 3$), será destinada à cultura j ($j = M$ -Milho, A -Arroz, F -Feijão).

TABELA 2.5 CONSUMO DE ÁGUA, ÁREA DE CULTIVO E LUCRO POR CULTURA

<i>Cultura</i>	<i>Área Máxima de Cultivo (Acre)</i>	<i>Consumo de Água (Litros por Acre)</i>	<i>Lucro (R\$/Acre)</i>
<i>Milho</i>	660	5,5	5.000
<i>Arroz</i>	880	4	4.000
<i>Feijão</i>	400	3,5	1.800

2. Elaboração da função objetivo

$z = \text{Maximizar } \{f(x) = 5.000 (x_{1M} + x_{2M} + x_{3M}) + 4.000 (x_{1A} + x_{2A} + x_{3A}) + 1.800 (x_{1F} + x_{2F} + x_{3F})\}$ – soma dos lucros em cada cultura em cada fazenda.

3. Formulação das restrições tecnológicas

a) Restrições associadas à área de cultivo:

- Fazenda 1

$$x_{1M} + x_{1A} + x_{1F} \leq 400$$

- Fazenda 2

$$x_{2M} + x_{2A} + x_{2F} \leq 650$$

- Fazenda 3

$$x_{3M} + x_{3A} + x_{3F} \leq 350$$

b) Restrições associadas ao consumo de água:

- Fazenda 1

$$5,5x_{1M} + 4x_{1A} + 3,5x_{1F} \leq 1.800$$

- Fazenda 2

$$5,5x_{2M} + 4x_{2A} + 3,5x_{2F} \leq 2.200$$

- Fazenda 3

$$5,5x_{3M} + 4x_{3A} + 3,5x_{3F} \leq 950$$

c) Restrições associadas ao plantio por cultura:

- Milho

$$x_{1M} + x_{2M} + x_{3M} \leq 660$$

- Arroz

$$x_{1A} + x_{2A} + x_{3A} \leq 880$$

- Feijão

$$x_{1F} + x_{2F} + x_{3F} \leq 400$$

d) Restrições associadas à proporção de área cultivada:

$$\frac{x_{1M} + x_{1A} + x_{1F}}{400} = \frac{x_{2M} + x_{2A} + x_{2F}}{650} = \frac{x_{3M} + x_{3A} + x_{3F}}{350}$$

4. Restrições de não negatividade

$$x_{1M} \geq 0, x_{1A} \geq 0, x_{1F} \geq 0, x_{2M} \geq 0, x_{2A} \geq 0, x_{2F} \geq 0, x_{3M} \geq 0, x_{3A} \geq 0, x_{3F} \geq 0$$

5 – O Problema da Mistura de Petróleo



Uma refinaria processa vários tipos de petróleo. Cada tipo de petróleo possui uma planilha de custos diferente, expressando condições de transporte e preços na origem. Por outro lado, cada tipo de petróleo representa uma configuração diferente de subprodutos para a gasolina. Na medida em que um certo tipo de petróleo é utilizado na produção da gasolina, é possível a programação das condições de oc-tanagem e outros requisitos. Esses requisitos implicam a classificação do tipo da gasolina obtida.

Supondo que a refinaria trabalhe com uma linha de quatro tipos diferentes de petróleo e deseje produzir as gasolinas amarela, azul e superazul, programar a mistura dos tipos de petróleo atendendo às condições que se seguem nas Tabelas 2.6 e 2.7:

TABELA 2.6 QUANTIDADE DISPONÍVEL DE PETRÓLEO

<i>Tipo de Petróleo</i>	<i>Quantidade Máxima Disponível (Barril/dia)</i>	<i>Custo por Barril/dia (R\$)</i>
1	3.500	19
2	2.200	24
3	4.200	20
4	1.800	27

TABELA 2.7 PERCENTUAIS PARA LIMITES DE QUALIDADE DAS GASOLINAS

<i>Tipo de Gasolina</i>	<i>Especificação</i>	<i>Preço de Venda R\$/Barril</i>
Superazul	Não mais que 30% de 1 Não menos que 40% de 2 Não mais que 50% de 3	35
Azul	Não mais que 30% de 1 Não menos que 10% de 2	28
Amarela	Não mais que 70% de 1	22

Solução

1. Escolha da variável de decisão

Para auxiliar na percepção da variável de decisão a adotar, o esquema a seguir poderá ser útil.

Verificamos no esquema que existe uma quantidade de cada tipo de petróleo em cada tipo de gasolina (no caso, a gasolina azul). Neste problema podemos adotar como variável de decisão:

x_{ij} \equiv número de barris de petróleo do tipo j , $j = 1, 2, 3, 4$, que serão destinados à produção da gasolina i , (i = A-gasolina Amarela, Z-gasolina aZul, S-gasolina Superazul).

2. Elaboração da função objetivo

$z = \text{Maximizar } \{f(x) = 22(x_{A1} + x_{A2} + x_{A3} + x_{A4}) + 28(x_{Z1} + x_{Z2} + x_{Z3} + x_{Z4}) + 35(x_{S1} + x_{S2} + x_{S3} + x_{S4}) - 19(x_{A1} + x_{Z1} + x_{S1}) - 24(x_{A2} + x_{Z2} + x_{S2}) - 20(x_{A3} + x_{Z3} + x_{S3}) - 27(x_{A4} + x_{Z4} + x_{S4}) = \text{soma do lucro resultante da venda dos diversos tipos de gasolina, abatidos os custos dos diversos tipos de petróleo utilizados.}$

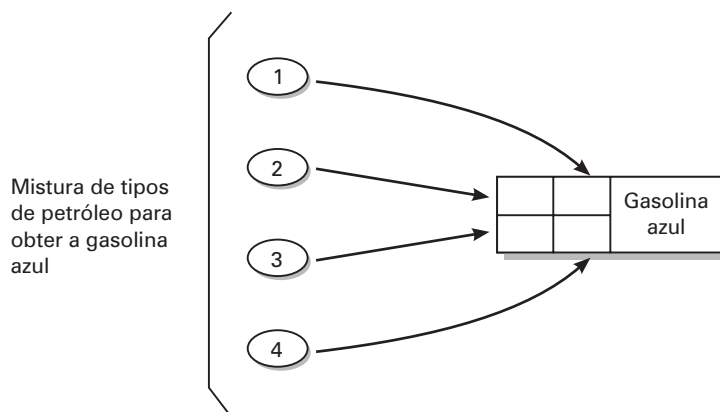


FIGURA 2.2 Percentagem de mistura de petróleos.

3. Formulação das restrições tecnológicas

a) Restrições associadas à quantidade de petróleo disponível:

- Tipo 1
 $x_{A1} + x_{Z1} + x_{S1} \leq 3.500$
- Tipo 2
 $x_{A2} + x_{Z2} + x_{S2} \leq 2.200$
- Tipo 3
 $x_{A3} + x_{Z3} + x_{S3} \leq 4.200$
- Tipo 4
 $x_{A4} + x_{Z4} + x_{S4} \leq 1.800$

b) Restrições associadas às especificações da mistura:

- Para a gasolina superazul
 $x_{S1} \leq 0,3 (x_{S1} + x_{S2} + x_{S3} + x_{S4})$
 $x_{S2} \geq 0,4 (x_{S1} + x_{S2} + x_{S3} + x_{S4})$
 $x_{S3} \leq 0,5 (x_{S1} + x_{S2} + x_{S3} + x_{S4})$
- Para a gasolina azul
 $x_{Z1} \leq 0,3 (x_{Z1} + x_{Z2} + x_{Z3} + x_{Z4})$
 $x_{Z1} \geq 0,1 (x_{Z1} + x_{Z2} + x_{Z3} + x_{Z4})$
- Para a gasolina amarela
 $x_{A1} \leq 0,7 (x_{A1} + x_{A2} + x_{A3} + x_{A4})$

4. Restrições de não negatividade

$$x_{A1} \geq 0, x_{A2} \geq 0, x_{A3} \geq 0, x_{A4} \geq 0, x_{Z1} \geq 0, x_{Z2} \geq 0, x_{Z3} \geq 0, x_{Z4} \geq 0,$$

$$x_{S1} \geq 0, x_{S2} \geq 0, x_{S3} \geq 0, x_{S4} \geq 0.$$

6 – O Problema do Fluxo de Petróleo na Refinaria



Uma refinaria produz dois tipos de óleo, I e II, que passam por refino em quatro centros de processamento, conforme a Figura 2.3. As linhas cheias do gráfico indicam o fluxo normal de refino para os óleos

do tipo I e II. Havendo capacidade ociosa, é possível processar o tipo de óleo I através do esquema alternativo representado pelas linhas tracejadas.

No esquema de produção, sabe-se que a distribuição custo \times capacidade de produção é:

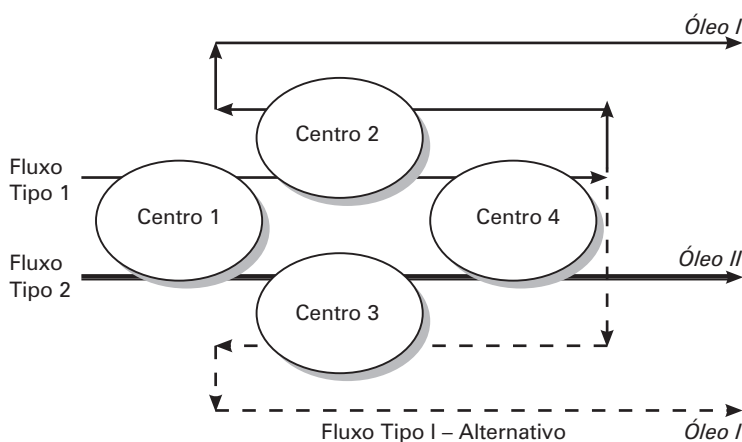


FIGURA 2.3 Fluxos de óleo.

TABELA 2.8 CUSTOS E CAPACIDADES DE PRODUÇÃO

Produto	Centro	Capacidade l/h	% de Recuperação	Custo \$/h
I	1	300	90	150
	2	450	95	200
	4	250	85	180
	2	400	80	220
	3	350	75	250
II	1	500	90	300
	3	480	85	250
	4	400	80	240

Por outro lado, as relações econômicas que regem a função lucro são:

TABELA 2.9 CUSTOS/PREÇOS DOS PRODUTOS

Produto	Custo da Matéria-prima (\$/l)	Preço de Venda (\$/l)	Venda Diária Máxima (l)
I	5	20	1.700
II	6	18	1.500

Os centros 1 e 4 operam 16 horas por dia. Os centros 2 e 3 operam 12 horas por dia. A refinaria possui a capacidade de transportar somente 2.500 l/dia, pois seu oleoduto está em manutenção. Formule matematicamente o problema de otimizar a produção dos dois tipos de óleo.

Solução

1. Escolha da variável de decisão

$x_i \equiv$ número de litros do óleo do tipo i , ($i = I$, para o óleo do tipo I, II, para o óleo do tipo II) refinados diariamente. Nesse caso, vamos necessitar distinguir na variável de decisão o esquema de refino. Então,

podemos desdobrar a variável x_I em uma soma de duas parcelas, a saber: a parcela do fluxo tipo I normal, x_I^N , e a parcela obtida via fluxo tipo I alternativo, x_I^A :

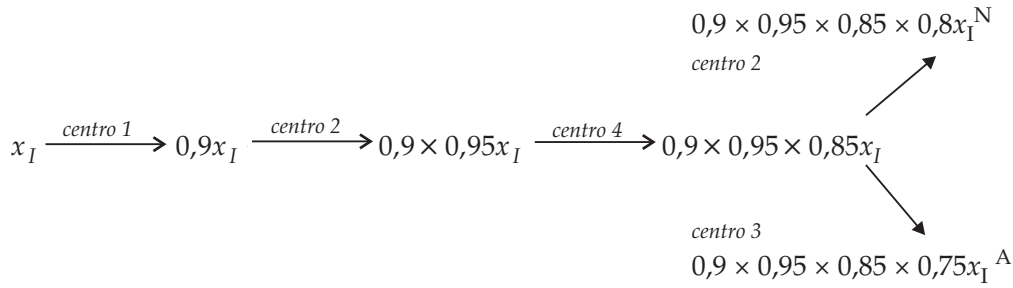
$$x_I = x_I^N + x_I^A$$

2. Formulação das restrições tecnológicas

Neste problema é mais interessante nos debruçarmos inicialmente sobre as restrições tecnológicas. A função objetivo dependerá dos coeficientes de aproveitamento de refino que serão calculados na ocasião da organização das restrições tecnológicas.

Inicialmente é indispensável perceber que à medida que o petróleo vai sendo refinado em cada centro, existe uma perda percentual na matéria-prima. Essa perda é acumulada em cada circuito. A seguir representamos o fluxo de perda de x_I e x_{II} .

a) Fluxo de perdas de x_I :



b) Fluxo de perdas de x_{II} :

$$x_{II} \xrightarrow{\text{centro 1}} 0,9x_{II} \xrightarrow{\text{centro 3}} 0,9 \times 0,85x_{II} \xrightarrow{\text{centro 4}} 0,9 \times 0,85 \times 0,80x_{II}$$

c) Restrições associadas à capacidade de processamento:

Centro 1:

$$\frac{x_I^A}{300} + \frac{x_I^N}{300} + \frac{x_{II}}{500} \leq 16$$

Centro 2:

$$\frac{0,9x_I^N}{450} + \frac{0,9x_I^A}{450} + \frac{0,9 \times 0,95 \times 0,85x_I^N}{400} \leq 12$$

Centro 3:

$$\frac{0,9x_{II}}{480} + \frac{0,9 \times 0,95 \times 0,85x_I^A}{350} \leq 12$$

Centro 4:

$$\frac{0,9 \times 0,95x_I^N}{250} + \frac{0,9 \times 0,95x_I^A}{250} + \frac{0,9 \times 0,85x_{II}}{400} \leq 16$$

d) Restrição de transporte:

$$0,9 \times 0,95 \times 0,85 \times 0,8x_I^N + 0,9 \times 0,95 \times 0,85 \times 0,75x_I^A + 0,9 \times 0,85 \times 0,80x_{II} \leq 2.500$$

e) Restrição de venda:

- Óleo tipo I

$$0,9 \times 0,95 \times 0,85 \times 0,8x_I^N + 0,9 \times 0,95 \times 0,85 \times 0,75x_I^A \leq 1.700$$

- Óleo tipo II

$$0,9 \times 0,85 \times 0,80x_{II} \leq 1.500$$

3. Restrições de não negatividade

$$x_I \geq 0, x_I^N \geq 0, x_I^A \geq 0$$

4. Elaboração da função objetivo

Denominando por:

$$\begin{aligned} a_N &= 0,9 \times 0,95 \times 0,85 \times 0,8 \\ a_A &= 0,9 \times 0,95 \times 0,85 \times 0,75 \quad e \\ b &= 0,9 \times 0,85 \times 0,80 \end{aligned}$$

$z = \text{Maximizar } \{f(x) = \text{Receita} - \text{Despesa}\}$, onde

Receita:

$$20a_N x_I^N + 20a_A x_I^A + 18bx_{II}$$

Despesa:

- Matéria-prima

$$5x_I^N + 5x_I^A + 6x_{II}$$

- Custos Operacionais:

Centro 1:

$$\frac{150x_I^N}{300} + \frac{150x_I^A}{300} + \frac{300x_I^N}{500}$$

Centro 2:

$$\frac{200 \times 0,9x_I^N}{450} + \frac{200 \times 0,9x_I^A}{450} + \frac{220 \times 0,9 \times 0,95 \times 0,85x_I^N}{400}$$

Centro 3:

$$\frac{250 \times 0,9x_{II}}{480} + \frac{250 \times 0,9 \times 0,95 \times 0,85x_I^A}{350}$$

Centro 4:

$$\frac{180 \times 0,9 \times 0,95x_I^N}{250} + \frac{180 \times 0,9 \times 0,95x_I^A}{250} + \frac{240 \times 0,9 \times 0,85x_{II}}{400}$$

Representação matricial

Como constatamos facilmente, o número de variáveis dos problemas de PL pode ser muito grande, de modo que a representação que adotamos nos exemplos da série introdutória pode tornar-se rapidamente impraticável, recomendando-se a forma matricial.

7 – O Problema da Otimização de Padrões de Corte



Uma fábrica necessita cortar uma fita de aço de 12 cm de largura em tiras de 2,4 cm, 3,4 cm e 4,5 cm de largura. As necessidades globais das tiras são:

TABELA 2.10 NECESSIDADES GLOBAIS DE TIRAS

Tipo de Tira	Largura (cm)	Comprimento Mínimo
Tira 1	2,4	2.500 m
Tira 2	3,4	4.500 m
Tira 3	4,5	8.000 m

Formule o problema que permite otimizar o consumo da fita a ser cortada minimizando a perda de material.

Solução

Diferentemente do modelo 11, as configurações possíveis para o corte não estão ainda definidas. A Figura 2.4 mostra uma das configurações viáveis de corte. A Tabela 2.11 resume as demais configurações:

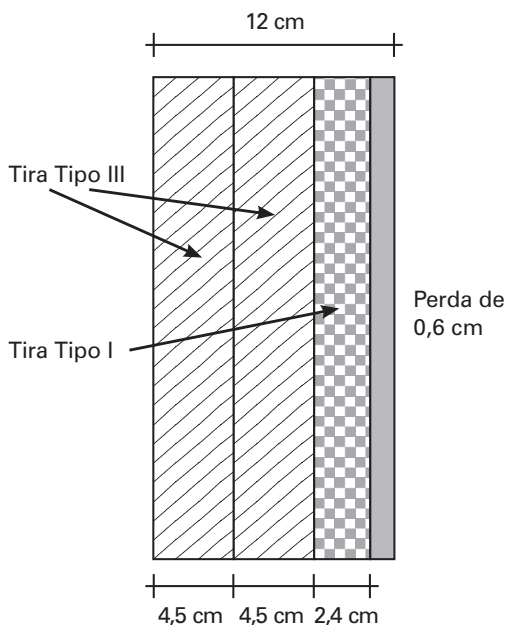


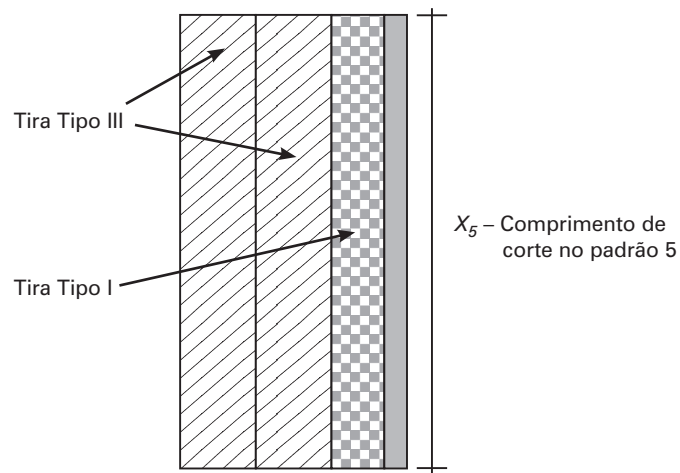
FIGURA 2.4 Padrão de corte número 5 (ver Tabela 2.11).

TABELA 2.11 PADRÕES DE CORTE DA FITA

<i>Padrão de Corte (i)</i>	<i>Número de Tiras Tipo 1 – 2,4 cm</i>	<i>Número de Tiras Tipo 2 – 3,4 cm</i>	<i>Número de Tiras Tipo 3 – 4,5 cm</i>	<i>Perda no Padrão (P_i)</i>
<i>Padrão 1</i>	5	0	0	0
<i>Padrão 2</i>	3	1	0	1,4
<i>Padrão 3</i>	3	0	1	0,3
<i>Padrão 4</i>	2	2	0	0,4
<i>Padrão 5</i>	1	0	2	0,6
<i>Padrão 6</i>	0	3	0	1,8
<i>Padrão 7</i>	0	2	1	0,7

1. Escolha da variável de decisão

$x_i \equiv$ comprimento de corte no padrão i , ($i = 1, \dots, 7$) e folga das restrições a), b) e c) ($i = 8, 9, 10$), a serem utilizadas na produção.

FIGURA 2.5 Variável de decisão: comprimento do corte no padrão i .

2. Formulação das restrições tecnológicas

a) Restrição associada à demanda da tira do tipo I:

$$5x_1 + 3x_2 + 3x_3 + 2x_4 + x_5 \geq 2.500$$

b) Restrição associada à demanda da tira do tipo II:

$$x_2 + 2x_4 + 3x_6 + 2x_7 \geq 4.500$$

c) Restrição associada à demanda da tira do tipo III:

$$x_3 + 2x_5 + x_7 \geq 8.000$$

3. Elaboração da função objetivo

$z = \text{Minimizar } \{\text{perdas nos padrões} + \text{perdas no excesso de tiras}\}$

a) Perdas nos padrões de corte:

$$1,4x_2 + 0,3x_3 + 0,4x_4 + 0,6x_5 + 1,8x_6 + 0,7x_7$$

b) Perda pelo excesso de tiras:

Chamando por x_8 , x_9 e x_{10} (em metros) as folgas das restrições a, b e c, respectivamente, temos:

$$2,4x_8 + 3,4x_9 + 4,5x_{10}$$

4. Restrições de não negatividade

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0, x_6 \geq 0, x_7 \geq 0, x_8 \geq 0, x_9 \geq 0, x_{10} \geq 0.$$

8 – O Problema da Competição entre os Engenheiros



Duas equipes de fabricantes de carros de corrida estão praticamente disputando sozinhas o campeonato. A cada mês são disputadas três provas. A equipe vencedora é aquela que acumula o maior número de vitórias durante todas as corridas do ano. Assim que o campeonato se iniciou, a Equipe B passou a vencer a Equipe A em cerca de 60% das provas. O carro da Equipe B acabava sempre ultrapassando nas curvas o carro da equipe A. Tentando anular essa vantagem, os engenheiros da Equipe A desenvolveram um aerofólio especial para dar mais estabilidade ao carro nas curvas. Apesar de ficar com o carro mais pesado, o piloto da Equipe A podia, agora, usar sua habilidade e impedir a ultrapassagem do carro da Equipe B nas curvas, mantendo ou ganhando sua posição nas retas. Com a melhoria tecnológica, a Equipe A passou a vencer cerca de 80% das corridas. Vendo o desastre se aproximar, os engenheiros da Equipe B contra-atacaram criando um estabilizador de suspensão para melhorar a manobrabilidade nas curvas. O resultado anulou a habilidade do piloto da Equipe A em bloquear o carro da Equipe B nas curvas e esta passou a vencer cerca de 80 % das provas. Acuada e sem ter como aumentar mais a aderência do carro nas curvas, a Equipe A, em uma reunião de emergência, tentou uma estratégia suicida: retirar o aerofólio para ganhar com o alívio de peso e tentar compensar nas retas o que fatalmente perderia nas curvas. Surpreendentemente, com essa estratégia passou a ganhar 80% das provas, pois o estabilizador de suspensão especial dos carros da Equipe B era muito pesado e inútil nas retas, reduzindo sobremaneira o desempenho do carro fora das curvas. Imediatamente, a Equipe B retirou seu equipamento e as escuderias voltaram ao ponto de partida com vantagem novamente para a Equipe B. A essa altura do campeonato, as duas equipes estavam rigorosamente empatadas em número de vitórias. O chefe da Equipe A resolveu contratar um assessor de Pesquisa Operacional para decidir a melhor política para sua equipe e, quem sabe, dar-lhe o título daquele conturbado ano. Formule e solucione o problema de encontrar a política ótima de equipagem para o carro da Equipe A. Sabe-se que a instalação do estabilizador especial e do aerofólio leva cerca de 4 horas e não pode ser *desfeita* ao longo da corrida, e que os boxes de cada equipe podem ser fechados, de forma que a equipe concorrente não saiba qual configuração será adotada em uma corrida pela sua adversária.

Solução

O que o consultor de Pesquisa Operacional deverá ser capaz de responder é a frequência com a qual os carros da Equipe A, seu contratante, deverão ser equipados com o aerofólio.

1. Escolha da variável de decisão:

Seja x_1 a probabilidade da Equipe A resolver equipar seu carro com o aerofólio. Nesse caso, $x_2 = 1 - x_1$ será a probabilidade de o aerofólio não ser empregado. Pelas experiências declaradas no enunciado, a situação do confronto tecnológico pode ser resumida como:

	<i>Estratégia</i>	<i>Equipe B</i>	
		<i>Sem Estabilizador</i>	<i>Com Estabilizador</i>
<i>Equipe A</i>	<i>Sem Aerofólio</i>	40 %	80%
	<i>Com Aerofólio</i>	80%	20%

A variável de decisão deve ser avaliada pelo valor esperado $E(.)$ do “ganho” obtido pela Equipe A nas condições de resposta da equipe B, ou seja, com ou sem estabilizador.

Vamos inicialmente considerar o evento S_1 , a condição de ganho da Equipe A, quando a Equipe B permanece sem seu fantástico estabilizador de suspensão. Nesse caso, a esperança de vitória da Equipe A será:

$$E(S_1) = 40x_1 + 80(1 - x_1) = 80 - 40x_1$$

Temos ainda uma outra condição a levar em conta: o caso de a Equipe B estar utilizando seu estabilizador. Vamos chamar esse evento de S_2 . Nessa alternativa, a esperança de vitória da Equipe A torna-se:

$$E(S_2) = 80x_1 + 20(1 - x_1) = 60x_1 + 20$$

2. Formulação das restrições tecnológicas

Seja G , o menor “ganho” obtido pela Equipe A quando usar sua política. Nesse caso:

$$E(S_1) \geq G$$

$$E(S_2) \geq G$$

ou seja:

$$40x_1 + G \leq 80$$

$$-60x_1 + G \leq 20$$

As equações acima podem ser consideradas restrições de um programa linear no qual a variável de decisão é a probabilidade da Equipe A utilizar seu aerofólio. Elas representam o conjunto das possibilidades para o valor do ganho G .

3. Função objetivo

É óbvio que a Equipe A deseja maximizar seu “ganho”, ou seja:

$$z = \text{Maximizar } \{f(x) = G\}$$

4. Restrição de não negatividade e definição do valor probabilístico

Como sabemos que a probabilidade varia entre 0 e 1:

$$1 \geq x_1 \geq 0$$

Para esclarecer melhor a solução desse interessante modelo, apresentamos na Figura 2.6 a solução gráfica do problema 8.

5. Solução Gráfica

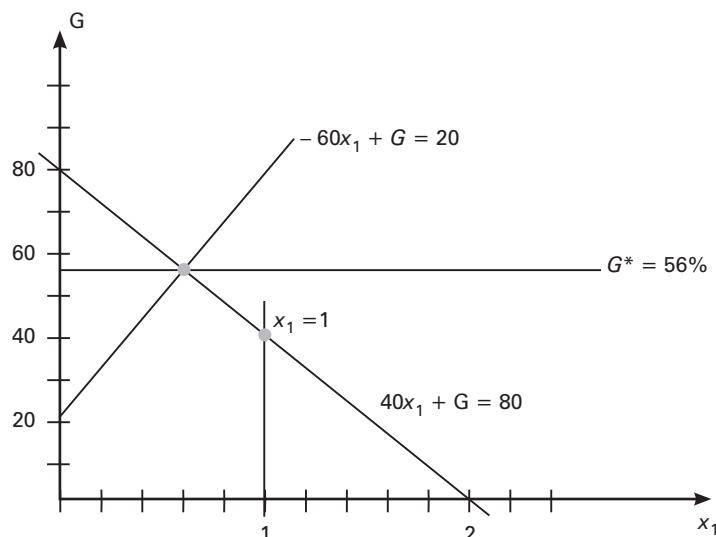


FIGURA 2.6 Solução gráfica do problema das equipes de corrida.

Que nos leva aos valores de $x_1 = 0,6$ e $G = 56\%$. Concluimos com isso que a Equipe A pode ser campeã se tiver, além do tal bom piloto, auxílio da Pesquisa Operacional!

2.2.3 – Exemplos de Modelagem de Problemas de PL Inteira com Possibilidade de Aproximação Contínua

1 – O Problema da Fábrica de Móveis



Uma grande fábrica de móveis dispõe em estoque de 250 metros de tábuas, 600 metros de pranchas e 500 metros de painéis de conglomerado. A fábrica normalmente oferece uma linha de móveis composta por um modelo de escrivaninha, uma mesa de reunião, um armário e uma prateleira. Cada tipo de móvel consome uma certa quantidade de matéria-prima, conforme a Tabela 2.12. A escrivaninha é vendida por 100 unidades monetárias (u. m.), a mesa por 80 u.m., o armário por 120 u.m. e a prateleira por 20 u.m. Pede-se exibir um modelo de Programação Linear que maximize a receita com a venda dos móveis.

TABELA 2.12 RESTRIÇÕES/CUSTOS DO EXEMPLO 2

	Quantidade de material em metros consumidos por unidade de produto				Disponibilidade do Recurso (m)
	Escrivaninha	Mesa	Armário	Prateleira	
Tábua	1	1	1	4	250
Prancha	0	1	1	2	600
Painéis	3	2	4	0	500
Valor de Venda (u.m.)	100	80	120	20	

Solução

Nesse caso, a escolha das variáveis de decisão segue o mesmo raciocínio do modelo anterior. O objetivo continua sendo maximizar a receita de uma produção cujos quantitativos relativos são passíveis de planejamento. A diferença ocorre por conta do maior número de quantitativos (ou variáveis a programar).

1. Escolha da variável de decisão

$x_i \equiv$ quantidade em unidades a serem produzidas do produto escrivaninha ($i = 1$), mesa ($i = 2$), armário ($i = 3$), prateleira ($i = 4$).

Com as variáveis de decisão escolhidas, devemos expressar a função objetivo como uma função dessas variáveis:

2. Elaboração da função objetivo

$$z = \text{Maximizar } \{f(x) = 100x_1 + 80x_2 + 120x_3 + 20x_4\}$$

Receita bruta em unidades monetárias em função do número de unidades produzidas de cada tipo de móvel.

O levantamento das restrições de produção apontam para a disponibilidade nos tipos de madeira.

3. Formulação das restrições tecnológicas

a) Restrição associada à disponibilidade de tábuas:

$$x_1 + x_2 + x_3 + 4x_4 \leq 250$$

b) Restrição associada à disponibilidade de pranchas:

$$x_2 + x_3 + 2x_4 \leq 600$$

c) Restrição associada à disponibilidade de painéis:

$$3x_1 + 2x_2 + 4x_3 \leq 500$$

4. Restrições de não negatividade

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$$

2 – O Problema do Atleta Indeciso



Um jovem atleta sente-se atraído pela prática de dois esportes: natação e ciclismo. Sabe por experiência que:

A *natação* exige um gasto em mensalidade do clube e deslocamento até a piscina que pode ser expresso em um custo médio de 3 (três) reais por seção de treinamento de duas horas.

O *ciclismo*, mais simples, acaba custando cerca de 2 (dois) reais pelo mesmo tempo de prática.

O orçamento do rapaz dispõe de 70 reais para seu treinamento.

Seus afazeres de aluno de graduação na universidade lhe dão liberdade de empregar, no máximo, 18 horas mensais e 80.000 calorias para os esforços físicos.

Cada seção de natação consome 1.500 calorias, enquanto cada etapa ciclística dispense 1.000 calorias. Considerando que o rapaz goste igualmente de ambos os esportes o problema consiste em planejar seu treinamento de forma a maximizar o número de seções de treinamento.

Solução

O presente modelo é um exemplo de programação de atividades. Obviamente as variáveis de decisão não estão limitadas a modelar quantitativos que dizem respeito apenas a produtos. O problema se propõe a tratar o número de seções esportivas.

1. Escolha da variável de decisão

$x_i \equiv$ número de práticas de natação ($i = 1$) e do ciclismo ($i = 2$).

2. Elaboração da função objetivo

$z = \text{Maximizar } \{f(x) = x_1 + x_2\}$

Número total de seções de prática desportiva.

Em relação às restrições, o presente problema diferencia-se dos anteriores por incluir vários aspectos associados à prática desportiva do rapaz.

3. Formulação das restrições tecnológicas

a) Restrição associada aos recursos financeiros do rapaz:

$$3x_1 + 2x_2 \leq 70$$

b) Restrição associada à disponibilidade de energia:

$$1.500x_1 + 1.000x_2 \leq 80.000$$

c) Restrição associada à disponibilidade de tempo:

$$2x_1 + 2x_2 \leq 18$$

4. Restrição de não negatividade

$$x_1 \geq 0, x_2 \geq 0.$$

3 – O Problema da Pequena Fábrica



Considere a situação de decidir sobre o número de unidades a serem produzidas por certo fabricante de dois diferentes tipos de produto.

Os lucros por unidade do produto_1 e do produto_2 são, respectivamente, 2 e 5 u.m. Cada unidade do produto_1 requer 3 horas de máquina e 9 unidades de matéria-prima, enquanto o produto_2 requer 4 horas de máquina e 7 unidades de matéria-prima.

Os tempos máximos disponíveis de horas de máquina e de matéria-prima são 200 horas e 300 unidades, respectivamente.

Formule o problema de forma a otimizar o lucro total.

Solução

1. Escolha da variável de decisão

$x_i \equiv$ quantidade de unidades do produto_1 ($i = 1$) e do produto_2 ($i = 2$) fabricadas.

2. Elaboração da função objetivo

$z = \text{Maximizar } \{f(x) = 2x_1 + 5x_2\}$

Número total de unidades monetárias obtidas como lucro pela venda dos produtos 1 e 2. Nesse caso “otimizar” o lucro total é sinónimo de maximizar. Em certas ocasiões, a função objetivo pode não ser explicitamente evidenciada.

3. Formulação das restrições tecnológicas

a) Restrição associada à disponibilidade do tempo de máquina:

$$3x_1 + 4x_2 \leq 200$$

b) Restrição associada à disponibilidade de matéria-prima:

$$9x_1 + 7x_2 \leq 300$$

4. Restrições de não negatividade

$$x_1 \geq 0, x_2 \geq 0.$$

4 – O Problema da Fábrica de Camisas



Uma companhia produz dois tipos de camisas: manga longa e manga curta. Na companhia, o único ponto crítico é a mão-de-obra disponível.

A camisa de manga longa consome 50% a mais de mão de obra do que a de manga curta. Sabe-se também que se toda a produção fosse concentrada na disponibilização de camisas de manga curta a companhia poderia entregar 400 camisas de manga curta por dia.

O mercado limita a produção diária das camisas em 150 mangas longas e 300 mangas curtas. O lucro bruto por camisa de manga longa é de 5,00 u.m. e por camisa de manga curta, 3,5 u.m.

Formular o problema de modo a permitir a determinação das quantidades de camisas a produzir de modo a otimizar o lucro.

Solução

1. Escolha da variável de decisão

$x_i \equiv$ quantidade de unidades de camisas de manga longa ($i = 1$) e de manga curta ($i = 2$) a serem fabricadas.

2. Elaboração da função objetivo

$z = \text{Maximizar } \{f(x) = 5x_1 + 3,5x_2\}$

Número total de unidades monetárias obtidas como lucro pela venda das camisas.

A novidade introduzida pelo presente problema é a existência de um acoplamento de variáveis. Uma restrição tecnológica evidencia limites para cada uma das variáveis; contudo, outra trata a produção das duas variáveis concomitantemente.

3. Formulação das restrições tecnológicas

a) Restrição associada à demanda de mercado:

$$x_1 \leq 150$$

$$x_2 \leq 300$$

b) Restrição associada ao acoplamento das variáveis:

$$(3/2)x_1 + x_2 = 400$$

Observar que o relacionamento entre as variáveis é *linear*. O enunciado do problema fornece o termo independente da equação geral da reta, 400, através do ponto (0,400) em que a produção de camisas de manga longa é nula. O segundo dado necessário à determinação da equação vem da relação entre os coeficientes das variáveis: 1 e 1 + 0,5 ou 1 e 3/2.

4. Restrições de não negatividade

$$x_1 \geq 0, x_2 \geq 0$$

5 – O Problema da Otimização de Padrões de Produção



Uma determinada fábrica produz painéis de metal médios e grandes a partir de elementos circulares de diâmetros de 0,25 e 0,40 metro, respectivamente. A primeira operação para obter as painéis é um corte desses elementos circulares sobre chapas de dimensão de 1,40 × 0,50 metro. Os elementos planos circulares são transformados em painéis em uma segunda operação de estamparia. Para o corte existem quatro tipos de matrizes conforme mostra a Figura 2.7. A fábrica deseja uma produção diária mínima de 500 painéis médios (obtidos do elemento circular de diâmetro 0,25) e 350 grandes (obtidos do elemento circular de diâmetro de 0,40). Os custos em reais por chapa pelo uso de cada matriz de corte são respectivamente: 1, 2, 3, 2. Elaborar o modelo de Programação Linear que planeje a produção de modo a minimizar o custo com o uso das chapas.

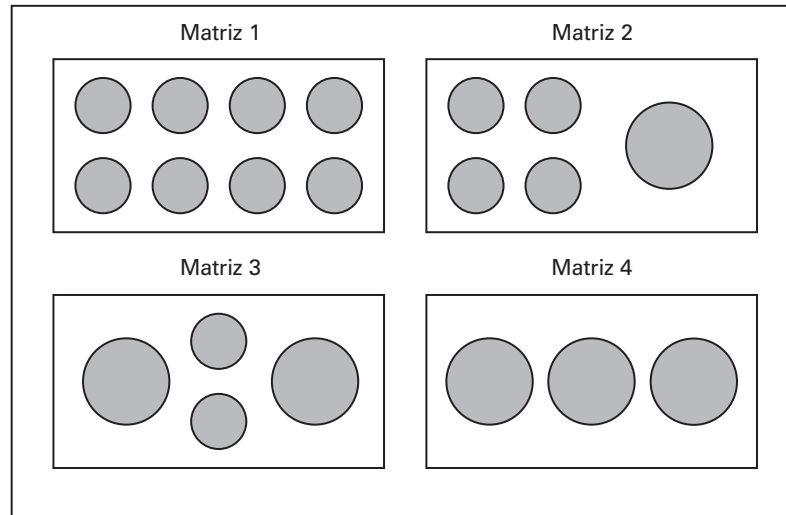


FIGURA 2.7 Padrões de corte do problema 5.

Solução

1. Escolha da Variável de Decisão

$x_i \equiv$ quantidade de chapas cortadas de acordo com a matriz, ($i = 1, 2, 3, 4$), a serem utilizadas na produção.

2. Elaboração da função objetivo

$z = \text{Minimizar } \{f(x) = x_1 + 2x_2 + 3x_3 + 2x_4\}$ – Número total de reais gastos com a produção.

3. Formulação das restrições tecnológicas

a) Restrição associada à demanda de painéis médios (elemento de 0,25 de diâmetro):

$$8x_1 + 4x_2 + 2x_3 \geq 500$$

b) Restrição associada à demanda de painéis grandes (elemento de 0,40 de diâmetro):

$$x_2 + 2x_3 + 3x_4 \geq 350$$

4. Restrições de não negatividade

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0.$$

2.2.4 – Exemplos de Modelagem de Problemas de PL Inteira sem Aproximação Contínua

1 – O Problema de Alocação de Pessoal



Um hospital trabalha com um atendimento variável em demanda durante as 24 horas do dia. As necessidades distribuem-se segundo a Tabela 2.13:

TABELA 2.13

<i>Turno de Trabalho</i>	<i>Horário</i>	<i>Número Mínimo de Enfermeiros</i>
1	08:00-12:00	50
2	12:00-16:00	60
3	16:00-20:00	50
4	20:00-00:00	40
5	00:00-04:00	30
6	04:00-08:00	20

O horário de trabalho de um enfermeiro é de oito horas quando ele entra nos turnos 1, 2, 3, 4 e 6. O enfermeiro que entra no turno 4 recebe uma gratificação de 50% sobre o salário e o enfermeiro que entra no turno 5 trabalha apenas quatro horas.

Elaborar o modelo de programação linear inteira que minimiza o gasto com a mão de obra.

Solução:

- Escolha da variável de decisão

$x_i \equiv$ número de enfermeiros que entram em serviço no início do turno i , ($i = 1, 2, \dots, 6$).

- Elaboração da função objetivo

Como praticamente todos os enfermeiros ganham o mesmo salário, poderemos considerar que minimizar o gasto com a mão de obra será equivalente a minimizar o número total de trabalhadores, ponderando-se o número de trabalhadores do quarto turno na medida de seu salário, então:

$z = \text{Minimizar } \{f(x) = x_1 + x_2 + x_3 + 1,5x_4 + 2x_5 + x_6\}$ número total de
enfermeiros
trabalhando em um dia.

O coeficiente de 1,5 para os enfermeiros que entram no quarto turno e 2 para os que ingressam no quinto igualam, em termos de salário, a contribuição desses turnos na despesa. Os trabalhadores do quinto turno ganham um salário completo por uma meia-jornada, ou seja, ganham o dobro dos enfermeiros dos turnos 1, 2, 3 e 6.

- Formulação das restrições tecnológicas

a) Restrição associada aos seis turnos:

✓ 1º turno:
 $x_6 + x_1 \geq 50$

✓ 2º turno:
 $x_1 + x_2 \geq 60$

✓ 3º turno:
 $x_2 + x_3 \geq 50$

✓ 4º turno:
 $x_3 + x_4 \geq 40$

✓ 5º turno:
 $x_4 + x_5 \geq 30$

✓ 6º turno:
 $x_6 \geq 20$

4. Condições de integralidade e não negatividade

$x_1, x_2 \in \mathbb{Z}^+$ (conjunto dos inteiros positivos).

2 – O Problema do Jantar de Nero



O imperador romano Nero, em um momento de inspiração, resolveu promover um jantar para eliminar seus “melhores” inimigos. Consultando seu médico de confiança, soube que ele dispunha de dois tipos de venenos, alfa e beta. Tratavam-se de fármacos próprios para serem misturados no molho de carneiro. Havia no estoque da farmácia do facultativo 0,5 kg do veneno alfa e 2 kg do veneno beta. Para que os convidados não sentissem o gosto do veneno, era indispensável misturar em peso três porções do veneno alfa para cada porção de beta. Cada 12 gramas de alfa ou 6 de beta eram capazes de sozinhas liquidarem um homem. O efeito do veneno sobre as mulheres era cerca de 50% mais poderoso do que sobre os homens. Nero, satisfeito com a informação, deu suas ordens ao médico: prepare a mistura mais eficiente e elimine pelo menos 20 homens e 10 mulheres!

Elaborar o modelo de programação matemática que maximize o efeito do veneno sobre os preza-dos inimigos do imperador e evite que o médico perca o emprego e acabe queimado vivo no dito jantar.

Solução:

1. Escolha da variável de decisão

$x_{ij} \equiv$ número de envenenados i , do sexo masculino ($i = 1$), do feminino ($i = 2$), pelo veneno j , veneno alfa ($j = A$), veneno beta ($j = B$).

2. Elaboração da função objetivo

$z = \text{Maximizar } \{f(x) = x_{1A} + x_{2A} + x_{1B} + x_{2B}\}$ número total de envenenados.

3. Formulação das restrições tecnológicas

a) Restrições de disponibilidade de veneno

✓ *Tipo alfa:*

$$12x_{1A} + 6x_{2A} \leq 500$$

✓ *Tipo beta:*

$$6x_{1B} + 3x_{2B} \leq 2.000$$

b) Restrição devida ao paladar

$$\frac{12x_{1A} + 6x_{2A}}{6x_{1B} + 3x_{2B}} = 3$$

c) Restrições de demanda

✓ *Homens envenenados:*

$$x_{1A} + x_{1B} \geq 20$$

✓ *Mulheres envenenadas*

$$x_{2A} + x_{2B} \geq 10$$

4. Condições de integralidade e não negatividade

$x_{1A}, x_{2A}, x_{1B}, x_{2B} \in \mathbb{Z}^+$ (conjunto dos inteiros positivos).

3 – O Problema da Otimização de Padrões de Corte



Uma metalúrgica deve entregar uma partida de 2.500 placas retangulares de 2×4 cm (placas do tipo I) e 1.000 de 4×11 cm (placas do tipo II). Existem, em estoque, uma tira metálica com 15 cm de largura e 20 metros de comprimento e outra com 14 centímetros de largura e 30 metros de comprimento. As tiras com 30 metros de comprimento são cerca de 20% mais caras por quilo do que as de 20 metros, devido a problemas de transporte. São possíveis as seguintes configurações de corte ou padrões de corte nos equipamentos da empresa:

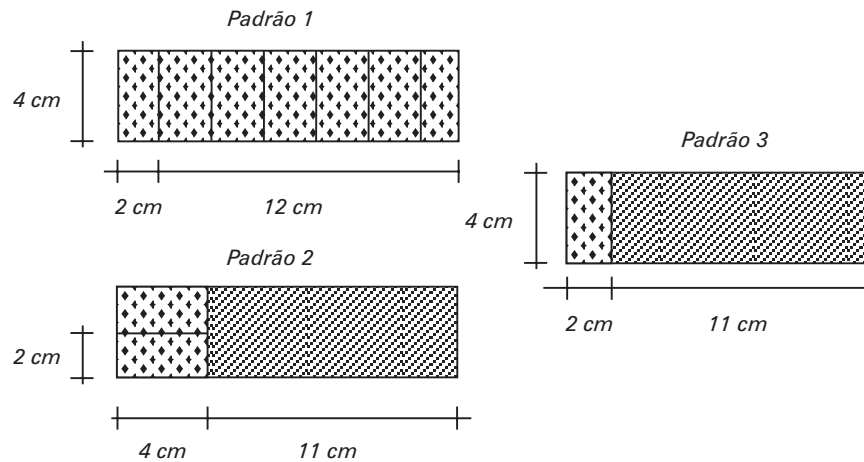


FIGURA 2.8 Padrões de corte.

Formular o modelo que permita minimizar a perda ao corte.¹

Solução:

- Escolha da variável de decisão

$x_{ij} \equiv$ número de vezes em que o padrão i , $i = 1, 2, 3$ de corte será aplicado à chapa j de 15×2.000 ($j = A$) e de 14×3.000 ($j = B$).

$c_{ij} \equiv$ perdas da configuração i aplicada na chapa j .

- Formulação das restrições tecnológicas

a) Restrições de demanda:

✓ Placas 2×4 :

$$7x_{1A} + 7x_{1B} + 2x_{2A} + x_{3A} + x_{3B} \geq 2.500$$

✓ Placas 4×11 :

$$x_{2A} + x_{3A} + x_{2B} + x_{3B} \geq 1.000$$

1. Interessantes trabalhos sobre o tema de perdas ao corte são encontrados em Hinxman (1980), Farley (1988) e (1990), Álvares-Valdi *et al.* (2002), Morabito e Arenales (2000).

b) Restrição de disponibilidade de estoque:

✓ Chapa 15 × 2.000:

$$4x_{1A} + 4x_{2A} + 4x_{3A} \leq 2.000$$

✓ Chapa 14 × 3.000:

$$4x_{1B} + 4x_{3B} \leq 3.000$$

3. Elaboração da função objetivo

$z = \text{Minimizar } \{f(x) = x_{1A} + x_{3A} + 1,2 x_{3B}\}$ perda total.

4. Condições de integralidade e não negatividade

$x_{1A}, x_{2A}, x_{3A}, x_{1B}, x_{2B}, x_{3B} \in \mathbb{Z}^+$ (conjunto dos inteiros positivos)

4 – O Problema da Câmara de Segurança



Um laboratório biológico está construindo um sala de desinfecção e segurança para proteger a entrada de suas instalações mais perigosas. A finalidade da sala é isolar agentes biológicos como vírus e bactérias que são manipulados em instalações hermeticamente fechadas, permitindo, contudo, o trânsito dos pesquisadores e demais materiais. A sala é composta de três câmaras visando a aumentar a segurança (ver Figura 2.9).

A desinfecção é basicamente realizada através de uma sofisticada unidade de irradiação. Como essa unidade é extremamente sensível e o objeto em trânsito pode ter um comportamento inesperado, existe uma probabilidade de falha da atuação de uma unidade isolada de desinfecção. Essa probabilidade varia basicamente em função da câmara em que a unidade esteja localizada. Existem restrições de espaço e peso para a distribuição das unidades de desinfecção no teto das câmaras. Os custos das unidades também variam por câmara devido à variação de temperatura e isolamento de umidade, tudo conforme a Tabela 2.14, a seguir.

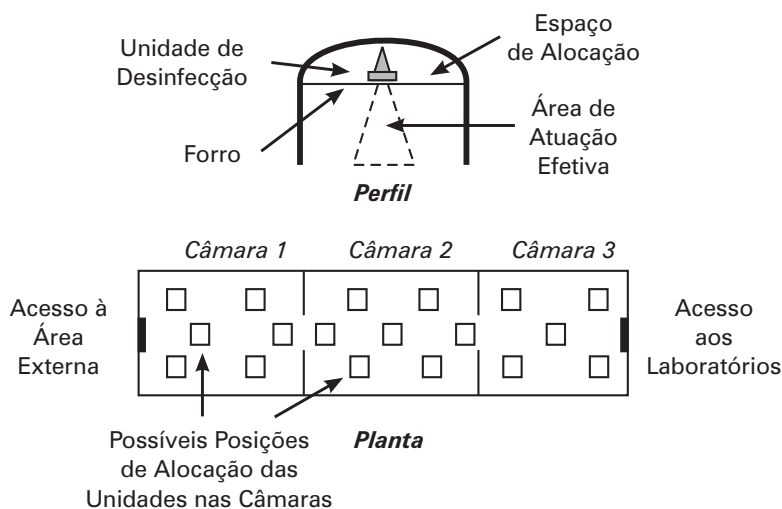


FIGURA 2.9 Planta e perfil da câmara.

TABELA 2.14

<i>Câmara</i>	<i>Espaço (m³)</i>	<i>Peso (kg)</i>	<i>Custo (R\$)</i>	<i>Probabilidade de Falha da Unidade</i>
1	2,5	150	30.000	0,20
2	4,0	130	70.000	0,15
3	3,0	100	40.000	0,30
Limitações	60	1.500	600.000	

Formular o problema de minimizar a probabilidade da ocorrência de um trânsito na sala de desinfecção sem que ocorra uma perfeita desinfecção, considerando-se inaceitável mais que 3% na probabilidade de que uma câmara isolada falhe na desinfecção.

1. Escolha da variável de decisão

$x_j \equiv$ número de unidades de desinfecção alocadas na câmara j , ($j = 1, 2, 3$).

$p_j^{x_j} \equiv$ probabilidade de falha ($j = 1, 2, 3$).

2. Formulação das restrições tecnológicas

a) Restrições de espaço:

$$2,5x_1 + 4x_2 + 3x_3 \leq 60$$

b) Restrição de peso:

$$150x_1 + 130x_2 + 100x_3 \leq 1.500$$

c) Restrição de custo:

$$30x_1 + 70x_2 + 40x_3 \leq 600$$

d) Restrição de falha de uma câmara:

$$\text{com } p_1 = 0,2; p_2 = 0,15; p_3 = 0,3$$

$$p_j^{x_j} \leq 0,03, j = 1, 2, 3$$

3. Elaboração da Função Objetivo

$$z = \text{Minimizar } \{f(x) = p_1^{x_1} p_2^{x_2} p_3^{x_3}\} - \text{probabilidade de falha total.}$$

4. Condições de integralidade e não negatividade

$$x_1, x_2, x_3 \in \mathbb{Z}^+ \text{ (conjunto dos inteiros positivos).}$$

5. Transformações de linearização

$$p_j^{x_j} \leq 0,03, j = 1, 2, 3 \rightarrow (\ln p_j)x_j \leq \log(0,03)$$

$$f(x) = p_1^{x_1} p_2^{x_2} p_3^{x_3} \rightarrow \ln(f(x)) = \sum_{j=1}^3 (\ln p_j)x_j$$

5 – O Problema de Carregamento de Aviões



Um avião de transporte possui quatro compartimentos para carga a saber: compartimento frontal, compartimento central, compartimento da cauda e porão de granel. Os três primeiros compartimentos só podem receber carga em *containers*, enquanto o porão recebe material em granel. A Tabela 2.15 resume a capacidade do aparelho:

TABELA 2.15

<i>Compartimento</i>	<i>Peso Máximo (ton)</i>	<i>Espaço Máximo (m³)</i>
<i>Compartimento Frontal</i>	5	35
<i>Compartimento Central</i>	7	55
<i>Compartimento da Cauda</i>	6	30
<i>Porão de Granel</i>	7	30

Objetivando o equilíbrio de voo, é indispensável que a distribuição da carga seja proporcional entre os compartimentos. Para carregar o avião, existem três tipos de *containers* e duas cargas em granel. Os dois tipos de carga em granel podem ser facilmente transportados conjuntamente, por isso essa carga é aceita em qualquer quantidade.

TABELA 2.16

<i>Carga Tipo</i>	<i>Peso por Container ou por m³ – (ton)</i>	<i>Volume por Container (m³)</i>	<i>Lucro \$/ton</i>
<i>1 (container)</i>	0,7	0,5	200
<i>2 (container)</i>	0,9	1	220
<i>3 (container)</i>	0,2	0,25	175
<i>4 (granel)</i>	1,2/m³	—	235
<i>5 (granel)</i>	1,7/m³	—	180

Elaborar o problema de programação linear que otimize a distribuição da carga de forma a maximizar o lucro do voo do cargueiro.

1. Escolha da variável de decisão

x_{ij} \equiv quantidade da carga do tipo i ($i = 1, 2, 3$ – números de *containers*, $i = 4, 5$ – toneladas do granel i), alocada ao compartimento j ($j = 1$ – frente, $j = 2$ – centro, $j = 3$ – cauda, $j = 4$ – porão).

2. Formulação das restrições tecnológicas

a) Restrições de peso por compartimento:

$$\sum_{i=1}^3 x_{ij} \leq \begin{cases} 5, & j=1 \\ 7, & j=2 \\ 6, & j=3 \end{cases}$$

$$1,2x_{4j} + 1,7x_{5j} \leq 7, j=4$$

b) Restrições de espaço:

$$0,5x_{1j} + x_{2j} + 0,25x_{3j} \leq \begin{cases} 35, & j=1 \\ 55, & j=2 \\ 30, & j=3 \end{cases}$$

$$x_{4j} + x_{5j} \leq 30, j=4,*$$

c) Restrições de equilíbrio geral

$$\frac{\sum_{i=1}^3 x_{i1}}{5} = \frac{\sum_{i=1}^3 x_{i2}}{7} = \frac{\sum_{i=1}^3 x_{i3}}{6} = \frac{1,2x_{44} + 1,7x_{54}}{7}$$

Cabe aqui observar que essa restrição deverá, provavelmente, possuir limites de tolerância, uma vez que as variáveis envolvidas no somatório são inteiras, o que tornaria, possivelmente, inviável o atendimento simultâneo desse conjunto de restrições. Essa tolerância é também perfeitamente justificada na visão prática do carregamento de aeronaves. Nesse caso uma das possibilidades de representar a situação seria como:

$$\frac{\sum_{i=1}^3 x_{i1}}{5} + x_{f1} = \frac{\sum_{i=1}^3 x_{i2}}{7} + x_{f2} = \frac{\sum_{i=1}^3 x_{i3}}{6} + x_{f3} = \frac{1,2x_{44} + 1,7x_{54}}{7} + x_{f4}$$

para adequadas variáveis de tolerância $x_{fi} \in \mathbb{R}, i = \{1, \dots, 4\}$

3. Elaboração da função objetivo

$z = \text{Maximizar}$

$$\{f(x) = 200 \sum_{j=1}^3 x_{1j} + 220 \sum_{j=1}^3 x_{2j} + 175 \sum_{j=1}^3 x_{3j} + 1,2 \times 235x_{44} + 1,7 \times 180x_{54}\}$$

4. Condições de integralidade e não negatividade

$$x_{1j}, x_{2j}, x_{3j} \in \mathbb{Z}^+, j = 1, 2, 3; x_{4j}, x_{5j} \leq 0, j = 4.$$

6 – O Problema da Auditoria Bancária



Fase 1: Processo normal de auditoria

Um banco deve decidir quantos auditores será necessário contratar em um horizonte de seis meses de operação a saber: março – agosto. As necessidades do esforço de auditoria são contabilizadas em termos de mão de obra de auditores experientes da seguinte forma:

*Os coeficientes volumétricos dos granéis são considerados iguais a 1 por segurança.

TABELA 2.17

<i>Mês</i>	<i>Necessidade em Homens x Hora</i>
<i>Março</i>	7.000
<i>Abril</i>	8.000
<i>Maio</i>	10.000
<i>Junho</i>	11.000
<i>Julho</i>	7.000
<i>Agosto</i>	11.000

Cada auditor contratado como funcionário do banco, apesar de formado e aprovado em concurso, tem de ser treinado por um mês antes de poder atuar plenamente em sua função. Nesse treinamento são utilizados auditores experientes do próprio banco que, deixando de trabalhar na auditoria normal, dedicam 100 horas para cada auditor a ser treinado. Um auditor trabalha 150 horas por mês. Em 1º de fevereiro o banco dispõe de 60 auditores experientes. O programa de contratação terá início em 1º de março.

Sabe-se também que o mercado de trabalho para os auditores está muito instável, de forma que 10% da força de trabalho desses profissionais experientes deixa o banco a cada mês em busca de melhores salários. Um auditor experiente recebe do banco cerca de R\$ 2.000,00 por mês, enquanto o auditor em treinamento só recebe uma ajuda de custo de R\$ 150,00. Quando o número de auditores excede as necessidades, a carga de trabalho é reduzida, mas não são feitas demissões devido ao elevado custo do processo de acordo na justiça. Quando isso acontece, novos auditores não são contratados e a evasão normal equilibra a força de trabalho.

Formular o problema objetivando minimizar os custos de operação do sistema de auditoria.

Fase 2: Utilizando o processo de terceirização

Paralelamente ao sistema de contratação formal para auditores existe a possibilidade de obter-se a mão de obra para as auditorias via uma empresa de terceirização: a Burocráticos ao seu Dispor. Essa organização oferece auditores experientes (possivelmente evadidos do sistema normal) e licenciados pela Câmara de Auditores Juramentados. Esse profissional custa R\$ 2.500,00 ao mês e pode ser incluído e retirado da folha a qualquer mês sem qualquer custo de admissão ou demissão. A Burocráticos só exige a garantia mínima de um mês de trabalho para o profissional e que ele não trabalhe em treinamentos, até porque a empresa promove o curso para licenciar auditores juramentados como um serviço adicional. Reformular o problema levando em conta essa nova possibilidade.

Desafio: Reformule o problema para o caso em que a empresa Burocráticos passe a exigir uma garantia mínima de três meses de emprego para seus auditores e uma indenização proporcional a K vezes o número de meses trabalhado além desses três meses.

Solução

Fase 1

1. Escolha da variável de decisão

$x_j \equiv$ quantidade de auditores em treinamento no mês j , ($j = 0, \dots, 6$).

i será utilizado como um índice auxiliar, $i = 1, \dots, 6$.

$d_i \equiv$ vetor demanda = (7.000, 8.000, 10.000, 11.000, 7.000, 11.000)

2. Formulação das restrições tecnológicas

a) Restrições de demanda:

$$x_0 = 60$$

$$150 \left(\sum_{j=0}^{i-1} 0,9^{i-j} x_j \right) - 100x_i \geq d_i, \quad i = 1, \dots, 6$$

O número de auditores experientes que inicializa o processo de planejamento pode ser considerado, em termos de modelagem, como treinados em fevereiro. Um auditor treinado, ou seja, experiente, incorpora-se ao contingente de evasão ao fim do mês. Em últimas palavras 10% dos profissionais treinados evade-se no primeiro mês comportando-se, para fins de força de trabalho, como auditores experientes. É necessário, contudo, lembrar que eles só recebem um ajuda de custo nesse primeiro mês.

b) Restrição de capacidade de treinamento

Considerando a restrição a) que estabelece as condições iniciais para o número de auditores seniores, então:

$$\sum_{j=0}^{i-1} 0,9^{i-j} x_j \geq 15x_i, \quad i = 1, \dots, 6$$

3. Configuração da função objetivo

$$z = \text{Minimizar } \{f(x) = \sum_{i=1}^6 \left(2000 \left(\sum_{j=0}^{i-1} 0,9^{i-j} x_j \right) + 150x_i \right)\}$$

4. Condições de integralidade e não negatividade

$$x_j \in \mathbb{Z}^+, j = 1, \dots, 6.$$

Fase 2

1. Escolha da variável de decisão

As mesmas anteriores e mais:

$y_i \equiv$ número de auditores juramentados contratados para trabalhar no mês $i, i = 1, \dots, 6$

2. Formulação das restrições tecnológicas

a) Restrições de demanda:

$$x_0 = 60$$

$$150 \left(\sum_{j=0}^{i-1} 0,9^{i-j} x_j + y_i \right) - 100x_i \geq d_i, \quad i = 1, \dots, 6$$

b) Restrição de capacidade de treinamento:

A mesma da primeira fase.

3. Elaboração da função objetivo

$$z = \text{Minimizar } \left\{ f(x) = \sum_{i=1}^6 \left(2.000 \left(\sum_{j=0}^{i-1} 0,9^{i-j} x_j \right) + 150x_i + 2.500y_i \right) \right\}$$

4. Condições de integralidade e não negatividade

$$x_j \in \mathbb{Z}^+, j = 1, \dots, 6; y_i \in \mathbb{Z}^+, i = 1, \dots, 6.$$

7 – O Problema do Incêndio Florestal – Ataque Massivo



Fase 1: Ataque massivo

Uma reserva florestal está em chamas e o governo planeja uma operação fulminante de combate ao fogo para amanhã. O incêndio é de pequenas proporções e está se propagando lentamente, devendo ser extinto em cerca de três horas de operação logo após o amanhecer. Estão sendo mobilizados aviões e helicópteros especializados nesse tipo de operações. As características dos aparelhos constam da Tabela 2.18:

TABELA 2.18

Aparelho	Eficiência no Incêndio (m ² /Hora)	Custo (R\$/Hora)	Necessidade em Pessoal
Helicóptero AH-1	15.000	2.000	2 Pilotos
Avião Tanque	40.000	4.000	2 Pilotos – 1 Operador
Avião B67	85.000	10.000	2 Pilotos – 3 Operadores

A área de floresta a ser coberta pelo combate ao fogo é de 3.000.000 m², envolvendo a frente de fogo (para paralisação do avanço do dano), áreas já queimadas que necessitam de rescaldo (para proteção de animais e segurança contra recrudescimento) e áreas de acero (proteção preventiva indispensável). Nas bases de apoio são disponíveis 14 pilotos de avião e 10 de helicóptero, bem como 22 operadores especializados em combate aéreo de fogo.

Formular o problema de programação matemática que minimize os custos da operação.

Solução:

1. Escolha da variável de decisão

$x_i \equiv$ número de aparelhos a serem empregados i ($i = 1$ – Helicóptero; $i = 2$ – Avião tanque; $i = 3$ – Avião B67).

2. Formulação das restrições tecnológicas

a) Restrição de demanda da área de fogo:

Como a operação deve ser concluída em três horas, podemos imaginar um horizonte de planejamento ótimo para uma hora sendo repetido três vezes, então:

$$15x_1 + 40x_2 + 85x_3 \geq 1.000$$

b) Restrição de disponibilidade de mão de obra

✓ *Helicóptero:*

$$2x_1 \leq 10 \text{ (pilotos)}$$

✓ *Aviões:*

$$2x_2 + 2x_3 \leq 14 \text{ (pilotos)}$$

$$x_2 + 3x_3 \leq 22 \text{ (operadores)}$$

3. Elaboração da função objetivo

$$z = \text{Minimizar } \{f(x) = 2.000x_1 + 4.000x_2 + 10.000x_3\}$$

4. Condições de integralidade e não negatividade

$$x_j \in \mathbb{Z}^+, j = 1, 2, 3.$$

8 – O Problema do Sistema de Defesa Antiaérea



Fase 1: Alocação arma × alvo

Um determinado conjunto de armas antiaéreas está distribuído de forma a defender uma cidade de um ataque. São n plataformas de mísseis. Sabe-se que d_{ij} é a distância entre a plataforma da arma i e a ameaça j (avião inimigo ou míssil), que o alcance máximo dos mísseis é de r_i , que o custo de cada tiro sobre uma ameaça j é de c_{ij} e o valor da neutralização da ameaça é v_j . Em cada ataque, o sistema de defesa deve selecionar, dentre m ameaças, apenas k possíveis alvos.

Elaborar o modelo matemático de alocação arma × alvo que minimiza o custo da defesa.

Fase 2: Consideração da eficácia

Sabe-se que a cada alocação arma × alvo devam ser disparados p_{ij} mísseis para que o alvo seja neutralizado. Sabendo-se que existem m_i mísseis disponíveis em cada plataforma, reformular o problema de minimizar o custo da defesa.

Solução:

Alocação arma × alvo

1. Escolha da variável de decisão

$$x_{ij} = \begin{cases} 1, & \text{caso a arma } i \text{ seja designada à ameaça } j \\ 0, & \text{caso contrário} \end{cases} \quad i = 1, \dots, n ; j = 1, \dots, m$$

2. Formulação das restrições tecnológicas

a) Restrição de alcance, visando a não permitir que uma arma atire em alvo fora do seu alcance:

$$(r_i - d_{ij}) x_{ij} \geq 0 \quad i = 1, \dots, n \quad j = 1, \dots, k, \dots, m$$

b) Restrição do número de alocações de alvos, visando que o sistema global atue sobre k alvos.

$$\sum_{i=1}^n \sum_{j=1}^m x_{ij} = k$$

c) Restrição de disparo que evita que um alvo seja designado a mais de uma arma.

$$\sum_{i=1}^n x_{ij} \leq 1, j = 1, \dots, m$$

d) Restrição de disparo que evita que uma arma seja designada a mais de um alvo.

$$\sum_{j=1}^m x_{ij} \leq 1, i = 1, \dots, n$$

3. Elaboração da função objetivo

$$z = \text{Maximizar } \{f(x) = \sum_{j=1}^m (v_j \sum_{i=1}^n x_{ij} - \sum_{i=1}^n c_{ij} x_{ij})\}$$

Consideração da eficácia

1. Escolha da variável de decisão

$$x_{ij} = \begin{cases} 1, & \text{caso a arma } i \text{ seja designada a ameaça } j \\ 0, & \text{caso contrário} \end{cases} \quad i = 1, \dots, n; \quad j = 1, \dots, m$$

2. Formulação das restrições tecnológicas

a) As restrições de alcance, alocação aos k alvos e uma só plataforma designada a um alvo e vice-versa permanecem as mesmas.

b) Restrição de número de disparos.

No modelo anterior a alocação arma \times alvo e o disparo constituíam uma mesma decisão. No presente caso a alocação implica p_{ij} disparos. Para uma alocação ser válida agora é necessário que $m_i \geq p_{ij}, j = 1, \dots, m$, então:

$$\sum_{j=1}^m x_{ij} p_{ij} \leq m_i \quad i = 1, \dots, n$$

3. Elaboração da função objetivo

$$z = \text{Maximizar } \{f(x) = \sum_{j=1}^m (v_j \sum_{i=1}^n x_{ij} - \sum_{i=1}^n c_{ij} p_{ij} x_{ij})\}$$

9 – O Problema dos Anúncios na Rede de TV



Uma rede de televisão resolveu estabelecer preços competitivos para o tempo de comercial em certos horários. Existem três horários para a propaganda em promoção na rede: horário nobre noturno (horário 1), horário da tarde em fins de semana (horário 2), horário da tarde em dias da semana (horário 3). O preço de um módulo mínimo de propaganda em cada horário é p_1 , p_2 e p_3 , respectivamente. A rede vende grandes espaços de tempo dentro da programação, denominados pacotes promocionais. Os anunciantes desejam entrar no programa de pacotes de propaganda promocional porque podem obter, com isso, melhores preços unitários. Um pacote representa um esquema de desconto baseado em economia de escala. Existem três faixas de preços sendo praticadas nos pacotes. A faixa 1, ou de desconto zero, a faixa 2, em que o desconto é de 10 unidades monetárias e a faixa 3, com desconto de 30 unidades monetárias. A tabela que se segue resume as regras da promoção:

TABELA 2.19 REGRAS DE PROMOÇÃO

<i>Descontos</i>	<i>Somatório de tempo</i>	<i>Valor do desconto (em unidades monetárias)</i>
Faixa 1	sem exigência	0
Faixa 2	comprar, no mínimo, f unidades de tempo	10
Faixa 3	comprar, no mínimo, g unidades de tempo	30

Os anunciantes que competem na disputa do uso da mídia são n e dispõem, cada um deles, de s_n unidades monetárias para investir em propaganda.

Os anunciantes, reunidos em uma associação, desejam estabelecer a melhor estratégia de negociação com a rede, objetivando maximizar, dentro das disponibilidades orçamentárias de cada cliente, o tempo global de utilização da mídia.

Formular o modelo que maximiza o ganho dos anunciantes.

Solução:

Sob o ponto de vista dos anunciantes, o desejável é maximizar o desconto, atendidas as restrições próprias de cada anunciante.

1. Escolha da variável de decisão

$t_i \equiv$ tempo disponível para propaganda no horário i , $i = 1, 2, 3$.

$x_{ik} \equiv$ tempo adquirido no horário i , $i = 1, 2, 3$, pelo anunciante k , $k = 1, \dots, n$.

$$y_k \equiv \begin{cases} 1, & \text{caso o anunciante } k \text{ tenha adquirido mais do que } f \text{ módulos de tempo} \\ 0, & \text{caso contrário} \end{cases}$$

$$w_k \equiv \begin{cases} 1, & \text{caso o anunciante } k \text{ tenha adquirido mais do que } g \text{ módulos de tempo} \\ 0, & \text{caso contrário} \end{cases}$$

$s_k \equiv$ montante máximo de recursos disponibilizado pelo anunciante k

$p_i \equiv$ preço do tempo de anúncio no horário i .

$c_k \equiv$ custo total que será pago pelo anunciante k , face ao total de desconto que lhe cabe.

2. Formulação das restrições tecnológicas

a) Restrições relativas ao montante pago pelo anunciante, face aos possíveis descontos

$$pg_k = \sum_{i=1}^3 x_{ik} p_i \quad k = 1, \dots, n$$

$$c_k = pg_k - 10y_k - 20w_k$$

$$s_k \geq c_k \quad k = 1, \dots, n$$

b) Restrição de tempo máximo disponível nos horários

$$\sum_{k=1}^n x_{ik} \leq t_i \quad i = 1, 2, 3$$

3. Elaboração da função objetivo

$$z = \text{Maximizar } \{f(x) = \sum_{i=1}^3 \sum_{k=1}^n x_{ik}\}$$

4. Condições de não negatividade

$$x_{ik} \geq 0, i = 1, 2, 3, \quad k = 1, \dots, n.$$

10 – O Problema das Equipes de Natação



Problema 1: A constituição de um grupo para uma competição

Um clube de natação, o Flamingo, foi desafiado pelo Botabolo em uma competição de revezamento quatro estilos em 200 m. Existem 100 nadadores treinando no clube. O departamento técnico possui a ficha de cada nadador onde constam os tempos nos estilos peito (P), costas (C), borboleta (B) e livre (L), para 200 metros. O treinador deseja selecionar, dentre os nadadores em treinamento, a melhor equipe possível. Formular o problema de programação matemática que auxilie o técnico em sua tarefa.

Problema 2: A constituição de um time de treinamento

Infelizmente, o Flamingo acabou derrotado no desafio com seu arqui-inimigo na natação por um acontecimento imprevisto: um de seus nadadores, o melhor no estilo borboleta, sentiu, na beira da piscina, uma pequena distensão no braço que passou a dificultar os movimentos para aquela especialidade. Como ele era um nadador que tinha excelente tempo no nado livre, o treinador tentou reorganizar na hora a designação dos estilos. O problema é que sua equipe era de especialistas. Os demais possuíam péssimos tempos fora de sua especialidade. A melhor troca possível foi feita, mas a equipe acabou derrotada pelo baixo tempo no nado borboleta. O treinador, revoltado pelo ocorrido, resolveu tomar duas providências:

a) Constituir um time de elite para treinamento com 4 equipes de 4 nadadores, evitando com isso problemas de saúde de última hora.

b) Identificar nadadores polivalentes.

Formular o problema de programação matemática que auxilie o técnico em sua tarefa.

Problema 3: A seleção de uma equipe de competição

O Flamingo não podia deixar sem uma resposta a derrota no torneio. Treinando a equipe de elite o técnico do clube sentiu-se seguro em devolver o desafio. Só que dessa vez as regras seriam diferentes: a equipe teria um reserva polivalente. Formular o problema de ajudar o Flamingo a escolher a melhor equipe para uma competição, considerando que o grupo inclui um nadador de reserva. Justifique os critérios que foram adotados.

Problema 4: A competição final (desafio)

Na prova de revezamento a equipe dos pênaltos (Flamingo) venceu o Botabolo. Dessa vez foi um dos nadadores do Botabolo que afundou na piscina. Os bolenses foram à loucura. A diretoria, irritada, culpou a escalação das equipes e ao seu treinador que “nada entendia de reservas”. Para tirar a dúvida sobre quais dos dois clubes seria o rei das piscinas o Botabolo desafiou dessa vez o Flamingo para uma grande gincana de natação. Um tudo ou nada. A ideia seria promover uma competição entre três equipes dos dois clubes. A pontuação seria feita por equipe do primeiro ao último lugar, de modo a eliminar a possibilidade de algum acidente ocasional e acabar com a história de reservas (que havia “claramente” beneficiado o Flamingo). Segundo as palavras do diretor do Botabolo: “Assim nós vamos ver quem é o bom dentro d’água... vamos acabar com a loteria dos torneios isolados e os truques dos técnicos.” A pontuação do primeiro ao último lugar para as seis equipes seria respectivamente: 8, 6, 4, 3, 2, 1. Humilhados, os bolenses resolveram jogar duro: conseguiram as fichas com os tempos dos pênaltos e contrataram você para formular o problema de escolher as suas três equipes de modo a maximizar as possibilidades de resultado positivo e afogarem os pênaltos na piscina com sete palmos de água em cima. Existem 100 nadadores treinando no Botabolo e seus melhores tempos estão igualmente disponíveis.

Solução:

Constituição de uma equipe de competição

1. Escolha da variável de decisão

$$x_{ij} \equiv \begin{cases} 1, & \text{caso o nadador } i \text{ seja designado a nadar o estilo } j \\ 0, & \text{caso contrário} \end{cases} \quad i = 1, \dots, 100; j \in \{P, C, B, L\}$$

$t_{ij} \equiv$ tempo do nadador i no estilo j (para 200 m).

2. Formulação das restrições tecnológicas

- a) Restrição que garante que todos os estilos terão um nadador alocado:

$$\sum_{i=1}^{100} x_{ij} = 1 \quad j \in \{P, C, B, L\}$$

- b) Restrição que garante que um nadador não será alocado a mais de um estilo:

$$\sum_{j=P}^L x_{ij} \leq 1, \quad i = 1, \dots, 100$$

3. Elaboração da função objetivo

$$z = \text{Minimizar } \{f(x) = \sum_{i=1}^{100} \sum_{j=P}^L t_{ij} x_{ij}\}$$

4. Condições de integralidade e não negatividade

$$x_{ij}^k \in \{0, 1\}$$

Constituição do time de treinamento com 16 nadadores polivalentes

1. Escolha da variável de decisão

$$x_{ij}^k = \begin{cases} 1, & \text{caso o nadador } i \text{ seja designado a nadar o estilo } j \\ & \text{na equipe } k \\ 0, & \text{caso contrário} \end{cases} \quad \left. \begin{matrix} i = 1, \dots, 100; j \in \{P, C, B, L\}; \\ k = 1, 2, 3, 4 \end{matrix} \right\}$$

$t_{ij} \equiv$ tempo do nadador i no estilo j (para 200 m).

2. Formulação das restrições tecnológicas

a) Restrição que garante que todos os estilos em cada equipe terão um nadador alocado:

$$\sum_{i=1}^{100} x_{ij}^k = 1 \quad j \in \{P, C, B, L\}; \quad k = 1, 2, 3, 4$$

b) Restrição que garante que um nadador não será alocado a mais de um estilo:

$$\sum_{j=P}^L x_{ij}^k \leq 1, \quad i = 1, \dots, 100; \quad k = 1, 2, 3, 4$$

3. Configuração da função objetivo das equipes

$$z = \text{Minimizar } \{f(x) = \sum_{k=1}^4 \sum_{i=1}^{100} \sum_{j=P}^L (t_{ij} x_{ij}^k)\}$$

4. Condições de integralidade e não negatividade (ressaltando a condição)

$$x_{ij}^k \in \{0, 1\}$$

5. Identificação dos nadadores polivalentes

Relaxar a restrição 2 b), permitindo que um nadador seja alocado a mais de um estilo na relação dos 16 nadadores de elite. Os polivalentes serão aqueles com múltiplas alocações.

Solução:**Constituição do time de competição 4×200 com um reserva**

Escolhida a equipe de competição segundo os critérios do problema 1, o reserva, um novo componente do grupo, poderia ser escolhido pelos critérios de polivalência do problema 2 ou minimizando o aumento de tempo da equipe pelo afastamento de um de seus componentes, como abaixo:

1. Escolha da variável de decisão

$$x_i \equiv \begin{cases} 1, & \text{caso o nadador } i \text{ seja designado como reserva} \\ 0, & \text{caso contrário} \end{cases} \quad i = 1, \dots, 96$$

t_{ij} , tempo do nadador i no estilo j (para 200 m).

Considerando o índice dos nadadores selecionados pelo critério dos especialistas como: $\alpha, \beta, \phi, \theta$

2. Formulação das restrições tecnológicas

a) Restrição que garante que um nadador é indicado como reserva:

$$\sum_{i=1}^{96} x_i = 1$$

3. Configuração da função objetivo que minimiza a redução do tempo pela substituição de um dos nadadores.

$z = \text{Minimizar}$

$$\{f(x) = \left(\sum_{i=1}^{96} t_{iP} x_i - t_{\alpha P} \right) + \left(\sum_{i=1}^{96} t_{iC} x_i - t_{\beta C} \right) + \left(\sum_{i=1}^{96} t_{iB} x_i - t_{\phi B} \right) + \left(\sum_{i=1}^{96} t_{iL} x_i - t_{\theta L} \right)\}$$

11 – O Problema da Expedição de Pesquisa a Marte

Após criteriosos estudos, um determinado país decidiu enviar uma expedição de pesquisa ao planeta Marte. Escolher uma tripulação adequada ao esforço da jornada e trabalho no planeta é uma das maiores dificuldades do comando da missão. O centro de treinamento, possuindo mais de 300 candidatos qualificados, resolveu utilizar um modelo matemático para auxiliar essa tarefa. Para adquirir confiança no modelo, o comando da missão achou por bem simulá-lo em um pequeno conjunto de candidatos altamente qualificados e, nesse conjunto menor, estudar as vantagens do modelo. A Tabela 2.20 resume os atributos do grupo de 11 pessoas selecionadas para testar o modelo:

TABELA 2.20

	<i>Homens</i>						<i>Mulheres</i>				
<i>Critério</i>	<i>Fábio</i>	<i>Gallo</i>	<i>Saulo</i>	<i>Paulo</i>	<i>José</i>	<i>Bruno</i>	<i>Maria</i>	<i>Ana</i>	<i>Linda</i>	<i>Dina</i>	<i>Carla</i>
<i>Peso</i>	80	85	70	65	90	75	50	45	55	60	65
<i>Saúde</i>	150	150	95	140	90	105	145	145	105	140	95

TABELA 2.20 CONTINUAÇÃO

	<i>Homens</i>						<i>Mulheres</i>				
<i>Critério</i>	<i>Fábio</i>	<i>Gallo</i>	<i>Saulo</i>	<i>Paulo</i>	<i>José</i>	<i>Bruno</i>	<i>Maria</i>	<i>Ana</i>	<i>Linda</i>	<i>Dina</i>	<i>Carla</i>
<i>Áreas de Habilitação</i>	E C I	M F C G	F B A	E A	C I A	A C	F B M I	F E G	M E I	P C I	P B M
<i>Treinamento no Simulador (h)</i>	150	200	180	240	220	100	200	250	200	180	100
<i>Incompat.</i>	José	–	–	–	Fábio	Linda	–	–	Bruno	Ana	–
<i>Consumo Energético (kcal)</i>	20	29	30	35	23	40	25	30	25	17	15
<i>Bagagem (kg)</i>	35	25	30	40	17	40	30	40	30	20	30

Siglas das especialidades: medicina (M), engenharia (E), geologia (G), psicologia (P), física (F), astronomia (A), comando (C), informática (I) e biologia (B).

As exigências do comando da missão podem ser resumidas, inicialmente, como:

A nave conduzirá 5 pessoas sendo pelo menos 2 homens e 1 mulher. Existem a bordo 2 vagas para pessoas habilitadas em comando (C) que deverão ser obrigatoriamente ocupadas. Outras habilidades são indispensáveis ao cumprimento da missão em Marte e na viagem. Para tal são ainda necessários a bordo pelo menos:

- 2 pessoas habilitadas em medicina (M).
- 1 pessoa habilitada em psicologia (P).
- 1 pessoa em astronomia (A).
- 1 pessoa em informática (I).

Para os engenheiros do comando da missão o fator primordial da otimização é o peso. Existe o peso fixo, independentemente da tripulação, e o peso variável. A otimização dessa etapa deve se concentrar sobre o peso da tripulação e de sua bagagem. Os custos estimados são de R\$10.000,00 por kg de peso da tripulação ou bagagem. O comando da missão entende por bagagem os elementos necessários ao exercício das habilitações de cada tripulante. Se o peso total das pessoas e bagagens exceder 1.000 kg, a missão necessitará de um módulo extra de acondicionamento ao custo de cerca de R\$10 milhões.

Existem também severas limitações para o uso do simulador de voo, um equipamento muito caro e que não possui capacidade para ser ampliado. Até o lançamento, serão disponíveis, de fato, 1.000 horas para treinamento. A tabela resume o tempo ainda necessário para treinar cada componente do grupo em análise.

Conforme estudos clínicos de contágio e progressão epidemiológica em populações isoladas, para proteger a missão, nenhum subgrupo de três tripulantes pode somar menos de 290 pontos no teste de saúde.

Se o consumo energético da tripulação e de suas atividades profissionais ultrapassar 150 kcal/dia, serão possíveis duas soluções: equipar a nave com uma bateria extra de energia a um custo de R\$10 milhões ou com um módulo biológico de equilíbrio e reciclagem de energia em que a cada kcal de demanda além de 150 kcal custará R\$55.000,00 a mais.

Formular o problema de escolher a tripulação de modo a minimizar os custos totais.

Após algumas reuniões técnicas, o comando da missão se viu obrigado a repensar alguns pontos nos critérios de seleção:

Com base em um estudo da psicologia do trabalho em times:

- Se o número de homens na tripulação exceder o de mulheres em mais de duas pessoas o comandante da nave será obrigatoriamente uma mulher e vice-versa.
- As pessoas designadas em vaga de comando não exercerão outra habilitação.
- As pessoas que possuem habilitação ao comando mas que não foram designadas para tal podem ocupar qualquer outra vaga de habilitação.

Com base em um estudo técnico:

- Na hipótese do aparelho de reciclagem ambiental equipar a nave, a presença de um biólogo será indispensável na tripulação.

Reformular o problema de escolher a tripulação de modo a atender também às restrições derivadas do estudo da psicologia de trabalho em times e da exigência de reciclagem ambiental.

Solução do primeiro pedido

Escolheremos uma variável binária para decidir a escolha de cada pessoa para compor a missão. Como as exigências envolvem distinção de sexo, as variáveis levarão isso em conta:

1. Escolha da variável de decisão

$$h_i \equiv \begin{cases} 1, & \text{caso o homem } i \text{ seja escolhido} \\ 0, & \text{caso contrário} \end{cases} \quad i = F, G, S, P, J, B, \text{ o índice dos homens desse grupo.}$$

$$m_j \equiv \begin{cases} 1, & \text{caso a mulher } j \text{ seja escolhida} \\ 0, & \text{caso contrário} \end{cases} \quad j = M, A, L, D, C, \text{ o índice das mulheres desse mesmo grupo.}$$

2. Formulação das restrições tecnológicas

a) Restrição associada ao número de homens e mulheres na missão:

$$\sum_{i=F}^B h_i \geq 2$$

$$\sum_{j=M}^C m_j \geq 1$$

(*) O enunciado não deixa claro por que o número de vagas mínimo para cada sexo possui essa distribuição aparentemente discriminatória.

b) Restrição associada à exigência da habilidade de comando:

$$h_F + h_G + h_J + h_B + m_D \geq 2$$

c) Restrição associada à exigência da habilidade de psicologia:

$$m_D + m_C \geq 1$$

d) Restrição associada à exigência da habilidade de astronomia:

$$h_P + h_B + h_S \geq 1$$

e) Restrição associada à exigência da habilidade de informática:

$$h_F + h_J + m_M + m_L + m_D \geq 1$$

f) Restrição associada ao uso do simulador:

$$15h_F + 20h_G + 18h_J + 24h_P + 22h_J + 10h_B + 20m_M + 25m_A + 20m_L + 18m_D + 10m_C \leq 1.000$$

g) Restrição de compatibilidade pessoal:

$$\begin{aligned} h_F + h_J &\leq 1 \text{ (Fábio e José)} \\ h_B + m_L &\leq 1 \text{ (Bruno e Linda)} \\ m_D + m_A &\leq 1 \text{ (Dina e Ana)} \end{aligned}$$

h) Restrição associada aos aspectos epidemiológicos:

Essa restrição pode ser um problema significativo para o modelo pois seu número é basicamente explosivo, uma vez que resulta da combinação do número de candidatos três a três. Considerando o desempenho sanitário s_i e s_j associado respectivamente aos tripulantes do sexo masculino (índice i) e feminino (índice j), considerando ainda os conjuntos $\varphi \subset \{F, G, S, P, J, B\}$ e $\pi \subset \{M, A, L, D, C\}$, podemos modelar a exigência através das seguintes restrições:

$$\sum_{\substack{i \in \varphi \\ |\varphi| + |\pi| = 3}} s_i h_i + \sum_{j \in \pi} s_j m_j \geq 290$$

No presente caso é visível que várias das restrições seriam redundantes.

3. Elaboração da função objetivo

Devido a complexidade da F.O. vamos dividi-la em três parcelas a saber:

$$z = \text{Minimizar } \{f(x) = A + B + C\}$$

onde:

A: É a parcela relativa ao custo pelo peso dos tripulantes e suas bagagens.

B: É a parcela do módulo adicional de transporte (agregado se a carga variável total exceder 1.000 kg).

C: É a parcela devida ao módulo de energia ou ao reciclador biológico.

✓ Parcela A

$$A = 10^4 [\text{somatório do peso transportado}] =$$

$$\begin{aligned} &= 10^4 [(35 + 80)h_F + (25 + 85)h_G + (30 + 70)h_S + (40 + 60)h_P + (17 + 90)h_J + (40 + 75)h_B + \\ &+ (30 + 50)m_M + (40 + 45)m_A + (30 + 55)m_L + (20 + 60)m_D + (30 + 65)m_C] \end{aligned}$$

✓ Parcela B

A modelagem dessa parcela necessita de um raciocínio complementar. O problema diz que se o peso contabilizado na parcela A ultrapassar 1.000 kg, uma nova parcela deve ser acrescentada à função objetivo. Resumindo:

Se [somatório do peso transportado] > 1.000, então existe a parcela B. Vamos modelar essa condição através da variável binária σ da seguinte forma:

$$\begin{aligned} \text{Se } [\sum \text{pesos} > 1.000] &\Rightarrow \sigma = 1 \\ \text{Se } [\sum \text{pesos} \leq 1.000] &\Rightarrow \sigma = 0 \end{aligned}$$

A condição lógica anteriormente definida pode ser representada pela restrição que se segue:

$$(\sum \text{pesos} - 1.000) \sigma \geq 0, \text{ para } \sigma \text{ uma variável binária } \{0,1\}.$$

Com a variável binária podemos então controlar a condição de ativação da parcela de R\$10⁷ na função objetivo. Nesse caso:

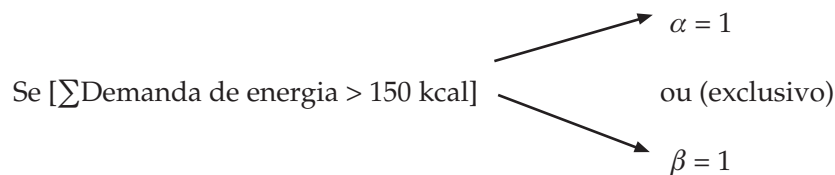
$$B = 10^7 \sigma$$

com a seguinte restrição lógica que será acrescentada no conjunto das restrições do problema:

$$\begin{aligned} \{ & (115h_F + 110h_G + 100h_S + 100h_P + 107h_J + 115h_B + 80m_M + \\ & + 85m_A + 85m_L + 80m_D + 95m_C) - 1.000 \} \sigma \geq 0 \end{aligned}$$

✓ Parcela C

A parcela C exige um controle da condição de ativação para duas situações diferentes. Quando o somatório do dispêndio energético ultrapassa 150 kcal/dia, duas hipóteses podem ser possíveis: a ativação de um módulo de reciclagem biológica ou um módulo-bateria. Nesse caso, necessitamos de duas variáveis binárias de controle e a condição da decisão lógica da modelagem pode ser assim descrita:



Condição que pode ser formulada como:

$$\begin{aligned} (\sum \text{Demanda} - 150) (\alpha + \beta) &\geq 0 \text{ e} \\ (\alpha + \beta) &\leq 1 \\ \text{com } \alpha, \beta &\in \{0, 1\} \end{aligned}$$

ou seja:

$$\begin{aligned} \text{a-} & \{ (20h_F + 29h_G + 30h_S + 35h_P + 23h_J + 40h_B + 25m_M + 30m_A + 25m_L + \\ & + 17m_D + 15m_C) - 150 \} (\alpha + \beta) \geq 0 \\ \text{b-} & (\alpha + \beta) \leq 1 \end{aligned}$$

Acrescentando as duas restrições anteriores ao conjunto de restrições do problema e utilizando as duas variáveis binárias para controlar a decisão da função objetivo podemos escrever a parcela C como:

$$C = 10^7 \alpha + 55.000 \beta \{ (20h_F + 29h_G + 30h_S + 35h_P + 23h_J + 40h_B + 25m_M + 30m_A + 25m_L + 17m_D + 15m_C) - 150 \}$$

Observamos que a inclusão de restrições lógicas dessa natureza torna o problema não linear.

Solução do segundo pedido

✓ Exigência da dedicação exclusiva ao comando

A consideração de uma distinção entre as pessoas que ocupam a vaga de comando na designação de tarefas não é possível de ser feita no modelo anterior. Para atender ao pedido do comando da missão, o modelo deverá receber uma variável que torne esse controle possível, então:

x_i , variável binária que assume o valor 1 quando o tripulante i é designado para a vaga de comando e 0 em caso contrário, $i = F, \dots, C$.

A consideração da variável x_i nos leva a uma implicação lógica. Se $x_i = 1$, obviamente os h_i e m_i correspondentes serão igualmente 1 (ninguém pode ocupar uma vaga de comando se não for designado à missão). Nesse caso, a restrição de comando, que foi no primeiro pedido escrita em termos de h_i e m_i , pode agora ser reescrita com a nova variável de decisão.

Para garantir a condição lógica de que se $x_i = 1 \Rightarrow h_i = 1$ ou $m_i = 1$, nos índices correspondentes temos a possibilidade de introduzir as seguintes restrições:

$$x_i - h_i \leq 0 \text{ ou } x_i - m_i \leq 0$$

Assim sendo, acrescentaremos as seguintes restrições ao modelo:

$$x_F - h_F \leq 0 ; x_M - m_M \leq 0 ; x_G - h_G \leq 0 \text{ ou } x_A - m_A \leq 0 \text{ etc.}$$

e mais a nova restrição para o comando que substitui a do pedido 1:

b) Restrição associada a exigência da habilidade de comando

$$x_F + x_G + x_J + x_B + x_D \geq 2$$

Nesse ponto, insistimos em ressaltar que o controle sobre o sexo do tripulante representado pela variável x é perfeitamente possível de ser realizado por seu índice. Isso poderia ter sido levado em conta desde o primeiro pedido.

✓ Exigência do equilíbrio de poder entre os sexos

O comando da missão deseja equilibrar o poder entre os sexos da tripulação. Chamando de R o seguinte balanço entre os tripulantes:

$$R = \left(\sum_{i=F}^B h_i - \sum_{i=M}^C m_i - 2 \right)$$

Então:

Se $R > 0$ pelo menos uma mulher será designada ao comando.

Se $R < 0$ pelo menos um homem será designado ao comando.

Se $R = 0$ permanece as condições do pedido 1.

O pedido nos leva à necessidade de controlar a ativação e desativação de restrições no problema da seguinte forma:

Se $R > 0 \Rightarrow \sum_{i \in V} x_i \geq 1$, onde V é o conjunto das mulheres habilitadas ao comando.

Se $R < 0 \Rightarrow \sum_{i \in W} x_i \geq 1$, onde W é o conjunto dos homens habilitados ao comando.

Para o caso:

a) $5x_D - R \geq 0$

b) $5(x_F + x_G + x_J + x_B) + R \geq 0$

Observar que se $R < 0$ a restrição a) é trivialmente atendida e, para o caso de $R > 0$, é a restrição (b) que será trivialmente atendida.

O valor do coeficiente 5 foi escolhido convenientemente igual ao número de tripulantes da nave para evitar problemas de inconsistência como, por exemplo, impedir que uma das pessoas do grupo minoritário fosse designada ao comando.

✓ Exigência da presença de um biólogo

A presença de um biólogo pode ser controlada pela variável β . A restrição da obrigatoriedade da presença de um biólogo será ativada quando $\beta = 1$. No caso específico:

$$(h_S + m_M + m_C) - \beta \geq 0$$

Observar que quando $\beta = 0$ a condição é trivialmente atendida.

12 – O Problema das Lagoas de Estabilização



Um determinado sistema de tratamento de esgotos foi concebido de forma a estruturar a coleta de efluentes industriais e domésticos e reduzir o débito de oxigênio nas águas servidas (DBO) através de tratamentos aeróbicos em duas lagoas de estabilização e um canal de ligação. A Figura 2.9 esclarece a arquitetura geral da coleta e tratamento:

As normas de proteção ambiental exigem que as águas lançadas pelo emissário no mar tenham no máximo 2% de resíduos não processados pelas lagoas e canal do sistema de tratamento. Ambas as lagoas recebem carga de esgoto bruto e efluentes industriais não tratados. O volume de poluição a ser tratado em cada descarga P_i , varia percentualmente em função do tipo de descarga. O tratamento consiste em uma decomposição da poluição através de bactérias e aeração. Basicamente os elementos principais do processo são uma profundidade conveniente (garantida pela engenharia) das lagoas e um tempo adequado de retenção do poluente. A parcela de aeração será garantida pela declividade do canal (possuirá degraus de aeração). A retenção será garantida pelo volume de cada lagoa. Considerando que as descargas industriais possuam um percentual de poluição de 40% em volume e que os esgotos domésticos de 10%. Considerando ainda que:

- As descargas de infiltração do lençol freático não possuam percentual de poluição.
- Em cada hora de retenção nas lagoas, o efluente perde, por decomposição biológica, cerca de 3% de seu conteúdo de poluição.
- Em cada quilômetro do canal de aeração o efluente é reduzido de 2% em poluição.

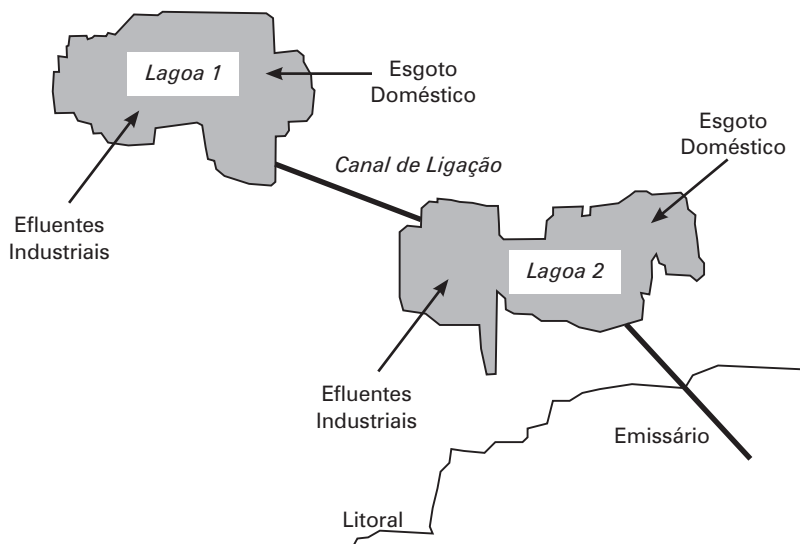


FIGURA 2.10 Conjunto de lagoas.

Formular o problema de determinar os volumes das duas lagoas e o comprimento do canal que minimize o custo da obra.

Outros dados: o comprimento do canal pode variar entre 1.500 m e 3.000 m. O volume das lagoas podem variar entre 10^6 m^3 e 10^8 m^3 . Cada metro do canal custa R\$1.200,00. Cada m^3 da primeira lagoa custa R\$1,02, e da segunda, R\$1,1. As vazões de efluentes são:

TABELA 2.21

Tipo de Descarga	Valor da Descarga (m^3)	Localização
Esgoto Doméstico	p_1	Lagoa 1
Efluente Industrial	p_2	Lagoa 1
Infiltração de Subsolo	p_3	Lagoa 1
Esgoto Doméstico	p_4	Lagoa 2
Efluente Industrial	p_5	Lagoa 2
Infiltração de Subsolo	p_6	Lagoa 2

As leis ambientais determinam que na primeira lagoa o índice de poluição não ultrapasse 5% e ao se lançar as águas tratadas ao mar, não exista mais de 1% de poluição no efluente.

Solução:

1– Escolha das variáveis de decisão

$v_i \equiv$ o volume da lagoa i , $i = 1, 2$

$L \equiv$ comprimento do canal de ligação.

$p_j \equiv$ volume da descarga de efluentes j , $j = 1, \dots, 6$.

$h_i \equiv$ tempo de retenção na lagoa i , $i = 1, 2$.

2. Formulação das restrições tecnológicas

a) Restrição relativa aos volumes das lagoas:

$$10^6 \leq v_i \leq 10^8, i = 1, 2$$

b) Restrição relativa ao comprimento do canal de aeração:

$$1.500 \leq L \leq 3.000$$

c) Restrições de retenção

$$\begin{aligned} v_1 &= h_1 (p_1 + p_2 + p_3) \\ v_2 &= h_2 (p_1 + p_2 + p_3 + p_4 + p_5 + p_6) \end{aligned}$$

d) Restrições de depuração biológica na primeira lagoa:

$$\left(\frac{0,1p_1 + 0,4p_2}{p_1 + p_2 + p_3} \right) (0,97)^{h_1} \leq 0,05$$

e) Restrições de depuração no canal:

Denominado por p_7 o índice de poluição do efluente ao final do canal temos:

$$p_7 = ((p_1 + p_2 + p_3) 0,05) 0,98 L$$

f) Restrições de depuração biológica na segunda lagoa:

$$\left(\frac{p_7 + 0,1p_4 + 0,4p_5 + p_6}{p_1 + p_2 + p_3 + p_4 + p_5 + p_6} \right) (0,97)^{h_2} \leq 0,01$$

3. Configuração da função objetivo

$$z = \text{Minimizar } \{f(x) = 1,02v_1 + 1,1v_2 + 1,2(10^6 L)\}$$

13 – O Problema da Programação de Dietas em uma Clínica de Repouso



Uma clínica de repouso recebe cerca de 50 pacientes em um fim de semana para exercícios e desintoxicação alimentar. Um dos pontos fundamentais do planejamento da clínica é a escolha das dietas e a preparação das refeições. Devido ao grande consumo de alimentos, a clínica compra normalmente os gêneros no atacado. A Tabela 2.22 descreve a constituição nutritiva das refeições que poderão ser preparadas dentro das dietas escolhidas para os próximos grupos de visitantes.

TABELA 2.22 CONSTITUIÇÃO NUTRITIVA

Grupos	Refeições	Proteínas g	Carboidratos g	Minerais mg	Vitaminas Unidades	Fibras g
I	<i>Dieta 1</i>	60	100	300	10 – A; 20 – C	150
	<i>Dieta 2</i>	80	50	350	20 – A; 10 – C	100
	<i>Dieta 3</i>	80	20	100	40 – A; 20 – C	90
	<i>Dieta 4</i>	100	25	100	25 – A; 10 – C	100
II	<i>Dieta 5</i>	40	30	300	10 – A; 20 – C	150
	<i>Dieta 6</i>	50	35	350	20 – A; 10 – C	200
	<i>Dieta 7</i>	–	50	100	40 – A; 20 – C	90
III	<i>Dieta 8</i>	40	25	100	25 – A; 10 – C	80
	<i>Dieta 9</i>	80	10	250	40 – A; 50 – C	100

Cada uma das dietas é composta pelos seguintes ingredientes ou alimentos:

TABELA 2.23 CONSUMO DE INGREDIENTES

Grupos	Refeições	Carne g	Cereais g	Massas g	Verduras g	Leite ml	Frutas g
I	<i>Dieta 1</i>	60	50	80	60	50	100
	<i>Dieta 2</i>	30	60	30	100	40	150
	<i>Dieta 3</i>	20	60	20	60	60	120
	<i>Dieta 4</i>	20	40	30	80	30	80
II	<i>Dieta 5</i>	40	80	–	80	–	150
	<i>Dieta 6</i>	50	50	–	100	–	100
	<i>Dieta 7</i>	–	100	–	100	–	100
III	<i>Dieta 8</i>	–	50	20	120	50	50
	<i>Dieta 9</i>	–	40	10	150	100	40
	<i>Preço por kg</i>	4,00	6,00	1,50	0,9	1,0	0,7

Para a execução do programa de treinamento, a clínica deve escolher uma dieta dentro de cada grupo (dejejum, almoço e jantar), para dar condições ao cozinheiro de preparar as refeições.

Pedido 1: sabendo-se que existem dois programas de treinamento e que para cada programa as necessidades em nutrientes diárias por pessoa está relacionada a seguir, programar a compra de ingredientes de forma a minimizar os gastos com alimentação.

TABELA 2.24 DEMANDA DOS PROGRAMAS DE TREINAMENTO

Programa de Treinamento	Proteínas g	Carboidratos g	Minerais mg	Vitaminas Unidades	Fibras g	Pessoas Matriculadas
<i>Programa 1</i>	150	200	400	30 – A 40 – C	250	20
<i>Programa 2</i>	180	220	500	20 – A 60 – C	300	30

Pedido 2: por um problema logístico, sabe-se que não é desejável preparar menos de 10 refeições com base na dieta do tipo 4, caso ela seja escolhida. Reformular o problema anterior levando-se isso em conta.

Pedido 3: após uma rápida análise do nutricionista, descobriu-se que a dieta 1 e a dieta 9 eram incompatíveis, e a dieta 2 deveria vir sempre acompanhada da dieta 8. Reformular o pedido 1 levando em conta a nova informação.

Solução do pedido 1

1. Escolha das variáveis de decisão

$$x_i \equiv \begin{cases} 1, & \text{caso a dieta seja escolhida para ser adotada} \\ 0, & \text{caso contrário} \end{cases} \quad i = 1, \dots, 9$$

$y_i \equiv$ número de refeições preparadas com base na dieta i , $i = 1, \dots, 9$.

$A = [a_{ij}]$ / matriz de constituição nutritiva, $i = 1, \dots, 9$ e $j = \text{Pr, Cr, Mi, Vi, Fi}$;

$B = [b_{ij}]$ / matriz de consumo de ingredientes, $i = 1, \dots, 9$ e $j = \text{Ca, Ce, Ma, Ve, Le, Fr}$;

$c_j \equiv$ custo da unidade do ingrediente j , $j = \text{Ca, Ce, Ma, Ve, Le, Fr}$;

$N_{sj} \equiv$ necessidade do programa s , $s = 1, 2$ com respeito ao nutriente $j = \text{Pr, Cr, Mi, Vi, Fi}$.

2. Formulação das restrições tecnológicas

a) Restrição associada às necessidades nutritivas dos programas de treinamento:

O programa de treinamento engloba dois dias, ou seja, duas vezes a demanda da tabela dos programas de treinamento; contudo, não há qualquer condição que obrigue o planejamento a distinguir os dias, portanto, o programa pode ser planejado para um dia e repetido no dia seguinte (pelas condições do pedido 1). A demanda geral é obtida pelo produto do número de pessoas matriculadas no programa vezes as exigências diárias em nutrientes.

Seja o cálculo da demanda de proteína por dia:

No programa 1 temos: 150 gramas \times 20 pessoas = 3.000 g de consumo.

No programa 2 temos: 180 gramas \times 30 pessoas = 5.400 g de consumo.

Repetindo esse cálculo para cada nutriente temos:

- *Programa 1:* 3.000 g de proteínas, 4.000 g de carboidratos, 8.000 U de minerais, 600 U de vitamina A, 800 U de vitamina C e 5.000 g de fibras.
- *Programa 2:* 5.400 g de proteínas, 3.600 g de carboidratos, 15.000 U de minerais, 600 U de vitamina A, 1.800 U de vitamina C e 9.000 g de fibras.

A demanda dos programas deverá ser atendida por três refeições diárias, uma dentro de cada grupo. Para o caso da proteína, por exemplo, as 3.000 g + 5.400 g = 8.400 g serão atendidas com as seguintes refeições:

$$60y_1 + 80y_2 + 80y_3 + 100y_4 + 40y_5 + 50y_6 + 40y_8 + 80y_9 \geq 8.400$$

Utilizando a notação matricial:

$$\sum_{i=1}^9 a_{ij}y_i \geq \sum_{s=1}^2 N_{sj}, j = \text{Pr, Ca, Mi, Vi, Fi.}$$

b) Restrição associada à distribuição das dietas pelas três refeições diárias:

Esse é um interessante caso em que devemos desenvolver variáveis lógicas independentes das variáveis reais do problema. O raciocínio a modelar é o seguinte:

Se a refeição que será escolhida pertence a uma dieta do grupo g ($g = \text{I, II, III}$) nenhuma outra dieta desse grupo poderá ser ativada. Se imaginarmos uma variável de decisão x_i binária para descrever essa condição de pertinência, então a condição lógica de modelagem será:

$$(x_1 \text{ ou } x_2 \text{ ou } x_3 \text{ ou } x_4) \text{ e } (x_5 \text{ ou } x_6 \text{ ou } x_7) \text{ e } (x_8 \text{ ou } x_9)$$

Onde a condição de ou é exclusiva, implica em apenas uma das variáveis ativadas. O conjunto de restrições que pode atender a essa condição lógica para as variáveis binárias é:

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 &= 1 \\ x_5 + x_6 + x_7 &= 1 \\ x_8 + x_9 &= 1 \end{aligned}$$

Não podemos esquecer que as variáveis binárias devem ser “ligadas” às variáveis reais do problema que decidem sobre o número de refeições a serem preparadas. A condição lógica dessa “ligação” diz que se uma dieta é escolhida, o número de refeições que ela dará origem está liberado para ser calculado pelo programa. Caso a dieta não seja escolhida, o número de refeições correspondente deve ser, obviamente, zero. A condição lógica pode ser reescrita como:

$$\begin{aligned} x_1 = 0 &\Rightarrow y_1 = 0 \\ x_1 = 1 &\Rightarrow y_1 \geq 0 \end{aligned}$$

Essa condição lógica pode ser modelada como:

$$y_i - Mx_i \leq 0$$

Onde M é um número adequadamente grande. No caso, 50 é um número adequado pois é o limite das refeições em cada grupo, de modo que a restrição pode ser escrita como:

$$y_i - 50x_i \leq 0, i = 1, \dots, 9$$

c) Restrição associada ao total de refeições diárias ($50 \times 3 = 150$):

$$\sum_{i=1}^4 y_i = 50$$

$$\sum_{i=5}^7 y_i = 50$$

$$\sum_{i=8}^9 y_i = 50$$

3. Elaboração da função objetivo

A função objetivo deve ser expressa em termos de ingredientes, que são os elementos cotados monetariamente e que devem ser adquiridos no atacado. O somatório das quantidades de cada ingrediente varia com o tipo de dieta que dá suporte à refeição, o que é resumido na tabela de consumo de ingredientes. Dessa forma, a quantidade de um certo ingrediente é dada por:

$$\sum_{i=1}^9 b_{ij} y_i \quad j = \text{Ca, Ce, Ma, Ve, Le, Fr}$$

Daí:

$$z = \text{Minimizar } \{f(x) = c_j \sum_{i=1}^9 b_{ij} y_i \quad j = \text{Ca, Ce, Ma, Ve, Le, Fr}\}$$

valor total gasto com os ingredientes.

4. Restrição de não negatividade

$$y_i \geq 0, x_i \geq 0, i = 1, \dots, 9$$

Solução do pedido 2

Só necessitamos alterar uma das restrições da solução do pedido 1. A condição lógica que unia as refeições era:

$$\begin{aligned} x_1 = 0 &\Rightarrow y_1 = 0 \\ x_1 = 1 &\Rightarrow y_1 \geq 0 \end{aligned}$$

Essa condição lógica deve ser reescrita, no caso da dieta 4, na forma:

$$\begin{aligned} x_4 = 0 &\Rightarrow y_4 = 0 \\ x_4 = 1 &\Rightarrow y_4 \geq 10 \end{aligned}$$

A restrição do pedido 1, $y_4 - 50x_4 \leq 0$, garante apenas um limite superior para $x_4 = 1$ na forma de:

$$y_4 \leq 50$$

O segundo pedido exige também um limite inferior para a variável número de refeições. Para tal, devemos acrescentar uma restrição do tipo:

$$y_i - Lx_i \geq 0$$

Nesse caso, com o valor do limite $L = 10$:

$$y_4 - 10x_4 \geq 0$$

Solução do pedido 3

Para a primeira exigência: se duas dietas $i-j$ são incompatíveis, então a condição lógica de modelagem será:

$$\begin{aligned}x_i = 1 &\Rightarrow x_j = 0 \\x_j = 1 &\Rightarrow x_i = 0\end{aligned}$$

No caso de variáveis binárias, essa condição de incompatibilidade pode ser expressa pela equação:

$$x_i + x_j \leq 1$$

Nesse caso, então, devemos acrescentar ao conjunto de restrições a seguinte inequação:

$$x_1 + x_9 \leq 1$$

Para a segunda exigência, a condição lógica é do tipo:

$$\begin{aligned}x_i = 1 &\Rightarrow x_j = 1 \\x_j = 1 &\Rightarrow x_i = 0 \text{ ou } 1\end{aligned}$$

No caso de variáveis binárias, essa condição é expressa pela equação:

$$x_i - x_j \leq 0$$

No caso do terceiro pedido, devemos, então, acrescentar ao conjunto de restrições a seguinte inequação:

$$x_2 - x_8 \leq 0$$

14 – O Problema do Casamento e da Casa Própria



Uma agência de casamento procura promover a melhor união possível entre os casais de seu vasto banco de dados (cerca de $R = \{1, \dots, r\}$ rapazes e $M = \{1, \dots, m\}$ moças). Para a solução desse problema existe uma série de critérios a adotar. O índice de compatibilidade final, que é calculado levando em conta vários condicionantes, é o fator preponderante. O sistema de dados da agência pode facilmente obter a adequação feminina f_{ij} , ou seja, a variável que diz o quanto a moça i é adequada ao rapaz j , e a masculina, h_{ij} , que expressa a situação inversa. Quando os valores de f e h estão entre determinados limites, o casal pode ser formado, sendo formação cada vez mais adequada, na medida do crescimento da soma desses fatores. Além desse indicador, existe uma enorme condicionante econômica a ser levada em conta: a casa própria. As experiências dessa agência dizem que um casal só é admissível quando ambos possuem interesse pela mesma casa, ela puder ser disponibilizada e seu preço estiver dentro da posse do casal. Existem $W = \{1, \dots, w\}$ casas no banco de dados.

Formular o problema de maximizar o número de uniões estáveis dentro do banco de dados.

Solução

1. Escolha das variáveis de decisão

$$x_{ij}^k \equiv \begin{cases} 1, & \text{caso o casal moça } i \times \text{rapaz } j, \text{ morando na} \\ & \text{casa } k \text{ seja formado} \\ 0, & \text{caso contrário} \end{cases} \quad i = 1, \dots, m$$

$j = 1, \dots, r ; k = 1, \dots, w$

$f_{ij} \equiv$ adequação da moça i ao rapaz j .

$h_{ji} \equiv$ adequação do rapaz j à moça i .

$l \equiv$ limite inferior da adequação das moças pelos rapazes.

$u \equiv$ limite inferior da adequação dos rapazes pelas moças.

$v_i \equiv$ capacidade econômica da moça $i = 1, \dots, m$.

$w_j \equiv$ capacidade econômica do rapaz $j = 1, \dots, r$.

$c_k \equiv$ custo da casa $k, k = 1, \dots, w$.

2. Formulação das restrições tecnológicas

a) Restrição associada aos limites de compatibilidade:

$$\sum_{k=1}^w \sum_{i=1}^m x_{ij}^k f_{ij} \geq l, j = 1, \dots, r$$

$$\sum_{k=1}^w \sum_{j=1}^r x_{ij}^k h_{ij} \geq u, i = 1, \dots, m$$

b) Restrições associadas à monogamia:

$$\sum_{k=1}^w \sum_{j=1}^r x_{ij}^k \leq 1, i = 1, \dots, m$$

$$\sum_{k=1}^w \sum_{i=1}^m x_{ij}^k \leq 1, j = 1, \dots, r$$

c) Restrição associada à capacidade financeira:

A condição pode ser expressa logicamente como:

$$\sum_{i=1}^m \sum_{j=1}^r x_{ij}^k (v_i + w_j) \geq c_k, k = 1, \dots, w$$

d) Restrição associada à disponibilidade da casa:

A condição pode ser interpretada como a impossibilidade de dois casais receberem a mesma casa, ou seja:

$$\sum_{i=1}^m \sum_{j=1}^r x_{ij}^k \leq 1, k = 1, \dots, w$$

3. Elaboração da função objetivo

$$z = \text{Maximizar } \{f(x) = \sum_{k=1}^w \sum_{i=1}^m \sum_{j=1}^r x_{ij}^k\} - \text{número total de uniões.}$$

2.3 – PROBLEMAS PROPOSTOS

1 – O Problema das Calças e das Camisas



Uma determinada confecção opera com dois produtos: calças e camisas. Como tratam-se de produtos semelhantes, possuem uma produtividade comparável e compartilham os mesmos recursos. A programação da produção é realizada por lotes de produto.

O departamento de produção informa que são necessários 10 homens \times hora para um lote de calças e 20 homens \times hora para um lote de camisas. Sabe-se que não é necessária mão de obra especializada para a produção de calças, mas são necessários 10 homens \times hora desse tipo de mão de obra para produzir um lote de camisas. O departamento de pessoal informa que a força máxima de trabalho disponível é de 30 homens \times hora de operários especializados e de 50 homens \times hora de não especializados.

Da planta de produção, sabemos que existem apenas duas máquinas com capacidade de produzir os dois tipos de produto, sendo que a máquina 1 pode produzir um lote de calças a cada 20 horas e um lote de camisas a cada 10 horas, não podendo ser utilizada por mais de 80 horas no período considerado. A máquina 2 pode produzir um lote de calças a cada 30 horas e um lote de camisas a cada 35 horas, não podendo ser utilizada por mais de 130 horas no período considerado.

São necessários dois tipos de matéria-prima para produzir calças e camisas. Na produção de um lote de calças são utilizados 12 quilos da matéria-prima A e 10 da B. Na produção de um lote de camisas são utilizados 8 quilos da matéria-prima A e 15 da B.

O almoxarifado informa que, por imposições de espaço, só pode fornecer 120 quilos de A e 100 quilos de B no período considerado.

Sabendo-se que o lucro pela venda é de 800 reais nos lotes de camisas e de 500 reais nos lotes de calças, formule o problema de maximizar o lucro da operação produtiva em pauta.

2 – O Problema do Pedido de Socorro



Um posto de rádio capta a seguinte mensagem: “Somos náufragos do navio Intrépido. Estamos isoladas em uma ilha de coordenadas... No momento sofremos com diversos problemas: as cobras venenosas, algumas pessoas com diarreia, os nativos canibais e a falta de água e alimentos. Temos duas armas, mas a munição está praticamente no fim. Só temos uma ampola de soro antiofídico e o suprimento de água e comida acabou. Somos 20 pessoas, nos enviem ajuda !!!”. O chefe da patrulha de salvamento da área só possui um helicóptero, que, por sua vez, só pode fazer uma viagem de socorro até a isolada e longínqua ilha. O grupo de salvamento sabe que o resgate por mar vai levar três dias para chegar. Para sobreviver, o grupo necessita de seis caixas de alimento, quatro de água, duas de munição e duas com remédios e soro antiofídico. Em valor para a sobrevivência, as caixas de munição e remédios são duas vezes mais importantes do que as de água e quatro vezes mais importantes do que as de alimentos. Sabendo-se que apenas sete caixas poderão ser transportadas no helicóptero, elaborar o programa que otimiza a carga de socorro.

3 – O Problema da Refinaria



Em uma determinada refinaria, o petróleo bruto sofre os seguintes processamentos antes de ser transformado em gás/óleo ou gasolina bruta:

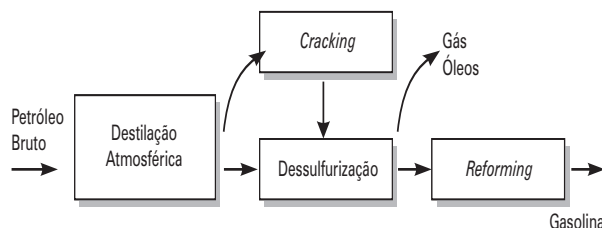


FIGURA 2.11 Esquema de refino.

A Tabela 2.25 representa a capacidade máxima de processamento de cada unidade de operação. Formular o problema de modo a maximizar os lucros totais, solucionando-o graficamente.

TABELA 2.25 CAPACIDADE MÁXIMA DE PROCESSAMENTO

<i>Operação</i>	<i>Produto Gasolina Bruta (Ton/ano)</i>	<i>Produto Gás/Óleo (Ton/ano)</i>
<i>Destilação</i>	500.000	600.000
<i>Dessulfuração</i>	700.000	500.000
<i>Reforming</i>	400.000	
<i>Cracking</i>		450.000
<i>Lucros Unitários</i>	10,00 reais/Ton	7,00 reais/Ton

4 – O Problema do Reflorestamento



Uma companhia de reflorestamento possui áreas de plantio em quatro municípios. A empresa considera o uso de espécies de árvores: pinus, carvalho, nogueira e araucária. A Tabela 2.26 resume os dados do problema:

TABELA 2.26 DADOS DE PLANTIO E PRODUTIVIDADE

<i>Cidade</i>	<i>Produção Anual Esperada m³/Hectare</i>				<i>Renda Anual Esperada Unidade Monetária/Hectare</i>			
<i>Área Disponível</i>	<i>Pinus</i>	<i>Carvalho</i>	<i>Nogueira</i>	<i>Araucária</i>	<i>Pinus</i>	<i>Carvalho</i>	<i>Nogueira</i>	<i>Araucária</i>
1 1.500	17	14	10	9	16	12	20	18
2 1.700	15	16	12	11	14	13	24	20
3 900	13	12	14	8	17	10	28	20
4 600	10	11	8	6	12	11	18	17
<i>Produção Mínima em 1.000 x m³</i>	225	9	4,8	3,5				

Formule o problema de designar as áreas de plantio por município de forma a maximizar a renda.

5 – O Problema do Hospital



O diretor de um hospital deve escolher um esquema de designação de leitos e quartos em uma nova ala que será construída.

Existem três tipos de quartos possíveis:

- Com um leito.
- Com dois leitos.
- Com três leitos.

O total de quartos a construir não pode ultrapassar 70. Por imposições de demanda deverão ser oferecidos pelo menos mais 120 leitos. A percentagem de quartos de um leito deve ser restrita entre 15% a 30% do total de quartos. A necessidade em área construída é de:

- 10 m² por quarto com um leito.
- 14 m² por quarto com dois leitos.
- 17 m² por quarto com três leitos.

Os pacientes dos quartos com dois e três leitos exigem apenas 80% da mão de obra que os do quarto individual. O que o hospital recebe por cada paciente internado é inversamente proporcional à capacidade do número de pessoas do quarto em que ele está internado. Considerando o hospital sempre lotado, formule o problema de forma a:

- Minimizar o esforço da mão de obra em apoio médico e administrativo.
- Maximizar a arrecadação global.
- Maximizar o número de leitos.
- Minimizar o espaço necessário para a nova ala.

6 – O Problema da Compra de Aviões da VAB



A Viação Aérea Brasileira está estudando a compra de três tipos de aviões: Boeing 717 para as pontes aéreas de curta distância, Boeing 737-500 para voos domésticos e internacionais de média distância e MD-11 para voos internacionais de longa distância. Em um estudo preliminar, considerou-se que a capacidade máxima dos aviões a serem comprados será sempre preenchida para efeito de planejamento. Os dados de planejamento constam da Tabela 2.27:

TABELA 2.27 DADOS OPERACIONAIS DOS AVIÕES

<i>Tipo do Avião</i>	<i>Custo Milhões US\$</i>	<i>Receita Teórica Milhões US\$</i>	<i>Pilotos Aptos</i>
BOEING 717	5,1	330	30
BOEING 737-500	3,6	300	20
MD-11	6,8	420	10

A verba disponível para as compras é de 220 milhões de dólares. Os pilotos de MD-11 podem pilotar todos os aviões da empresa, mas os demais pilotos só podem ser escalados às aeronaves a que foram habilitados. Cada aeronave necessita de dois pilotos para operar. As oficinas de manutenção podem suportar até 40 Boeings 717. Um Boeing 737-500 equivale, em esforço de manutenção, a 3/4, e um MD-11 a 5/3, quando referidos ao Boeing 717. Formular o modelo de PL do problema de otimizar as aquisições de aviões.

7 – O Problema da Frota de Ônibus



Suponhamos que o número de ônibus requerido para o atendimento da demanda de um determinado corredor de transporte para a i -ésima hora do dia seja b_i , $i = 1, 2, \dots, 24$, em que cada ônibus trafega seis horas consecutivas. Se o número de ônibus no período i excede o mínimo requerido b_i , isso resultaria em um custo adicional c_i por ônibus/hora. Formule o problema como um PPL de modo que o custo adicional resultante seja minimizado.

8 – O Problema do Corredor de Transporte

☹

Um sistema ferroviário de transporte suburbano dá escoamento principalmente às demandas d_a , d_b e d_c de três grandes bairros, A, B e C, em direção ao centro da cidade. Entre cada um desses bairros existe uma pequena redução na demanda de transporte devido ao tráfego regional. Essa redução é calculada percentualmente em 5% a cada intervalo entre os grandes bairros. O sistema ferroviário é, em princípio, insuficiente para arcar com toda a demanda e necessita de uma complementação rodoviária, sob pena de entrar em colapso. Em virtude desse fato, podemos considerar que o sistema ferroviário trabalha sempre em sua capacidade máxima. Junto a cada estação ferroviária existe uma opção de embarque para um sistema rodoviário de desafogo. A percentagem da redução da demanda proporcionada pelo corredor rodoviário sobre a demanda do sistema ferroviário é denominada nível de desafogo. O custo total do sistema rodoviário é diretamente proporcional à soma das parcelas de desafogo em cada estação. A capacidade de operação do sistema ferroviário é associada ao trecho considerado no transporte, sendo denominadas por c_1 , c_2 , c_3 unidades de demanda, como mostra a Figura 2.12.

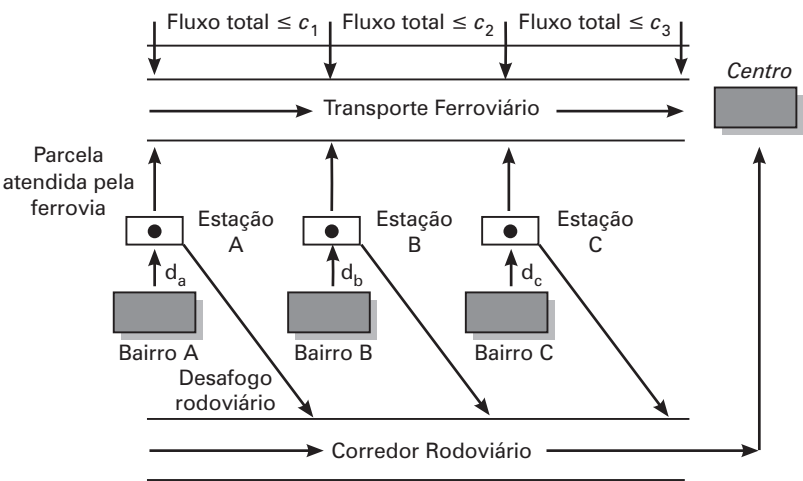


FIGURA 2.12 Esquema do sistema de transporte.

Determinar o nível percentual de desafogo que deve possuir o sistema rodoviário em cada estação para que, em qualquer trecho da ferrovia, a demanda não ultrapasse os valores estipulados e de modo a minimizar o custo do sistema global. O trajeto de transporte inicia-se sem demanda anterior na estação A.

9 – O Problema da Fábrica de Plásticos

☹

Uma empresa fabrica malas, bolsas, pastas e sacolas de plástico. Ela compra sua matéria-prima em rolos com uma certa largura e corta em tiras adequadas a cada tipo de objeto produzido. Sabendo-se que existem três tamanhos para cada item, as possibilidades de cortes estão resumidas na Tabela 2.28:

TABELA 2.28 QUANTITATIVOS PARA OS CORTES

		Quantidade de Itens Dentro de Cada Método de Corte								
		1	2	3	4	5	6	7	8	9
Malas	P	2	1		1	–	–	–	–	1
	M	1	2	1	1	–	–	–	3	–
	G	1	–	–	–	–	2	2	–	–
Bolsas	P	2	1	–	2	–	–	–	–	1
	M	3	–	1	4	1	2	–	3	–
	G	–	–	1	1	–	2	2	–	–

TABELA 2.28 CONTINUAÇÃO

		<i>Quantidade de Itens Dentro de Cada Método de Corte</i>								
		1	2	3	4	5	6	7	8	9
Pastas	P	6	4	2	1	–	–	–	–	1
	M	1	1	1	1	–	2	–	4	1
	G	–	2	1	1	2	2	3	–	1
Sacolas	P	–	2	–	1	2	–	–	–	1
	M	–	–	2	1	2	–	–	2	1
	G	–	1	1	–	2	–	3	–	1
Perda		3	5	5	2	4	7	1	3	8

Em um determinado dia os pedidos para a fabricação são (pequeno, médio, grande):

malas: 10, 20, 13; bolsas: 5, 2, 6; pastas: 4, 3, 12; sacolas: 5, 5, 3.

Formular o problema de PL associado a cada um dos seguintes critérios:

- Minimizar as perdas de material.
- Minimizar o volume não vendido das peças somado com as perdas de material.
- Minimizar o estoque não vendido.

10 – O Problema das Camisetas



Uma certa fábrica de camisetas deseja aproveitar as finais de um campeonato de futebol para vender camisetas dos times envolvidos. Os jogos vão durar quatro semanas. O custo de produção de cada camiseta é R\$2,00 nas duas primeiras semanas e subirá para R\$2,50 nas duas últimas, quando a concorrência demandar por material no mercado. A demanda semanal de camisetas será de 5.000, 10.000, 30.000 e 60.000. A capacidade máxima de produção da empresa é de 25.000 camisetas. Na primeira e na segunda semana a empresa poderá, em um esforço excepcional, carrear mão de obra em horas extras e fabricar mais 10.000 camisetas em cada semana. Nesse caso, o custo dessas camisetas será de R\$2,80. O excesso de produção pode ser estocado a um custo de R\$0,20 por unidade por semana.

Pedido 1: Formular o modelo de PL que minimize os custos.

Pedido 2: Após o planejamento anterior, a direção da empresa verificou que a demanda iria variar substancialmente dentro dos quatro modelos de camiseta que representavam os quatro times disputando as finais. Apesar de a demanda total ser exatamente aquela anteriormente levantada, o valor das camisetas iria variar em conformidade com o time e sua posição no campeonato. Nas duas primeiras semanas todos os times estariam em pé de igualdade até que fosse decidido os dois finalistas. A partir daí, as camisetas dos times eliminados cairiam em valor e em demanda no mercado, e as dos times finalistas subiriam conforme a Tabela 2.29:

TABELA 2.29 DEMANDA DE CAMISETAS

	<i>Semana</i>							
	<i>1</i>		<i>2</i>		<i>3</i>		<i>4</i>	
	Demanda	Valor	Demanda	Valor	Demanda	Valor	Demanda	Valor
<i>Time A</i>	1.250	5	2.500	6	500	3	–	–
<i>Time B</i>	1.250	5	2.500	6	500	3	–	–
<i>Time Finalista C</i>	1.250	5	2.500	6	14.500	8	30.000	9
<i>Time Finalista D</i>	1.250	5	2.500	6	14.500	8	30.000	9

Sabendo-se que existe um completo equilíbrio entre os quatro finalistas, formular o modelo que maximize os lucros da empresa produtora de camisetas.

11 – O Problema dos Ovos de Páscoa



Um fabricante de ovos de páscoa trabalha durante o ano para atender à grande demanda de ovos por ocasião da Páscoa. A fábrica produz quatro tipo de ovos e dois bolos. Basicamente os insumos críticos de fabricação são os mesmos, variando apenas quantitativamente e podem ser resumidos na tabela a seguir. Os custos finais dos produtos estão cotados em unidades monetárias.

TABELA 2.30 CONSUMO DOS INSUMOS PARA OS OVOS DE PÁSCOA

<i>Produtos</i>	<i>Insumos Críticos</i>				
	<i>Chocolate</i>	<i>Recheio</i>	<i>Embalagem</i>	<i>Mão de Obra</i>	<i>Custo Final</i>
<i>Ovo 15</i>	100 g	5 Unidades	1 Folha	0,05h	1 U.M.
<i>Ovo 20</i>	150 g	10 Unidades	1,5 Folhas	0,08h	2 U.M.
<i>Ovo 25</i>	200 g	20 Unidades	2,0 Folhas	0,1h	3 U.M.
<i>Ovo 30</i>	400 g	30 Unidades	2,5 Folhas	0,12h	4 U.M.
<i>Bolo Tipo1</i>	100 g	7 Unidades	2,0 Folhas	0,1h	3 U.M.
<i>Bolo Tipo2</i>	200 g	12 Unidades	3,0 Folhas	0,15h	4 U.M.

Os quantitativos totais disponíveis dos insumos para a empresa, dentro da qualidade exigida, são:

TABELA 2.31 QUANTITATIVOS DISPONÍVEIS DOS INSUMOS

<i>Períodos</i>	<i>Chocolate</i>	<i>Recheio</i>	<i>Embalagem</i>	<i>Mão de Obra</i>
<i>Abr-Mai</i>	5.000 kg	10 ⁴ Unidades	5 x 10 ³ Folhas	1.000 hs
<i>Jun-Set</i>	70.000 kg	10 ⁴ Unidades	8 x 10 ⁵ Folhas	1.500 hs
<i>Out-Jan</i>	40.000 kg	10 ⁴ Unidades	7 x 10 ³ Folhas	1.000 hs
<i>Fev-Mar</i>	20.000 kg	10 ⁴ Unidades	8 x 10 ⁴ Folhas	2.000 hs

As entregas dos ovos e bolos fabricados podem ser realizadas no seguinte esquema:

TABELA 2.32 DEMANDA (D) E PREÇO (P) DE OVOS E BOLOS

	Ovo 15		Ovo 20		Ovo 25		Ovo 30		Bolo T1		Bolo T2	
Períodos	D	P	D	P	D	P	D	P	D	P	D	P
Abr-Mai	4×10^5	3U.M.	2×10^5	5U.M.	10^5	7U.M.	4×10^4	9U.M.	2×10^5	6U.M.	2×10^4	7U.M.
Jun-Set	(*) –	–	–	–	–	–	–	–	–	–	–	–
Out-Jan	10^4	2U.M.	10^4	3U.M.	10^3	4U.M.	10^3	6U.M.	10^3	4U.M.	–	–
Fev-Mar	10^5	2U.M.	10^5	3U.M.	2×10^4	4U.M.	10^4	6U.M.	4×10^4	4U.M.	10^3	5U.M.
Custos de Armazenagem (por período)	0,2 U.M.		0,3 U.M.		0,4 U.M.		0,5 U.M.		0,5 U.M.		0,5 U.M.	
Capacidade de Armazenagem Total (em Unidades)	4×10^5		6×10^5		4×10^5		6×10^4		10^5		4×10^4	

(*) As células vazias indicam que não existe demanda prevista no período, contudo pode haver produção para o estoque.

Formule o problema de planejar o melhor esquema de produção e venda dos ovos de páscoa.

12 – O Problema da Evacuação de Emergência



Uma determinada região está sendo ameaçada pela ruptura de uma barragem e deve ser evacuada em, no máximo, dez horas. São no total 8.000 homens, 7.900 mulheres e 1.850 crianças a transportar. Cada pessoa poderá levar até dez quilos de bagagem pessoal. Toda a região foi isolada e só circulam veículos autorizados para que se evitem acidentes e engarrafamentos. Para efetuar a evacuação estão disponíveis os meios:

TABELA 2.33 RESUMO DOS DADOS OPERACIONAIS

Tipo Informação	Veículo de 6 Ton do Exército	Veículo de 1/4 Ton do Exército	Helicóptero	Ônibus	Micro-ônibus	Veículo de Passeio
Quantidade de Unidades Disponíveis	10	20	15	10	5	60
Capacidade de Transporte	20 pessoas	5 pessoas	10 pessoas	30 pessoas	15 pessoas	5 pessoas
Capacidade para Bagagem	1 ton	20 kg	50 kg	1 ton	500 kg	100 kg
Custo por Viagem	10 u.m.	4 u.m.	75 u.m.	5 u.m.	3 u.m.	2 u.m.
Tempo de Viagem	1 h	45 min	10 min	45 min	30 min	30 min

Para minimizar o pânico, as crianças deverão viajar acompanhadas por suas mães. Existem 10 famílias com 5 filhos, 25 com 4 filhos, 150 com 3, 450 com 2 e 350 com 1. Os carros de passeio só poderão fazer uma viagem de evacuação, ficando, por segurança, retidos fora da área de perigo.

Formular o programa de evacuação que minimize os custos finais da operação.

13 – O Problema do Aluno Atarefado



Um certo aluno está cursando o último ano do segundo grau e se preparando para o vestibular. Nessa difícil fase, ele precisa equilibrar sua carga de estudo e lazer, pois esse é um dos fatores importantes para se obter sucesso nesse exame. A Tabela 2.34, fornecida por seu colégio, mostra a distribuição da carga de aulas ao longo dos meses do ano:

TABELA 2.34 DISTRIBUIÇÃO DA CARGA HORÁRIA

	<i>Mar</i>	<i>Abr</i>	<i>Mai</i>	<i>Jun</i>	<i>Ago</i>	<i>Set</i>	<i>Out</i>	<i>Nov</i>	<i>Dez</i>
<i>Matemática</i>	24	24	30	24	30	24	20	28	24
<i>Química</i>	20	18	24	18	20	18	18	28	20
<i>Física</i>	20	18	20	18	18	24	18	20	36
<i>Biologia</i>	24	18	20	20	18	18	18	24	36
<i>Português</i>	16	16	16	16	14	16	16	16	16
<i>Línguas</i>	8	8	8	8	6	8	8	8	8
<i>Geografia</i>	8	10	8	10	6	10	8	10	8
<i>História</i>	10	8	10	8	8	8	10	8	10

E a seguinte distribuição de trabalhos (T) e provas (P) no segundo grau:

TABELA 2.35 DISTRIBUIÇÃO DAS PROVAS

	<i>Mar</i>	<i>Abr</i>	<i>Mai</i>	<i>Jun</i>	<i>Ago</i>	<i>Set</i>	<i>Out</i>	<i>Nov</i>	<i>Dez</i>
<i>Matemática</i>	P	P	P	P	2P	T	P	P	P
<i>Química</i>	T	P	P	T	2P	T	P	P	P
<i>Física</i>	P	T	P	P	2P	P	T	P	P
<i>Biologia</i>	T	P	P	T	2P	T	P	P	P
<i>Português</i>	P	T	P	P	2P	P	T	P	P
<i>Línguas</i>	P	P	P	P	P	P	P	P	P
<i>Geografia</i>	T	P	P	T	P	T	P	T	P
<i>História</i>	P	T	T	P	P	P	T	P	T

Cada prova exige uma preparação em estudo de 10 horas. Cada trabalho escolar consome 4 horas de esforço. Considera-se o mês com quatro semanas de sete dias e que o aluno dorme 8 horas por dia, não estuda no domingo, gasta uma hora por dia em deslocamentos e duas com as refeições. Sabe-se ainda que é desejável que o aluno tenha pelo menos 20 horas de lazer por mês e 250 horas ao longo dos nove meses de preparação, além do descanso do dia de domingo e das férias de julho, para evitar a estafa. Formular o problema de maximizar o aproveitamento do estudo, levando-se em conta que a carga horária de estudo despendida em um mês em que o descanso é igual a 20 horas possui apenas 50% da eficiência dos meses em que o descanso é maior ou igual a 40 horas e 70% dos meses em que esse valor varia entre 20 e 40 horas.

14 – O Problema da Distribuição de Facilidades



Cenário 1

Uma companhia possui duas fábricas, uma em São Paulo e outra em Curitiba. A empresa possui também quatro depósitos que facilitam a distribuição de seus produtos: um em Manaus e os outros em Natal, Rio de Janeiro e Belo Horizonte. A companhia vende seus produtos no atacado nas cidades em que possui fábricas ou depósitos e em mais seis outras praças. A distribuição dos custos unitários origem x destino, incluído o transporte e as taxas está resumida na tabela abaixo (custos em R\$10 por tonelada):

TABELA 2.36 CUSTO DE OPERAÇÃO DOS DEPÓSITOS

<i>Destino</i> \ <i>Origem</i>	<i>Fábricas</i>			<i>Depósitos</i>		
	<i>SP</i>	<i>CRT</i>	<i>MA</i>	<i>RJ</i>	<i>NAT</i>	<i>BH</i>
<i>Manaus</i>	1,5	–				
<i>Rio</i>	0,5	0,3				
<i>Natal</i>	1,0	0,5				
<i>Belo Horizonte</i>	0,2	0,2				
<i>P₁</i>	1,0	2,9	–	1,0	–	–
<i>P₂</i>	2,5	–	1,0	3,5	2,5	–
<i>P₃</i>	1,5	–	0,5	0,5	2,0	0,2
<i>P₄</i>	2,5	–	1,5	1,0	1,5	1,0
<i>P₅</i>	–	3,5	–	2,0	0,5	1,5
<i>P₆</i>	1,0	–	1,0	–	1,5	1,5

Certos clientes expressam sua preferência em serem atendidos da seguinte forma:

TABELA 2.37 EXIGÊNCIAS DOS CLIENTES

<i>Origem Cliente</i>	<i>SP</i>	<i>CRT</i>	<i>RJ</i>	<i>NAT</i>
<i>P₁</i>	x			
<i>P₂</i>		x		
<i>P₃</i>			x	
<i>P₄</i>			x	
<i>P₅</i>				x
<i>P₆</i>				x

Cada fábrica possui uma capacidade mensal de produção e os depósitos um limite de armazenagem do seguinte modo:

São Paulo: 450.000 toneladas
 Curitiba: 250.000 toneladas
 Manaus: 200.000 toneladas
 Rio de Janeiro: 100.000 toneladas
 Natal: 80.000 toneladas
 Belo Horizonte: 150.000 toneladas

Cada praça demanda o seguinte volume de produção:

São Paulo: 50.000 toneladas

Curitiba: 20.000 toneladas

Manaus: 100.000 ton

Rio de Janeiro: 60.000 ton

Natal: 50.000 ton

Belo Horizonte: 30.000 ton

P_1 : 50.000 ton; P_2 : 10.000 ton;

P_3 : 40.000 ton; P_4 : 35.000 ton;

P_5 : 60.000 ton; P_6 : 80.000 ton

1. Formular o modelo que minimiza o custo total da distribuição da produção.

2. Considerando que exista um custo de transbordo e estocagem, não incluído nas tabelas anteriores, C_i , $i = 1, 2, 3, 4$, associado a cada depósito, reformular o problema.

3. Qual o efeito imediato de uma possível ampliação na capacidade dos depósitos? Considerar que a política de investimento para a ampliação dos depósitos seria motivo de um outro estudo mais aprofundado.

4. Seria possível atender a todas as preferências dos clientes? Essas preferências poderiam ser incluídas de alguma forma no modelo de programação?

Cenário 2

No problema anterior é possível abrir um novo depósito em Salvador e um em Recife ou ainda ampliar os depósitos de Natal e do Rio de Janeiro. A Tabela 2.38 resume os impactos, apresentando os custos finais das alternativas.

TABELA 2.38 ESTUDO DE MODIFICAÇÃO DO SISTEMA

Consequência da Ampliação x Novos Depósitos		
	<i>Custos</i> (R\$10 ⁴)	<i>Capacidade</i> (1.000 ton)
Salvador	12	40.000
Recife	10	30.000
Natal	5	40.000
Rio de Janeiro	3	35.000
Economia Gerada com o Fechamento (R\$10 ⁴)		
Manaus	10	
Belo Horizonte	5	

Como a empresa não admite operar mais de cinco depósitos, poderão ser fechados os depósitos de Belo Horizonte ou Manaus, alcançando-se alguma economia face a esse fechamento. A distribuição dos custos operacionais dos novos depósitos e suas capacidades de atendimento estão resumidas na tabela a seguir:

TABELA 2.39 DISTRIBUIÇÃO DE CUSTOS OPERACIONAIS

<i>Origem</i>	<i>Fábricas</i>		<i>Depósitos</i>	
<i>Destino</i>	SP	CRT	SAL	REC
<i>Salvador</i>	1,0	1,5		
<i>Recife</i>	1,5	2,0		
P_1			–	–
P_2			1,5	2,5
P_3			0,3	0,5
P_4			0,5	0,5
P_5			0,0	0,3
P_6			0,5	0,0
<i>Manaus</i>			2,5	2,0
<i>Belo Horizonte</i>			2,0	2,5

Formule o programa de Programação Matemática para otimizar a tomada de decisão na escolha da política de localização de recursos.

3

MÉTODO SIMPLEX

3.1 – SOLUÇÃO EXATA PARA OS MODELOS DE PL

O modelo de programação linear reduz um sistema real a um conjunto de equações ou inequações em que pretendemos otimizar uma função objetivo. O conjunto de equações deverá ser, em princípio, um conjunto indeterminado, de forma que o número das soluções ditas “viáveis” é infinito. Mesmo sabendo que alguma solução ótima será encontrada em um ponto extremo do conjunto das soluções viáveis, nosso trabalho não será pequeno se desejarmos determiná-lo. Infelizmente, esse número de pontos extremos ou vértices pode ser muito grande, em número exponencial, em relação às variáveis. Podemos resumir em dois blocos as dificuldades que deverão ser vencidas por quem deseja encontrar a melhor solução possível para um sistema indeterminado de equações lineares, segundo um critério qualquer.

- ✓ Como obter soluções viáveis básicas do sistema de equações.
- ✓ Como evitar o teste de todas as soluções viáveis básicas possíveis para garantir a otimização do sistema.

É nesse contexto que o *algoritmo simplex* destaca-se como uma das grandes contribuições à Programação Matemática deste século. Trata-se de um algoritmo geral *extremamente eficiente* para a solução de sistemas lineares e adaptável ao cálculo computacional (apesar de algumas dificuldades clássicas), e cuja compreensão funcional embasará vários outros métodos. O estudo desse algoritmo é indispensável para o profissional que deseja dominar as técnicas quantitativas de *análise e solução de problemas* em um contexto razoavelmente avançado.

3.2 – FUNDAMENTOS TEÓRICOS DO SIMPLEX

O simplex é um algoritmo. Genericamente, podemos entender por algoritmo qualquer estratégia para solucionar problemas; contudo, seremos mais precisos, reservando para a palavra algoritmo um conceito diferente de procedimento. Diremos que:

Um procedimento é uma sequência finita de instruções.

e que:

Um algoritmo é um procedimento que termina em um número finito de operações (passos).

O simplex é um algoritmo que se utiliza de um ferramental baseado na Álgebra Linear para determinar, por um método iterativo, a solução ótima de um Problema de Programação Linear (PPL). Sua concepção básica é simples e, por isso mesmo, eficiente. Em linhas bastante gerais, o algoritmo parte de uma solução viável do sistema de equações que constituem as restrições do PPL, solução essa normalmente extrema (vértice). A partir dessa solução inicial vai identificando novas soluções viáveis de valor igual ou melhor que a corrente. O algoritmo, portanto, possui um critério de escolha que permite encontrar sempre novos e melhores vértices da envoltória convexa do problema, e um outro critério que consegue determinar se o vértice escolhido é ou não um vértice ótimo.

Antes de desenvolvermos o algoritmo propriamente dito é necessário recordar alguns conceitos matemáticos que serão utilizados pelo método.

Definição 3.1:

Uma base de uma matriz A ($m \times n$) é uma matriz quadrada de m vetores coluna linearmente independentes em \mathfrak{R}^m . As variáveis associadas a essas colunas denominaremos *variáveis básicas*.

Podemos decompor o vetor das variáveis x em (ver Figura 3.1):

$x = (x_B, x_R)$, onde:

x_B representa o vetor das variáveis básicas de m componentes e

x_R é o vetor das restantes $n - m$ variáveis não básicas.

I é o conjunto dos índices correspondentes às variáveis básicas.

J é o índice das variáveis não básicas.

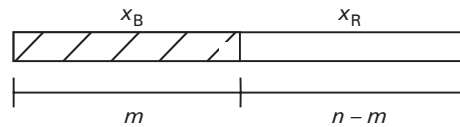


FIGURA 3.1 Decomposição do vetor de variáveis.

Como podemos solucionar o conjunto de equações $m \times m$ somente em função das variáveis básicas, temos: $x = (x_B, 0)$. Se estendermos o raciocínio à matriz A , podemos igualmente dividi-la em uma matriz $m \times m$, matriz que denominaremos por B , de $\text{rank} = m$ e outra $m \times (n - m)$, matriz que denominaremos R .

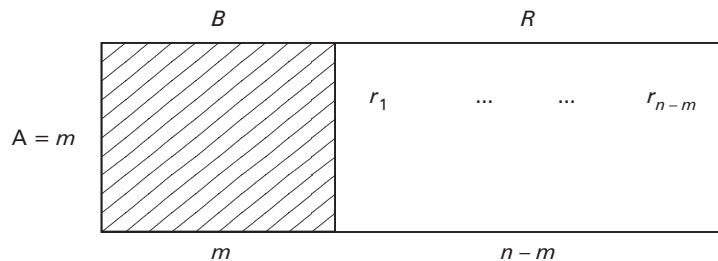


FIGURA 3.2 Decomposição da matriz A .

Em relação à divisão proposta pela Figura 3.2 podemos afirmar que:

Definição 3.2:

Seja B uma base associada à matriz A . O vetor composto de $x_B = B^{-1}b$ e $x_R = 0$ é chamado de *solução básica*.

E ainda que:

Definição 3.3:

Uma Solução Básica sem componentes negativas é denominada *solução básica viável*.

$$\bar{x} = \begin{array}{c} B^{-1}b \\ \boxed{} \\ m \end{array} \quad \begin{array}{c} x_R \\ \boxed{00\dots00} \\ n-m \end{array}$$

FIGURA 3.3 Decomposição do vetor x .

Para fixar as definições anteriores; seja o seguinte sistema de equações que constitui um conjunto de restrições de algum problema de Programação Linear:

$$\begin{aligned} x_1 + 2x_2 &\leq 4 \\ x_2 &\leq 1 \end{aligned}$$

Passando para a forma padrão de representação temos:

$$\begin{aligned} x_1 + 2x_2 + x_3 &= 4 \\ x_2 + x_4 &= 1 \end{aligned}$$

Nesse pequeno exemplo a divisão da matriz A poderia ser:

$$A = \begin{bmatrix} +1 & +2 & +1 & 0 \\ 0 & +1 & 0 & +1 \end{bmatrix} B = \begin{bmatrix} +1 & 0 \\ 0 & +1 \end{bmatrix}$$

Nesse caso $x_B = (x_3, x_4)$ e $x_R = (x_1, x_2)$

Definição 3.4:

O conjunto $C = \{x \text{ tal que } Ax = b, x \geq 0\}$ denomina-se *conjunto de soluções viáveis*.

Da Definição 3.4 decorrem os seguintes teoremas e corolários importantes para o entendimento do método simplex:

Teorema 3.1:

O conjunto C das soluções viáveis de um modelo de programação *linear* é um conjunto *convexo*.

Demonstração:

Seja C o conjunto formado pelos pontos x tais que:

$$\begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned}$$

Se C é convexo então (ver definição de conjunto convexo no anexo) para qualquer conjunto composto por dois pontos distintos x_1, x_2 pertencentes a C , a combinação linear convexa desses pontos também pertence a C , o que é equivalente a dizer que:

$$\{x_1, x_2\} \in C \Rightarrow \begin{cases} x = \alpha x_1 + (1 - \alpha)x_2 \in C \\ 0 \leq \alpha \leq 1 \end{cases}$$

Sejam duas soluções viáveis de C , x_1, x_2 , tais que $x_1 \neq x_2$, então:

$$\begin{aligned} Ax_1 &= b & Ax_2 &= b \\ x_1 &\geq 0 & x_2 &\geq 0 \end{aligned}$$

e seja:

$$\begin{aligned} x &= \alpha x_1 + (1 - \alpha)x_2 \\ 0 &\leq \alpha \leq 1 \end{aligned}$$

Então:

$$\begin{aligned} Ax &= A[\alpha x_1 + (1 - \alpha)x_2] = \\ &= \alpha Ax_1 + (1 - \alpha)Ax_2 = \\ &= \alpha b + (1 - \alpha)b = b \end{aligned}$$

e $x \geq 0$, uma vez que:

$$x = \alpha x_1 + (1 - \alpha)x_2 \geq 0$$

e

$$x_1 \geq 0, x_2 \geq 0 \text{ e } 0 \leq \alpha \leq 1$$

Teorema 3.2:

Toda *solução básica* viável do sistema $Ax = b$ é um *ponto extremo* do conjunto de soluções viáveis, ou seja, um extremo do conjunto C .

Demonstração:

Seja C o conjunto formado pelos pontos x tais que:

$$\begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned}$$

Seja, ainda, uma solução viável qualquer x , de dimensão n , na qual, sem perda de generalidade, as variáveis básicas são as m primeiras:

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_m \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{ com todos as componentes } x_i \geq 0$$

Suponhamos, por absurdo, que x não seja um ponto extremo do conjunto convexo C , definido anteriormente. Então x pode ser obtido por uma combinação convexa de outros dois pontos distintos desse mesmo conjunto. Chamando de y e z esses dois pontos, temos:

$$\begin{aligned} x &= \alpha y + (1 - \alpha)z \\ 0 &\leq \alpha \leq 1 \end{aligned}$$

Como y e z pertencem ao conjunto C , as seguintes relações de pertinência são válidas:

$$\begin{aligned} Ay &= b & e & & Az &= b \\ y &\geq 0 & & & z &\geq 0 \end{aligned}$$

A relação $x = \alpha y + (1 - \alpha)z$, colocada em termos das coordenadas de cada um dos três vetores, fornece as seguintes relações:

$$\begin{aligned} x_1 &= \alpha y_1 + (1 - \alpha)z_1 \\ x_2 &= \alpha y_2 + (1 - \alpha)z_2 \\ &\dots\dots\dots \\ x_m &= \alpha y_m + (1 - \alpha)z_m \\ 0 &= \alpha y_{m+1} + (1 - \alpha)z_{m+1} \\ &\dots\dots\dots \\ 0 &= \alpha y_n + (1 - \alpha)z_n \end{aligned}$$

Devido às relações $0 \leq \alpha \leq 1$, $y \geq 0$ e $z \geq 0$ as últimas $(n - m)$ relações do conjunto acima descrito só podem ser satisfeitas em um dos seguintes casos:

1. $0 < \alpha < 1$ e $y_{m+i} = z_{m+i} = 0$ para $i = 1, \dots, n - m$.

Nesse caso teríamos $x = y = z$, pois tanto y quanto z são soluções básicas do sistema em análise, calculados com as mesmas variáveis básicas.

2. $\alpha = 0$ e $z_{m+i} = 0$ para $i = 1, \dots, n - m$.

Por raciocínio análogo ao caso anterior, deduzimos que $x = z$. Além disso, como $\alpha = 0$, segue-se que $x = y = z$.

3. $\alpha = 1$ e $y_{m+i} = 0$ para $i = 1, \dots, n - m$.

Por razões análogas conclui-se que $x = y = z$.

Dessa forma, demonstra-se que não existem soluções viáveis y e z , distintas da solução básica x que satisfaçam à relação $x = \alpha y + (1 - \alpha)z$. Por contradição com a hipótese inicial, demonstra-se, então, que x é um ponto extremo do conjunto convexo C .

A relação entre pontos extremos de C e as soluções viáveis básicas de um PPL é uma correspondência biunívoca, como mostra a Figura 3.4.

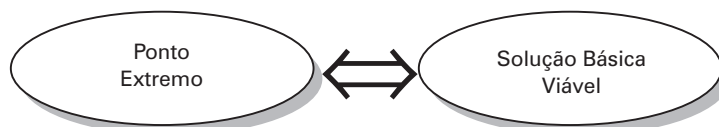


FIGURA 3.4 Correspondência solução básica x ponto extremo de um PPL.

O Teorema 3.2 demonstra a correspondência no sentido de solução básica viável para ponto extremo, e o Teorema 3.3 completa a associação.

Teorema 3.3:

Um ponto x é extremo em um conjunto de soluções viáveis de um problema de programação linear se e somente se $x \geq 0$ for uma solução básica do sistema de equações lineares $Ax = b$

Observamos também que um ponto x é extremo em um conjunto de soluções viáveis de um PPL se e somente se $x \geq 0$ for uma solução básica do sistema de equações lineares $Ax = b$.

Corolário 3.1:

O conjunto dos pontos extremos de um conjunto de soluções viáveis é finito e limitado em C_n^m .

Corolário 3.2:

Se existe uma solução viável, então existe uma solução básica viável.

Finalmente, o último respaldo necessário da teoria para entendermos e justificarmos a estratégia do método simplex diz respeito ao valor ótimo alcançado pela função objetivo. Qual seria a associação entre os pontos extremos e o valor da função objetivo? Existe alguma garantia de que o valor ótimo seja alcançado nos pontos extremos de C ? O Teorema 3.4 esclarece a questão colocada.

Teorema 3.4:

1. Se uma função objetivo possui um máximo ou um mínimo finito, então pelo menos uma solução ótima é um ponto extremo do conjunto convexo C do Teorema 1.
 2. Se a função objetivo assume o máximo ou o mínimo em mais de um ponto extremo, então ela toma o mesmo valor para qualquer combinação convexa desses pontos.
-

A Figura 3.5 resume simplificadamente os aspectos teóricos que garantem o bom funcionamento do método simplex. O método vai, basicamente, experimentar uma sequência de soluções básicas viáveis na busca do valor ótimo para a função objetivo, e isso é um dos aspectos que permitem ao algoritmo um bom desempenho e a garantia de sua convergência em número finito de passos.



FIGURA 3.5 Correspondência com a solução ótima.

3.3 – O ALGORITMO PRIMAL SIMPLEX

3.3.1 – A Visão Geral do Algoritmo

O algoritmo simplex descreve uma sequência de passos para a solução de sistemas de equações lineares sujeitos a uma função objetivo. Basicamente, ele dispõe sobre três situações:

1. O método de inversão da matriz básica $m \times m$ deduzida a partir de A , uma matriz de restrições $m \times n$.
2. As condições de troca de variáveis dentro da matriz básica, para que exista garantia de melhoria da solução ao longo do desenvolvimento dos passos do algoritmo.
3. As regras de parada do algoritmo e a interpretação dessa situação final.

Eventualmente, o algoritmo pode ser adaptado para promover a escolha da base viável inicial, ou solução viável de partida. Contudo, a essência do método desconhece o problema da escolha de uma base viável inicial.

A discussão do primeiro aspecto envolvido no algoritmo, ou seja, o método de inversão da matriz básica, é bastante evidente na apresentação dos “quadros” do simplex e de suas operações de “pivoteamento”. O método normalmente sugerido na literatura é o das *operações elementares*. Essa técnica permite que a cada passo do algoritmo o esforço de inversão já despendido em iterações anteriores seja completamente aproveitado. É importante que se entenda que não existe, de fato, um compromisso do algoritmo com um método de inversão específico para a matriz básica. Em última análise, o método simplex não obriga que a matriz tenha de ser invertida por um método de pivoteamento, apesar de que todo o seu raciocínio seja aplicado tradicionalmente junto com essa técnica.

O segundo ponto é abordado por um critério bastante simples que envolve o cálculo da possível contribuição para o acréscimo ou decréscimo da função objetivo (conforme o caso: maximização ou minimização) com a possível entrada na base de uma variável não básica. O critério aponta a escolha da variável de maior contribuição imediata. A eficiência do método simplex e seu extraordinário poder de funcionar na prática estão associados ao critério adotado nesse cálculo.

O terceiro tema diz respeito ao teste de parada que inclui a identificação das condições em que não existe mais a possibilidade de que uma troca de variáveis na base possa melhorar o critério de otimização, ou ainda situações em que é identificado um comportamento patológico de crescimento indefinido ou de inviabilidade do problema.

3.3.2 – Um Exemplo de Solução de um Programa de Programação Linear

Um pequena metalúrgica deseja maximizar sua receita com a venda de dois tipos de finas fitas de aço que se diferenciam em qualidade no acabamento de corte. As fitas são produzidas a partir do corte de bobinas de grande largura. Existem duas máquinas em operação. Uma das máquinas é mais antiga e permite o corte diário de 4.000 m de fita. A outra, mais nova, corta até 6.000 m. A venda das chapas no mercado varia com a qualidade de cada uma. Fitas produzidas na máquina antiga permitem um lucro de 3 u.m. por mil metros de produção. Fitas cortadas na máquina moderna produzem um lucro de 5 u.m. por mil metros de produção. Cada mil metros de fita cortada na máquina antiga consome 3 homens \times hora de mão de obra. Na máquina moderna são gastos apenas 2 homens \times hora. Diariamente são disponíveis 18 homens \times hora para a operação de ambas as máquinas. Determinar a produção que otimiza o lucro da metalúrgica.

A modelagem deste problema de programação linear, que denominaremos “das fitas” ou Fita_1, é simples e pode ser realizada da seguinte forma:

1. Escolha da variável de decisão

$x_i \equiv$ comprimento da fita cortada na máquina velha ($i = 1$) e comprimento da fita cortada na máquina nova ($i = 2$).

2. Elaboração da função objetivo

$z = \text{Maximizar } \{f(x) = 3x_1 + 5x_2\}$ – número total de unidades monetárias obtidas com a venda das fitas em seus diversos comprimentos.

3. Formulação das restrições tecnológicas

Considerando a unidade de comprimento das fitas igual a 1.000 m temos:

a) Restrição associada à disponibilidade de mão de obra:

$$3x_1 + 2x_2 \leq 18$$

b) Restrição do comprimento das chapas fabricadas na máquina velha:

$$x_1 \leq 4$$

c) Restrição do comprimento das chapas fabricadas na máquina nova:

$$x_2 \leq 6$$

4. Restrições de não negatividade

$$x_1 \geq 0, x_2 \geq 0.$$

Podemos resumir o modelo de Fita_1 da seguinte forma:

$$\text{Maximizar } z = 3x_1 + 5x_2$$

Sujeito a:

$$x_1 \leq 4$$

$$x_2 \leq 6$$

$$3x_1 + 2x_2 \leq 18$$

$$x_1 \geq 0, x_2 \geq 0$$

Como nos foi solicitado que, além do modelo, fosse exibido o melhor esquema para a produção, vamos buscar a solução do problema. Examinado o modelo de Fita_1 encontramos a necessidade de maximizar uma função linear sujeita a um conjunto de três inequações. Colocado sob a forma de equações, o sistema de restrições é matricial com m linhas e n colunas, onde $n > m$. Supondo que esse sistema de equações possua alguma solução, então essa solução poderá ser obtida através de um determinante da ordem de m , correspondendo a uma certa submatriz $m \times m$ derivada da matriz de restrições de Fita_1. Nesse caso, deduzimos que a matriz de restrições de Fita_1 representa um conjunto de equações indeterminado, ou seja, possuindo infinitas soluções aceitáveis. Resumindo: se Fita_1 possui solução, então teremos de buscá-la maximizando a função z sobre um conjunto infinito de soluções viáveis. Vamos exemplificar a natureza da dificuldade do problema proposto através da representação gráfica de Fita_1. Representaremos a variável x_1 no eixo dos x e a variável x_2 no eixo dos y .

Como podemos observar na Figura 3.6 os pontos O, A, B, C, D e E são exemplos de soluções para o conjunto de restrições de Fita_1. A Tabela 3.1 mostra o valor assumido da função objetivo z em cada um dos pontos selecionados:

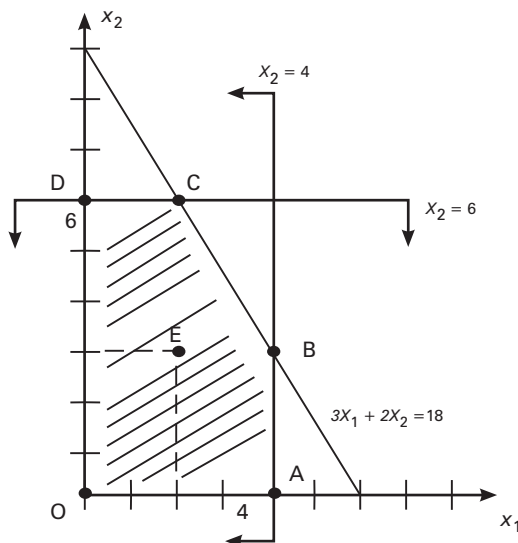
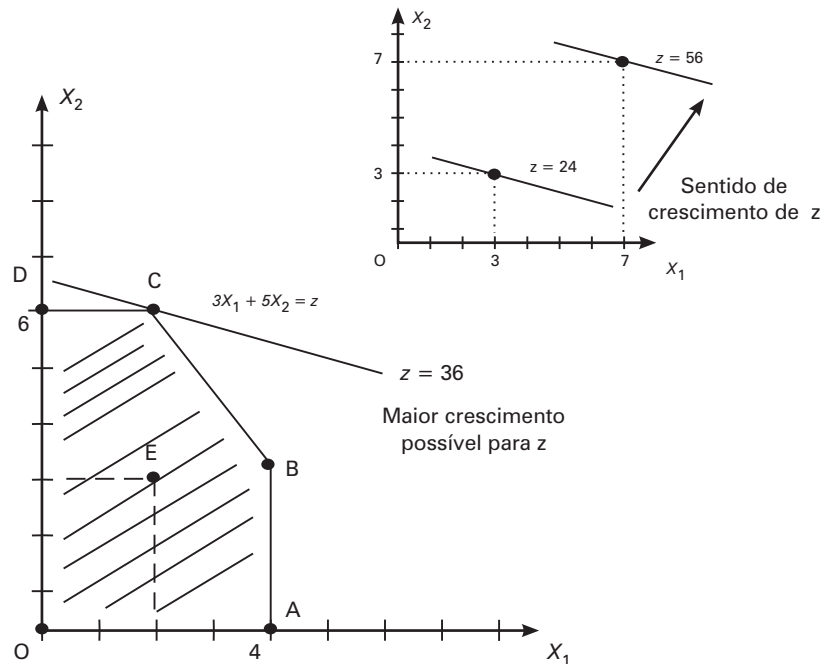


FIGURA 3.6 Representação gráfica de Fita_1.

TABELA 3.1

Pontos Examinados	Coordenadas (x_1 , x_2)	Valor da Função $z = 3x_1 + 5x_2$
O	(0, 0)	0
A	(4, 0)	12
B	(4, 3)	27
C	(2, 6)	36
D	(0, 6)	30
E	(2, 3)	21

Dentre os pontos examinados, a solução que leva ao maior valor de z é a que corresponde ao vértice C. Seria esse o valor máximo da função z ? Obviamente, não podemos examinar os infinitos pontos do poliedro OABCD, apesar de todos eles serem soluções possíveis ou viáveis. Felizmente, o esquema da Figura 3.5 nos garante que a solução ótima de Fita_1 está em um dos pontos extremos do seu poliedro de soluções viáveis. De fato, não faz qualquer sentido o exame de um ponto interno como o ponto E (apenas sentido didático), o ponto ótimo de z jamais ocorrerá ali. A Figura 3.7 é útil para que possamos visualizar graficamente o processo de otimização de z sobre o conjunto das soluções viáveis. No atual exemplo bidimensional, ele é representado pelo deslocamento de uma reta. No caso geral teremos um hiperplano representando uma curva de nível da função objetivo z que passe por algum ponto do poliedro de restrições.


FIGURA 3.7 Crescimento da função z em Fita_1.

Examinando a equação geral das retas associadas às curvas de nível da função objetivo, percebemos que cada valor numérico de z corresponde ao termo independente de alguma reta perpendicular à direção do gradiente, configurada pelos coeficientes dos lucros no modelo. Especificamente, para o exemplo em pauta, maximizar z corresponde a deslocar a direção proporcionada pela reta $3x_1 + 5x_2$ até obter o máximo valor para seu termo independente. É claro que a reta z deve possuir pelo menos um ponto comum com a figura geométrica correspondente às restrições ou não será capaz de atendê-las. Esse ponto em

comum é a solução que procuramos. É claro que nem todos os sentidos de deslocamento serão úteis para maximizar o termo independente de z . No caso em exame, a melhor direção de deslocamento é a do gradiente que corresponde a um sentido ortogonal ao suporte da reta z . A direção de máximo crescimento da função z não depende do formato ou do número de equações da matriz de restrições do modelo de programação linear. O coeficiente angular de $z = c_1 x_1 + c_2 x_2$ é independente do valor de z e igual a c_1/c_2 . A direção de maior crescimento de z corresponde a uma reta ortogonal a z , sendo, portanto, igual a c_2/c_1 . Nesse caso, otimizar z sobre um conjunto de restrições lineares resume-se a encontrar, na direção do maior crescimento da função, o mais afastado ponto do universo de soluções viáveis configurado pela matriz de restrições. A Figura 3.7 nos sugere visualmente que a melhor solução para Fita_1 estará localizada nas bordas do polígono OABCD. O Teorema 3.4 nos garante que esse nosso “sentimento” é sempre verdadeiro, já que o polígono OABCD é convexo!

Como afirma a segunda parte do Teorema 4, é possível que a direção de busca encontre infinitos pontos que levariam z a assumir um valor ótimo. Se o lucro com a venda das fitas da máquina nova de nosso exemplo fosse 2 unidades monetárias e não 5, um fato interessante ocorreria. Passando a nova função z a ser $3x_1 + 2x_2$, sua direção tornar-se-ia paralela à restrição imposta pela mão de obra. O polígono das soluções viáveis não teria sua forma alterada, uma vez que as restrições não foram mudadas, mas a nova direção de crescimento da função objetivo encontraria simultaneamente os pontos extremos C e B. Para este exemplo, qualquer ponto da face CB é uma solução ótima, conforme mostra a Figura 3.8. Em termos do algoritmo simplex isso é caracterizado, como veremos oportunamente, pela existência de variáveis não básicas $j \in J$ com valor de seus custos reduzidos iguais a zero ($z_j - c_j = 0$) na solução ótima.

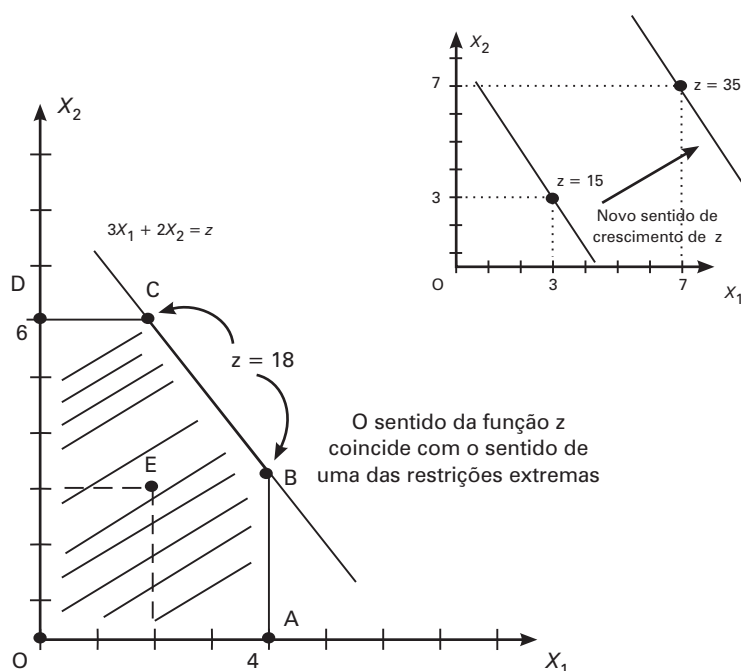


FIGURA 3.8 Caso de infinitas soluções para Fita_1.

Uma outra ocorrência possível para o conjunto das soluções viáveis de um PPL diz respeito à sua forma. Eventualmente a matriz de restrições pode não corresponder a uma figura fechada. A Figura 3.9 mostra a situação em que o conjunto das soluções viáveis é não limitado. Apesar de representar um caso peculiar, nem sempre essa característica introduzirá qualquer dificuldade para o processo de solução. Examinando o exemplo proposto, verificamos facilmente que, se a direção de “crescimento” da função z tiver o sentido contrário do mostrado na figura, apontando conseqüentemente para a região do ponto B, a solução desse problema será limitada. Para a região do ponto B o conjunto de restrições é limitado, uma

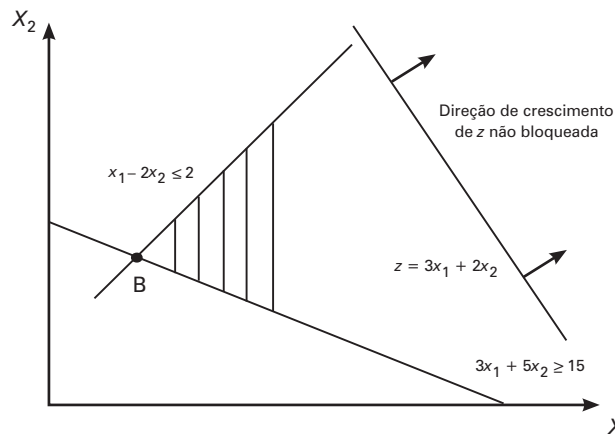


FIGURA 3.9 PPL ilimitado.

vez que a direção de crescimento de z acabará bloqueada pelo encontro de faces do poliedro (duas restrições para o caso bidimensional) ou um ponto extremo.

Uma outra possibilidade topológica para o conjunto das soluções viáveis é a do conjunto vazio. De um modo geral, nada garante que um problema corretamente formulado tenha uma solução. Nesse caso, o PPL é denominado *inviável* ou *impossível*. Na interpretação geométrica, tal fato é caracterizado por uma *região viável vazia*. A Figura 3.10 mostra um exemplo de um PPL inviável.

Infelizmente, os problemas de programação linear não admitem, na esmagadora maioria dos casos, a possibilidade de solução gráfica, já que esta só é praticável quando o problema possui duas variáveis. Uma forma de encontrar eficientemente o ponto extremo de C em que a solução ótima de z ocorrerá tem sido motivo de grande esforço e pesquisa. É nesse sentido que, há mais de meio século, a contribuição do algoritmo simplex é tão importante.

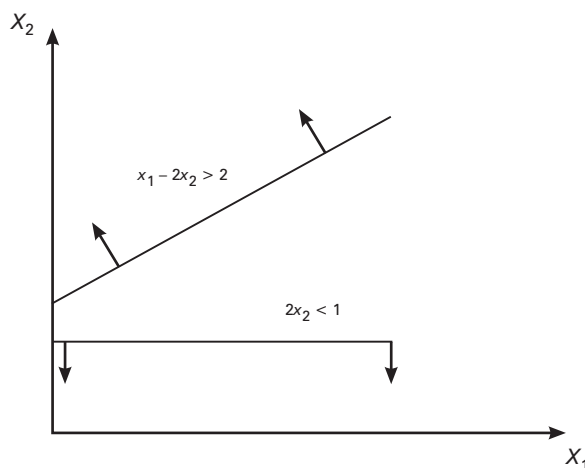


FIGURA 3.10 PPL inviável.

3.3.3 – O Critério de Melhoria de uma Solução Básica

Considerando que exista uma solução básica viável para o seguinte problema de programação linear:

$$\begin{aligned} &\text{Minimize } z = cx \\ &\text{Sujeito a: } Ax = b \\ &\quad x \geq 0 \end{aligned}$$

onde A é uma matriz $m \times n$, com $\text{rank } m$ (todas as linhas linearmente independentes), podemos decompor o vetor c em suas componentes básicas e não básicas, $c = (c_B, c_R)$, e supor que a solução básica viável existente seja representada por um vetor $\bar{x} = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$ cujo valor associado é dado pela seguinte expressão:

$$z_0 = c \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix} = (c_B, c_R) \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix} = c_B B^{-1}b$$

Pode-se escrever o vetor x , em função das variáveis básicas e não básicas da seguinte forma:

$$\bar{x} = \begin{pmatrix} x_B \\ x_R \end{pmatrix} \text{ e } b = Ax = Bx_B + Rx_R.$$

Multiplicando-se por B^{-1} a expressão de z_0 tem-se:

$$\begin{aligned} x_B &= B^{-1}b - B^{-1}Rx_R \\ &= B^{-1}b - \sum_{j \in J} B^{-1}a_jx_j \end{aligned}$$

dessa forma pode-se reescrever a expressão $z = cx$ como se segue:

$$\begin{aligned} z &= cx \\ &= c_Bx_B + c_Rx_R \\ &= c_B \left(B^{-1}b - \sum_{j \in J} B^{-1}a_jx_j \right) + \sum_{j \in J} c_jx_j \\ &= z_0 - \sum_{j \in J} (z_j - c_j)x_j \end{aligned} \tag{3.1}$$

onde $z_j = c_B B^{-1}a_j$ para cada variável não básica.

A Equação (3.1) mostra a possibilidade do estabelecimento de um critério de verificação de otimalidade ou de melhoria de uma solução básica. Quando o valor do termo $z_j - c_j$ é estritamente maior que zero existe a possibilidade de, com a entrada da variável de índice j na base, reduzir o valor da função objetivo em $(z_j - c_j)x_j$, desde que essa variável possa assumir um valor positivo. O termo $z_j - c_j$ pode ser interpretado como o coeficiente de utilidade relativa da variável x_j , e seu valor negativo $c_j - z_j$ é comumente denominado custo reduzido. Se denominarmos k o índice dessa variável não básica teremos:

$$z = z_0 - (z_k - c_k)x_k \tag{3.2}$$

Examinando (3.2) é fácil concluir que, de uma forma geral, para o processo de otimização será interessante que a variável x_k seja incrementada ao máximo. Com o crescimento de x_k , o valor de z diminui na nova solução básica, proporcionalmente ao valor do custo reduzido associado. Como sabemos que:

$$x_B = B^{-1}b - B^{-1}a_kx_k = \bar{b} - y_kx_k \tag{3.3}$$

onde

$$y_k = B^{-1}a_k \text{ e } \bar{b} = B^{-1}b \tag{3.4}$$

Denotando as componentes do vetor x_B e \bar{b} respectivamente por $x_{B_1}, x_{B_2}, \dots, x_{B_m}$ e $\bar{b}_1, \bar{b}_2, \dots, \bar{b}_m$ temos, finalmente, a seguinte expressão:

$$\begin{bmatrix} x_{B_1} \\ x_{B_2} \\ \vdots \\ x_{B_m} \end{bmatrix} = \begin{bmatrix} \bar{b}_1 \\ \bar{b}_2 \\ \vdots \\ \bar{b}_m \end{bmatrix} - \begin{bmatrix} y_{1k} \\ y_{2k} \\ \vdots \\ y_{mk} \end{bmatrix} x_k \quad (3.5)$$

A Expressão (3.5) nos mostra que se existir algum elemento de y_{ik} , $y_{ik} \leq 0$, então o x_{B_i} associado pode crescer indefinidamente com o crescimento de x_k . Se existir $y_{ik} > 0$, então x_{B_i} decresce com o incremento de x_k . Para satisfazer as condições de não negatividade de uma solução básica viável, a nova variável x_k só poderá crescer até que a primeira componente x_{B_i} seja reduzida a zero, o que corresponde ao mínimo entre todos os \bar{b}_i / y_{ik} para os valores positivos de y_{ik} , ou:

$$\frac{\bar{b}_s}{y_{sk}} = \min_{1 \leq i \leq m} \left\{ \frac{\bar{b}_i}{y_{ik}} : y_{ik} > 0 \right\} \Rightarrow x_s \quad (3.6)$$

Note que para garantir a independência linear da coluna k com as demais colunas existentes na base é indispensável que $y_{sk} \neq 0$. Pelo critério sugerido em (3.2), a variável x_k seria a que entraria na base melhorando o valor da função objetivo, e a variável x_s , linearmente dependente de x_k , deixaria a base ao ter o seu valor numérico esgotado completamente pelo crescimento de x_k .

Exemplo da aplicação do critério de melhoria

Seja o problema da fabricação das fitas do item 3.2. Podemos reescrevê-lo na forma padrão, considerando a transformação para um PPL de minimização, da seguinte forma:

$$\begin{aligned} &\text{Minimizar } z = -3x_1 - 5x_2 \\ &\text{Sujeito a:} \\ &x_1 + x_3 = 4 \\ &x_2 + x_4 = 6 \\ &3x_1 + 2x_2 + x_5 = 18 \\ &x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{aligned}$$

Onde identificamos a matriz de restrições, de custos e dos termos independentes como se segue:

$$A = [a_1, a_2, a_3, a_4, a_5] = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 3 & 2 & 0 & 0 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} 4 \\ 6 \\ 18 \end{bmatrix} \text{ e } c = [-3, -5, 0, 0, 0]$$

Consideremos a base composta pelas variáveis x_3, x_2, x_5 , ou seja:

$$B = [a_3, a_2, a_5] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

Nesse caso as variáveis x_1 e x_4 são não básicas. $I = \{3, 2, 5\}$, $J = \{1, 4\}$
Calculando a inversa da base temos:

$$B^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}$$

Podemos com a inversa da base calcular o vetor x_B , x_R e z da seguinte forma:

$$x_B = \begin{bmatrix} x_3 \\ x_2 \\ x_5 \end{bmatrix} = B^{-1} b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 6 \\ 18 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 6 \end{bmatrix}$$

$$x_R = \begin{bmatrix} x_1 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$z_D = cx = c_B x_B + c_R x_R = [0 \ -5 \ 0] \begin{bmatrix} 4 \\ 6 \\ 6 \end{bmatrix} = -30$$

A base escolhida corresponde ao ponto D da Figura 3.6. Para melhorar a solução z_D será necessário calcular os valores $(z_j - c_j)$ para $j \in J$, ou seja:

$$\begin{aligned} z_1 - c_1 &= c_B B^{-1} a_1 - c_1 \\ &= [0 \ -5 \ 0] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix} - (-3) \\ &= 3 \end{aligned}$$

$$\begin{aligned} z_4 - c_4 &= c_B B^{-1} a_4 - c_4 \\ &= [0 \ -5 \ 0] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} - (0) \\ &= -5 \end{aligned}$$

Pelo critério adotado, a variável x_1 entrará na base saindo a variável $\frac{\bar{b}_s}{y_{sk}} = \min_{1 \leq i \leq m} \left\{ \frac{\bar{b}_i}{y_{ik}} : y_{ik} > 0 \right\}$, que no caso é x_5 .

$$\begin{array}{c} \leftarrow \begin{bmatrix} x_3 \\ x_2 \\ x_5 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 6 \end{bmatrix} / \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix} = \begin{bmatrix} 4 \\ - \\ 2 \end{bmatrix} \leftarrow \end{array}$$

A nova base será formada pelas variáveis x_3 , x_2 , x_1 , ou seja:

$$B' = [a_3, a_2, a_1] = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 2 & 3 \end{bmatrix}$$

Correspondendo à solução ótima do problema, ou o vértice C da Figura 3.6.

Podemos formalizar o processo de escolha da base inicial de cálculo, critério de troca de variáveis na base e regra de parada em um algoritmo da seguinte forma:

Inicialização:

Determinar uma solução básica inicial \bar{x}_B . Seja I o conjunto de índices das colunas de A pertencentes à base e $J = N \setminus I$ (operação de diferença de conjuntos).

Passo 1: Calcular a matriz $Y = (y_j) = (y_{sj})$, $s \in I$ e $j \in J$ e os valores $z_j - c_j$, $\forall j \in J$ como se segue:

$$Y = B^{-1} R$$

$$z_j = c_B y_j, j \in J$$

• Se $z_j - c_j \leq 0 \forall j \in J$ então a solução básica viável \bar{x}_B é ótima. **Pare!**

• Caso contrário, fazer $J_1 = \{j \in J \mid z_j - c_j > 0\}$

Passo 2:

• Se $y_i \leq 0$ para pelo menos um $j \in J_1$, não existe solução ótima finita. **Pare!**

• Caso contrário, determinar k de modo que $z_k - c_k = \max_{j \in J} \{z_j - c_j\}$

Na coluna k encontrar a relação: $\frac{\bar{b}_s}{y_{sk}} = \frac{\bar{x}_s}{y_{sk}} = \min_{1 \leq i \leq m} \left\{ \frac{\bar{b}_i}{y_{ik}} : y_{ik} > 0 \right\} \Rightarrow x_s$

Passo 3: Considere a nova base B deduzida a partir da anterior pela substituição de a_s por a_k .

$$B = (B \setminus \{a_s\} \cup \{a_k\})$$

Calcular a nova solução básica viável:

$$\bar{x}_B = B^{-1} b$$

$$z_0 = z_0 - (z_k - c_k) \frac{\bar{x}_{B_s}}{y_{sk}}$$

Atualizar:

$$R = (R \setminus \{a_k\}) \cup \{a_s\}$$

$$I = (I \setminus \{s\}) \cup \{k\}$$

$$J = (J \setminus \{k\}) \cup \{s\}$$

Voltar para o passo 1.

Quadro Geral:

QUADRO 3.3 QUADRO GERAL DO SIMPLEX

		$x_1 \dots x_k \dots x_s \quad x_{s+1} \dots x_{s+r} \dots x_n$	
	z	$z_1 - c_1 \dots z_k - c_k \dots z_s - c_s \quad z_{s+1} - c_{s+1} \dots z_{s+r} - c_{s+r} \dots z_n - c_n$	
x_{B_1}	\bar{b}_1	$y_{11} \dots y_{1k} \dots y_{1s}$	B^{-1}
\vdots	\vdots	$\vdots \quad \vdots \quad \vdots$	
x_{B_r}	\bar{b}_r	$y_{r1} \dots y_{rk} \dots y_{rs}$	
\vdots	\vdots	$\vdots \quad \vdots \quad \vdots$	
x_{B_m}	\bar{b}_m	$y_{m1} \dots y_{mk} \dots y_{ms}$	
			$\frac{\bar{b}_s}{y_{sk}}$

Com o auxílio do quadro anterior seguiremos estes passos para a solução do sistema de inequações lineares sujeitas a uma função objetivo:

Passo 1: Organizar o quadro inicial como indicado, partindo de um PPL na forma canônica.

Passo 2: Realizar o teste de parada:

- Se todos os $z_j - c_j \leq 0$ ($j \in J$), então a solução ótima foi alcançada.
- Caso contrário, escolha o maior $z_j - c_j \geq 0$, digamos $z_k - c_k$, escolhendo o vetor associado x_k para entrar na base.

Passo 3: Definir a variável que sairá da base:

- Se $y_{ik} \leq 0$ para todo $i = 1, \dots, m$, então a variável x_k poderá ser diminuída indefinidamente e o valor de z tenderá ao infinito negativo. Nesse caso, a solução será ilimitada.
- Se $y_{ik} > 0$ para algum i , $i = 1, \dots, m$, então faça:
Calcule r , onde r é a variável básica relacionada ao mínimo entre os coeficientes $\frac{\bar{b}_i}{y_{ik}}$. O elemento y_{rk} é denominado “pivô”.

Passo 4: Substituir a r -ésima variável, correspondente a r -ésima equação pela variável x_k , que passará a integrar a nova base, e recalculas as matrizes B^{-1} , Y e os vetores $z_j - c_j$, \bar{x}_B e z_0 . Retornar ao passo 2.

Para recalculas as matrizes e os vetores do método, podemos utilizar diretamente as fórmulas deduzidas no item 3.3 ou, com base no quadro anteriormente proposto, utilizar o método de inversão de matrizes por operações elementares, considerando que as variáveis que permaneceram na base já possuem suas colunas canonizadas, ou seja, transformadas em vetores canônicos. Para canonizar a coluna da nova variável dividiremos a linha r pelo elemento pivô, obtendo 1 na posição do pivô. Para reduzir a zero os demais elementos da coluna utilizaremos a adição de múltiplos da linha (mais dúvidas, ver o método de inversão de matrizes por operações elementares no anexo).

A utilização direta das equações algébricas possui uma grande desvantagem para o cálculo manual, pois dificulta o acompanhamento das operações. Esse método é próprio para as implementações computacionais. O formato em quadro, apesar de exigir um número maior de operações algébricas e de espaço na memória (caso venha a ser automatizado), uma vez que trabalha com toda a matriz Y , permite uma melhor visão da evolução lógica do algoritmo. A propriedade algébrica de inversão da matriz aumentada permite que os vetores $z_j - c_j$, \bar{x}_B e z_0 possam igualmente ser obtidos por ocasião da inversão da nova matriz B através de operações elementares (ver anexo – Tópicos em Álgebra Linear e Convexidade).

Quadro de Pivoteamento

Quando o algoritmo simplex introduz uma nova variável na base e estamos usando o quadro para o recálculo, tudo se passa como se uma das colunas da matriz ainda tivesse de ser calculada pelo método de inversão por operações elementares para completar a obtenção da inversa de B . As variáveis que não são alteradas preservam suas colunas canonizadas, o que é muito conveniente ao cálculo, pois aproveita o trabalho de inversão realizado no passo anterior. A operação que denominamos de “pivoteamento” representa, em última análise, a aplicação das regras do método da inversão de matrizes por operações elementares ao quadro proposto.

A primeira providência para a inversão da nova base formada com a entrada da variável x_k e a saída de x_r é exatamente obter um valor unitário na posição do elemento pivô. Isso é alcançado pela divisão de toda a linha do pivô pelo valor desse elemento na coluna de x_k . Como o quadro é uma matriz aumentada, a linha do pivô estende-se ao termo independente associado. Pelo mesmo raciocínio, podemos facilmente entender por que a posição da linha dos $z_j - c_j$ fará parte também das colunas das variáveis.

Infelizmente, o aspecto matemático das equações que se formam ao tratarmos algebricamente uma operação de pivoteamento pode ser considerado um pouco complicado, mas é somente através da compreensão dessas expressões que seremos capazes de automatizar o processo.

Suponhamos que a variável x_k tenha sido escolhida, em uma determinada iteração do método simplex, para entrar na base, pois possuía o maior valor de $z_j - c_j$ no quadro, e a variável x_r foi a primeira que esgotou seu valor positivo com o crescimento de x_k , devendo, portanto, deixar a base. Nesse caso, o pivô será o elemento y_{rk} (coluna k , linha r). Como a nova coluna vai, quando canonizada, ocupar a posição r dos vetores canônicos que compõem a base, o elemento y_{rk} deve ser transformado em um valor unitário por uma operação elementar válida. A operação mais indicada para o caso é a divisão da linha do pivô pelo valor dele mesmo.

Após a adequação do pivô urge reduzir a zero toda a coluna do pivô, de modo a completar a operação de inversão preconizada pelo método das operações elementares. Como o pivô é unitário, cada posição poderá ser anulada pela soma de um valor simétrico obtido pela multiplicação da linha do pivô por -1 vezes o valor do elemento a ser anulado. Todas as operações necessárias são independentes entre si e suficientes para completar a inversão da base. A seguinte sequência de quadros esclarece algebricamente a operação de pivoteamento:

QUADRO 3.4 IDENTIFICAÇÃO DAS VARIÁVEIS QUE ENTRAM E SAEM DA BASE

		<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px;"> Maior valor no quadro </div>									
		x_1	x_k	x_s	x_{s+1}	x_{s+r}	x_n
	z	$z_1 - c_1$...	$z_k - c_k$...	$z_s - c_s$	$z_{s+1} - c_{s+1}$...	$z_{s+r} - c_{s+r}$...	$z_n - c_n$
x_{s+1}	\bar{b}_1	y_{11}	y_{1k}	y_{1s}	1	...	0	...	0
\vdots	\vdots	\vdots		\vdots		\vdots	0	...	0	...	0
$\leftarrow x_{s+r}$	\bar{b}_r	y_{r1}	y_{rk}	y_{rs}	0	...	1	...	0
\vdots	\vdots	\vdots		\vdots		\vdots	\vdots		\vdots		\vdots
x_n	\bar{b}_m	y_{m1}	y_{mk}	y_{ms}	0	...	0	...	1

Linha associada
ao mínimo valor $\frac{\bar{b}_s}{y_{sk}}$

Quadro após a Divisão pelo Pivô

QUADRO 3.5 OPERAÇÃO DE PIVOTEAMENTO

		$x_1 \dots x_k \dots x_s \quad x_{s+1} \dots x_{s+r} \dots x_n$	
	z	$z_1 - c_1 \dots z_k - c_k \dots z_s - c_s \quad z_{s+1} - c_{s+1} \dots z_{s+r} - c_{s+r} \dots z_n - c_n$	
x_{s+1}	\bar{b}_1	$y_{11} \dots y_{1k} \dots y_{1s}$	$1 \dots 0 \dots 0$
\vdots	\vdots	$\vdots \quad \vdots \quad \vdots$	$0 \dots 0 \dots 0$
x_k	$\frac{\bar{b}_r}{y_{rk}}$	$\frac{y_{r1}}{y_{rk}} \dots \frac{y_{rk}}{y_{rk}} \dots \frac{y_{rs}}{y_{rk}}$	$0 \dots \frac{1}{y_{rk}} \dots 0$
\vdots	\vdots	$\vdots \quad \vdots \quad \vdots$	$\vdots \quad \vdots \quad \vdots$
x_n	\bar{b}_m	$y_{m1} \dots y_{mk} \dots y_{ms}$	$0 \dots 0 \dots 1$

1

Elemento Pivô

O Quadro após a Canonização da Coluna do Pivô

QUADRO 3.6 QUADRO FINAL

		$x_1 \dots x_k \dots x_s \quad x_{s+1} \dots x_{s+r} \dots x_n$	
	$c_B \bar{B} -$ $-(z_k - c_k) - \frac{\bar{b}_r}{y_{rk}}$	$(z_j - c_j) - \frac{y_{rj}}{y_{rk}} (z_k - c_k)$ 0	$0 \dots \frac{c_k - z_k}{y_{rk}} \dots 0$
x_{s+1}	$\bar{b}_1 - \frac{y_{1k}}{y_{rk}} \bar{b}_r$	$y_{11} + \frac{y_{r1}}{y_{rk}} (-y_{1k}) \quad 0 \quad y_{1s} + \frac{y_{rs}}{y_{rk}} (-y_{1k})$	$1 \dots \frac{-y_{1k}}{y_{rk}} \dots 0$
\vdots	\vdots	$\vdots \quad \vdots \quad \vdots$	$0 \quad \vdots \quad 0$
x_k	$\frac{\bar{b}_r}{y_{rk}}$	$\frac{y_{r1}}{y_{rk}} \dots 1 \dots \frac{y_{rs}}{y_{rk}}$	$0 \dots \frac{1}{y_{rk}} \dots 0$
\vdots	\vdots	$\vdots \quad \vdots \quad \vdots$	$0 \quad \vdots \quad 0$
x_n	$\bar{b}_m - \frac{y_{mk}}{y_{rk}} \bar{b}_r$	$y_{m1} \frac{y_{r1}}{y_{rk}} (-y_{mk}) \quad 0 \quad y_{ms} \frac{y_{r1}}{y_{rk}} (-y_{mk})$	$0 \dots \frac{-y_{mk}}{y_{rk}} \dots 1$

3.3.5 – Um Exemplo Numérico

Seja o seguinte problema de programação linear:

$$\text{Maximizar } z = 4x_1 + 5x_2 + 9x_3 + 11x_4$$

sujeito a:

$$x_1 + x_2 + x_3 + x_4 \leq 15$$

$$7x_1 + 5x_2 + 3x_3 + 2x_4 \leq 120$$

$$3x_1 + 5x_2 + 10x_3 + 15x_4 \leq 100$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$$

1. Transformação do modelo para o formato padrão:

$$\text{Maximizar } z = 4x_1 + 5x_2 + 9x_3 + 11x_4 + 0x_5 + 0x_6 + 0x_7$$

sujeito a:

$$x_1 + x_2 + x_3 + x_4 + x_5 = 15$$

$$7x_1 + 5x_2 + 3x_3 + 2x_4 + x_6 = 120$$

$$3x_1 + 5x_2 + 10x_3 + 15x_4 + x_7 = 100$$

2. Quadro inicial de cálculo, Passo 1 do algoritmo:

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
z	0	4	5	9	11	0	0	0
x_5	15	1	1	1	1	1	0	0
x_6	120	7	5	3	2	0	1	0
x_7	100	3	5	10	15	0	0	1

Observar que no Quadro 1 as variáveis de folga constituem uma base canônica, sendo desnecessária a determinação de sua inversa.

3. Passo 2 do algoritmo: regra de parada.

Como existe $z_j - c_j > 0$ uma variável deve entrar na base. A escolhida é x_4 pois $z_4 - c_4 = 11$ é o maior valor entre os $z_j - c_j$. A seta indica a entrada de x_4 na base.

Maior valor $z_j - c_j$ no quadro

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
z	0	4	5	9	11	0	0	0
x_5	15	1	1	1	1	1	0	0
x_6	120	7	5	3	2	0	1	0
x_7	100	3	5	10	15	0	0	1

4. Passo 3 do algoritmo: determinação da variável que sai da base (determinação do pivô).

A variável x_7 sai da base uma vez que $100/15$ é o menor valor para o crescimento de x_4 . A seta horizontal indica a saída de x_7 da base.

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	
z	0	4	5	9	11	0	0	0	
x_5	15	1	1	1	1	1	0	0	15/1
x_6	120	7	5	3	2	0	1	0	120/2
x_7	100	3	5	10	15	0	0	1	100/15
s					Pivô				

5. Passo 4 do algoritmo: operação de cálculo dos valores de solução associados à nova base.

No caso do uso do quadro simplex, essa fase corresponde ao denominado pivoteamento, que permite o cálculo das matrizes do problema através do método das operações elementares. A linha do pivô é dividida por 15, que é o valor do pivô.

Etapa 1 – Construção do Pivô:

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	
z	0	4	5	9	11	0	0	0	
x_5	15	1	1	1	1	1	0	0	
x_6	120	7	5	3	2	0	1	0	
x_7	20/3	1/5	1/3	2/3	1	0	0	1/15	

Etapa 2 – Construção da Coluna do Pivô:

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	
z	-220/3	9/5	4/3	5/3	0	0	0	-11/15	
x_5	25/3	4/5	2/3	1/3	0	1	0	-1/15	
x_6	320/3	33/5	13/3	5/3	0	0	1	-2/15	
x_4	20/3	1/5	1/3	2/3	1	0	0	1/15	

6. Passo 2 do algoritmo: aplicação da regra de parada.

Nesse caso o maior $z_j - c_j$ corresponde a $z_1 - c_1 = 9/5$, que conduz a variável x_1 a entrar na base.

7. Passo 3 do algoritmo: determinação da variável que sai da base (determinação do pivô).

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	
z	-220/3	9/5	4/3	5/3	0	0	0	-11/15	
x_5	25/3	4/5	2/3	1/3	0	1	0	-1/15	125/12
x_6	320/3	33/5	13/3	5/3	0	0	1	-2/15	1600/99
x_4	20/3	1/5	1/3	2/3	1	0	0	1/15	100/3

8. Passo 4 do algoritmo: operação de cálculo dos valores de solução associados à nova base.

Etapa 1 – Construção do Pivô:

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	
z	-220/3	9/5	4/3	5/3	0	0	0	-11/15	
x_1	125/12	1	5/6	5/12	0	5/4	0	-1/12	
x_6	320/3	33/5	13/3	5/3	0	0	1	-2/15	
x_4	20/3	1/5	1/3	2/3	1	0	0	1/15	

Etapa 2 – Construção da Coluna do Pivô:

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	
z	-1105/12	0	-1/6	11/12	0	0	0	-11/15	
x_1	125/12	1	5/6	5/12	0	5/4	0	-1/12	
x_6	455/12	0	-7/6	-13/12	0	-33/4	1	5/12	
x_4	55/12	0	1/6	7/12	1	-1/4	0	1/12	

9. Passo 2 do algoritmo: aplicação da regra de parada.

Ainda temos $z_3 - c_3 = 11/12$, que conduz a variável x_3 a entrar na base.

10. Passo 3 do algoritmo: determinação da variável que sai da base (determinação do pivô).

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	
z	-1105/12	0	-1/6	11/12	0	-9/4	0	-7/12	
x_1	125/12	1	5/6	5/12	0	5/4	0	-1/12	25
x_6	455/12	0	-7/6	-13/12	0	-33/4	1	5/12	
x_4	55/12	0	1/6	7/12	1	-1/4	0	1/12	55/7

Pivô

11. Passo 4 do algoritmo: operação de cálculo dos valores de solução associados à nova base.

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	
z	$-695/7$	0	$-3/7$	0	$-11/7$	$-13/7$	0	$-5/7$	
x_1	$50/7$	1	$5/7$	0	$-5/7$	$10/7$	0	$-1/7$	
x_6	$325/7$	0	$-6/7$	0	$13/7$	$-61/7$	1	$4/7$	
x_3	$55/7$	0	$2/7$	1	$12/7$	$-3/7$	0	$1/7$	

12. Passo 2 do algoritmo: aplicação da regra de parada.

Todos os $z_j - c_j \leq 0$, o que conduz à regra de parada. A solução ótima foi encontrada.

O valor ótimo de z é $695/7$, com $x_1 = 50/7$; $x_6 = 325/7$ e $x_3 = 55/7$.

3.3.6 – O Caso em que a Base Viável Inicial não Está Disponível

No algoritmo descrito, a fase de obtenção de uma base viável para o problema foi considerada *superada*. Realmente, quando, por meio da inclusão de variáveis de folga na passagem da descrição da forma canônica para a padrão obtemos uma base inicial (e ainda por cima canônica), o algoritmo descrito soluciona bem o problema. Mas e quando isso não acontece? Seja o exemplo que se segue:

$$\text{Minimizar } z = -3x_1 - 5x_2$$

sujeito a:

$$x_1 \leq 4$$

$$x_2 \leq 6$$

$$3x_1 + 2x_2 \geq 18$$

$$x_1 \geq 0, x_2 \geq 0$$

Que ao ser convertido para a forma padrão será:

$$\text{Minimizar } z = -3x_1 - 5x_2 + 0x_3 + 0x_4 + 0x_5$$

sujeito a:

$$x_1 + x_3 = 4$$

$$x_2 + x_4 = 6$$

$$3x_1 + 2x_2 - x_5 = 18$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0$$

Obviamente a matriz formada pelas variáveis de folga não é uma base viável:¹

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

1. B é uma base, contudo não atende à exigência de conduzir a uma solução em que todas as variáveis serão maiores ou iguais a zero.

A Figura 3.11 mostra graficamente que o ponto origem (todas as variáveis reais iguais a zero) não pertence ao espaço das soluções viáveis desse problema, ou seja, a base tradicionalmente considerada como inicial falha em ser viável.

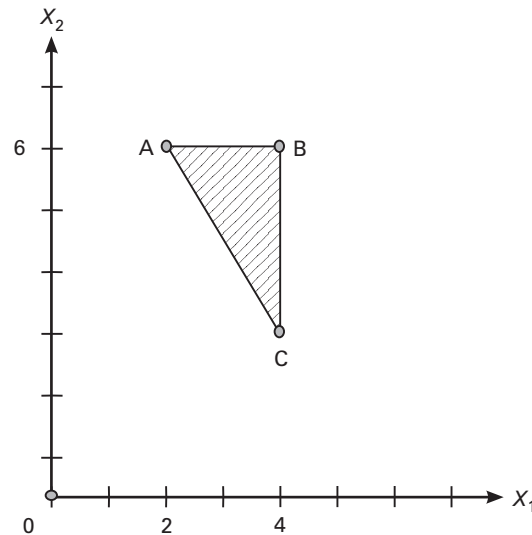


FIGURA 3.11 Espaço das soluções viáveis do exemplo.

Nesse ponto, temos dois caminhos a adotar:

- Descobrir uma base viável (pontos A, B ou C) por um outro método qualquer e então aplicar o simplex a partir dessa base.
- Usar o próprio simplex para gerar uma base viável, a partir de uma base qualquer inviável.

Optaremos pela segunda alternativa. Nesse caso, inicialmente, o algoritmo buscará uma base viável para, posteriormente, buscar o ótimo do programa em si. Tal método de solução terá genericamente duas fases. Podemos descrever essas etapas da seguinte forma:

- **Primeira Fase:** o procedimento utilizará o simplex para buscar uma base viável de solução.
- **Segunda Fase:** de posse da base calculada na primeira fase aplicaremos o método simplex tradicional em busca do ótimo do programa.

Para operacionalizar a busca da base viável inicial existem duas alternativas, todas com base na ideia de “variáveis artificiais”. Vamos definir como variáveis artificiais aquelas que faremos introduzir no programa de otimização para que tenhamos a tão necessária base viável inicial. Essas variáveis não necessitarão possuir qualquer ligação com a realidade da modelagem do problema, daí o nome de “artificiais”. No exemplo anterior, como apenas uma das colunas da base apresenta valor negativo em sua componente canônica (terceira coluna), ou seja, existe somente uma variável de folga negativa, nos bastará uma artificial para corrigir essa anomalia. Podemos obter uma base viável canônica para o programa do exemplo com a introdução de uma variável artificial como representada a seguir:

$$\text{Minimizar } z = -3x_1 - 5x_2 + 0x_3 + 0x_4 + 0x_5 + ?x_6^a$$

sujeito a:

$$x_1 \quad \quad \quad x_3 \quad \quad = 4$$

$$\quad x_2 \quad \quad \quad x_4 \quad = 6$$

$$3x_1 + 2x_2 \quad - x_5 + x_6^a = 18$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0, x_6^a \geq 0$$

Consequentemente podemos agora dispor da seguinte base canônica:

$$B = \begin{matrix} & x_3 & x_4 & x_6^a \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

O ponto de interrogação no custo da variável artificial é um aspecto fundamental para nossa discussão, e é nele que os dois métodos de solução mais conhecidos diferem. Enquanto as variáveis de folga não possuem custos, apesar de estarem ligadas ao sentido lógico do modelo, as artificiais estão ligadas apenas à estratégia adotada para a obtenção de uma base viável para o simplex. Como elas possuem um sentido dentro de um estratagema matemático, seu “custo” associado pode ser útil nessa estratégia.

O raciocínio é simples: se desejamos apenas, em uma primeira fase, obter uma base viável real para o programa, teremos de nos livrar, em algum momento, dessas tais variáveis artificiais. Pensando em termos do funcionamento do algoritmo simplex, será de todo o nosso interesse que as variáveis artificiais deixem o mais rapidamente possível a base que será calculada a cada iteração. Nenhuma dessas bases artificialmente constituídas poderá ser usada diretamente na otimização do problema real (pelo menos em tese) e elas representam esforço de cálculo. Uma maneira natural de pensar uma saída seria associar a essas variáveis “intrusas” um custo extraordinariamente alto. Obviamente, se utilizarmos o simplex para minimizar um problema em que existem variáveis extraordinariamente grandes na base, o método terá como prioritário trocá-las por outras mais baratas, o que, em tese, conduziria a uma solução viável.

A estratégia anteriormente descrita é conhecida como o *método do grande M*, em que M é o valor gigantesco que atribuímos aos custos das variáveis artificiais. A principal desvantagem desse método reside na possibilidade de dificuldades numéricas quando implementado computacionalmente. Quando M é um valor muito grande podem ocorrer erros numéricos de arredondamento nas divisões decorrentes, especialmente quando a diferença entre M e os valores dos demais custos do quadro é excessivamente grande.

Por outro lado, quando M é adotado inconvenientemente pequeno podem decorrer efeitos de modelagem. Nessa alternativa, as variáveis artificiais podem ser tomadas pelo programa por variáveis reais. Para evitar as dificuldades que o método do grande M pode introduzir na escolha do valor de M, temos a possibilidade de separar completamente as duas fases de cálculo, escolhendo, inclusive, uma função objetivo diferente para cada fase.

Na fase inicial podemos usar uma função objetivo que minimize o número de variáveis artificiais existentes na base. Como uma base real não possui qualquer variável artificial não nula, concluída a primeira fase, prosseguiremos normalmente com os cálculos com a função objetivo original do problema. Esse procedimento geral de dividir a otimização em duas etapas, com funções objetivos diferentes, é denominado *método das duas fases*. O nome do método está vinculado à existência de duas funções objetivo, uma para cada fase, uma vez que o método do grande M também opera basicamente em duas fases.

Para esclarecermos como podemos realizar, na prática, esses cálculos, vamos prosseguir na solução do exemplo sugerido. Inicialmente, utilizaremos o quadro simplex para acompanhar os cálculos de cada iteração. Introduziremos uma pequena alteração no quadro de modo a permitir que possamos acompanhar as duas fases do método, a saber: uma linha para cada função objetivo conforme constante no Quadro 3.7. A função objetivo da primeira fase será minimizar o número de variáveis artificiais. Denominaremos a função da primeira fase por q e a expressaremos, para o exemplo, da seguinte forma:

$$\text{Minimizar } q = 0x_1 + 0x_2 + 0x_3 + 0x_4 + 0x_5 + 1x_6^a$$

que nos levará ao Quadro 3.7:

QUADRO 3.7 UM EXEMPLO DO MÉTODO DAS DUAS FASES

		x_1	x_2	x_3	x_4	x_5	x_6^a	
q		0	0	0	0	0	-1	Função da 1ª Fase
z		3	5	0	0	0	0	Função da 2ª Fase
x_3	4	1	0	1	0	0	0	
x_4	6	0	1	0	1	0	0	
x_6^a	18	3	2	0	0	-1	1	

Observamos que, ao simplesmente transcrevermos os custos das variáveis básicas para o quadro, como fazíamos no método tradicional, deixamos de considerar que o vetor das variáveis básicas não é nulo. Nos quadros do método clássico, os custos das variáveis eram sempre zero, o que tornava nulo o termo z_j na primeira iteração. Nesse caso, era correto transcrever para o quadro a linha da função objetivo com o sinal trocado. Agora a variável básica artificial x_6^a possui um custo unitário e z_j não será automaticamente igual a zero, devendo ser calculado. Isso ocorre exatamente porque estamos “forçando” a variável artificial a pertencer à base. Notamos, contudo, que o termo c_j é nulo para todas as variáveis, com exceção da variável x_6^a , o que nos conduz a:

$$z_j - c_j = c_B B^{-1} a_j - c_j$$

que, nesse caso, será:

$$z_6 - c_6 = [0 \ 0 \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - 1 = 0$$

$$z_1 - c_1 = [0 \ 0 \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix} - 0 = 3$$

$$z_2 - c_2 = [0 \ 0 \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} - 0 = 2$$

$$z_3 - c_3 = [0 \ 0 \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - 0 = 0$$

$$z_4 - c_4 = [0 \ 0 \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} - 0 = 0$$

$$z_5 - c_5 = [0 \ 0 \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} - 0 = -1$$

O valor dos custos reduzidos das variáveis não básicas, para esse caso, acaba igual ao valor dos coeficientes das variáveis na linha em que a variável artificial é introduzida. Se houver mais de uma variável artificial, o produto $c_B B^{-1} a_j$ corresponde a simplesmente somar esses coeficientes para as linhas em que essas variáveis forem necessárias, uma vez que haverá um elemento unitário no vetor de custos básicos para cada uma dessas linhas. Essa é a razão matemática que leva alguns textos a enunciar que, para obter a função objetivo da primeira fase do quadro simplex, devemos *somar as linhas das variáveis artificiais à linha da função objetivo*.

Outra forma de justificar essa regra é lembrar que no quadro simplex estamos sempre trabalhando com uma matriz aumentada, ou seja, devemos canonizar as colunas das variáveis artificiais (o que nos levará a somar as linhas da matriz A à função objetivo, exatamente como prevê a regra anterior quando formos anular o custo reduzido da coluna artificial).

Pivoteando no elemento unitário de x_6^a chegamos ao quadro inicial:

		x_1	x_2	x_3	x_4	x_5	x_6^a		
(+)	q	18	3	2	0	0	-1	0	Função da 1ª Fase
	z		3	5	0	0	0	0	Função da 2ª Fase
	x_3	4	1	0	1	0	0	0	
	x_4	6	0	1	0	1	0	0	
←	x_6^a	18	3	2	0	0	-1	1	

Que vai conduzir x_1 a entrar e x_3 a sair da base, conforme o quadro que se segue:

		x_1	x_2	x_3	x_4	x_5	x_6^a		
	q	18	3	2	0	0	-1	0	Função da 1ª Fase
	z		3	5	0	0	0	0	Função da 2ª Fase
←	x_3	4	1	0	1	0	0	0	$4/1 = 4$
	x_4	6	0	1	0	1	0	0	-
	x_6^a	18	3	2	0	0	-1	1	$18/3 = 6$

Que resultará em:

		x_1	x_2	x_3	x_4	x_5	x_6^a	
q	6	0	2	-3	0	-1	0	Função da 1ª Fase
z	-12	0	5	-3	0	0	0	Função da 2ª Fase
x_1	4	1	0	1	0	0	0	-
x_4	6	0	1	0	1	0	0	6
x_6^a	6	0	2	-3	0	-1	1	3

O que, finalmente, nos levará ao quadro em que a função objetivo da primeira fase foi otimizada (término da primeira fase-ponto c da Figura 3.11).

		x_1	x_2	x_3	x_4	x_5	x_6^a	
q	6	0	0	0	0	0	-1	Função da 1ª Fase
z	-27	0	0	9/2	0	5/2	-1	Função da 2ª Fase
x_1	4	1	0	1	0	0	0	-
x_4	3	0	0	3/2	1	1/2	-1/2	6
x_2	3	0	1	-3/2	0	-1/2	1/2	3

A partir desse quadro, a variável x_6^a pode ser desconsiderada do processo sem qualquer perda de informações para o programa. Recomenda-se, portanto, que ela seja marcada para que não se realizem cálculos inúteis de “custos reduzidos”. Retomando o processo, na segunda fase temos:

		x_1	x_2	x_3	x_4	x_5	x_6^a	
z	-27	0	0	9/2	0	5/2		Função da 2ª Fase
x_1	4	1	0	1	0	0	0	4
x_4	3	0	0	3/2	1	1/2	-1/2	2
x_2	3	0	1	-3/2	0	-1/2	1/2	-

Que leva ao quadro:

		x_1	x_2	x_3	x_4	x_5	x_6^a	
z	-36	0	0	0	-3	1		Função da 2ª Fase
x_1	2	1	0	1	-2/3	-1/3	1/3	-
x_3	2	0	0	1	2/3	1/3	-1/3	6
x_2	6	0	1	0	1	0	0	-

Que nos conduz ao quadro ótimo a seguir:

		x_1	x_2	x_3	x_4	x_5	x_6^a	
z	-42	0	0	-3	-5	0		Função da 2ª Fase
x_1	4	1	0	0	0	0	0	-
x_5	6	0	0	0	2	1/3	-1	
x_2	6	0	1	0	1	0	0	-

O gráfico da Figura 3.12 mostra, geometricamente, a sequência das bases geradas nas duas fases do método:

Sugere-se que o leitor resolva este mesmo exemplo utilizando algum custo elevado M , associado à variável artificial x_6^a .

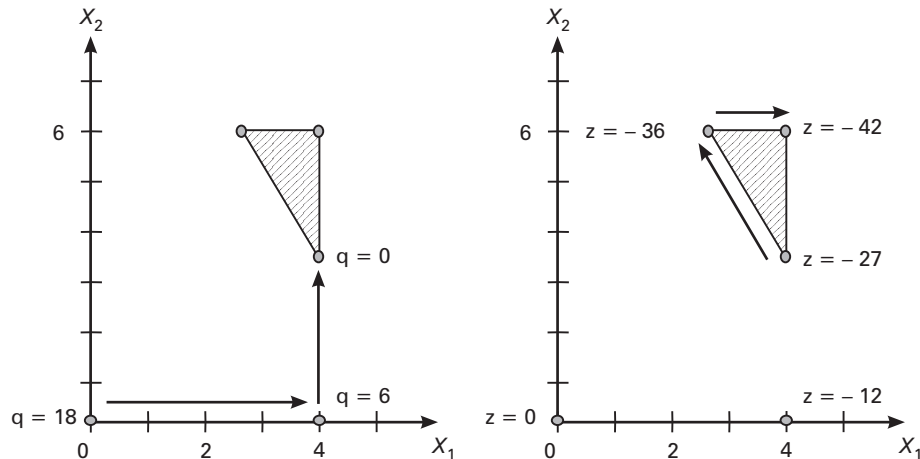


FIGURA 3.12 Caminho das sequências de bases.

3.4 – CASOS ESPECIAIS PARA O SIMPLEX

3.4.1 – Degeneração e Ciclagem

A convergência do algoritmo simplex é garantida pela certeza de que, a cada passo, o critério de escolha de troca de uma variável da base vai redundar em melhoria para o valor da função objetivo. Mesmo que fosse necessário examinar um número exponencial de combinações de variáveis para formar a base ótima (também denominadas configurações de solução), em passo finito, o algoritmo terminaria. Entretanto, em algumas situações especiais, podemos encontrar configurações que invalidem o uso perfeito dos critérios anteriormente descritos. Essas configurações patológicas decorrem de situações em que os $z_j - c_j$ associados a algumas variáveis possuem valores iguais, e o mesmo acontece em relação ao mínimo valor $\frac{\bar{b}_s}{y_{sk}}$. O quadro abaixo mostra a situação descrita:

QUADRO 3.8 EXEMPLO DE EMPATES NOS CRITÉRIOS DE SELEÇÃO DE VARIÁVEIS

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	
z		0	0	0	1	1	-1	0	
x_1	0	1	0	0	1	3	1	0	0
x_2	1	0	1	0	0	-2	-1	0	
x_7	0	0	0	0	1	2	0	1	0
x_3	2	0	0	1	1	0	0	0	

Em relação ao critério de saída da base, o que ocorre é que existem variáveis básicas com valor nulo. Quando isso acontece, o produto $(z_j - c_j) x_j$, responsável pela garantia do decréscimo da função objetivo (critério de minimização), vai falhar, não porque deixe de existir a possibilidade de que uma troca de variáveis possa melhorar a função objetivo, e sim porque a variável x_j que entra não poderá assumir valor diferente de zero. O chamado “empate” de saída é também conhecido como degeneração e corresponde a uma situação no hiperespaço das atividades, como representado na Figura 3.13.

Quando um certo ponto extremo do conjunto C das soluções viáveis do modelo de programação matemática está associado a mais restrições do que as estritamente necessárias para a definição desse mesmo ponto, o conjunto de restrições é linearmente dependente e existe a possibilidade de que ocorra, em um certo estágio do desenvolvimento do algoritmo, uma degeneração no critério de saída da

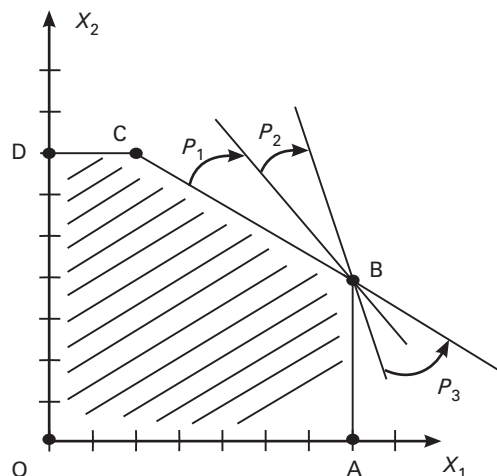


FIGURA 3.13 Ciclagem do algoritmo Simplex.

base. O surgimento do valor nulo no quadro é decorrente do ato de que as folgas associadas às várias restrições redundantes são esgotadas simultaneamente.

Apesar de alguns autores afirmarem que a ciclagem do algoritmo simplex é um fato desconhecido em problemas práticos, o primeiro autor deste texto deparou-se com vários exemplos de ciclagem na solução de problemas de particionamento não ponderado de grande porte. Em ocasiões nas quais é esperado que o processo de solução lide com muitas variáveis candidatas simultaneamente a entrar na base e alta percentagem de degeneração,² a Regra de Bland torna-se de utilização indispensável (ver Bland [1977]).

Regra de Bland

- Entre todas as candidatas a entrar na base, selecione a variável x_k que possui o menor índice.
- Entre todas as candidatas a sair da base, selecione a variável x_r que possui o menor índice.

A simples regra do menor índice é capaz de esgotar toda a discussão da ciclagem no algoritmo simplex, como o método lexicográfico (ver Chvátal [1983]). Observar que não existe qualquer problema se trocarmos a regra do menor índice pela do maior índice.

3.4.2 – Múltiplas Soluções Ótimas

A Figura 3.8 exemplifica graficamente um caso de múltiplas soluções ótimas em um modelo de programação linear. De uma forma intuitiva, o leitor é levado a entender que nesse caso a direção da função objetivo vai obrigatoriamente alcançar mais de um ponto extremo da envoltória convexa do modelo. Como por combinação linear dos pontos extremos podemos gerar infinitos pontos que serão igualmente pertencentes à envoltória convexa do modelo (uma face dessa envoltória), o modelo terá infinitas soluções ótimas.

A pergunta que se coloca é como poderíamos identificar uma tal situação no desenvolvimento do simplex. Para exemplificar o caso em estudo, vamos solucionar o problema a seguir:

2. O autor refere-se ao caso da solução de problemas de particionamento não ponderado em que a percentagem de degeneração das linhas situam-se em valores no entorno de 95% (ver Goldbarg [1990]).

Minimizar $z = -x_1 - 2x_2$

sujeito a:

$$x_1 \leq 3$$

$$x_2 \leq 4$$

$$x_1 + 2x_2 \leq 9$$

$$x_1 \geq 0, x_2 \geq 0$$

Que corresponde ao seguinte quadro inicial, após a inclusão das três variáveis de folga necessárias:

		x_1	x_2	x_3	x_4	x_5	
z	0	1	2	0	0	0	
x_3	3	1	0	1	0	0	
x_4	4	0	1	0	1	0	$4/1 = 4$
x_5	9	1	2	0	0	1	$9/2 = 4,5$

Que, após a entrada de x_2 na base e o pivoteamento, é reduzido a:

		x_1	x_2	x_3	x_4	x_5	
z	-8	1	0	0	-2	0	
x_3	3	1	0	1	0	0	
x_2	4	0	1	0	1	0	
x_5	1	1	0	0	-2	1	

No novo quadro, a variável x_1 é candidata a entrar na base e a variável x_5 deverá sair.

		x_1	x_2	x_3	x_4	x_5	
z	-8	1	0	0	-2	0	
x_3	3	1	0	1	0	0	3
x_2	4	0	1	0	1	0	
x_5	1	1	0	0	-2	1	1

Resultando, após o pivoteamento, no quadro abaixo.

		x_1	x_2	x_3	x_4	x_5	
z	-9	0	0	0	0	-1	
x_3	2	0	0	1	2	-1	
x_2	4	0	1	0	1	0	
x_1	1	1	0	0	-2	1	

O quadro anterior é ótimo e a regra de parada do simplex atesta isso. A solução final é:

$$z = -9, x_3 = 2; x_2 = 4; x_1 = 1$$

Contudo, é possível gerar uma outra base (ponto extremo) com o mesmo valor da função objetivo, ou seja, com o mesmo valor ótimo. Para tal vamos continuar o processo de pivoteamento incluindo a variável x_4 na solução. O quadro que se segue mostra que a variável que deverá sair é x_3 .

		x_1	x_2	x_3	x_4	x_5	
z	-9	0	0	0	0	-1	
x_3	2	0	0	1	2	-1	$2/2 = 1$
x_2	4	0	1	0	1	0	(valor nulo)
x_1	1	1	0	0	-2	1	(valor negativo)

Que leva ao quadro também ótimo a seguir com $z = -9$, $x_4 = 1$; $x_2 = 3$; $x_1 = 3$:

		x_1	x_2	x_3	x_4	x_5	
z	-9	0	0	0	0	-1	
x_4	1	0	0	1/2	1	-1/2	
x_2	3	0	1	-1/2	0	1/2	
x_1	3	1	0	1	0	0	

A Figura 3.14 mostra os dois pontos extremos representados pela base do primeiro e do segundo quadro ótimo. Note que a função z tem direção paralela à restrição representada pelos pontos B e C, ou seja, $x_1 + 2x_2 \leq 9$. A variável x_3 é a folga associada à restrição representada pelo segmento de reta CD e, por isso, ela está presente na base que representa o ponto C. Quando, por pivoteamento, o algoritmo escolhe o ponto B como solução, a variável x_3 é substituída por x_4 , a folga da restrição associada à CB.

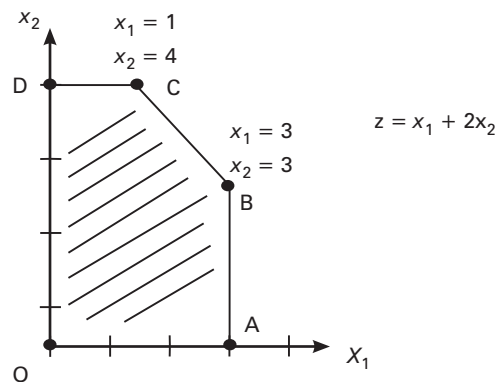


FIGURA 3.14 Caso de infinitas soluções ótimas.

3.5 – PROBLEMAS PROPOSTOS

1 – Revisão da Teoria



Responda com Certo ou Errado justificando sua escolha:

1. Se em um sistema com n incógnitas e m equações ($n > m$) é possível selecionar um conjunto de m incógnitas que forme uma matriz $m \times m$ com determinante diferente de zero, então:

- a. ☐ O sistema $m \times m$ é determinado.
 - b. ☐ O sistema é determinado, mas pode não ter solução.
 - c. ☐ O sistema é determinado e pode ter uma solução infinita.
 - d. ☐ O sistema pode ter infinitas soluções.
 - e. ☐ O simplex pode solucionar esse sistema desde que a solução seja não negativa.
2. Seja x^* uma solução ótima de um PPL e M o conjunto de todas as suas soluções viáveis. Então:
- a. ☐ x^* pode ser sempre obtida por uma combinação linear convexa de dois ou mais pontos distintos de M .
 - b. ☐ x^* só pode ser obtida por uma combinação linear convexa de dois ou mais pontos distintos de M se M for limitado.
 - c. ☐ x^* corresponde sempre a um vértice da envoltória convexa de M , mesmo quando M é ilimitado.
 - d. ☐ x^* é um vértice apenas no caso de não haver degeneração.
3. Se o conjunto de soluções viáveis do PPL não for limitado, então:
- a. ☐ A solução do problema é impossível ou ilimitada.
 - b. ☐ A solução pode não ocorrer em um vértice porque o espaço das soluções não é convexo.
4. Se o conjunto de soluções viáveis do PPL não for limitado em alguma iteração do algoritmo simplex:
- a. ☐ Utilizamos o método das duas fases para obter a inviabilidade.
 - b. ☐ Pelo menos uma das colunas correspondentes às variáveis não básicas com $z_j - c_j \geq 0$ é formada por elementos nulos e negativos.
 - c. ☐ Somente a variável não básica com maior $z_j - c_j$ é formada por elementos nulos ou negativos.
 - d. ☐ Existe mais de uma variável não básica candidata a entrar na base possuindo o mesmo valor de $z_j - c_j \geq 0$.
5. Se, em uma determinada iteração do algoritmo simplex:
- a. ☐ Uma variável deixa a base, na próxima iteração ela não poderá entrar novamente na base.
 - b. ☐ Uma variável entra na base, na próxima iteração ela não poderá deixá-la.
 - c. ☐ Existe uma variável artificial na base e existem variáveis não básicas com $z_j - c_j \geq 0$, a primeira fase do método das duas fases ainda não chegou ao fim.
 - d. ☐ Todas as variáveis não básicas possuem $z_j - c_j \leq 0$, a solução ótima foi alcançada.
 - e. ☐ Todas as variáveis não básicas possuem $z_j - c_j \leq 0$, a solução é limitada.
6. Quando existem variáveis não básicas com $z_j - c_j = 0$ no quadro ótimo do simplex, então:
- a. ☐ Existe degeneração.
 - b. ☐ Existem infinitas soluções ótimas para o problema.
 - c. ☐ Esse fato é absolutamente normal na condição de parada e a solução é única.
 - d. ☐ As variáveis não básicas com $z_j - c_j = 0$ podem ser introduzidas na base gerando, com isso, novas soluções ótimas.
7. Quando existe empate no critério de saída da base, então:
- a. ☐ Existe degeneração.
 - b. ☐ Existem infinitas soluções ótimas para o problema.
 - c. ☐ Esse fato é absolutamente normal na condição de parada e a solução é única.
 - d. ☐ O conjunto das restrições é linearmente dependente.
 - e. ☐ Podemos eliminar do problema uma das restrições que geraram o empate.
 - f. ☐ A partir da ocorrência do empate, as folgas das linhas empatadas serão sempre iguais a zero para todas as demais iterações do simplex.

2 – Solução de Problemas de Programação Linear



Solucione pelo método simplex e graficamente os seguintes problemas de programação linear:

a) Maximizar $z = 4x_1 + 7x_2$
 sujeito a:

$$\begin{aligned} x_1 &\leq 6 \\ x_2 &\leq 8 \\ 4x_1 - 2x_2 &\leq 10 \\ x_1 \geq 0, x_2 &\geq 0 \end{aligned}$$

b) Minimizar $z = -2x_1 - 3x_2$
 sujeito a:

$$\begin{aligned} x_1 &\leq 8 \\ x_2 &\leq 4 \\ -2x_1 + x_2 &\leq 5 \\ x_1 \geq 0, x_2 &\geq 0 \end{aligned}$$

c) Uma empresa produz dois tipos de bolsas de plástico (B_1, B_2) cujos mercados absorvem respectivamente 80 e 60 unidades diárias. O processo de produção consome dois tipos de matéria-prima: folhas de plásticos e fechos. Cada unidade de B_1 consome duas folhas de plástico e quatro fechos. Cada unidade de B_2 consome três folhas de plástico e três fechos. São disponíveis diariamente 200 folhas de plástico e 240 fechos. Os lucros unitários pelas vendas dos produtos são, respectivamente, R\$20 e R\$25. Qual deve ser o esquema de produção que conduza ao maior lucro possível?

3 – Solução de Problemas de Programação Linear



Determine todas as soluções básicas viáveis do seguinte conjunto de restrições:

$$\begin{aligned} -x_1 + x_2 + x_3 + x_4 - 2x_5 &= 4 \\ +x_1 - 2x_2 + x_4 - x_5 &= 3 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 &\geq 0 \end{aligned}$$

4 – Método das Duas Fases



Solucione pelo método simplex os seguintes problemas de programação linear:

a) Maximizar $z = 2x_1 + 5x_2 + x_3$
 sujeito a:

$$\begin{aligned} x_1 + x_2 &\geq 6 \\ x_2 - x_3 &\geq 4 \\ 4x_1 + 2x_2 + x_3 &\leq 15 \\ x_1 \geq 0, x_2 \geq 0, x_3 &\geq 0 \end{aligned}$$

b) Minimizar $z = x_1 - 3x_2 - 5x_3 - x_4$
 sujeito a:

$$\begin{aligned} x_1 + x_4 &\leq 8 \\ x_2 + x_3 &\leq 8 \\ -x_1 + x_2 + x_3 + x_4 &\geq 4 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 &\geq 0 \end{aligned}$$

5 – Interpretação Gráfica



Sabendo-se que a Figura 3.15 representa graficamente a região das soluções viáveis de um PPL e que a sequência de pontos extremos indicada mostra a trajetória de solução do algoritmo simplex utilizado (B é o ponto ótimo), pede-se:

- Representar as bases associadas aos pontos extremos visitados pelas iterações.
- Representar matricialmente a solução associada ao ponto F (ponto interno).
- Como poderíamos transformar a solução associada ao ponto F em uma solução básica viável?

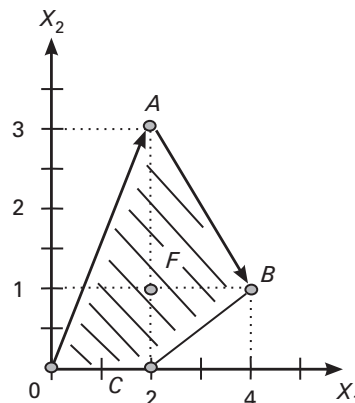


FIGURA 3.15 Uma trajetória do simplex.

6 – Método Simplex – Fundamentos Teóricos



João, muito nervoso, está terminando sua prova de programação linear quando descobre que errou, logo no primeiro quadro, as contas no cálculo dos custos reduzidos ($z_j - c_j$) de uma das questões. Desesperado, correu até o professor e pediu mais tempo para que pudesse refazer os quadros e não ser prejudicado por um erro tão simples. O professor, examinando o quadro final e inicial do problema, deu a seguinte solução:

“João, refaça apenas os cálculos para o último quadro. Isso será suficiente. Nesse caso considerarei a questão correta! Nem precisava, mas vou lhe dar mais cinco minutos... boa sorte”.

Ajude João a corrigir o seu erro sem ter de refazer todos os pivoteamentos anteriores.

Quadro inicial copiado com erro (célula sombreada):

		x_1	x_2	x_3	x_4	x_5	x_6	
z	0	-1	-3	4	0	0	0	
x_4	9	1	1	2	1	0	0	
x_5	2	1	1	-1	0	1	0	
x_6	4	-1	1	1	0	0	1	

Quadro final com os erros acumulados decorrentes

		x_1	x_2	x_3	x_4	x_5	x_6	
z	-17	0	-6	0	-1	0	-2	
x_1	1/3	1	-1/3	0	1/3	0	-2/3	
x_5	6	0	2	0	0	1	1	
x_3	13/3	0	2/3	1	1/3	0	1/3	

Erro cometido: o valor correto de $z_2 - c_2$ no primeiro quadro era -1.

7 – Método Simplex – Fundamentos Teóricos



Seja o problema que se segue:

Maximizar $z = 3x_1 + 2x_2 - 2x_3 + 5x_4$
 sujeito a:

$$\begin{aligned} x_1 + x_2 + 3x_3 + x_4 &\leq 8 \\ 3x_1 - x_2 + x_3 + 2x_4 &\leq 10 \\ x_1 - x_3 &\leq 6 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0 \end{aligned}$$

Determine uma solução para o problema com as variáveis x_1, x_2, x_3 . Essa solução é ótima? Se não, tomando-a por base inicial, determine a solução ótima.

8 – Método Simplex – Fundamentos Teóricos



		x_1	x_2	x_3	x_4	x_5	x_6	
z	-40	b	d	0	0	f	0	
x_2	4	c	1	0	e	2	g	
x_4	-1	-5	0	1	-1	1	0	
x_6	a	2	0	0	0	3	1	

Determine os parâmetros livres no quadro de modo que:

- O quadro seja ótimo e conduza a uma solução única.
- A solução do quadro seja ótima, mas existe pelo menos uma outra solução ótima.
- A solução seja inviável.
- A solução seja viável mas ilimitada.
- O quadro represente uma solução viável que pode ser melhorada pela entrada de x_1 na base e a saída de x_6 .

9 – Método Simplex – Fundamentos Teóricos



Solucione o problema abaixo pelo método das duas fases introduzindo uma variável artificial de índice 5 (x_5) de custo igual a 1 e com um vetor de atividades igual a: $\begin{bmatrix} -1 \\ -1 \end{bmatrix}$.

$$\text{Minimizar } z = 2x_1 + 3x_2$$

sujeito a:

$$-x_1 - x_2 + x_3 = -3$$

$$-2x_1 - x_2 + x_4 = -2$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$$

Esse método poderia ser generalizado para qualquer PPL? Se possível descreva a generalização; caso contrário, justifique a impossibilidade.

10 – Método Simplex – Interpretação Geométrica



Determine o menor valor de K para que o problema abaixo seja viável:

$$\text{Minimizar } z = 2x_1 + 3x_2$$

sujeito a:

$$-2x_1 - x_2 + 3x_3 \geq 4$$

$$x_1 + 2x_2 \geq 6$$

$$3x_1 - x_2 + 2x_3 \leq 7$$

$$-x_1 + 5x_2 + x_3 \leq K$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$$

11 – Método Simplex – Aplicação



- Solucione manualmente os problemas de 1 a 7 da série de exemplos resolvidos do Capítulo 2.
- Solucione os demais problemas propostos no Capítulo 2 com o auxílio de um programa computacional adequado.

12 – Método Simplex – Aplicação



Uma companhia deve produzir 500 ferramentas para um de seus clientes em um certo espaço de tempo. Ela possui duas máquinas que podem produzir as ferramentas segundo a distribuição de custos constantes da tabela abaixo:

TABELA 3.2 CUSTOS E LIMITES DE PRODUÇÃO

Máquina	Custo unitário de produção (US\$)	Custo de montagem das máquinas (US\$)	Limite de produção no tempo disponível
1	1,12	60	300
2	1,23	50	270

- a) Solucione o problema de forma a minimizar o custo total da produção.
b) Seria possível solucionar esse problema com o uso do simplex? Justifique a resposta.

13 – Método Simplex – Aplicação



Determine as condições necessárias e suficientes para os valores dos parâmetros k e p no problema a seguir de modo que o modelo seja:

Minimizar $z = x_1 + x_2$

sujeito a:

$$kx_1 + px_2 \geq 1$$

$$x_1 \geq 0, x_2 \geq 0$$

1. Ilimitado.
2. Impossível.
3. Inviável.

4

DUALIDADE E SENSIBILIDADE

4.1 – CONCEITO DE DUALIDADE EM PROGRAMAÇÃO MATEMÁTICA

Dualidade é um conceito amplo que engloba a possibilidade do tratamento de duas naturezas distintas de uma mesma entidade. Inúmeros fenômenos físicos e químicos podem ser representados por modelos cujas estruturas e comportamentos são iguais; contudo, podem ser interpretados de modo completamente diferente. Na economia, engenharia e física esses casos são até comuns. Vamos, a título de exemplo, mostrar esse fenômeno no contexto dos circuitos elétricos. Sejam os circuitos representados na Figura 4.1 compostos por uma resistência R , uma indutância L e uma capacitância C .

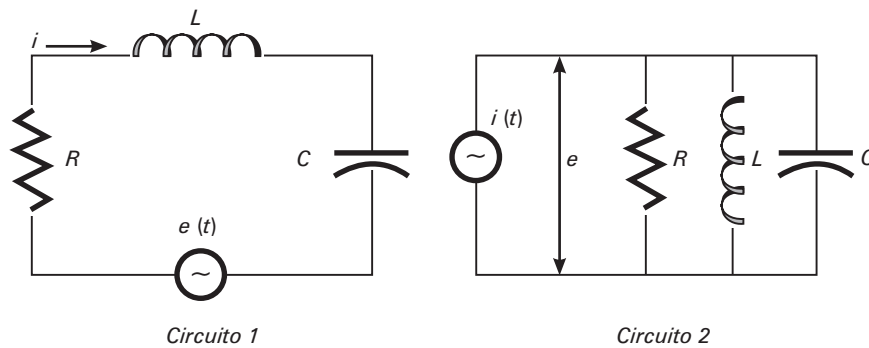


FIGURA 4.1 Circuitos duais.

A equação que relaciona a corrente à tensão como uma função do tempo para o circuito 1 é:

$$L \frac{di}{dt} + Ri + \frac{1}{C} \int i dt = e(t) \quad (4.1)$$

A equação que relaciona a tensão à corrente aplicada em função do tempo para o circuito 2 é:

$$C \frac{de}{dt} + Ge + \frac{1}{L} \int e dt = i(t) \quad (4.2)$$

onde $G = 1/R$. É evidente que as equações (12) e (13) têm uma mesma forma geral, a saber:

$$a \frac{dx}{dt} + bi + c \int u \, dt = y(t) \quad (4.3)$$

Como a equação geral (4.3) descreve tanto o circuito 1 como o circuito 2, podemos dizer que eles são duais. No caso dos modelos matemáticos, a definição de *dualidade* tem um cunho próprio e associado ao processo de solução e aplicação prática dos modelos. Em ambas as vertentes, a dualidade é hoje um instrumento indispensável.

Definiremos como *duais* um par de modelos de programação matemática denominados *primal* e *dual*, que preservam as seguintes condições:

- Possuem funções objetivo simétricas, ou seja, se o primal for de minimização o dual será de maximização e vice-versa.

Alguns autores fixam o primal como um problema de minimização ou de maximização, contudo isso não é necessário para caracterizar a dualidade. Como temos utilizado a forma padrão como um problema de minimização, consideraremos por conveniência, quase sempre, o primal como um problema de minimização. Fixar um modelo como representante de uma das classes dessa relação é no mínimo inconveniente porque a relação é reflexiva, ou seja, o dual do dual é o primal.

- Possuem simetria na descrição das restrições, ou seja, se na forma canônica o primal possui restrições \leq então o dual possuirá restrições \geq .
- Os termos independentes no primal surgem como os coeficientes da função objetivo no dual e vice-versa.
- O número de restrições do primal é igual ao número de variáveis do dual e vice-versa.
- A matriz de restrição do primal é a transposta da matriz de restrição do dual e vice-versa.

Considerando a seguinte convenção nos índices:

$$M = \{1, \dots, m\}; N = \{1, \dots, n\}; A = [a_{ij}];$$

$$\overline{M}_1 = M \setminus M_1 \text{ e } \overline{N}_1 = N \setminus N_1$$

Lembrando que x é um vetor coluna e u um vetor linha, definiremos o par de modelos primal \times dual como mostrado na Tabela 4.1.

TABELA 4.1 MODELOS PRIMAL \times DUAL

<i>Primal</i>		<i>Dual</i>
Minimizar: $z = \sum_{j \in N} c_j x_j$		Maximizar: $w = \sum_{i \in M} b_i u_i$
sujeito a		sujeito a
$\sum_{j \in N} a_{ij} x_j \geq b_i$	$i \in M_1$	$u_i \geq 0$
$\sum_{j \in N} a_{ij} x_j = b_i$	$i \in \overline{M}_1$	$u_i \in R$
$x_j \geq 0$	$j \in N_1$	$\sum_{i \in M} a_{ij} u_i \leq c_j$
$x_j \in R$	$j \in \overline{N}_1$	$\sum_{i \in M} a_{ij} u_i = c_j$

Ou ainda, considerando-se separadamente as forma padrão e canônica:

TABELA 4.2 FORMA PADRÃO E CANÔNICA

	<i>Primal</i>	<i>Dual</i>
<i>Forma Canônica</i>	$\text{Min } z = cx$ sujeito a: $Ax \geq b$ $x \geq 0$	$\text{Max } w = ub$ sujeito a: $uA \leq c$ $u \geq 0$
<i>Forma Padrão</i>	$\text{Min } z = cx$ sujeito a: $Ax = b$ $x \geq 0$	$\text{Max } w = ub$ sujeito a: $uA \leq c$ $u \in R$

4.2 – EXEMPLOS DE PARES PRIMAL X DUAL

Exemplo 1: Determinar o dual do seguinte modelo de programação linear:

$$\begin{aligned}
 & \text{(P) Maximizar } z = 6x_1 + 2x_2 + x_3 \\
 & \text{sujeito a:} \\
 & \quad x_1 - x_2 + 7x_3 \leq 4 \\
 & \quad 2x_1 + 3x_2 + x_3 \leq 5 \\
 & \quad x_1 \geq 0, x_2 \geq 0, x_3 \geq 0
 \end{aligned}$$

Aplicando as regras da dualidade temos que:

$$\begin{aligned}
 & \text{(D) Minimizar } w = 4u_1 + 5u_2 \\
 & \text{sujeito a:} \\
 & \quad u_1 + 2u_2 \geq 6 \\
 & \quad -u_1 + 3u_2 \geq 2 \\
 & \quad 7u_1 + u_2 \geq 1 \\
 & \quad u_1 \geq 0, u_2 \geq 0
 \end{aligned}$$

Exemplo 2: Determinar o dual do seguinte modelo de programação linear, representando graficamente ambos os modelos.

$$\begin{aligned}
 & \text{(P) Maximizar } z = 3x_1 + 4x_2 \\
 & \text{sujeito a:} \\
 & \quad x_1 - x_2 \leq -1 \\
 & \quad -x_1 + x_2 \leq 0 \\
 & \quad x_1 \geq 0, x_2 \geq 0
 \end{aligned}$$

Solução(D) Minimizar $w = -u_1 + 0u_2$

sujeito a:

$$u_1 - u_2 \geq 3$$

$$-u_1 + u_2 \geq 4$$

$$u_1 \geq 0, u_2 \geq 0$$

A Figura 4.2 compara os gráficos dos dois modelos. Notar que ambos são inconsistentes.

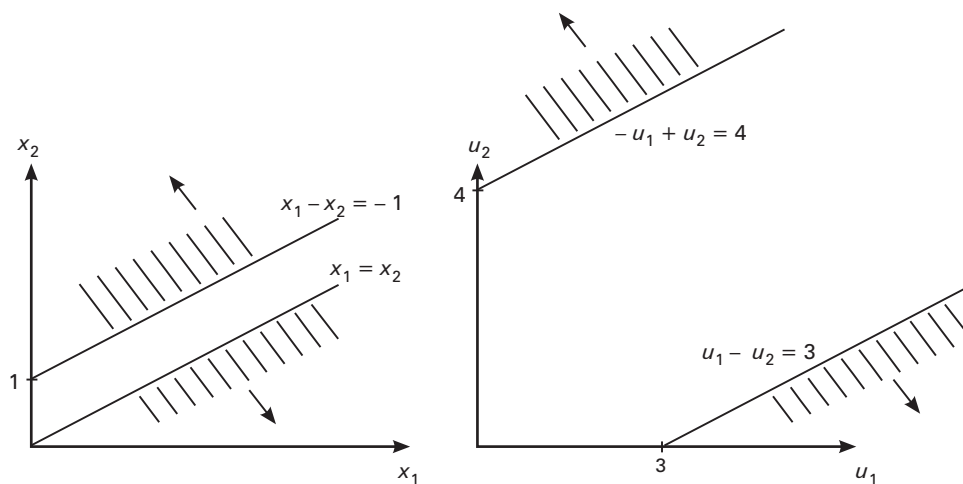


FIGURA 4.2 Representação gráfica do par primal x dual.

4.3 – TEOREMA DAS FOLGAS COMPLEMENTARES

4.3.1 – Fundamentação Teórica

Algumas importantes propriedades ligam os pares primal x dual. A primeira delas diz respeito ao valor das funções z_0 e w_0 quando alcançam seus valores ótimos. Seja o seguinte par de modelos primal x dual:

(P) Min $z = cx$	(D) Max $w = ub$
sujeito a:	sujeito a:
$Ax \geq b$	$uA \leq c$
$x \geq 0$	$u \geq 0$

Proposição 4.1

Se \bar{x} e \bar{u} são soluções viáveis dos problemas (P) e (D) respectivamente, então: $c\bar{x} \geq \bar{u}b$.

Prova:

Sabemos das restrições do problema primal que $A\bar{x} \geq b$.

Se pré-multiplicarmos a expressão por \bar{u} temos:

$$\bar{u}A\bar{x} \geq \bar{u}b, \text{ uma vez que } \bar{u} \geq 0 \quad (4.4)$$

Como $\bar{u}A \leq c$ chegamos, por raciocínio semelhante, a:

$$\bar{u}A\bar{x} \leq \bar{c}\bar{x}, \text{ uma vez que } \bar{x} \geq 0 \quad (4.5)$$

Comparando as expressões (10) e (11) temos:

$$\bar{u}b \leq \bar{u}A\bar{x} \leq \bar{c}\bar{x} \quad \therefore \quad \bar{c}\bar{x} \geq \bar{u}b$$

Proposição 4.2

Se \bar{x} e \bar{u} são soluções viáveis dos problemas (P) e (D), respectivamente, satisfazendo a $\bar{c}\bar{x} = \bar{u}b$, então ambos são soluções ótimas dos correspondentes problemas.

Prova:

Suponhamos que \bar{x} não seja uma solução ótima do primal e $\bar{c}\bar{x} = \bar{u}b$.

Se \bar{u} é solução ótima do dual e \bar{x} não é solução ótima do primal, então, nesse caso, existe x^* tal que $cx^* < \bar{u}b$, o que contraria a proposição 1.

Suponhamos agora que \bar{u} não é solução ótima do dual, mas $\bar{c}\bar{x} = \bar{u}b$, sendo \bar{x} solução ótima do primal.

Nesse caso, existirá u^* tal que $u^*b > \bar{u}b$, e, portanto, $u^*b > \bar{c}\bar{x}$, o que contraria novamente a proposição 1.

Logo, se não existe x^* e u^* que possam violar as duas suposições anteriores, \bar{x} e \bar{u} são soluções ótimas.

O par de problemas primal \times dual é também unido através de propriedades mais amplas que ligam seus universos de soluções viáveis. O Quadro 4.1 resume as situações possíveis.

QUADRO 4.1 SITUAÇÕES DE VIABILIDADE DOS PROBLEMAS PRIMAL \times DUAL

		Dual	
		Viável	Inviável ($U = \emptyset$)
Primal	Viável	Min $cx = \text{Max } ub$	$cx = -\infty$
	Inviável ($X = \emptyset$)	$ub = +\infty$	possível

O teorema a seguir garante a validade do Quadro 4.1:

Teorema da Existência

Para um par de problemas duais, uma e somente uma das alternativas abaixo é verdadeira:

- Nenhum dos problemas tem solução.
- Um deles não tem solução viável e o outro tem solução ótima ilimitada.
- Ambos possuem solução ótima finita.

Prova:

Sejam os conjuntos de soluções viáveis para (P) e (D):

$$\begin{aligned} X &= \{x/Ax \geq b, x \geq 0\} \\ U &= \{u/uA \leq c, u \geq 0\} \end{aligned}$$

Suponhamos que o programa dual (D) tenha solução viável, mas que não tenha solução ótima finita. Então:

$$\text{Max } \{w = ub\} > +\infty$$

Para que a proposição 4.1 se verifique, deve existir $x \in X$ tal que $cx > +\infty$, o que não faz sentido, uma vez que o primal (P) é de minimização, daí:

$$X = \emptyset$$

Se supusermos que o programa primal não tem solução ótima finita o mesmo raciocínio será aplicável ao dual que, obrigatoriamente, deverá ter $U = \emptyset$.

Por outro lado, se $X = \emptyset$ implica que não existam limites para o valor das soluções do problema dual, ou seja, esse problema será vazio ou ilimitado. Como a relação de dualidade é reflexiva o mesmo raciocínio se aplica para o primal quando $U = \emptyset$.

Finalmente, para demonstrar a terceira afirmativa, seja $Ax - Ix_e = b$ o sistema primal incluindo as variáveis de folga x_e .

Consideremos B uma base viável ótima do primal e $\bar{x}_B = B^{-1}b$ a respectiva solução básica viável ótima. Como a solução é ótima, por hipótese, temos:

$$z_j - c_j \leq 0, \quad \forall j \in N = I \cup J$$

ou seja:

$$c_B B^{-1} a_j - c_j \leq 0 \quad \therefore$$

$$c_B B^{-1} a_j \leq c_j, \quad \forall j \in N$$

Fazendo:

$$c_B B^{-1} = \bar{u}$$

temos:

$$\bar{u} a_j \leq c_j, \quad \forall j \in N \quad \therefore$$

$$\bar{u} A \leq c$$

ou seja, \bar{u} é uma solução do programa dual. Para verificarmos a viabilidade, consideremos os $z_j - c_j$ para as variáveis de folga. Então:

$$c_B B^{-1} a_j \leq c_j, \text{ onde } a_j = -e_i \text{ e } c_j = 0 \quad \therefore$$

$$-c_B B^{-1} e_i \leq 0 \quad \therefore$$

$$\bar{u} e_i \geq 0 \text{ ou } \bar{u}_i \geq 0 \text{ ou ainda } \bar{u} I \geq 0 \text{ e } \bar{u} \geq 0$$

Assim,

$$U \neq \emptyset$$

Resta-nos mostrar que a solução \bar{u} é ótima. Representando por (*) a situação de ótimo associada ao vetor ou matriz em consideração temos:

$$w = \bar{u}b = c_B B^{-1}b = c_B \bar{x}_B \text{ e}$$

$$z^* = c_B \bar{x}_B$$

Logo $w^* = z^*$ e, portanto, \bar{u} é ótima.

A título de esclarecimento apresentamos o exemplo 3 que aborda uma situação de existência primal \times dual $\infty \Rightarrow \emptyset$.

Exemplo 3: Determinar o dual do modelo de programação linear a seguir, representando ambos os programas graficamente.

(P) Maximizar $z_0 = -3x_1 + 2x_2$
sujeito a:

$$\begin{aligned} x_1 &\leq 3 \\ x_1 - x_2 &\leq 0 \\ x_1 &\geq 0, x_2 \geq 0 \end{aligned}$$

Solução:

O dual de (P) será um programa de minimização cuja matriz de restrições será a transposta de (P), terá o termo independente igual ao vetor custos, o vetor custos igual ao termo independente de (P). Daí:

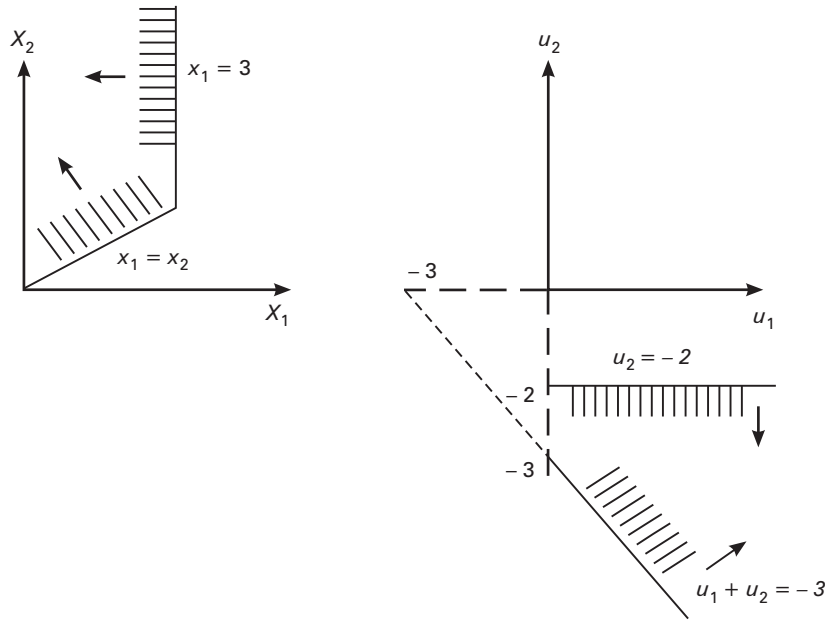
(D) Minimizar $w_0 = -3u_1 + 0u_2$
sujeito a:

$$\begin{aligned} u_1 + u_2 &\geq -3 \\ u_2 &\leq -2 \\ u_1 &\geq 0, u_2 \geq 0 \end{aligned}$$

A Figura 4.3 mostra que o programa primal é ilimitado e o dual é inviável, ou seja, $X = \infty$ e $U = \emptyset$. Observar que o dual não foi descrito na forma canônica.

4.3.2 – Teorema das Folgas Complementares

No item anterior desenvolvemos os elementos que permitiram estabelecer uma relação entre o valor das soluções ótimas do par primal \times dual. Além dos valores numéricos em si, observamos que existe, inclusive, uma dependência entre a condição de viabilidade de cada um desses modelos. No presente item iremos mais longe, explorando as condições de comportamento entre as variáveis desses programas.

FIGURA 4.3 Exemplo de par primal \times dual.

Teorema das Folgas Complementares

Dado um par de programas duais, uma condição necessária e suficiente para que as soluções \bar{x} e \bar{u} sejam ótimas é que se verifiquem as seguintes relações de complementaridade de folga:

$$\begin{aligned} u(A\bar{x} - b) &= 0 \\ (c - \bar{u}A)\bar{x} &= 0 \end{aligned}$$

Prova:

Sendo \bar{x} e \bar{u} soluções viáveis teremos:

$$\begin{cases} A\bar{x} \geq b & \therefore A\bar{x} - b \geq 0 & \therefore \bar{u}(A\bar{x} - b) \geq 0 \\ \bar{u}A \leq c & \therefore c - \bar{u}A \geq 0 & \therefore (c - \bar{u}A)\bar{x} \geq 0 \end{cases}$$

Fazendo

$$\begin{cases} \alpha = \bar{u}(A\bar{x} - b) \\ \beta = (c - \bar{u}A)\bar{x} \end{cases}$$

teremos:

$$\begin{aligned} \alpha + \beta &= \bar{u}(A\bar{x} - b) + (c - \bar{u}A)\bar{x} = \\ &= -\bar{u}b + c\bar{x} \geq 0 \end{aligned}$$

Se \bar{x} e \bar{u} forem soluções ótimas, teremos:

$$\bar{u}b = c\bar{x}$$

Logo,

$$\alpha + \beta = 0 \text{ com } \alpha, \beta \geq 0, \text{ ou seja:}$$

$$\alpha = \beta = 0$$

Assim chegamos finalmente a:

$$\begin{cases} \alpha = \bar{u}(A\bar{x} - b) = 0 \\ \beta = (c - \bar{u}A)\bar{x} = 0 \end{cases}$$

4.3.3 – Uma Aplicação do Teorema das Folgas Complementares

Vamos esclarecer as aplicações matemáticas dos teoremas da dualidade através de um exemplo numérico. Em tópico específico faremos comentários sobre interpretações econômicas dos teoremas.

Exemplo 4: Solucione o modelo de programação linear que se segue com auxílio das relações expressas pela dualidade.

$$\begin{aligned} \text{(P) Minimizar } z_0 &= 2x_1 + x_2 + 4x_3 \\ \text{sujeito a:} \\ -2x_1 + x_2 + x_3 &\geq 1 \\ -x_1 + 2x_2 - x_3 &\leq 1 \\ x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0 \end{aligned}$$

Solução:

Vamos optar por solucionar graficamente (P) através de seu dual. Como já sabemos, o dual desse programa possuirá apenas duas variáveis reais (número de linhas do primal) sendo, portanto, solucionável graficamente. O dual de (P) será:

$$\begin{aligned} \text{(D) Maximizar } w_0 &= u_1 - u_2 \\ \text{sujeito a:} \\ -2u_1 + u_2 &\leq 2 \\ u_1 - 2u_2 &\leq 1 \\ u_1 + u_2 &\leq 4 \\ u_1 \geq 0, \quad u_2 \geq 0 \end{aligned}$$

cujas representação gráfica está representada na Figura 4.4.

Com o valor das variáveis duais conhecidas poderemos calcular as variáveis primais? Aplicando as relações da complementaridade de folga temos:

$$\bar{u}(A\bar{x} - b) = 0 \text{ e } (c - \bar{u}A)\bar{x} = 0$$

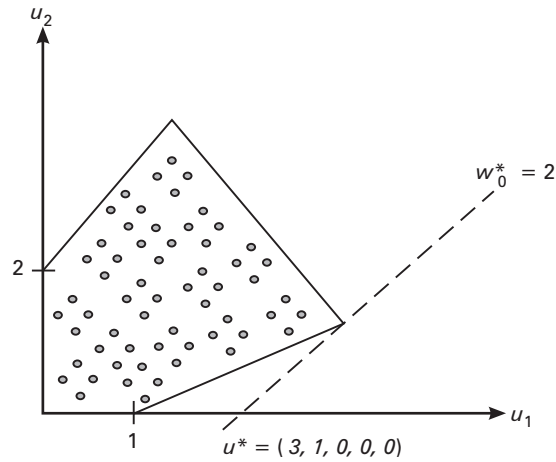


FIGURA 4.4 Solução gráfica.

Em consequência,

$$\begin{cases} u_1^*(-2x_1^* + x_2^* + x_3^* - 1) = 0, & \text{como } u_1^* = 3, \text{ então } -2x_1^* + x_2^* + x_3^* - 1 = 0 \\ u_2^*(x_1^* - 2x_2^* + x_3^* + 1) = 0, & \text{como } u_2^* = 1, \text{ então } x_1^* - 2x_2^* + x_3^* + 1 = 0 \\ x_1^*(-2u_1^* + u_2^* - 2) = 0 \therefore x_1^* = 0 \\ x_2^*(u_1^* - 2u_2^* - 1) = 0 \therefore x_2^* \geq 0 \\ x_3^*(u_1^* + u_2^* - 4) = 0 \therefore x_3^* \geq 0 \end{cases}$$

O que nos permite concluir em relação ao primal que:

$$\begin{aligned} x_2^* + x_3^* &= 1 \\ -2x_2^* + x_3^* &= -1 \end{aligned}$$

Sistema determinado que, quando solucionado, nos leva a:

$$\begin{aligned} x_2^* &= 2/3 \\ x_3^* &= 1/3 \end{aligned}$$

e, finalmente, à solução primal:

$$z_0^* = 2, x^* = (0, 2/3, 1/3, 0, 0)$$

4.4 – UM ALGORITMO DUAL PARA O MÉTODO SIMPLEX

4.4.1 – Um Exemplo do Funcionamento do Primal Simplex no Dual

Um modo simples e direto de percebermos o que significa realmente o processo de pivoteamento do algoritmo simplex é acompanhar seu desenvolvimento simultâneo em um par de problemas primal \times dual. Sejam os seguintes problemas:

(P) Max: $z_0 = 6x_1 + 2x_2 + x_3$
sujeito a:

$$\begin{aligned} x_1 - x_2 + 7x_3 &\leq 4 \\ 2x_1 + 3x_2 + x_3 &\leq 5 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{aligned}$$

(D) Min: $w_0 = 4u_1 + 5u_2$
sujeito a:

$$\begin{aligned} u_1 + 2u_2 &\geq 6 \\ -u_1 + 3u_2 &\geq 2 \\ 7u_1 + u_2 &\geq 1 \\ u_1 \geq 0, u_2 \geq 0 \end{aligned}$$

que podem ser escritos em dois quadros de cálculo sem as variáveis artificiais e com as variáveis de folga positivas (obtidas com a multiplicação da primeira e segunda linhas do dual por -1) da seguinte forma:

		u_1	u_2	u_3	u_4	u_5
w		-4	-5	0	0	0
u_3	-6	-1	-2	1	0	0
u_4	-2	1	-3	0	1	0
u_5	-1	-7	-1	0	0	1

QUADRO 4.2

		x_1	x_2	x_3	x_4	x_5
z		6	2	1	0	0
x_4	4	1	-1	7	1	0
x_5	5	2	3	1	0	1

QUADRO 4.3

Se tentarmos aplicar as regras de pivoteamento nos quadros formados não teremos problema com o Quadro 4.3, que corresponde ao primal; contudo, o Quadro 4.2 não é viável! Existem valores no vetor \bar{x}_B negativos. É mais interessante ainda é que os “custos reduzidos” desse primeiro quadro indicam regra de parada, com todos os $z_j - c_j \leq 0$. Sem condições de solucionar o Quadro 4.2, vamos realizar então um pivoteamento no Quadro 4.3 e acompanhar o que acontece no Quadro 4.2 quando procuramos compatibilizá-lo com o Quadro 4.3.

Escolha das variáveis que entram e saem da base:

↓

		u_1	u_2	u_3	u_4	u_5
w		-4	-5	0	0	0
u_3	-6	-1	-2	1	0	0
u_4	-2	1	-3	0	1	0
u_5	-1	-7	-1	0	0	1

QUADRO 4.2 PIVÔ

↑

		x_1	x_2	x_3	x_4	x_5
z		6	2	1	0	0
x_4	4	1	-1	7	1	0
x_5	5	2	3	1	0	1

QUADRO 4.3 PIVÔ

Pivoteamento:

		u_1	u_2	u_3	u_4	u_5
w	15	-3/2	0	-5/2	0	0
u_2	3	1/2	1	-1/2	0	0
u_4	7	5/2	0	-3/2	1	0
u_5	2	13/2	0	1/2	0	1

QUADRO 4.2 APÓS O PIVOTEAMENTO FORÇADO

		x_1	x_2	x_3	x_4	x_5
z	-15	0	-7	-2	0	-3
x_1	3/2	0	-5/2	13/2	1	-1/2
x_5	5/2	1	3/2	1/2	0	1/2

QUADRO 4.3 APÓS O PIVOTEAMENTO PRIMAL

No caso, encontramos agora condições de viabilidade e regra de parada nos dois quadros. Se imaginarmos que o nosso problema era constituído pelo Quadro 4.2, o processo de compatibilização deste com o Quadro 4.3 acabou por forçar um pivoteamento sobre uma linha inviável, o que não era uma operação válida para o algoritmo primal simplex. Apesar de esse fato singular ocorrer no pri-

mal, no dual as regras do simplex estão sendo perfeitamente atendidas. Nesse caso, se prosseguirmos aplicando o algoritmo simplex ao dual, chegaremos à solução ótima, o que nos sugere que o processo adaptado ao primal será também capaz de convergir para uma solução. No exemplo, o programa primal (Quadro 4.2) não possuía uma solução viável de partida, mas o dual sim. De fato, ao utilizarmos o algoritmo simplex para solucionar, via problema dual, um problema primal cuja base viável inicial não era disponível, estamos desenhando um novo algoritmo. Podemos descrever esse algoritmo da seguinte forma:

- Escolha o termo de \bar{x}_B mais negativo para sair da base.
- Se existem coeficientes ≤ 0 (negativos) na linha, escolha a menor razão entre os coeficientes e os “custos reduzidos”, ou seja: $\frac{z_s - c_s}{y_{sk}} = \underset{1 \leq i \leq m}{\text{mínimo}} \left\{ \frac{z_i - c_i}{y_{ik}} : y_{ik} > 0 \right\} \Rightarrow x_s$, determinando com isso a variável que sai da base.
- Caso contrário, o problema é ilimitado. Pare.
- Realize a operação de pivoteamento (sobre o pivô negativo) e canonize a coluna do pivô que entra na base, como previsto no algoritmo simplex.
- Verifique as condições de viabilidade do problema. Se $\bar{x}_B \geq 0$, então o ótimo foi encontrado; caso contrário, retorne ao procedimento de troca de variáveis na base.

Verificamos que a condição de parada no primal corresponde à condição de viabilidade no dual e vice-versa.

4.5 – INTERPRETAÇÃO ECONÔMICA

4.5.1 – Aplicações dos Métodos Quantitativos no Contexto de uma Economia Racional

A escala de globalização e busca da eficácia organizacional tem forçado cada vez mais a utilização de técnicas quantitativas avançadas para a tomada de decisão nos sistemas de produção. Nesse contexto, a programação matemática torna-se um instrumento seguro para uma análise econômica racional, principalmente porque permite a criação de modelos bastante sofisticados e específicos para a representação dos fenômenos econômicos.

A filosofia de substrato para a utilização do método simplex no auxílio a uma análise econômica está ancorada no conceito do perfeito equilíbrio entre o valor agregado por um processo a um produto e sua planilha de decomposição em insumos de constituição (teoria do custo de oportunidade, Menger [1871]).

Se um sistema de produção é modelado por um programa linear cuja função objetivo é, por exemplo, maximizar o valor dos produtos realizados e cujas restrições são de disponibilidade de fatores, veremos que, no equilíbrio da solução ótima, o valor de uma variável dual associada a um fator satisfaz à seguinte afirmação daquela teoria:

O valor da unidade marginal de um fator de produção é igual ao máximo valor da produção que poderia ser obtida usando essa unidade do fator.

Além dessa clássica interpretação da teoria marginalista, existem outras propriedades econômicas da programação linear que não têm sido frisadas em textos sobre o assunto. Além de estarem associadas a valores marginais, as variáveis duais também estão associadas a valores de equilíbrio financeiro

(isto é, tais que o custo seja igual à receita) para as atividades econômicas em consideração. Assim, com as variáveis duais obtém-se um tipo de informação local (isto é, marginal) sobre o sistema. Mas há também um tipo de informação global que desejamos enfatizar em nossa interpretação econômica do método simplex e da dualidade em programação linear.

Propomos uma abordagem complementar à desenvolvida classicamente. Trocaremos a visão de maximização de lucros (diferença entre benefício e custo) na face primal, pela minimização de custos. Consideraremos na vertente dual o benefício da produção.

Esse modelo está voltado para o entendimento do processo de interação entre o sistema e seu meio ambiente, em perfeita sintonia com as mais modernas filosofias da qualidade total e das gestões para a excelência. Ao se propor a produzir, o sistema assume certos compromissos com o meio ambiente organizacional que denominaremos *vínculos*. Admitamos que tais vínculos sejam mensuráveis e que o sistema tenha escalas lineares de utilidade (preços) dos produtos. Admitamos igualmente que cada produto tenha escalas lineares de participação dentro de cada vínculo.

Nesse modelo, o sistema tomará decisões guiado pelo objetivo de maximizar a utilidade de produção ou minimizar os custos dos vínculos. Uma solução qualquer em que a utilidade da produção (receita) seja maior ou igual ao custo dos vínculos será compensadora. Por outro lado, uma solução em que a utilidade de transformar é inferior ao custo não será realizável. Veremos que a uma solução genérica do método simplex, aplicado a um problema primal cujo objetivo é maximizar a utilidade dos produtos, correspondem variáveis duais que medem os limites máximos dos custos dos vínculos para os quais a produção calculada ainda é economicamente viável ou não deficitária. Em outras palavras, a cada solução do método simplex calculam-se custos dos recursos para os quais o preço de cada produto realizado é exatamente igual ao seu custo de produção.

Se a solução corrente ainda não é ótima, o teste de otimização do simplex identifica, pelo menos, um produto não presente na solução (não básico) cuja utilidade relativa (lucro) é positiva. Isso significa que se esse produto participar de uma nova solução em detrimento das participações dos produtos da presente solução, teremos uma melhora em relação a essa escala linear com o coeficiente igual à utilidade relativa. Essa utilidade relativa de um produto corresponde ao grau da inviabilidade da restrição do dual associada a esse produto. Ou seja, o nível da inviabilidade de uma restrição do dual associada a um determinado produto não produzido é igual à utilidade marginal de sua produção.

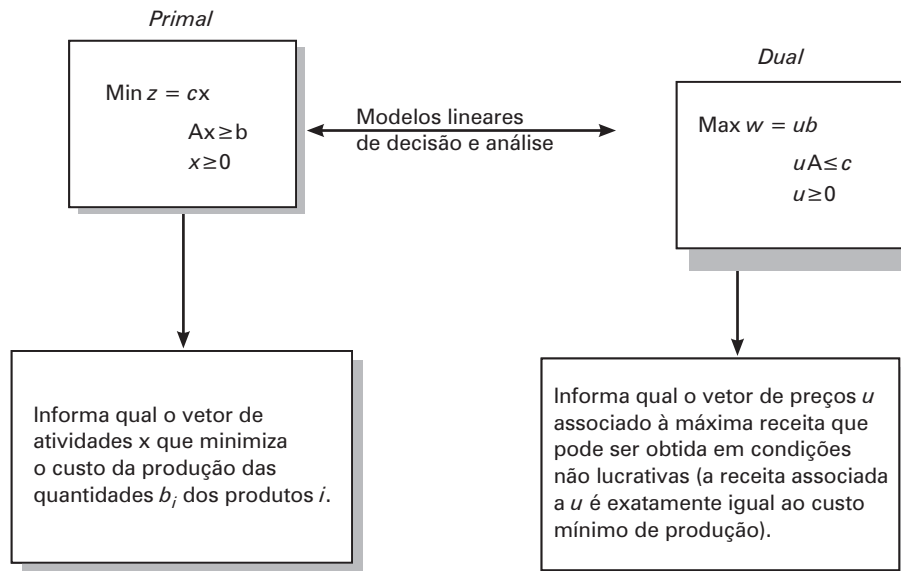
A consequência imediata da não completa realização da utilidade marginal é a negação da solução ótima. Enquanto existir utilidade marginal positiva, ainda é possível melhorar a solução. A restrição do dual no quadro simplex nos diz que essa utilidade marginal é a diferença entre a utilidade (receita) de uma unidade do produto e o custo associado à produção dessa unidade, se o custo dos vínculos forem avaliados de modo a permitirem equilíbrio financeiro para a solução corrente.

A Figura 4.5 resume o aspecto do fluxo de informações que transita entre o par primal \times dual durante um processo de solução do algoritmo simplex. Notamos claramente a dinâmica da busca do equilíbrio de um mínimo \times máximo global.

4.5.2 – Exemplo de Interpretação Econômica do Método Simplex

Suponhamos o caso hipotético de uma empresa pública que monopolize o setor petroquímico de uma economia nacional ou regional. Digamos que o objetivo da companhia seja minimizar o custo global da produção de gasolina e de óleo, estando comprometida em satisfazer às demandas desses produtos aos níveis estipulados de respectivamente 5 milhões de barris e 3 milhões de barris. Suponhamos que a companhia decida levar em frente duas atividades principais com o objetivo de atender à demanda. A primeira atividade é explorar e refinar petróleo nacional. Cada 1 milhão de barris processados custa 8 milhões de dólares e gera 1 milhão de barris de gasolina e 1 milhão de barris de óleo. A segunda atividade é importar e refinar petróleo estrangeiro. Nesse caso, cada 1 milhão de barris importados custa 12 milhões de dólares e gera 2 milhões de barris de gasolina e 1 milhão de barris de óleo.

Denominando por x_i , $i = 1, 2$, o valor de operação de cada uma dessas atividades, o problema de programar a compra e refino do petróleo pode ser modelado e solucionado pelo programa descrito a seguir:

FIGURA 4.5 Informações entre o par primal \times dual.

$$\text{Minimizar } z = 8x_1 + 12x_2$$

sujeito a:

$$x_1 + 2x_2 \geq 5$$

$$x_1 + x_2 \geq 3$$

$$x_1 \geq 0, x_2 \geq 0$$

cuja solução é $z_0^* = 32, x_1 = 1, x_2 = 2$ como podemos ver na Figura 4.6.

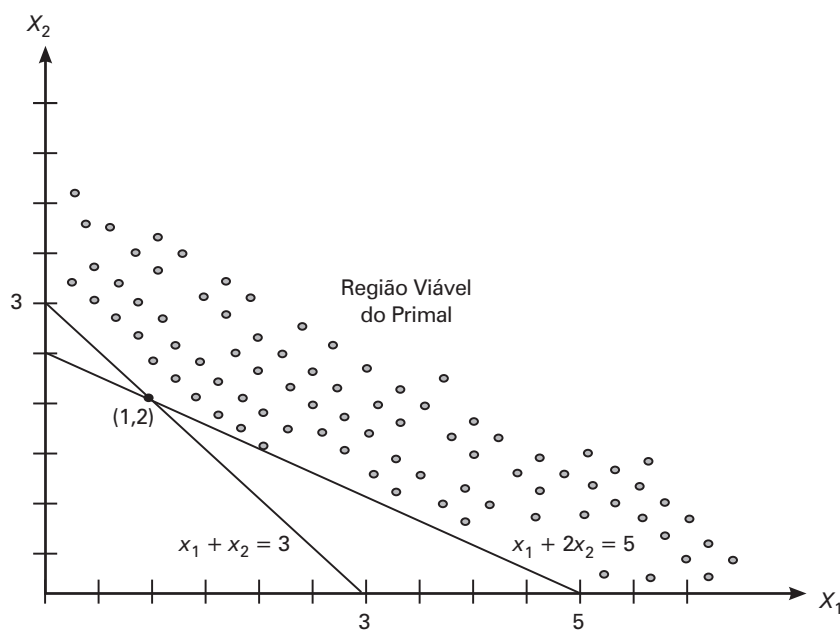


FIGURA 4.6 Representação gráfica do problema.

O setor de novos projetos da empresa está desenvolvendo um processo de transformação de xisto betuminoso no qual gasta 15 milhões de dólares em cada milhão de toneladas de xisto processado, gerando-se 2 milhões e 1,5 milhão de toneladas de gasolina e óleo. O problema a resolver, nesse novo cenário, passa a ser:

$$\text{Minimizar } z = 8x_1 + 12x_2 + 15x_3$$

sujeito a:

$$x_1 + 2x_2 + 2x_3 \geq 5$$

$$x_1 + x_2 + 3/2x_3 \geq 3$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$$

Podemos resolver o problema agora pelo método simplex, acrescentando as variáveis de folga x_4 e x_5 , associadas às restrições 1 e 2. Devemos também acrescentar variáveis artificiais x_6 e x_7 para obter uma base viável inicial, já que as variáveis de folga são negativas devido às restrições de maior ou igual. Podemos usar o artifício de penalizar fortemente as variáveis artificiais para que elas logo deixem a base e permitam a constituição de uma base viável real no desenvolver do algoritmo. Podemos então constituir o programa abaixo:

$$\text{Minimizar } z = 8x_1 + 12x_2 + 15x_3 + 0x_4 + 0x_5 + 40x_6 + 18x_7$$

sujeito a:

$$x_1 + 2x_2 + 2x_3 - x_4 + x_6 = 5$$

$$x_1 + x_2 + 3/2x_3 - x_5 + x_7 = 3$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0, x_6 \geq 0, x_7 \geq 0$$

É interessante notar que, no programa, todas as variáveis utilizadas possuem interpretação econômica. As variáveis de folga x_4 e x_5 podem ser vistas como quantidades de estocagem de óleo e gasolina. Nesse caso, supõe-se nulo o custo da estocagem de cada tipo de produto. As variáveis artificiais representam os custos exorbitantes de 40 e 18 milhões de dólares por 1 milhão de barris de, respectivamente, gasolina e óleo importados. A ideia de fase 1 pelo método do M grande é de que, ao se estabelecer um custo exorbitante (M grande) para cada variável artificial se iniba o uso dessa variável.

Atividade	Explorar Petróleo Nacional	Importar e Refinar Petróleo	Explorar Xisto	Estocar Gasolina	Estocar Óleo	Importar Gasolina	Importar Óleo	Demanda (Milhões de Barris)
Produtos	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
Gasolina	1	2	2	-1	0	1	0	5
Óleo	1	1	3/2	0	-1	0	1	3
Custos	8	12	15	0	0	40	18	-

Vamos agora solucionar o programa utilizando o método simplex revisado. Faremos as devidas intervenções para justificar os passos de solução adotados sob a ótica da visão econômica.

Passo 0: Inicialmente identifiquemos os elementos de cálculo, que são a matriz A , o veto b e o vetor c :

$$A = [a_1, a_2, a_3, a_4, a_5, a_6, a_7] = \begin{bmatrix} 1 & 2 & 1 & -1 & 0 & 1 & 0 \\ 1 & 1 & 3/2 & 0 & -1 & 0 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} 5 \\ 3 \end{bmatrix} \text{ e } c = [8 \quad 12 \quad 15 \quad 0 \quad 0 \quad 40 \quad 18]$$

Consideremos a base composta pelas variáveis x_6, x_7 , ou seja:

$$B = [a_6, a_7] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = B^{-1}$$

$$u = (u_1 u_2) = c_B \times B^{-1} = (40 \ 18) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = (40 \ 18)$$

Nesse caso, as variáveis x_1, x_2, x_3, x_4 e x_5 não são básicas. $J = \{1, 2, 3, 4, 5\}$, $I = \{6, 7\}$

$$x_b = \begin{bmatrix} x_6 \\ x_7 \end{bmatrix} = B^{-1}b = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$$

$$x_R = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Passemos agora para a fase de solução.

Passo 1: escolher a variável que entrará na base.

Como o vetor c_B é composto por variáveis artificiais que possuem custos exorbitantes (c_B é diferente de zero desde a primeira iteração) temos de calcular os $z_j - c_j$ para $j \in J$, a fim de obter o critério de escolha da variável que deverá entrar na base como se segue:

Escolher a variável k para entrar na base tal que $z_k - c_k = \max_{j \in J_1} \{z_j - c_j\}$. Daí:

$$\begin{aligned} z_1 - c_1 &= c_B B^{-1} a_1 - c_1 \\ &= [40 \ 18] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} - (8) = 50 \end{aligned}$$

$$\begin{aligned} z_2 - c_2 &= \\ &= [40 \ 18] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} - (12) = 86 \end{aligned}$$

$$\begin{aligned} z_3 - c_3 &= \\ &= [40 \ 18] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3/2 \end{bmatrix} - (15) = 92 \end{aligned}$$

$$\begin{aligned} z_4 - c_4 &= \\ &= [40 \ 18] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \end{bmatrix} - (0) = -40 \end{aligned}$$

$$\begin{aligned} z_5 - c_5 &= \\ &= [40 \ 18] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \end{bmatrix} - (0) = -18 \end{aligned}$$

Do cálculo anterior escolheremos a variável 3 para entrar na base porque ela possui o máximo entre os $z_j - c_j$ positivos, ou seja, $z_3 - c_3 = 92$. A interpretação econômica é de que, se os preços da gasolina e do óleo são de, respectivamente, 40 e 18, a atividade de explorar o xisto daria o lucro fabuloso de $z_3 - c_3 = 92$, já que a esses preços dos produtos a receita unitária da operação do xisto seria de $40.2 + 18.3/2 = 107$, contra um custo de apenas 15.

Ao escolhermos uma variável para ingressar na base estamos procurando atividades mais lucrativas do que as da solução corrente. As atividades 1, 2 e 3 são mais lucrativas do que a importação exorbi-

tante. Isso era um fato esperado; contudo, não pudemos evitá-lo já que não estava disponível, inicialmente, um esquema viável. Partindo de um “absurdo” pensamos em utilizar o próprio simplex para encontrar a tal solução “razoável” (básica viável) que nos faltava.

Essa estratégia praticamente está sugerindo duas fases de solução:

- Fase da busca da viabilidade econômica.
- Fase da busca da melhor solução viável possível.

A interpretação econômica neste instante vai mais longe. Se, por algum motivo, o algoritmo oferecer, ao final de seu desenvolvimento, ainda uma solução “absurda”, ou seja, incluindo a importação exorbitante, então podemos ter certeza de que não existe nenhuma solução “razoável” melhor. Em outras palavras, se o algoritmo não for capaz de encontrar pelo menos uma saída “razoável”, é porque ela não existe, e o problema é economicamente absurdo (inviável) desde o começo.

Continuando o cálculo, vamos para o Passo 2.

Passo 2: verificar viabilidade e escolha da variável que deixa a base.

Pelo critério adotado para a melhoria, a variável x_3 entrará na base saindo a variável

$\frac{\bar{b}_s}{y_{sk}} = \text{mínimo}_{1 \leq i \leq m} \left\{ \frac{\bar{b}_i}{y_{ik}} : y_{ik} > 0 \right\}$, que no caso é x_7 , conforme o cálculo abaixo demonstra:

$$\bar{b} = \begin{bmatrix} 5 \\ 3 \end{bmatrix} \text{ e } y_3 = \begin{bmatrix} 2 \\ 3/2 \end{bmatrix} \text{ logo:}$$

$$\leftarrow \begin{bmatrix} x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \end{bmatrix} / \begin{bmatrix} 2 \\ 3/2 \end{bmatrix} = \begin{bmatrix} 2,5 \\ 2 \end{bmatrix}$$

Valor mínimo, ($s = 2$ – segunda linha)

A nova base será formada pelas variáveis x_6, x_3 , ou seja:

$$B' = [a_6, a_3] = \begin{bmatrix} 1 & 2 \\ 0 & 3/2 \end{bmatrix}$$

Passo 3: atualizar os elementos de cálculo.

O elemento fundamental de cálculo é a inversa da base. É com ela que poderemos atualizar os demais vetores. O método de inversão da base, de fato, não caracteriza o método simplex; contudo, tradicionalmente ele é feito por pivoteamento pelo motivo que debatemos no capítulo anterior. Podemos inverter a base utilizando o pivoteamento no contexto do método simplex revisado. Nesse caso, faremos a inversão da matriz aumentada $B^{-1} | y_3$, através da canonização da coluna 3 (genericamente coluna k) junto à matriz já invertida no passo anterior (notar que só no primeiro passo vamos ter $B = B^{-1}$). A seta significa a operação de pivoteamento:

$$\left[\begin{array}{c|c} \text{antiga } B^{-1} & y_k \end{array} \right] \xrightarrow{\text{Pivoteamento}} \left[\begin{array}{c|c} \text{nova } B^{-1} & \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \end{array} \right]$$

ou seja:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2/3 \end{bmatrix} \xrightarrow{\text{Pivoteamento}} \begin{bmatrix} 1 & -4/3 \\ 0 & 2/3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot$$

$$B^{-1} = \begin{bmatrix} 1 & -4/3 \\ 0 & 2/3 \end{bmatrix}$$

$$\bar{x}_B = \bar{b} = B^{-1}b = \begin{bmatrix} 1 & -4/3 \\ 0 & 2/3 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$u = (u_1 u_2) = c_B B^{-1} = [40 \quad 15] \begin{bmatrix} 1 & -4/3 \\ 0 & 2/3 \end{bmatrix} = [40 \quad -130/3]$$

$$z = z - (z_3 - c_3) \frac{\bar{x}_{B2}}{y_{23}} = 254 - 92 \frac{2}{1} = 70$$

Podemos também calcular z através de:

$$z = c_B \bar{x}_B = [40 \quad 15] \begin{bmatrix} 1 \\ 2 \end{bmatrix} = 70$$

Neste ponto, as variáveis x_1, x_2, x_4, x_5 e x_7 não são básicas. $J = \{1, 2, 4, 5, 7\}$, $I = \{6, 3\}$.

Vamos, novamente, ao Passo 1.

Passo 1: escolha da nova variável que entrará na base.

$$z_1 - c_1 = ua_1 - c_1 = [40 \quad -130/3] \begin{bmatrix} 1 \\ 1 \end{bmatrix} - (8) = -34/3$$

$$z_2 - c_2 = [40 \quad -130/3] \begin{bmatrix} 2 \\ 1 \end{bmatrix} - (12) = 74/3$$

$$z_4 - c_4 = [40 \quad -130/3] \begin{bmatrix} -1 \\ 0 \end{bmatrix} - (0) = -40$$

$$z_5 - c_5 = [40 \quad -130/3] \begin{bmatrix} 0 \\ -1 \end{bmatrix} - (0) = 130/3$$

$$z_7 - c_7 = [40 \quad -130/3] \begin{bmatrix} 0 \\ 1 \end{bmatrix} - (18) = -184/3$$

Do cálculo anterior escolheremos a variável 5 para entrar na base porque possui o máximo entre os $z_j - c_j$ positivos, ou seja, $z_5 - c_5 = 130/3$.

Continuando o cálculo vamos para o Passo 2.

Passo 2: verificar viabilidade e escolha da variável que deixa a base.

Pelo critério adotado para a melhoria, a variável x_5 entrará na base saindo a variável $\frac{\bar{b}_s}{y_{sk}} = \text{mínimo} \left\{ \frac{\bar{b}_i}{y_{ik}} : y_{ik} > 0 \right\}$. O cálculo não pode ser feito sem antes obtermos a coluna transformada da variável x_5 diretamente da matriz A . Na iteração anterior a inversa da base era a própria identidade e $y_i = r_i \quad \forall i \in J$. Na atual iteração:

$$y_5 = B^{-1}a_5 = \begin{bmatrix} 1 & -4/3 \\ 0 & 2/3 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 4/3 \\ -2/3 \end{bmatrix}$$

Daí,

$$\bar{b} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \text{ e } y_5 = \begin{bmatrix} 4/3 \\ -2/3 \end{bmatrix} \text{ logo}$$

$$\begin{array}{c} \leftarrow \begin{bmatrix} x_6 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} / \begin{bmatrix} 4/3 \\ -2/3 \end{bmatrix} = \begin{bmatrix} 3/4 \\ - \end{bmatrix} \rightarrow \\ \uparrow \qquad \qquad \qquad \uparrow \\ \text{Valor negativo} \qquad \qquad \text{Valor mínimo (s = 1)} \end{array}$$

Portanto, deve ser x_6 a variável que deixará a base.

A nova base será formada pelas variáveis x_5 e x_3 , ou seja:

$$B' = [a_5, a_3] = \begin{bmatrix} 0 & 2 \\ -1 & 3/2 \end{bmatrix}$$

Passo 3: atualizar os elementos de cálculo.

Vamos inverter a nova base aproveitando os cálculos da iteração anterior, ou seja,

$$\begin{bmatrix} 1 & -4/3 \\ 0 & 2/3 \end{bmatrix} \begin{bmatrix} 4/3 \\ -2/3 \end{bmatrix} \xrightarrow{\text{Pivoteamento}} \begin{bmatrix} 3/4 & -1 \\ 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \therefore$$

$$B^{-1} = \begin{bmatrix} 3/4 & -1 \\ 1/2 & 0 \end{bmatrix}$$

$$\bar{x}_B = \bar{b} = B^{-1}b = \begin{bmatrix} 3/4 & -1 \\ 1/2 & 0 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \end{bmatrix} = \begin{bmatrix} 3/4 \\ 5/2 \end{bmatrix}$$

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = c_B B^{-1} = [0 \quad 15] \begin{bmatrix} 3/4 & -1 \\ 1/2 & 0 \end{bmatrix} = [15/2, \quad 0]$$

$$z_0 = z_0 - (z_5 - c_5) \frac{\bar{x}_{B1}}{y_{15}} = 70 - (130/3) (3/4) = 75/2$$

ou:

$$z_0 = c_B \bar{x}_B = [0 \quad 15] \begin{bmatrix} 3/4 \\ 5/2 \end{bmatrix} = 75/2$$

Nesse ponto as variáveis x_1, x_2, x_4, x_6 e x_7 e não são básicas. $I = \{1, 2, 4, 6, 7\}, J = \{5, 3\}$. Após a segunda iteração, as variáveis artificiais deixaram a base e a primeira fase do simplex está concluída. De fato, não existe mais necessidade de levarmos em conta essas variáveis. Vamos novamente ao Passo 1.

Passo 1: escolher a nova variável que entrará na base.

$$z_1 - c_1 = ua_1 - c_1$$

$$= [15/2 \quad 0] \begin{bmatrix} 1 \\ 1 \end{bmatrix} - (8) = -1/2$$

$$z_2 - c_2 = [15/2 \quad 0] \begin{bmatrix} 2 \\ 1 \end{bmatrix} - (12) = 3$$

$$z_4 - c_4 = [15/2 \quad 0] \begin{bmatrix} -1 \\ 0 \end{bmatrix} - (0) = -15/2$$

Do cálculo anterior escolheremos a variável 2 para entrar na base porque ela possui o máximo entre os $z_j - c_j$ positivos, ou seja, $z_2 - c_2 = 3$. Continuando o cálculo, vamos para o Passo 2.

Passo 2: verificar viabilidade e escolha da variável que deixa a base.

Pelo critério adotado para a melhoria, a variável x_2 entrará na base saindo a variável

$$\frac{\bar{b}_s}{y_{sk}} = \min_{1 \leq i \leq m} \left\{ \frac{\bar{b}_i}{y_{ik}} : y_{ik} > 0 \right\}.$$

Novamente, necessitamos obter a coluna da variável x_2 como fizemos na iteração anterior. Nesse caso:

$$y_2 = B^{-1}a_2 = \begin{bmatrix} 3/4 & -1 \\ 1/2 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 1 \end{bmatrix}$$

Daí,

$$\bar{b} = \begin{bmatrix} 3/4 \\ 5/2 \end{bmatrix} \text{ e } y_2 = \begin{bmatrix} 1/2 \\ 1 \end{bmatrix} \text{ logo}$$

$$\begin{array}{c} \longleftarrow \begin{bmatrix} x_5 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3/4 \\ 5/2 \end{bmatrix} / \begin{bmatrix} 1/2 \\ 1 \end{bmatrix} = \begin{bmatrix} 3/2 \\ 5/2 \end{bmatrix} \longrightarrow \\ \uparrow \\ \text{Valor mínimo (s = 1)} \end{array}$$

A nova base será formada pelas variáveis x_2, x_3 , ou seja:

$$B' = [a_2, a_3] = \begin{bmatrix} 2 & 2 \\ 1 & 3/2 \end{bmatrix}$$

Passo 3: atualizar os elementos de cálculo.

Vamos inverter a nova base pelo método das operações elementares, ou seja:

$$\begin{bmatrix} 3/4 & -1 \\ 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1/2 \\ 1 \end{bmatrix} \xrightarrow{\text{Pivoteamento}} \begin{bmatrix} 3/2 & -2 \\ -1 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot$$

$$B^{-1} = \begin{bmatrix} 3/2 & -2 \\ -1 & -2 \end{bmatrix}$$

$$\bar{x}_B = \bar{b} = B^{-1}b = \begin{bmatrix} 3/2 & -2 \\ -1 & -2 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \end{bmatrix} = \begin{bmatrix} 3/2 \\ -11 \end{bmatrix}$$

$$u = (u_1 u_2) = c_B B^{-1} = [12 \quad 15] \begin{bmatrix} 3/2 & -2 \\ -1 & -2 \end{bmatrix} = [3 \quad -54]$$

$$z_0 = c_B \bar{x}_B = [12 \quad 15] \begin{bmatrix} 3/2 \\ 1 \end{bmatrix} = 33$$

Neste ponto as variáveis x_1, x_5, x_7, x_4 e x_6 não são básicas. $I = \{1, 4, 5, 6, 7\}, J = \{2, 3\}$. Voltamos ao Passo 1.

Passo 1: escolher a nova variável que entrará na base.

$$z_1 - c_1 = ua_1 - c_1 = [3 \quad 6] \begin{bmatrix} 1 \\ 1 \end{bmatrix} - (8) = 1$$

$$z_4 - c_4 = [3 \quad 6] \begin{bmatrix} -1 \\ 1 \end{bmatrix} - (0) = -3$$

$$z_5 - c_5 = [3 \quad 6] \begin{bmatrix} 0 \\ -1 \end{bmatrix} - (0) = -6$$

A variável de índice 1 entra na base, pois é a única que possui $z_j - c_j$ positivo. Continuando o cálculo, vamos para o Passo 2.

Passo 2: verificar viabilidade e escolha da variável que deixa a base.

Novamente necessitamos obter a coluna da variável x_1 como fizemos na iteração anterior para calcular a variável de bloqueio ao crescimento de x_1 . Nesse caso,

$$y_1 = B^{-1}a_1 = \begin{bmatrix} 3/2 & -2 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1/2 \\ 1 \end{bmatrix}$$

Daí,

$$\bar{b} = \begin{bmatrix} 3/2 \\ 1 \end{bmatrix} \text{ e } y_1 = \begin{bmatrix} -1/2 \\ 1 \end{bmatrix} \text{ logo}$$

$$\begin{bmatrix} x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3/2 \\ 1 \end{bmatrix} / \begin{bmatrix} -1/2 \\ 1 \end{bmatrix} = \begin{bmatrix} - \\ 1 \end{bmatrix}$$

A nova base será formada pelas variáveis x_2 e x_1 , ou seja:

$$B' = \begin{bmatrix} a_2 & a_1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

Passo 3: atualizar os elementos de cálculo.

Invertendo a base temos:

$$\begin{bmatrix} 3/2 & -2 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} -1/2 \\ 1 \end{bmatrix} \xrightarrow{\text{Pivoteamento}} \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot$$

$$B^{-1} = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$$

$$\bar{x}^B = \bar{b} = B^{-1}b = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$u = (u_1 u_2) = c_B B^{-1} = [12 \quad 8] \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} = [4 \quad 4]$$

$$z_0 = c_B \bar{x}_B = [12 \quad 8] \begin{bmatrix} 2 \\ 1 \end{bmatrix} = 32$$

Neste ponto: $I = \{3, 4, 5, 6, 7\}$, $J = \{2, 1\}$ e voltamos ao Passo 1.

Passo 1: escolher a nova variável que entrará na base.

$$z_3 - c_3 = u a_1 - c_1 = [4 \quad 4] \begin{bmatrix} 2 \\ 3/2 \end{bmatrix} - (15) = -1$$

$$z_4 - c_4 = [4 \quad 4] \begin{bmatrix} -1 \\ 0 \end{bmatrix} - (0) = -4$$

$$z_5 - c_5 = [4 \quad 4] \begin{bmatrix} 0 \\ -1 \end{bmatrix} - (0) = -4$$

Como todos os $z_j - c_j \leq 0$, estamos em condição de parada. O ótimo foi alcançado.

Vamos agora explorar as relações duais e sua interpretação econômica. O dual do problema anteriormente solucionado é:

Maximizar $w = 5u_1 + 3u_2$
 sujeito a:

$$\begin{aligned} u_1 + u_2 &\leq 8 \\ 2u_1 + u_2 &\leq 12 \\ 2u_1 + 3/2u_2 &\leq 15 \\ u_1 &\geq 0, u_2 \geq 0 \end{aligned}$$

onde as variáveis u estão associadas aos preços dos produtos demandados pelo mercado. De fato, o objetivo de maximização pode ser interpretado como a busca do maior valor possível para a oferta de combustível sem que se possa contabilizar lucro. O problema dual admite uma solução gráfica como mostra a Figura 4.7.

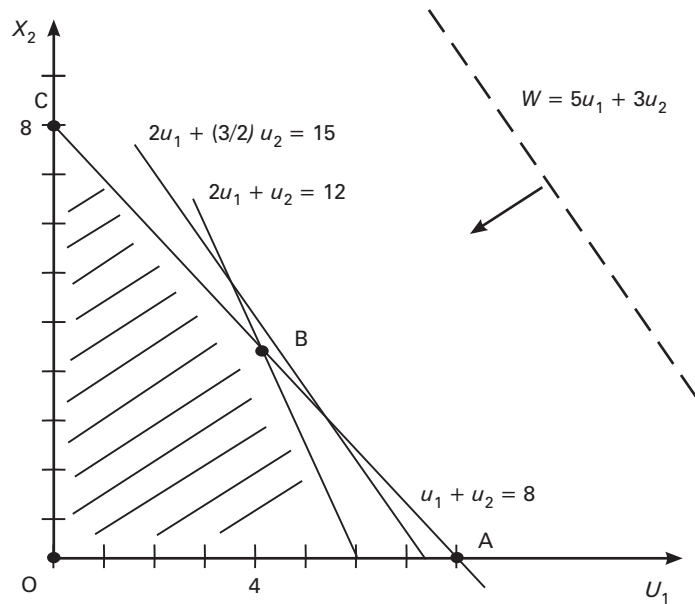


FIGURA 4.7 Solução gráfica do dual.

Observe que qualquer ponto do conjunto de soluções viáveis do problema dual está associado a preços de gasolina e de óleo para os quais nenhuma atividade gera lucro positivo. No exemplo, a solução dual ótima nos fornece os preços dos produtos demandados para o qual existe um perfeito equilíbrio econômico, com custo total igual à receita total. No ótimo, nenhuma atividade básica é deficitária ou lucrativa. Por outro lado, seria deficitária a introdução de qualquer atividade não básica no esquema de produção. A solução ótima do dual, que pode ser obtida graficamente, nos leva a $w = 32$ e $u_1 = 4$ e $u_2 = 4$. Por esse resultado, poderíamos ter evitado o cálculo de otimização após a inclusão da atividade de exploração do xisto. A primeira solução obtida sem a alternativa do xisto seria mantida.

4.6 – ANÁLISE DE SENSIBILIDADE

4.6.1 – Considerações Gerais sobre o Tema

A análise de sensibilidade é basicamente uma técnica utilizada para avaliar os impactos que o programa sofre quando existem modificações nas condições de modelagem. Análise de sensibilidade é, basicamente, o estudo de um modelo de programação matemática submetido a mudanças em suas condições iniciais. As mudanças poderão abranger:

- Mudança no vetor de custos.
- Mudança no vetor de termos independentes.
- Mudança nos coeficientes das variáveis.
- Acréscimo de restrições.
- Acréscimo de novas variáveis.

Para ilustrar as mudanças, examinaremos cada caso dentro de um exemplo numérico.

Minimizar $z = 8x_1 + 12x_2 + 15x_3 + 0x_4 + 0x_5 + 40x_6 + 18x_7$
 sujeito a:

$$\begin{aligned}x_1 + 2x_2 + 2x_3 - x_4 + x_6 &= 5 \\x_1 + x_2 + (3/2)x_3 - x_5 + x_7 &= 3 \\x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0, x_6 \geq 0, x_7 \geq 0\end{aligned}$$

com o seguinte sistema associado à solução ótima:

		x_1	x_2	x_3	x_4	x_5
z	32	0	0	-1	-4	-4
x_4	1	1	0	1	4	-2
x_5	2	0	1	1/2	-1	1

4.6.2 – Mudanças no Vetor de Custos

As mudanças no vetor de custos poderão incidir sobre dois tipos de variáveis: as básicas e as não básicas.

• Mudança de custos de variáveis não básicas

A fórmula básica para o cálculo dos custos reduzidos é:

$$z_j - c_j = c_B B^{-1} a_j - c_j$$

Observamos que para uma variável não básica o termo $c_B B^{-1} a_j$ não se altera, o que significa que as alterações nos custos reduzidos se propagam diretamente com a variação do termo c_j . Seja a possível variação do custo igual a δ . Para que a variável de índice $j \in J$ seja candidata a entrar na base é necessário que $z_j - (c_j + \delta) > 0$ ou $\delta < z_j - c_j$.

Exemplo 1: Vamos estudar a possibilidade de mudanças no vetor custo no problema de combustível do item anterior. Sabemos que o programa inicial era:

Minimizar $z = 8x_1 + 12x_2 + 15x_3 + 0x_4 + 0x_5 + 40x_6 + 18x_7$
 sujeito a:

$$\begin{aligned}x_1 + 2x_2 + 2x_3 - x_4 + x_6 &= 5 \\x_1 + x_2 + 3/2x_3 - x_5 + x_7 &= 3 \\x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0, x_6 \geq 0, x_7 \geq 0\end{aligned}$$

com o seguinte sistema associado à solução ótima:

		x_1	x_2	x_3	x_4	x_5
z	32	0	0	-1	-4	-4
x_1	1	1	0	1	1	-2
x_2	2	0	1	1/2	-1	1

Examinemos o caso de possíveis alterações no custo do xisto.

Denominando por δ a variação de c_3 (custo do xisto) teremos que $z_3 - (c_3 + \delta) > 0$ é a condição de alterar a solução ótima encontrada. Substituindo os valores temos:

$$-\delta + (-1) > 0 \Rightarrow -\delta > 1 \text{ ou } \delta < -1.$$

Nesse caso, se o custo do xisto cair 2 unidades, por exemplo, ele tornar-se-á atrativo e fará parte da solução ótima, levando ao seguinte quadro ótimo:

		x_1	x_2	x_3	x_4	x_5
z	31	-1	0	0	-8	-2
x_3	1	1	0	1	4	-2
x_2	3/2	-1/2	1	0	-3	3

• Mudança de custos de variáveis básicas

Obviamente, se os valores dos custos das variáveis básicas sofrerem alterações no sentido de se tornarem ainda mais desejáveis, elas jamais deixarão a base; contudo, não podemos saber o que aconteceria no caso inverso. O que desejamos nesse caso é saber qual o intervalo de variação admissível no vetor custo para que certa variável permanecesse na base sem alterar a base calculada.

Exemplo 2: Vamos examinar mudanças de preço no petróleo importado (variável x_2). Da solução ótima temos que:

$$u = (u_1 \ u_2) = c_B B^{-1} = [12 + \delta \quad 8] \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} = [4 + \delta \quad 4 - \delta]$$

$$z_3 - c_3 = u a_1 - c_1 = [4 + \delta \quad 4 - \delta] \begin{bmatrix} 2 \\ 3/2 \end{bmatrix} - (15) = -1 + \delta/2$$

$$z_4 - c_4 = [4 + \delta \quad 4 - \delta] \begin{bmatrix} -1 \\ 0 \end{bmatrix} - (0) = -4 - \delta$$

$$z_5 - c_5 = [4 + \delta \quad 4 - \delta] \begin{bmatrix} 0 \\ -1 \end{bmatrix} - (0) = -4 + \delta$$

Daí retiramos o seguinte conjunto de relações para se manter a mesma base ótima.

$$-1 + \delta/2 \leq 0 \therefore \delta \leq 2$$

$$-4 - \delta \leq 0 \therefore \delta \geq -4$$

$$-4 + \delta \leq 0 \therefore \delta \leq 4$$

O que significa que no intervalo $-4 \leq \delta \leq 2$ a solução não sofrerá alteração em sua base. Fora desse intervalo, a base não é ótima. Com essas equações poderemos examinar a variação simultânea de custos em mais de uma variável (básica ou não) através da constituição de sistemas de equações.

4.6.3 – Mudança no Vetor de Demandas ou Termo Independente

O impacto da mudança no vetor b é calculado através da seguinte fórmula:

$$B^{-1} (b + \Delta b)$$

onde Δb indica a variação no vetor b .

Para o caso de exigirmos a viabilidade primal então:

$$B^{-1} (b + \Delta b) \geq 0$$

Exemplo 3: Para o problema do combustível,

$$B^{-1} (b + \Delta b) = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 5 & + & \Delta b_1 \\ 3 & + & \Delta b_2 \end{bmatrix} \geq 0$$

$$2 + \Delta b_1 - \Delta b_2 \geq 0$$

$$1 - \Delta b_1 - 2\Delta b_2 \geq 0$$

que implica a solução de um sistema de inequações para a análise da mudança do termo independente.

Exemplo 4: Nesse caso, vamos examinar detalhadamente o programa representado a seguir.

Maximizar $z = x_1 + x_2$
sujeito a:

$$\begin{aligned} x_1 &\leq 3 \\ 2x_1 + 3x_2 &\leq 24 \\ x_1 \geq 0, x_2 &\geq 0 \end{aligned}$$

com o seguinte sistema associado à solução ótima:

		x_1	x_2	x_3	x_4
z	9	0	0	-1/3	-1/3
x_1	3	1	0	1	0
x_2	6	0	1	-2/3	1/3

✓ Suponhamos que o vetor $b = \begin{bmatrix} 3 \\ 24 \end{bmatrix}$ seja alterado para $b = \begin{bmatrix} 1 \\ 18 \end{bmatrix}$, o que acontecerá?

O impacto da alteração do vetor b pode ser facilmente calculado:

$$\bar{x}_B = B^{-1}b = \begin{bmatrix} 1 & 0 \\ -2/3 & 1/3 \end{bmatrix} \begin{bmatrix} 1 \\ 18 \end{bmatrix} = \begin{bmatrix} 1 \\ 16/3 \end{bmatrix}$$

Assim,

$$\bar{x}_B = \begin{bmatrix} \bar{x}_{B_1} \\ \bar{x}_{B_2} \end{bmatrix} = \begin{bmatrix} 1 \\ 16/3 \end{bmatrix}$$

A base permaneceu ótima (por coincidência). Então, como vimos no exemplo anterior, podemos estar interessados nos intervalos em que as componentes de b poderiam variar para manter a viabilidade da solução corrente.

✓ Determinemos, para o caso corrente, o menor valor de b_2 para que a solução permaneça ótima.

$$\bar{x}_B = B^{-1}b = \begin{bmatrix} 1 & 0 \\ -2/3 & 1/3 \end{bmatrix} \begin{bmatrix} 3 \\ \Delta \end{bmatrix} = \begin{bmatrix} 3 \\ -2 + \Delta/3 \end{bmatrix} \geq 0 \therefore$$

$$\Delta \geq 6$$

✓ Supondo agora que $b = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ o que aconteceria?

Nesse caso:

$$\bar{x}_B = B^{-1}b = \begin{bmatrix} 1 & 0 \\ -2/3 & 1/3 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1/3 \end{bmatrix}$$

Ou seja, a base ótima, com a modificação do vetor b , deixa de ser viável. Para solucionarmos esse problema, podemos aplicar o algoritmo dual simplex da seguinte forma:

		x_1	x_2	x_3	x_4
z	1	0	0	-1/3	-1/3
x_1	2	1	0	1	0
→ x_2	-1/3	0	1	-2/3	1/3

que nos leva ao seguinte quadro ótimo:

		x_1	x_2	x_3	x_4
z	1/2	0	-1/2	0	-1/2
x_1	1/2	1	3/2	0	1/2
x_2	3/2	0	-3/2	1	-1/2

4.6.4 – Mudança nas Restrições

✓ O primeiro caso a estudar diz respeito ao acréscimo de uma nova restrição ao problema. Sempre que isso ocorre em um programa matemático, acontece uma redução no espaço das soluções viáveis, o que, ou aumenta/reduz (minimização/maximização) o valor da função objetivo, ou não altera a solução encontrada. Em nenhuma hipótese o acréscimo de uma restrição “melhora” o valor numérico da função objetivo. Como um procedimento geral para esse caso, podemos sugerir que se verifique se o acréscimo da restrição altera a solução ou não. No caso da alteração se configurar, devemos prosseguir na solução do “novo” problema.

Exemplo 5: Nesse caso, vamos examinar o programa a seguir.

$$\begin{aligned} &\text{Maximizar } z = x_1 + 3x_2 \\ &\text{sujeito a:} \\ &\quad x_1 \leq 3 \\ &\quad 2x_1 + 3x_2 \leq 24 \\ &\quad x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

que, quando solucionado, possui o seguinte quadro ótimo:

		x_1	x_2	x_3	x_4
z	24	-1	0	0	-1
x_3	3	1	0	1	0
x_2	8	2/3	1	0	1/3

Suponhamos agora que desejamos acrescentar ao programa a seguinte restrição:

$$x_1 + 6x_2 \leq 30$$

Para verificar se houve o corte da solução ótima calculada anteriormente basta substituir os valores de $\bar{x}^B = \begin{bmatrix} \bar{x}_{B_1} \\ \bar{x}_{B_2} \end{bmatrix} = \begin{bmatrix} 3 \\ 8 \end{bmatrix}$ na restrição, o que resultará em: $0 + 6 \times 8 \geq 30$, provando que haverá alteração na solução, uma vez que a solução encontrada no quadro ótimo causa a violação da nova restrição. A nova restrição reduzirá o tamanho do espaço das soluções viáveis do problema e, consequentemente, diminuirá seu valor no ótimo. Para solucionar o problema decorrente da agregação da nova restrição vamos acrescentá-la ao quadro inicial. Como uma nova linha está sendo criada na matriz de restrições, devemos ter o cuidado de introduzir a respectiva variável de folga para manter uma base viável inicial. Nesse exemplo, a variável de folga será x_5 . Assim teremos:

		x_1	x_2	x_3	x_4	x_5
z	24	-1	0	0	-1	0
x_3	3	1	0	1	0	0
x_2	8	2/3	1	0	1/3	0
x_5	30	1	6	0	0	1

A coluna de x_2 , uma variável básica, não é canônica, pois existe o valor 6 na última linha. Usando o método das operações elementares para completar a inversão da base temos:

		x_1	x_2	x_3	x_4	x_5
z	24	-1	0	0	-1	0
x_3	3	1	0	1	0	0
x_2	8	2/3	1	0	1/3	0
x_5	-18	-3	0	0	-2	1

Aplicando o algoritmo dual simplex, após dois pivoteamentos, chegaremos ao seguinte quadro:

		x_1	x_2	x_3	x_4	x_5
z	$33/2$	0	0	$-1/2$	0	$-1/6$
x_4	$9/2$	0	0	$-3/2$	1	$1/2$
x_2	$9/2$	0	1	$-1/6$	0	$1/6$
x_1	3	1	0	1	0	0

✓ O segundo caso corresponde à eliminação de uma restrição. Nessa alternativa encontramos alguns subcasos, a saber:

- A restrição eliminada é de desigualdade.

Se a variável básica relacionada com essa restrição for a variável de folga a ela associada, então a restrição é inativa e sua eliminação não causará qualquer modificação para a solução ótima.

Caso a variável básica seja real, nada podemos afirmar e devemos retomar o processo de solução como um problema novo.

- A restrição eliminada é de igualdade.

Nesse caso, devemos recomeçar a solução a partir da eliminação da restrição.

✓ O terceiro caso corresponde à alteração de coeficientes em uma restrição já existente. Nessa hipótese não será mais necessária a introdução da variável de folga para completar a base. A partir daí recaímos no caso de acréscimo de uma nova restrição.

4.6.5 – Mudança nas Variáveis

✓ O primeiro caso possível de mudança na variável é a eliminação que comporta dois subcasos:

- A variável eliminada é não básica.

Evidentemente, quando retiramos do programa uma variável não básica no quadro ótimo, não alteramos a solução, uma vez que as variáveis não básicas são nulas.

- A variável eliminada é básica.

Não podemos simplesmente apagar uma variável básica porque a inversa da base foi calculada em sua função. Se agirmos assim teremos de recomeçar a solução do passo inicial. O que podemos fazer é migrar para uma base vizinha que não contenha essa variável e, posteriormente, eliminá-la do programa como variável não básica. Para realizar essa migração deveremos forçar um pivoteamento. Os critérios para a escolha desse pivoteamento serão discutidos a seguir.

Exemplo 6:

$$\text{Maximizar } z = 5x_1 + 5x_2 + 13x_3$$

sujeito a:

$$-x_1 + x_2 + 3x_3 \leq 20$$

$$12x_1 + 4x_2 + 10x_3 \leq 90$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$$

que possui o seguinte quadro ótimo:

		x_1	x_2	x_3	x_4	x_5
z	-100	0	0	-2	-3	0
x_2	20	-1	1	3	1	0
x_5	10	16	0	-2	-4	1

Supondo que a variável x_2 deva ser eliminada do problema, então poderemos substituí-la na base por x_3 que, dentre as variáveis não básicas, possui o custo reduzido menos desfavorável. Em uma situação em que se apresente a condição de parada para o simplex (quadro ótimo) as variáveis não básicas ou possuem custos reduzidos nulos ou negativos. Nessa situação, substituir uma variável básica por uma não básica irá, provavelmente, piorar o valor da solução representada no quadro. O pivoteamento se configura da seguinte forma:

		x_1	x_2	x_3	x_4	x_5
z	-100	0	0	-2	-3	0
x_2	20/3	-1	1	3	1	0
x_5	10	16	0	-2	-4	1

o que nos leva ao seguinte vértice vizinho ao anterior, ótimo dentro do novo quadro:

		x_1	x_2	x_3	x_4	x_5
z	-260/3	-2/3	2/3	0	-7/3	0
x_3	20/3	-1/3	1/3	1	1/3	0
x_5	50/3	46/3	2/3	0	-10/3	1

✓ O segundo caso refere-se ao acréscimo de uma nova variável.

De fato, a introdução de uma nova variável (x_l) exige o recálculo dos custos reduzidos e da coluna (l) da matriz Y .

Exemplo 7:

$$\text{Maximizar } z = x_1 + 2x_2 + 7x_3$$

sujeito a:

$$x_1 + 2x_3 \leq 4$$

$$x_2 + 3x_3 \leq 5$$

$$-x_1 + x_2 \leq 0$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$$

que nos leva ao seguinte quadro ótimo:

		x_1	x_2	x_3	x_4	x_5	x_6
z	-13	0	0	0	-2	-1	-1
x_1	2	1	0	0	3	-2	2
x_2	2	0	1	0	3	-2	3
x_3	1	0	0	1	-1	1	-1

Suponhamos que desejamos acrescentar mais uma atividade, x_7 , com os seguintes elementos:

$$x_7 = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} \text{ e } c_7 = 4$$

Podemos calcular o valor de $z_7 - c_7$ de sua expressão geral:

$$z_7 - c_7 = c_B B^{-1} a_7 - c_7 = [-1 \quad -2 \quad -7] \begin{bmatrix} 3 & -2 & 2 \\ 3 & -2 & 3 \\ -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} - (-4) = 1$$

De modo semelhante, calculamos a coluna y_7 .

$$y_7 = \begin{bmatrix} 3 & -2 & 2 \\ 3 & -2 & 3 \\ -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ -1 \end{bmatrix}$$

Acrescentando a nova coluna no quadro temos:

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
z	-13	0	0	0	-2	-1	-1	1
x_1	2	1	0	0	3	-2	2	2
x_2	2	0	1	0	3	-2	3	4
x_3	1	0	0	1	-1	1	-1	-1

cujo pivoteamento conduz a:

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
z	-27/2	0	-1/4	0	-11/4	-1/2	-7/4	0
x_1	1	1	-1/2	0	3/2	-1	1/2	0
x_7	1/2	0	1/4	0	3/4	-1/2	3/4	1
x_3	3/2	0	1/4	1	-1/4	1/2	-1/4	0

✓ O terceiro caso de mudança nas variáveis aborda a alteração dos coeficientes de uma variável já existente.

Dois subcasos são possíveis:

- A variável modificada é não básica.

Nesse caso, basta recalculer a coluna (l) como no exemplo anterior.

- A variável modificada é básica.

Podemos recalculer a coluna (l), introduzi-la no quadro e completar o processo de inversão da base.

Exemplo 8:

Seja o quadro ótimo do exemplo 6:

		x_1	x_2	x_3	x_4	x_5
z	-100	0	0	-2	-3	0
x_2	20	-1	1	3	1	0
x_5	10	16	0	-2	-4	1

Suponhamos agora que $x_2 = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$. Sabemos que:

$$y_2 = B^{-1}a_2 = \begin{bmatrix} 1 & 0 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Contudo, como a variável é básica temos:

		x_1	x_2	x_3	x_4	x_5
z	-100	0	0	-2	-3	0
x_2	20	-1	1	3	1	0
x_5	10	16	-1	-2	-4	1

E podemos prosseguir na inversão da base pelo método das operações elementares chegando a:

		x_1	x_2	x_3	x_4	x_5
z	-100	0	0	-2	-3	0
x_2	20	-1	1	3	1	0
x_5	30	15	0	1	-3	1

4.7 – PROBLEMAS PROPOSTOS

1 – Revisão da Teoria



Responda com Certo ou Errado justificando sua escolha:

1. Para um par de problemas primal \times dual na solução ótima temos:
 - a. ☐ Quando a solução de cada problema é finita os valores das soluções ótimas do par é igual.
 - b. ☐ O número de variáveis de folga de um problema correspondem ao número de variáveis reais do outro.
 - c. ☐ Sempre que uma variável de folga é diferente de zero, a variável dual correspondente é igual a zero.
 - d. ☐ Se um dos sistemas tiver infinitas soluções, o outro também terá.
 - e. ☐ Se um dos sistemas for inviável, o outro também será.
 - f. ☐ O número de iterações do algoritmo simplex necessárias para a solução de um problema será o mesmo para cada sistema.
 - g. ☐ A condição de parada no problema primal corresponde à condição de viabilidade no problema dual.

2 – Aplicação da Teoria da Dualidade



Para o problema abaixo:

Maximizar $z = 2x_1 + 3x_2 + 6x_3 + 8x_4$
sujeito a:

$$\begin{aligned}x_1 + 2x_2 + 3x_3 + x_4 &\geq 3 \\ -2x_1 + x_2 - x_3 + 3x_4 &\leq -4 \\ x_j &\geq 0, j = 1, 2, 3, 4\end{aligned}$$

1. Escreva o problema dual.
2. Solucione o dual graficamente.
3. A partir da solução do dual obtenha a solução do primal.

3 – Aplicação da Teoria da Dualidade



Para o problema abaixo:

Maximizar $z = 2x_1 + 3x_2 + 6x_3$
sujeito a:

$$\begin{aligned}x_1 + 2x_2 + x_3 &\leq 12 \\ x_1 - x_2 + 3x_3 &\leq 7 \\ x_j &\geq 0, j = 1, 2, 3\end{aligned}$$

1. Solucione-o pelo algoritmo simplex.
2. Escreva seu dual.
3. Solucione o problema dual graficamente.
4. Calcule, utilizando as condições de dualidade e a solução gráfica, o quadro simplex ótimo para o dual.

4 – Aplicação da Teoria da Dualidade



Para o problema abaixo:

Maximizar $z = x_1 - x_2 + 2x_3$
sujeito a:

$$\begin{aligned}x_1 + x_2 + x_3 &\leq 6 \\ -x_1 + 2x_2 + 3x_3 &\leq 9 \\ x_j &\geq 0, j = 1, 2, 3\end{aligned}$$

1. Solucione-o pelo método simplex.
2. Escreva o problema dual e determine graficamente sua solução.
3. Supondo que b_1 (primeiro elemento do termo independente) seja acrescido de uma unidade, o que acontecerá com o valor da função objetivo?
4. Supondo que o vetor de custos seja alterado para $(2, 1, 1)$, o que acontecerá com a solução encontrada no primeiro pedido?
5. Supondo que o vetor de termos independentes seja alterado para $(5 \ 10)^t$, o que acontecerá com a solução encontrada no primeiro pedido?
6. Se uma nova variável for incluída no problema com o custo igual a -1 e com um vetor de restrição igual a $(2 \ 1)^t$, o que acontecerá com a solução encontrada no primeiro pedido?
7. Se o vetor custo do problema é alterado para $(1, -1, 2) + \lambda (1, 4, -1)$, determine as soluções ótimas para todos os valores de λ .

5 – Exercício Prático



Uma pequena siderúrgica recebe encomenda de um lote de lingotes de ferro que deverá totalizar 240 toneladas de conteúdo do elemento ferro (Fe). O cliente admitirá que o lote homogêneo tenha quantidades adicionais do elemento silício (Si), mas para cada tonelada de Si deverá haver na liga pelo menos 15 toneladas de Fe. A firma tem em estoque quantidade mais que suficiente:

- Minério do tipo A (min A), que custa R\$6.000,00 cada centena de toneladas e que tem 2% de Si e 60% de Fe.
- Minério do tipo B (min B), que custa R\$3.000,00 cada centena de toneladas e que tem 4% de Si e 40% de Fe.

A firma tem ainda a oportunidade de usar como matéria-prima uma sucata de boa qualidade, que custa R\$2.500,00 a tonelada, e que possui praticamente 100% de Fe.

1. Formule o problema de programação linear que calcula a mistura de mínimo custo de matérias-primas necessárias para a produção dos lingotes encomendados.
2. Formule o problema dual.
3. Coloque os problemas na forma padrão.
4. Resolva o problema pelo método simplex.
5. De quanto varia o custo mínimo por tonelada de Fe a ser acrescida ao lote encomendado?
6. Qual a interpretação econômica que você pode dar ao problema dual e suas variáveis?
7. Suponha que apareça um novo fornecedor de um minério de tipo C (min C), que custa R\$4.000,00 por centena de toneladas e que possui 2% de Si e 50% de Fe. Haverá mudança na composição da liga ótima? Se sim, qual será a nova composição?
8. Qual o máximo preço que a sucata pode ter a fim de que seja economicamente vantajosa para a produção da liga em questão?
9. Dentro de que intervalo de custo o minério do tipo A (min A) será atrativo para permanecer na solução ótima?

6 – Exercício Prático



Uma fábrica manufatura cinco tipos de prateleiras (p_1, \dots, p_5) utilizando dois processos de produção (processo normal – N e o processo acelerado – A). Cada produto requer um certo número de horas para ser trabalhado dentro de cada processo e alguns produtos só podem ser fabricados através de um dos tipos de processos. O Quadro 4.4 resume o consumo (em horas) dentro de cada esquema de fabricação e os lucros obtidos (em reais) após a dedução dos custos de produção.

QUADRO 4.4 Consumo de horas nos esquemas de fabricação

Prateleiras	p_1	p_2	p_3	p_4	p_5
Lucros/Unidade (R\$)	550	600	350	400	200
Processo Normal (horas)	12	20	–	25	15
Processo Acelerado (horas)	10	8	16	–	–

A montagem final de cada prateleira requer 20 horas de mão de obra por unidade. A fábrica possui três máquinas para o processo normal e duas para o processo acelerado. As máquinas trabalham em

dois turnos de 8 horas por dia, em um regime de 6 horas semanais. Uma equipe de 8 homens trabalha em turno único de 8 horas e durante seis dias, na montagem das prateleiras junto aos clientes.

1. Formule o problema de programação linear que calcula o melhor esquema de produção.
2. Formule o problema dual.
3. Coloque os problemas na forma padrão.
4. Resolva ambos os problemas pelo método simplex.
5. Existe algum processo que não está sendo totalmente utilizado? Por que isso está acontecendo?
6. O que aconteceria ao lucro total se um novo trabalhador fosse contratado temporariamente (pagamento por hora trabalhada).
7. Qual deveria ser o preço dos produtos que não foram escolhidos para serem fabricados para que eles se tornassem economicamente atrativos?
8. Qual o valor econômico de uma hora extra de capacidade de produção em cada processo?
9. Justifique o custo da prateleira 1 através da análise do valor das horas de trabalho agregadas pelos processos de fabricação e montagem.

7 – Exercício Prático



Considere o seguinte problema de programação linear:

$$\text{Maximizar } z = 2x_1 + 4x_2 + x_3 + x_4$$

sujeito a:

$$x_1 + 3x_2 + x_4 \leq 8$$

$$2x_1 + x_2 \leq 6$$

$$x_2 + 4x_3 + x_4 \leq 6$$

$$x_j \geq 0, j = 1, \dots, 4$$

O modelo corresponde a um processo de produção em que as variáveis são certos produtos manufaturados que são vendidos no mercado. O retorno da venda dos produtos está indicado, em unidades monetárias, pela função objetivo. As restrições representam as limitações de disponibilidade do insumo A (restrição 1), do insumo B (restrição 2) e do insumo C (restrição 3), insumos esses que são consumidos no processo de obtenção de cada produto.

Denominamos x_5 , x_6 e x_7 as variáveis de folga associadas respectivamente à primeira, segunda e terceira restrições e $b = (b_1, b_2, b_3)$ o termo independente. Sabendo-se que a inversa da base ótima (composta por x_1 , x_3 e x_2) corresponde a:

$$B^{-1} = \begin{bmatrix} -\frac{1}{5} & \frac{3}{5} & 0 \\ -\frac{1}{10} & \frac{1}{20} & \frac{1}{4} \\ \frac{2}{5} & -\frac{1}{5} & 0 \end{bmatrix}$$

Responder às perguntas que se seguem:

1. O que aconteceria com o retorno do sistema de produção se fosse disponibilizada mais uma unidade do insumo

- 1.1 A.
- 1.2 B.
- 1.3 C.
2. Em que intervalo de variação de disponibilidade o insumo A poderá oscilar para que a base ótima não seja alterada?
3. Qual será a solução ótima se $b_1 = 5$? E se $b_1 = 25$?
4. Se estão disponíveis mais 3 unidades do insumo 1, qual será o valor do retorno ótimo obtido com a venda dos produtos?
5. Quanto vale para a companhia a unidade do insumo A? E a unidade do B?
6. A companhia resolveu introduzir um novo produto em sua linha de produção (x_8). Cada unidade de x_8 poderá ser comercializada por 3 unidades monetárias e consumirá 1 unidade do insumo A, 1 do insumo B e nenhuma do C. Partindo da situação inicial de disponibilidade de insumos, qual será a nova política de produção da companhia? Nesse caso, valerá a pena aumentar a disponibilidade do insumo B? Por quê? Nesse caso, o valor do insumo A mudou para a companhia?
7. Uma súbita mudança de preferência da clientela valorizou o produto 4, dando a ele um retorno de 5 unidades monetárias. Estudar o efeito dessa valorização sobre a solução inicial. Nesse caso, qual seria o valor do insumo C para a produção da companhia?

8 – Aplicação da Teoria da Dualidade



Abaixo está transcrito o quadro ótimo de determinado problema de programação linear, cuja função objetivo era de maximização e as restrições eram todas de \leq .

		x_1	x_2	x_3	x_4	x_5
z	-20	-2	0	-2	-3	0
x_2	5	-2	1	1	1	0
x_5	10	4	0	-2	4	1

1. Escreva o problema original.
2. Escreva o dual desse problema.
3. Obtenha a solução ótima do problema dual a partir do quadro acima.

9 – Aplicação da Teoria da Dualidade



Solucione o problema abaixo:

$$\text{Maximizar } z = x_1 + 8x_2$$

sujeito a:

$$x_1 + x_2 \geq 2$$

$$x_1 + 4x_2 \leq 4$$

$$x_j \geq 0, j = 1, 2$$

10 – Aplicação da Teoria da Dualidade



É necessário efetuar a programação diária da montagem de placas-mãe para microcomputadores em uma pequena manufatura. Cada tipo de placa pode ser montada com dois tipos de CPU. O quadro a seguir representa a distribuição de agregação de valor em US\$ com a montagem de uma placa-mãe em função do esquema Tipo de Placa-Mãe \times CPU utilizado.

TABELA 4.3 ESQUEMA PLACA-MÃE \times CPU/US\$

	<i>Tipo 1</i>	<i>Tipo 2</i>
<i>CPU 1</i>	50	70
<i>CPU 2</i>	25	60

Cada combinação CPU \times Tipo de Placa-Mãe exige um tempo de montagem e testes peculiares que podem ser resumidos em homens \times horas por lote de 25 placas no quadro que se segue:

TABELA 4.4 ESQUEMA PLACA-MÃE \times CPU/HOMENS \times HORA

	<i>Tipo 1</i>	<i>Tipo 2</i>
<i>CPU 1</i>	1	3
<i>CPU 2</i>	0,7	5

São disponíveis 50 homens \times hora de mão de obra técnica para a montagem. O fluxo de montagem dos computadores exige, no mínimo, 500 placas (consideradas em qualquer esquema de montagem) e as quantidades máximas por esquema, conforme tabela abaixo:

TABELA 4.5 ESQUEMA PLACA-MÃE \times CPU/QUANTIDADES MÁXIMAS

	<i>Tipo 1</i>	<i>Tipo 2</i>
<i>CPU 1</i>	250	250
<i>CPU 2</i>	150	250

1. Solucione o problema de maximizar a agregação de lucros com a montagem das placas-mãe, utilizando o algoritmo dual simplex.
2. Qual deveria ser o mínimo valor associado ao esquema de Placa Tipo 1 \times CPU 2, para que ele fosse de montagem atrativa?
3. O que aconteceria se, em um certo dia, o fluxo de montagem exigisse a preparação de 600 placas? Nesse caso como se comportaria o valor agregado médio por placa? (Aumenta? Diminui?)

11 – Exercício Abrangente



Uma empresa siderúrgica, produtora de aços especiais, precisa programar sua produção para os próximos três meses. Todos os seus produtos, essencialmente obtidos das matérias-primas Ni, Cr e sucata de Fe, passam pelo laminador desbastador, cuja capacidade de laminação é de, no máximo, 50 mil toneladas de aço por mês. O estoque atual de Ni é de 6 mil toneladas e seu único fornecedor desse metal tem capacidade de mineração e beneficiamento de, no máximo, 3 mil toneladas por mês. O estoque atual de Cr é de 10 mil toneladas, mas a possibilidade de importar esse metal no prazo mínimo de um mês viabiliza seu consumo em quaisquer quantidades a partir do segundo mês do horizonte de planeja-

mento. A sucata de Fe e as demais matérias-primas estão disponíveis em quantidades folgadas. A empresa, detentora do monopólio de aços especiais, considera para os próximos três meses apenas as seguintes alternativas de produção:

- ✓ Aço tipo ABNT 301, que contém no máximo 0,15% de C, no máximo 2% de Mn e no máximo 1% de Si. Esse tipo de aço, utilizado para fins estruturais, deve conter de 16% a 18% de Cr e de 6% a 8% de Ni.
- ✓ Aço tipo ABNT 302, que deve atender aos mesmos limites do aço ABNT 301 para o C, Mn e Si. Mas como deve ser utilizado em equipamentos hospitalares e de indústria alimentícia (deve ser muito resistente a corrosão) deverá conter de 17% a 19% de Cr e de 8% a 10% de Ni.
- ✓ Aço tipo ABNT 409, que contém no máximo 0,08% de C, 1% de Mn e 1% de Si. Esse tipo de aço é utilizado para exaustão de gases e motores a explosão, devendo conter de 10,5% a 11,75% de Cr, ser isento de Ni e possuir uma percentagem de Ti pelo menos 6 vezes maior que a percentagem de C e no máximo igual a 0,75 do peso total da liga.

Suponha que a empresa tenha assumido compromissos de venda de pelo menos 10 mil toneladas por mês de aço ABNT 301 e que tenha se comprometido a fornecer, até o fim do terceiro mês, pelo menos 30 mil toneladas de aço tipo ABNT 302. A empresa sabe ainda que, embora seu mercado seja bom, não conseguirá vender mais do que o triplo das quantidades já pedidas desses dois tipos de aços com Ni. Por outro lado, a demanda pelo aço tipo ABNT 409 é praticamente ilimitada, face a seus bons preços e a uma conjuntura favorável nos mercados (nacional e internacional) dos motores de explosão.

Assumindo e explicitando as hipóteses que você julgar necessárias (para os estoques iniciais dos produtos etc.), responda às seguintes questões:

1. Desprezando as restrições relativas ao conteúdo de Ti no aço tipo ABNT 409, cite as principais variáveis de decisão no problema em questão.
2. Quais são os recursos limitados nesse problema de planejamento da produção e estocagem dessa empresa? Escreva explicitamente as principais relações de disponibilidade desses recursos.
3. Escreva implicitamente as principais restrições de atendimento ao cliente.
4. Arbitrando preços por tonelada para os diversos tipos de aço (da ordem de alguns milhares de US\$), formule uma função objetivo que maximize a receita total da empresa nos três meses em questão.
5. Mostre o *tableau* do problema de programação matemática que modela esse problema de planejamento e produção.
6. Assumindo que a redução de C numa liga seja um procedimento custoso e que o preço do Ti seja alto, como se pode acomodar no modelo de PL a exigência de qualidade que relaciona teores de C e Ti na produção do aço ABNT 409.
7. Resolva o problema utilizando algum *software* adequado.
8. Qual é a melhor política para produzir e estocar? Existem alternativas?
9. Se existir, cite um tipo de aço cuja produção seja antieconômica em algum período. Qual é o prejuízo, por unidade no nível da atividade, que esse tipo de aço daria se a gerência de produção insistisse na sua fabricação nesse período? Se o preço do aço subir, a partir de que acréscimo ele passa a fazer parte do esquema ótimo de produção?
10. Existe algum recurso folgado? Qual? Até quanto esses recursos podem ter sua disponibilidade reduzida sem alterar o esquema de produção ótima?
11. Assumindo que todas as demais condições se mantenham, dentro de que intervalo de preços cada atividade básica pode variar sem que ela seja descartada do programa ótimo de produção?
12. Cite preços de equilíbrio financeiro para a empresa, aí incluindo os valores dos produtos e dos insumos. Mantendo esses valores como base, suponha que atualmente o custo do Ni para a empresa seja de US\$8.000,00 a tonelada. Até quando esse preço pode subir, sem que incorra em prejuízos para a empresa?
13. Quais são os valores marginais dos diversos produtos e insumos?

5

PROGRAMAÇÃO INTEIRA E PROGRAMAÇÃO DINÂMICA

5.1 – CARACTERÍSTICAS DOS MODELOS LINEARES DE PROGRAMAÇÃO INTEIRA

Seja o problema de programação linear que se segue:

Uma confeitaria produz dois tipos de bolos de sorvete: chocolate e creme. Cada lote de bolo de chocolate é vendido com um lucro de 3 unidades monetárias e os lotes de creme com um lucro de 1. Contratos com várias lojas impõem que sejam produzidos no mínimo 10 lotes de bolos de chocolate por dia e que o total de lotes fabricados nunca seja menor que 20. O mercado só é capaz de consumir até 40 lotes de bolos de creme e 60 de chocolate. As máquinas de preparação do sorvete disponibilizam 180 horas de operação, sendo que cada lote de bolos de chocolate consome 2 horas de trabalho e cada lote de bolos de creme 3 horas. Determinar o esquema de produção que maximize os lucros com a venda dos bolos de sorvete.

A modelagem deste problema de programação matemática que denominaremos “dos lotes de bolos de sorvete”, ou BS, é simples e pode ser realizada da seguinte forma:

1. Escolha da variável de decisão

$x_i \equiv$ quantidade de lotes de bolos de sorvete do tipo creme ($i = 1$) e chocolate ($i = 2$) fabricados.

2. Elaboração da função objetivo

$z = \text{Maximizar } \{f(x) = x_1 + 3x_2\}$

Número total de unidades monetárias obtidas como a venda dos lotes dos bolos tipo creme e chocolate.

3. Formulação das restrições tecnológicas

- a) Restrição associada à disponibilidade de maquinaria:

$$3x_1 + 2x_2 \leq 180$$

- b) Restrição do número de lotes de bolos de creme no mercado:

$$x_1 \leq 40$$

- c) Restrição do número de lotes de bolos de chocolate no mercado:

$$x_2 \leq 60$$

- d) Restrições associadas aos contratos com as lojas:

$$x_2 \geq 10$$

$$x_1 + x_2 \geq 20$$

4. Restrições de não negatividade

$$x_1 \geq 0, x_2 \geq 0$$

Podemos resumir o modelo de BS da seguinte forma:

$$\begin{aligned} \text{(BS) Maximizar } z &= x_1 + 3x_2 \\ \text{sujeito a:} \\ x_1 &\leq 40 \\ x_2 &\leq 60 \\ x_2 &\geq 10 \\ x_1 + x_2 &\geq 20 \\ 3x_1 + 2x_2 &\leq 180 \\ x_1 \geq 0, x_2 &\geq 0 \end{aligned}$$

Como nos foi solicitado que, além do modelo, fosse exibido o melhor esquema de solução para a produção, vamos buscar a solução do problema. Uma primeira dúvida surge imediatamente ao analisarmos o conjunto de restrições: seria possível produzirmos meio lote de bolos de chocolate? Se o lote possuir um número ímpar de bolos, então alguém deveria comprar uma fração de bolo de chocolate.

Esse problema é de extraordinária importância. Em inúmeras situações, as variáveis de decisão não poderão admitir um valor contínuo. Ocasões em que estamos lidando com pessoas, configurações, objetos físicos etc., soluções fracionárias perdem o sentido prático. Poderíamos pensar que esse problema não seria tão grave se trabalhássemos com uma formulação contínua e, após a solução final, providenciássemos alguma estratégia de arredondamento. O que ingenuamente pode parecer uma solução “razoável” pode ser uma péssima ideia na prática. Vamos exemplificar a inadequação da solução via arredondamento mais adiante, mas no momento cabe “corrigir” nosso modelo substituindo a condição de não negatividade por uma condição mais exigente:

4. Condições de integralidade

$x_1, x_2 \in \mathbb{Z}^+$ (conjunto dos inteiros positivos).

A Figura 5.1 mostra a representação gráfica da matriz de restrição do problema BS.

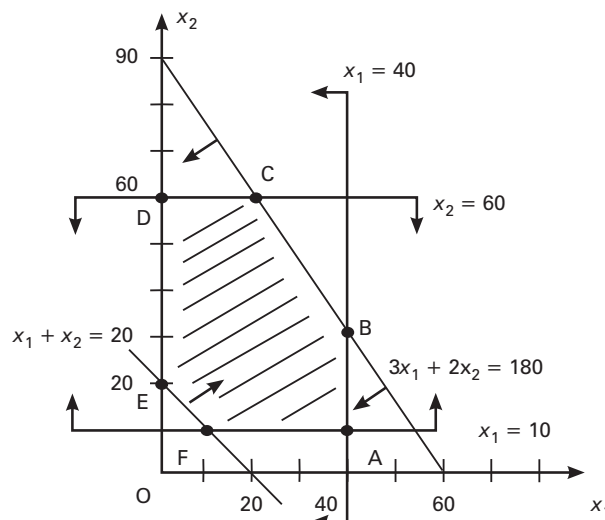


Figura 5.1 Representação gráfica de BS.

Como podemos observar na Tabela 5.1, os pontos A, B, C, D, E e F do polígono de soluções possuem coordenadas inteiras. Tal fenômeno é um fato raro. Obviamente, quando os pontos extremos do poliedro de soluções viáveis possuem valores inteiros, solucionar um problema de programação linear com variáveis contínuas é equivalente a solucionar um problema de programação inteira.

TABELA 5.1

<i>Pontos Examinados</i>	<i>Coordenadas (x_1, x_2)</i>	<i>Valor da função $z = x_1 + 3x_2$</i>
A	(40,10)	70
B	(40,30)	130
C	(20,60)	200
D	(0,60)	180
E	(0,20)	60
F	(10,10)	40

A solução que leva ao maior valor é a correspondente ao vértice C. Mas e o problema do arredondamento? Ainda não vimos por que ele é grave. Seja o programa que se segue:

$$\begin{aligned}
 &\text{Maximizar } z = x_1 + 19x_2 \\
 &\text{sujeito a:} \\
 &\quad x_1 + 20x_2 \leq 50 \\
 &\quad x_1 + x_2 \geq 20 \\
 &\quad x_1, x_2 \text{ variáveis inteiras}
 \end{aligned}$$

cuja solução ótima é: $x_1^* = 18\frac{8}{9}$, $x_2^* = 1\frac{11}{19}$ e $z_0^* = 48\frac{8}{19}$.

Aplicando a estratégia de arredondamento, uma vez que os valores ótimos são fracionários, e providenciando uma busca racional no entorno do ponto ótimo contínuo teríamos:

TABELA 5.2

<i>Pontos Examinados – Coordenadas</i>		<i>Valor da função $z = x_1 + 19x_2$</i>
$x_1^* = 19$	$x_2^* = 2$	<i>inviável</i>
$x_1^* = 19$	$x_2^* = 1$	$z = 38$
$x_1^* = 18$	$x_2^* = 2$	<i>inviável</i>
$x_1^* = 18$	$x_2^* = 1$	$z = 37$

Contudo, a solução ótima é obtida com, $x_1^* = 10$, $x_2^* = 2$ e $z_0^* = 48$, ou seja, o erro é de 21% no arredondamento. Com um número maior de variáveis e com essa margem de erro a técnica de arredondamento pode resultar em uma derrocada completa no esforço de modelagem e solução, impondo-se outros métodos de solução.

Existem vários métodos específicos para a obtenção da solução inteira exata de um problema de programação linear (ver Dantzig [1959], Gomory [1960], Glover [1965], Lawler e Wood [1966], Reiter e Rice [1966], Shapiro [1968], Nemhauser e Ullman [1968], Padberg [1970], Wolsey [1972], Pierce e Lasky [1973], Nemhauser e Garfinkel [1972], Geoffrion e Marsten [1972], Shamir [1984]) e soluções aproximativas (ver Chvatal [1979], Davis [1987], Goldberg [1988]). Apresentaremos, a seguir, uma técnica exata e uma heurística para a solução desse tipo de problema, mas antes exemplificaremos o amplo espectro do potencial de modelagem através de uma série de exemplos.

5.2 – O PROBLEMA DA MOCHILA E MÉTODOS DE SOLUÇÃO EXATA PARA A PROGRAMAÇÃO INTEIRA

O presente tópico vai debater um importante modelo de programação linear inteira (PLI) denominado o problema da mochila (PK) e métodos exatos para a solução dos modelos de PLI.

5.2.1 – O Problema da Mochila (Knapsack Problem – PK)

✓ A Importância do Problema e seu Contexto

O chamado problema da mochila ou *knapsack problem* caracteriza-se pelo estreito relacionamento com um grande número de outros modelos de programação. Sua importância está associada exatamente a esse fato. Metaforicamente podemos entendê-lo como o desafio de encher uma mochila sem ultrapassar um determinado limite de peso, otimizando o valor do produto carregado. Foi possivelmente reportado pela primeira vez na literatura por Dantzig (1957) e constitui um marco das técnicas de programação inteira, otimização combinatória e programação dinâmica. Além do aspecto matemático, o modelo em si pode ser aplicado diretamente em casos práticos como:

- Investimento de capital (ver Weingartner [1968]).
- No problema de corte e empacotamento (ver Gilmore e Gomory [1965] e [1963], Pirce [1964], Goulimis [1990], Vahrenkamp [1996]).
- Carregamento de veículos (ver Bellman e Dreyfus [1962] e Capítulo 9).
- Orçamento.

Um texto aprofundado no tema pode ser encontrado em Kellorer (2004).

✓ A Formulação do Problema

Podemos formular o problema da mochila (PK) da seguinte forma:

$$\begin{aligned}
 \text{(PK) Maximizar } z &= \sum_{j=1}^n c_j x_j \\
 \text{sujeito a:} \\
 \sum_{j=1}^n w_j x_j &\leq b \\
 x_j &\geq 0 \text{ e inteiro.}
 \end{aligned}$$

onde x_j representa o número de objetos do tipo j selecionados para serem incluídos em uma mochila, representada metaforicamente pela restrição do modelo. A mochila possui uma capacidade total de b unidades. A variável c_j representa o valor econômico de cada item, e w_j o peso do item. A literatura utiliza intensamente e, sem qualquer perda de generalidade, os coeficientes da restrição como pertencentes ao conjunto dos inteiros. Nesse caso:

$$\sum_{j=1}^n w_j > b \text{ e } w_j \leq b, j = 1, \dots, n$$

e $w \in \mathbb{S}$, $b \in \mathbb{S}$.

Considerando que exista apenas um objeto de cada tipo para ser escolhido, define-se o problema da mochila 0-1, em que a restrição da variável inteira é substituída por $x_j \in \{0, 1\}$. Denominaremos essa formulação de (PKI). O problema assim definido é também muitas vezes denominado de problema da mochila unidimensional (por possuir apenas uma restrição tipo mochila).

$$\begin{aligned} \text{(PKI) Maximizar } z &= \sum_{j=1}^n c_j x_j \\ \text{sujeito a:} \\ \sum_{j=1}^n w_j x_j &\leq b \\ x_j &\in \{0, 1\} \quad j = 1, \dots, n \end{aligned}$$

Um caso particular bastante conhecido para PKI é aquele em que as variáveis de decisão são inteiras e limitadas em certos valores máximos. Esse problema é denominado da mochila com limites (PKL) e pode ser formulado do seguinte modo:

$$\begin{aligned} \text{(PKL) Maximizar } z &= \sum_{j=1}^n c_j x_j \\ \text{sujeito a:} \\ \sum_{j=1}^n w_j x_j &\leq b \\ x_j &\leq l_j \quad j = 1, \dots, n \\ x_j &\in \mathbb{S}^+ \end{aligned}$$

Os limites impostos pelos valores l_j não permitem que a mochila seja preenchida com um número qualquer dos objetos x_j .

O (PK) é *NP-árduo* (ver Garey e Johnson [1979]). Já o caso da mochila linear, ou seja, aquele em que as variáveis são contínuas, pode ser solucionado de modo extremamente eficiente, em $O(n)$, onde n representa o número de variáveis do problema, sendo passível de solução tanto pelo algoritmo simplex (ver Chvátal [1983]), como, por exemplo, pelo método dos pontos interiores (ver Gonzaga [1989]). Esse talvez seja um dos problemas de otimização em que mais facilmente podemos perceber a diferença de dificuldade de solução entre os problemas de programação inteira e de programação linear, uma vez que somente uma restrição é considerada. Para exemplificar a natureza combinatória do problema PKI apresentamos, na Figura 5.2, a árvore de enumeração do seguinte caso:

$$\begin{aligned} \text{Maximizar } z &= 7x_1 + 10x_2 + 12x_3 + 14x_4 \\ \text{sujeito a:} \\ 41x_1 + 55x_2 + 60x_3 + 70x_4 &\leq 160 \end{aligned}$$

✓ Problemas Correlatos

Apresentaremos no presente item uma série de variantes para o problema da mochila. Essa coletânea, além de comprovar as várias aplicações do modelo, permite uma visão de sua importância.

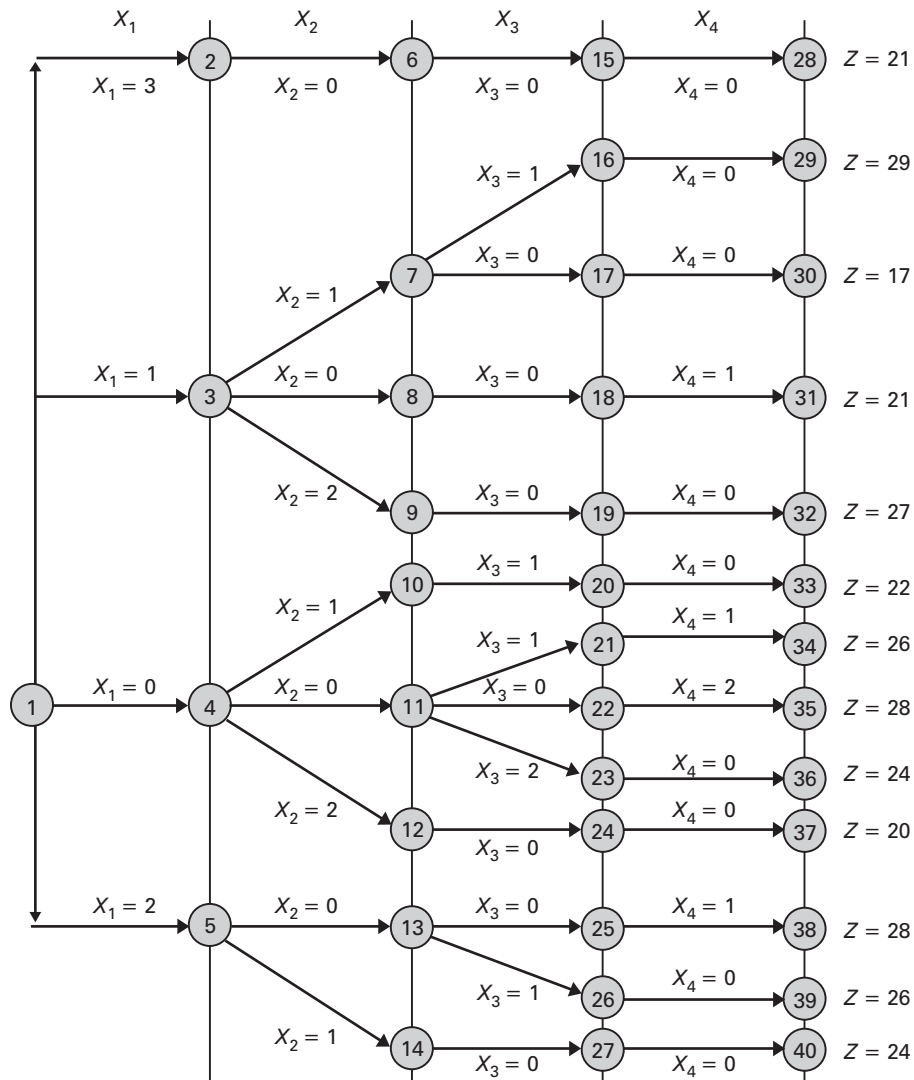


FIGURA 5.2 Árvore de enumeração do exemplo do problema da mochila.

• Subset-Sum Problem (SSP):

Dois problemas são fortemente correlacionados ao PK. O primeiro deles é um caso especial quando os custos possuem o mesmo valor dos pesos. Nesse caso, $c_j = w_j$ e assim define-se o *subset-sum problem* (SSP) (também denominado por Christofides [1979] por *Value-Independent Knapsack Problem*):

$$(\text{SSP}) \text{ Maximizar } z = \sum_{j=1}^n w_j x_j$$

sujeito a:

$$\sum_{j=1}^n w_j x_j \leq b$$

$$x_j \in \{0, 1\} \quad j = 1, \dots, n.$$

• Mochila Múltipla 0-1 (PKM)

Este problema caminha no sentido contrário de SSP, pois trata-se de uma generalização de PK. Nesse caso existem m mochilas a serem carregadas, cada uma com capacidade b_i , $i = 1, \dots, m$. PKM se reduz a PK quando $m = 1$. As variáveis de decisão do problema são afetadas por dois índices; ou seja, x_{ij} é uma variável binária com 0, 1 representando a inclusão do produto i na mochila j . Os pesos dos produtos e seus valores são os mesmos para todas as mochilas. É necessário incluir no modelo uma restrição adicional que evite a inclusão de um mesmo produto em mais de uma mochila. Podemos formular esse problema da seguinte forma:

$$\begin{aligned}
 \text{(PKM) Maximizar } z &= \sum_{i=1}^m \sum_{j=1}^n c_j x_{ij} \\
 \text{sujeito a:} \\
 \sum_{j=1}^n w_j x_{ij} &\leq b_i \quad i = 1, \dots, m \\
 \sum_{i=1}^m x_{ij} &\leq 1 \quad j = 1, \dots, n \\
 x_{ij} &\in \{0, 1\} \quad i = 1, \dots, m; j = 1, \dots, n.
 \end{aligned}$$

Existem as versões limitada ou não limitada para o caso múltiplo.

• Mochila 0-1 Multidimensional (PK- n -Dimensional)

Se para carregarmos a mochila com os objetos seja exigido um pagamento p_j a cada item adquirido e exista uma limitação p no capital disponível para tal aquisição, um novo problema pode ser definido: carregar a mochila com a carga de maior valor possível atendendo à disponibilidade de orçamento. Podemos formular o problema como um caso sujeito a duas restrições ou bidimensional (PKB), da seguinte forma:

$$\begin{aligned}
 \text{(PKB) Maximizar } z &= \sum_{j=1}^n c_j x_j \\
 \text{sujeito a:} \\
 \sum_{j=1}^n w_j x_j &\leq b \\
 \sum_{j=1}^n p_j x_j &\leq p \\
 x_j &\in \{0, 1\} \quad j = 1, \dots, n.
 \end{aligned}$$

O modelo poderá ser generalizado na medida em que existam outras restrições que limitem a utilização dos objetos para o preenchimento da mochila, daí a denominação multidimensional para esse tipo de modelo. Como no caso da mochila múltipla, podemos encontrar para o problema as versões limitada e não limitada.

• Mochila Max-Min 0-1 (PKMM)

Em diversas situações os problemas modelados pela mochila podem envolver itens que possuem dependências dos valores de um cenário. Um exemplo dessa situação diz respeito à preparação de uma bagagem para uma excursão. O valor de um capote impermeável ou de um casaco pesado depende da probabilidade associada à chuva ou ao frio no intervalo de viagem.

Dados n itens e um conjunto S de cenários, e uma mochila que possua o valor do item associado a v_i^s , o valor do item i dentro do cenário S , e considerando também w_i o peso do item i , e b a capacidade da mochila, uma mochila max-min 0-1 é definida da seguinte forma:

$$(PKMM) \text{ Maximizar } \left\{ \underset{x}{\text{Mínimo}} \left(\sum_{s \in S} v_i^s x_i \right) \right\}$$

sujeito a:

$$\sum_{i=1}^n w_i x_i \leq b \quad i = 1, \dots, m$$

$$x_i \in \{0, 1\} \quad i = 1, \dots, m;$$

O PKI é um caso particular de PKMM quando existe apenas um cenário a ser considerado. A formulação PKMM é aplicada em problemas de carteira de investimento em que o valor do retorno esperado depende do cenário futuro. A consideração de um objetivo min-max é um dos três critérios usados em um contexto maior de problemas de otimização denominado mais recentemente de otimização robusta (ver Kouvelis e Yu [1993]). Os problemas min-max ou max-min com variáveis contínuas foram estudados, dentre outros, por Kaplan (1974), Luss e Smith (1986), Pang e Yu (1989), Klein *et al.* (1993). O problema min-max discreto foi abordado por Jacobsen (1971), Porteus e Yormark (1972), Ichimori (1984), Tang (1988). Um trabalho interessante abordando o problema max-min é o de Rangan e Govindan (1992). O problema contínuo max-min da mochila foi estudado por Eiselt (1986). Yu (1996) apresenta um algoritmo B&B para a solução de PKMM, bem como uma heurística míope para a geração de soluções viáveis e limites inferiores.

• Mochila de Escolha Múltipla (PKEM)

Se o conjunto N das variáveis for particionado em m classes $N_k, k = 1, \dots, m$ e for exigido que exatamente uma variável de cada classe seja escolhida para integrar uma solução, então denominaremos esse modelo de mochila de escolha múltipla. Dudzinski e Walukiewicz (1987) formulam o PKEM da seguinte forma:

$$(PKEM) \text{ Maximizar } z = \sum_{k=1}^m \sum_{j \in N_k} c_j x_j$$

sujeito a:

$$\sum_{k=1}^m \sum_{j \in N_k} w_j x_j \leq 1$$

$$\sum_{j \in N_k} x_j = 1 \quad k \in M = \{1, \dots, m\}$$

$$x_i \in \{0, 1\} \quad j \in N = \bigcup_{k=1}^m N_k = \{1, \dots, n\}.$$

Sinha e Zoltner (1979) e Nauss (1979) sugerem várias aplicações para o PKEM. Zemel (1984) apresenta um algoritmo $O(n)$ para a solução do caso contínuo do problema. Johnson e Padberg (1981) tra-

tam o problema com restrições de empacotamento $\sum_{j \in N_k} x_j \leq 1$. Ibarra e Kim (1978) substituem as restri-

ções de escolha múltipla pela única exigência de que pelo menos uma das variáveis pertencentes a $N_k, k = 1, \dots, m$, seja positiva na solução (não obrigatoriamente inteira). A relaxação linear desse problema

possui um eficiente método dual de solução com complexidade $O(n)$ (Dyer [1984]). Reportam-se excelentes resultados na redução de PKEM (Sinha e Zoltner [1979]). O problema pode ser solucionado através de programação dinâmica em $O(nb)$ operações (ver Dudzinski [1984]).

• A Mochila Encapsulada (PKE)

Esse modelo se enquadra na classe daqueles direcionados à formulação de problemas de sequenciamento de tarefas. Dudzinski e Walukiewicz (1987) formulam o PKE da seguinte forma:

$$\begin{aligned}
 & \text{(PKE) Maximizar } z = \sum_{j \in S} c_j x_j \\
 & \text{sujeito a:} \\
 & \sum_{j \in S_i} w_j x_j \leq b_i \quad i = 1, \dots, m \\
 & x_j \in \{0, 1\}, \quad j \in S
 \end{aligned}$$

onde $S_i \cap S_k = \emptyset$ ou $(S_i \subseteq S_k \text{ ou } S_k \subseteq S_i)$ para $i \neq k$, com $S = \bigcup_{i=1}^m S_i$. Podemos supor sem qualquer perda de

generalidade que $S_i \neq S_{k'}$ e, se $S_i \subset S_{k'}$, então $b_i \leq b_{k'}$ para cada $i \neq k$. PK é um caso especial de PKE quando $m = 1$. A restrição de mochila do modelo comporta-se de modo “aninhado” ou “encapsulado” (daí seu nome) e pode ser representada em um grafo direcionado $G = (V, A)$, onde $V = \{1, \dots, m\}$ e as arestas $(i, k) \in A$, se e somente se $S_i \subset S_k$ e não existe $r \in V$ tal que $S_i \subset S_r \subset S_k$, sendo S um subconjunto de N . O grafo G é uma floresta. Dois casos particulares de PKE são importantes por sua aplicação prática e apresentados por Dudzinski e Walukiewicz (1987), a saber:

• O Problema da Mochila Decomposta (PKD)

$$\begin{aligned}
 & \text{(PKD) Maximizar } z = \sum_{j \in S} c_j x_j \\
 & \text{sujeito a:} \\
 & \sum_{j \in S_i} w_j x_j \leq b_i \quad i = 1, \dots, m \\
 & \sum_{j \in S} w_j x_j \leq b \\
 & x_i \in \{0, 1\}, \quad j \in S
 \end{aligned}$$

onde $S_i \cap S_k = \emptyset$ para $i, k = 1, \dots, m, i \neq k$, com $S = \bigcup_{i=1}^m S_i$ e $S_{m+1} = S$

• O Problema da Mochila Multiperíodo (PKMP)

Trata-se de um modelo com aplicações na programação industrial e mercado de capitais (ver também Salkin [1975]).

$$\begin{aligned}
 & \text{(PKMP1) Maximizar } z = \sum_{k=1}^m \sum_{j \in R_k} c_j x_j \\
 & \text{sujeito a:} \\
 & \sum_{k=1}^i \sum_{j \in R_k} w_j x_j \leq b_i \quad i = 1, \dots, m \\
 & x_j \in \{0, 1\}, \quad j \in R_k, k = 1, \dots, m
 \end{aligned}$$

onde para $i, k = 1, \dots, m, i \neq k$, com $S_i = \bigcup_{k=1}^i R_k$ e $S = S_m$ com $R_i \cap R_k = \emptyset$, para $i \neq k$.

O problema de sequenciamento de tarefas com prazos limite é um caso especial da mochila multi-período. Nesse caso, existem n tarefas a serem executadas em uma máquina com um tempo de processamento associado igual a t_j e um lucro resultante de p_j . As restrições de prazo são representadas por $d_j \geq t_j$ para todas as tarefas $j, j = 1, \dots, m$. Se uma tarefa é completada dentro do prazo, então o lucro p_j é realizado, sendo o retorno nulo em caso contrário. Se as tarefas são reindexadas de forma que:

$$d_1 \leq d_2 \leq \dots \leq d_n$$

podemos reformular o PKMP1 da seguinte forma:

$$(\text{PKMP2}) \text{ Maximizar } z = \sum_{j=1}^n p_j x_j$$

sujeito a:

$$\sum_{j=1}^i t_j x_j \leq d_i \quad i = 1, \dots, n$$

$$x_j \in \{0, 1\}, \quad j = 1, \dots, n.$$

onde $x_i = 0$ se a tarefa viola a restrição de tempo e $x_i = 1$ se está no prazo. Nesse caso $R_i = \{i\}$ e $S = \{1, \dots, i\}$ para $i = 1, \dots, n$ e $m = n$.

• O Problema Quadrático da Mochila (PKQ)

Este problema generaliza a mochila 0-1 multidimensional de forma que as variáveis são inteiras e a função objetivo é uma função quadrática. Esse modelo é aplicado em:

- Estudo de cenários de investimento de capital (Djerdjour *et al.* [1998]).
- Localização de estações de recepção em terra para comunicação via satélite (Witzgall [1975]).
- Localização de estações rodoviárias e pontes móveis de acesso aos aviões (*fingers*) (Billionnet e Calmels [1996]).
- Localização de estações de medição pluviométrica e seleção de carteiras de investimento (Laughunn [1970]).
- Roteamento de mensagem por satélite com limitação de memória (Billionnet *et al.* [1989]).

Gallo *et al.* (1980) formulam o PKQ como se segue:

$$(\text{PKQ}) \text{ Maximizar } z = \sum_{i=1}^n q_{ii} x_i + \sum_{1 \leq i < j \leq n} q_{ij} x_i x_j$$

sujeito a:

$$\sum_{i=1}^n w_i x_i \leq b$$

$$x_i \in \{0, 1\}, \quad i = 1, \dots, n.$$

Onde o coeficiente q_{ij} e o valor b são não negativos e inteiros tal que $0 < b < \sum_{i=1}^n w_i$. PKQ é

NP-completo, pois para o caso em que $q_{ij} = 0$ ele se reduz ao tradicional PK. Esse problema tem sido estudado por vários autores como Carter (1984), Barahona *et al.* (1989), Pardalos e Rodgers (1990), Billionnet *et al.* (1994), Chardaire e Sutter (1995) e Holmberg *et al.* (2000).

Para exemplificar o emprego do modelo PKQ vamos examinar o roteamento de mensagem por satélite com limitação de memória proposto por Billionnet *et al.* (1989). Esse problema tem uma descrição bastante interessante e rica. Presumindo-se que um satélite de comunicações possua um programa de trabalhos a cumprir, programa esse particionado em tarefas $T = \{1, \dots, n\}$, a questão que se coloca é como alocar essas tarefas entre o processador do satélite e o processador em terra. Normalmente, os processadores são homogêneos, mas as tarefas, obviamente, exigem um tempo diferente de processamento. O custo de processar a tarefa i no processador j é conhecido e igual a e_{ij} . Cada tarefa i requer a utilização de um segmento m_i da memória do processador. O processador do satélite possui um limite de memória igual a M sendo que o processador em terra tem uma capacidade considerada ilimitada. As capacidades para o fluxo de dados entre as ligações terra \times satélite e satélite \times terra são também consideradas ilimitadas (além da possível demanda gerada através do satélite). Se as tarefas i e j são transmitidas, então c_{ij} representa um custo conhecido de comunicação entre as tarefas i e j quando elas não são designadas no mesmo processador (custo apenas no chaveamento). O problema é definido como objetivando minimizar os custos totais de comunicação atendendo as condições de limite para a memória do processador do satélite. Billionnet e Camels (1996) formulam o problema como se segue:

$$\begin{aligned}
 \text{(PKSat) Maximizar } z = & \sum_{i=1}^n e_{i2} - \sum_{i=1}^n x_i \left(e_{i1} - e_{i2} + \sum_{j=i+1}^n c_{ij} + \sum_{j=1}^{i-1} c_{ij} \right) \\
 & + 2 \sum_{1 \leq i < j \leq n} c_{ij} x_i x_j \\
 \text{sujeito a:} & \\
 & \sum_{i=1}^n m_i x_i \leq M \\
 & x_i \in \{0, 1\}, \quad i = 1, \dots, n.
 \end{aligned}$$

onde $x_i = 1$ se a tarefa i foi alocada ao processador do satélite i e 0 se ela foi alocada ao processador em terra.

• Algoritmos de Solução

Os trabalhos de Wolsey (1988) e Martello e Toth (1990) apresentam uma extensa bibliografia que resume várias abordagens. De um modo resumido podemos classificar os principais algoritmos de solução, como normalmente é feito, nas abordagens exatas e heurísticas como mostra a Tabela 5.3.

TABELA 5.3 ALGORITMOS DE SOLUÇÃO PARA O PK

Problema	Ano	Pesquisador	Trabalho
Problema da Mochila	Abordagem Exata		
	1973	Ingargiola e Korsh	Algoritmos de redução
	1974	Horowitz e Sahni	Enumeração implícita
	1975	Ahrens e Finke	Programação dinâmica
	1977	Martello e Toth	Enumeração implícita

TABELA 5.3 CONTINUAÇÃO

<i>Problema</i>	<i>Ano</i>	<i>Pesquisador</i>	<i>Trabalho</i>
Problema da Mochila	Abordagem Exata		
	1977	Martello e Toth	Enumeração implícita
	1978	Zoltners	B&B
	1978	Martello e Toth	B&B
	1980	Toth	Programação dinâmica
	1980	Balas e Zemel	B&B
	1981	Martello e Toth	B&B e Relaxação lagrangeana
	1982	Fayard	Relaxação lagrangeana
	1983	Maculan	Relaxação lagrangeana
	1984	Dyer	Método dual para a relaxação linear
	1984	Camerini e Vercellis	Mochila em estrutura matroide
	1985	Martello e Toth	B&B
	1987	Dudzinski e Walukiewicz	Vários métodos exatos
	1988	Djerdjour <i>et. al.</i>	Relaxações <i>surrogate</i>
	1996	Bagchi <i>et. al.</i>	Relaxação linear
	1996	Amado e Barcia	Limites para a mochila matroide
	Abordagem Heurística		
	1975	Ibarra e Kim	Heurística IK
	1975	Sahni	Heurística SA – $O(n^3)$
	1979	Lawler	Melhoria da heurística IK
	1981	Magazine e Oguz	Heurística MO

✓ Transformação dos Problemas de Programação Inteira em Problemas da Mochila

Consideremos o problema de programação inteira (PPI) clássico s seguir:

$$\begin{aligned}
 \text{(PPI) Maximizar } z &= \sum_{j=1}^n c_j x_j \\
 \text{sujeito a:} \\
 \sum_{j=1}^n w_{ij} x_j &= b_i, \quad i = 1, 2, \dots, m \\
 x_j &\geq 0 \text{ e inteiro} \\
 0 &\leq x_j \leq u_j
 \end{aligned}$$

onde u_j é um limite superior para a variável x_j com valor inteiro e todos os b_i e w_{ij} têm igualmente valores inteiros. Para que o modelo de PPI seja equivalente ao de PKI é necessário que seja possível transformar PPI em PK retirando-se duas diferenças básicas:

1. As restrições de PPI são de estrita igualdade.
2. PPI possui m restrições.

A primeira diferença pode ser facilmente eliminada através da inclusão de variáveis de folga e do recálculo do limite superior. A segunda diferença pode ser eliminada com o seguinte procedimento: Substituímos as m restrições do modelo PPI por apenas uma da forma:

$$\sum_{j=1}^n w_j x_j \leq b$$

$$x_j \in \{0, 1\}, \quad j = 1, \dots, n.$$

onde cada w_j tem valor inteiro. Os dois problemas são equivalentes porque têm a mesma função objetivo e todas as soluções viáveis para o conjunto de restrições de PPI são também viáveis para PKI, e vice-versa. Vamos exemplificar o processo de agregação de duas restrições em uma só, através da manipulação dos limites e do valor do vetor w_j . Sejam duas restrições a saber:

$$\sum_{j=1}^n r_j x_j = b_1$$

$$\sum_{j=1}^n s_j x_j = b_2$$

Façamos então:

$$m_1 = \sum_{j=1}^n [\text{Máximo}(0, s_j)] U_j - b_1$$

$$m_2 = \sum_{j=1}^n [\text{Mínimo}(0, s_j)] U_j - b_1$$

$$m = \text{Máximo}(m_1, |m_2|)$$

E podemos substituir as duas restrições pela seguinte restrição derivada dos valores anteriores:

$$\sum_{j=1}^n (r_j - Ms_j)x_j = b_1 + Mb_2$$

onde M é qualquer número inteiro tal que $|M| > m$.

5.2.2 – Visão Geral dos Métodos de Solução para os PPI

Existem diversas técnicas desenvolvidas para a busca da solução inteira dos problemas de programação linear, das quais podemos citar:

Técnicas de Enumeração:

- Separação e avaliação progressiva ou *Branch-and-Bound* (B&B).
- Enumeração implícita.
- Restrições *surrogate*.

Técnicas de Cortes:

- Cortes inteiros (primais e duais).
- Cortes combinatórios.
- Cortes de interseção.
- Método de decomposição de Benders.

Técnicas Híbridas:

- *Branch-and-Cut*.
- Teoria de Grupo.

Na maioria das ocasiões, as técnicas de solução são especializadas para os inúmeros tipos de problemas de programação inteira, desenvolvendo-se abordagens e algoritmos específicos para cada situação. Apresentaremos agora duas técnicas exatas muito empregadas na solução de problemas de PLI.

5.2.3 – Branch-and-Bound

O método denominado de *Branch-and-Bound* (B&B) baseia-se na ideia de desenvolver uma enumeração inteligente dos pontos candidatos à solução ótima inteira de um problema. O termo *branch* refere-se ao fato de que o método efetua partições no espaço das soluções. O termo *bound* ressalta que a prova da otimalidade da solução utiliza-se de limites calculados ao longo da enumeração. Definindo:

$$(P) = \text{Maximizar } \{cx \mid Ax = b, x \geq 0, x \in Z^+\}$$

$$\text{e}$$

$$(\bar{P}) = \text{Maximizar } \{cx \mid Ax = b, x \geq 0, x \in R^+\}$$

Definindo ainda $V^*(P)$ e $V^*(\bar{P})$ os valores das funções objetivo no ótimo de (P) e (\bar{P}) respectivamente temos que:

$$V^*(P) \leq V^*(\bar{P})$$

Considerando ainda qualquer solução viável \tilde{x} de (P) e chamando de $V(\tilde{x})$ o valor da função objetivo no ponto \tilde{x} , então:

$$V(\tilde{x}) \leq V^*(P)$$

e dessa forma $V^*(\bar{P})$ é um limite superior para (P) e qualquer de suas soluções viáveis. Se \bar{x} é a solução ótima de (\bar{P}) tal que \bar{x}_j é não inteiro teremos:

$$x_j \geq \lfloor \bar{x}_j \rfloor + 1 \text{ ou } x_j \leq \lfloor \bar{x}_j \rfloor \quad (5.1)$$

em toda solução viável de (P) . Dessa forma, o problema (P) pode ser dividido em dois novos problemas (P_1) e (P_2) em que a envoltória convexa C de $(P_1) \cup (P_2)$ ou $\text{Conv}((P_1) \cup (P_2))$ está estritamente contida na envoltória de (P) ou $\text{Conv}((P_1) \cup (P_2)) \subset \text{Conv}((P))$.

Vamos exemplificar o processo de divisão da *envoltória convexa* de (P) com o seguinte exemplo:

Maximizar $z = 5x_1 + 8x_2$
sujeito a:

$$\begin{aligned} x_1 + x_2 &\leq 6 \\ 5x_1 + 9x_2 &\leq 45 \\ x_1, x_2 &\in Z^+ \end{aligned}$$

Que pode ser representado graficamente como na Figura 5.3. A solução ótima contínua do problema é encontrada em: $x_1 = \frac{9}{4}$; $x_2 = \frac{15}{4}$ levando a $z = 41\frac{1}{4}$. Desenvolvendo a ideia de separação da envoltória convexa em relação a variável x_2 podemos organizar a equação disjuntiva a seguir:

$$x_2 \geq \left\lfloor \frac{15}{4} \right\rfloor + 1 \geq 4 \text{ ou } x_2 \leq \left\lfloor \frac{15}{4} \right\rfloor \leq 3$$

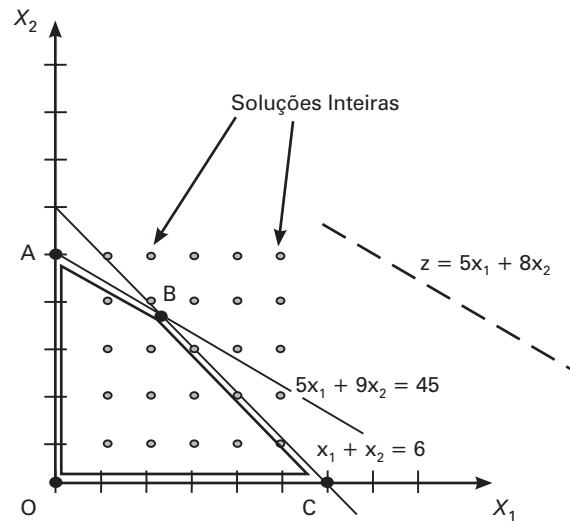


FIGURA 5.3 Solução gráfica do exemplo.

A equação anterior produz duas restrições disjuntivas que, quando acrescentadas ao problema original são capazes de criar dois novos problemas que não mais possuem a solução ótima contínua em sua envoltória convexa.

A Figura 5.4 mostra o efeito da aplicação da equação disjuntiva gerada em relação à variável x_2 , no exemplo.

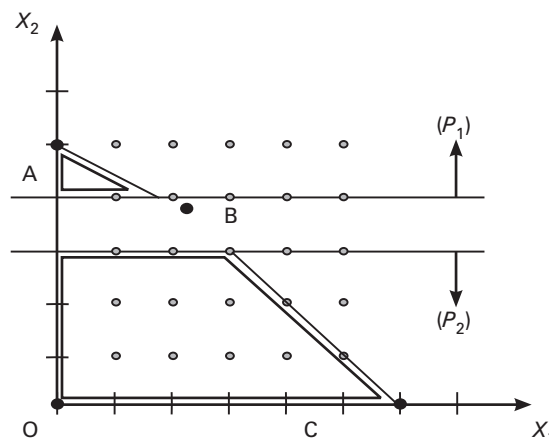


FIGURA 5.4 Resultado do branch (divisão ou ramificação).

Com a consideração da disjunção, o problema original será reduzido então a dois novos problemas, a saber:

$$\begin{array}{ll}
 (P_1) \text{ Maximizar } z = cx & (P_2) \text{ Maximizar } z = cx \\
 \text{sujeito a:} & \text{sujeito a:} \\
 Ax \leq b & Ax \leq b \\
 x_i \leq \lfloor x_i^0 \rfloor & x_i \leq \lfloor x_i^0 \rfloor + 1 \\
 x_i \in \mathbb{Z} & x_i \in \mathbb{Z}
 \end{array}$$

A estratégia de separação cria novos e mais restritos problemas que, normalmente, serão de mais fácil solução. No exemplo, o problema (P) é separado em dois problemas (P_1) e (P_2) com x_2 sendo a variável escolhida para separação. A estratégia de separação pode ser reaplicada a esses problemas em função, por exemplo, da variável x_1 . Vamos enumerar através de uma árvore as possibilidades de solução dos problemas que serão gerados pela divisão de (P) . Na árvore da Figura 5.5, cada nível representa uma separação ou *branch* em relação a uma variável.

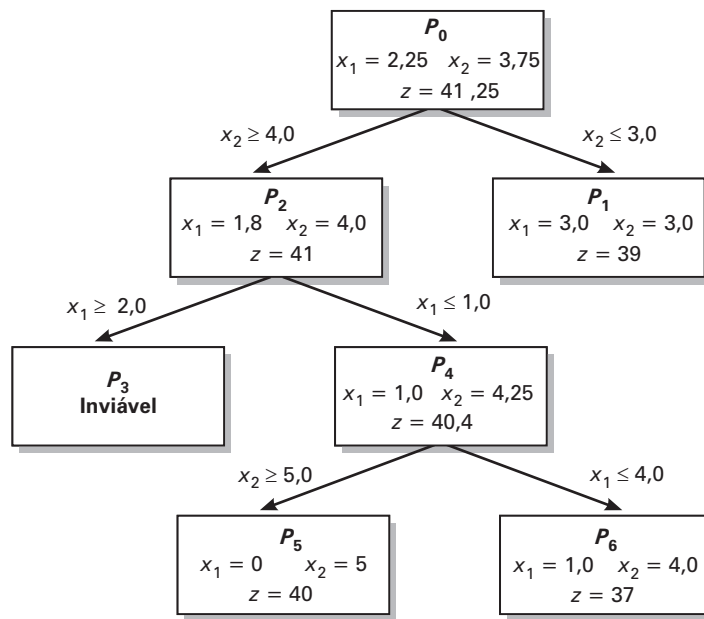


FIGURA 5.5 Árvore branch do exemplo.

Para entendermos o efeito do *bound*, suponhamos que tenhamos escolhido a sequência de *branches* da Figura 5.6, deixando de solucionar os problemas marcados:

As soluções contínuas são um limite superior para o valor de z_0^* , sob as condições estabelecidas nos vértices da árvore, enquanto as soluções inteiras geram um limite inferior. Como (P_4) , um problema com solução contínua, possui $Z = 40,4$ e (P_5) , um problema com solução inteira, possui $Z = 40$, o problema (P_6) não precisa mais ser solucionado, uma vez que entre $40,4$ e 40 não existe a possibilidade de uma outra solução inteira melhor que 40 ($40 \leq z_0^* \leq 40,4$). O problema (P_2) , com $z = 41$ pode dar origem, contudo, ainda a um problema com uma solução inteira de valor 41 ($40 \leq z_0^* \leq 41$), o que obriga ao desenvolvimento de (P_3) . De modo semelhante, (P_0) , com $z = 41,25$ pode dar origem a um problema com a solução também de valor 41 ($40 \leq z_0^* \leq 41,25$), o que obriga ao desenvolvimento de (P_1) .

A redução pelo limite inferior (*bound*) de apenas um vértice da árvore de enumeração do exemplo pode parecer pequena, mas devemos lembrar que esse problema é pequeno também. Em muitos casos reais, o poder de simplificação do limite inferior (ou superior no problema de minimização) se mostra significativo, sendo extremamente útil no processo de solução.

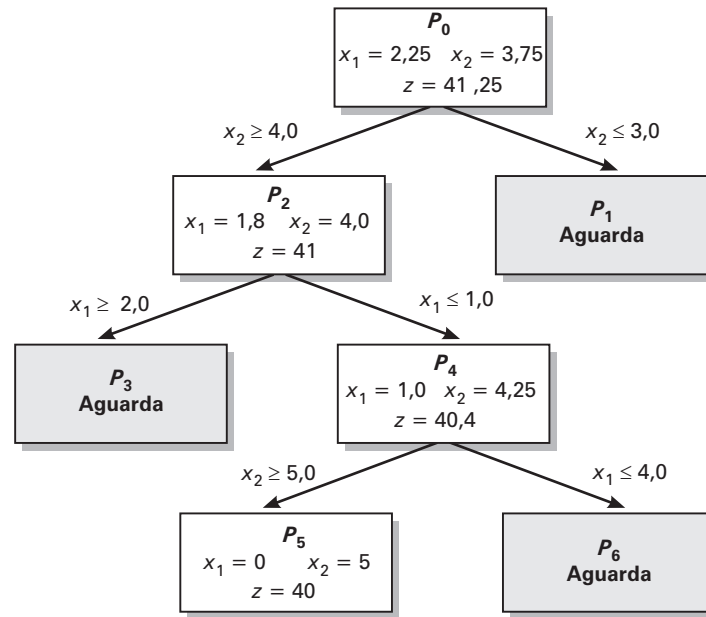


FIGURA 5.6 O efeito de redução do bound (limite).

Um dos pontos fundamentais para o sucesso do B&B é a qualidade do limite gerado pela solução inteira. Em várias situações, esses limites podem ser alcançados através de procedimentos heurísticos. A qualidade do limite alcançado normalmente depende, para cada problema, da estratégia de desdobramento da árvore de busca. Existem basicamente duas grandes estratégias de divisão ou *branch*. A Figura 5.7 apresenta o aspecto das árvores desenvolvidas pela busca em profundidade e pela busca em largura.

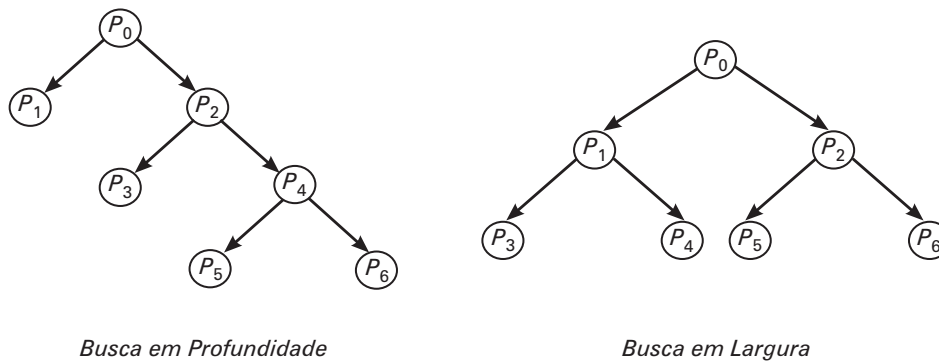


FIGURA 5.7 Estratégias de divisão.

O B&B é uma técnica de ampla aplicação. A ideia geral é sujeita a inúmeras adaptações e estratégias de implementação. Basicamente, os aspectos envolvidos são:

Técnicas de desenvolvimento da árvore de enumeração (escolha do problema):

- Busca em profundidade.
- Busca em largura.
- Variantes híbridas.

Técnicas de formação da árvore (escolha da variável de separação):

- Variante de Dank (1960).
- Variante de Land e Doig (1965).
- Variante de Spielberg (1968).
- Método das Penalidades (1965).
- Método de Taha (1971).
- Estratégias dinâmicas (1976).
- Outras variantes.

Técnicas complementares para obtenção dos limites:

- Relaxação linear (ver Pirce [1964], Bagchi *et al.* [1996], Pardulos *et al.* [1996]).
- Relaxação lagrangeana (ver Fisher [1981], Fisher [1985] e Beasley [1985], Desrosiers *et al.* [1988], Kohl e Madsen [1997] e Holmberg e Yuan [2000]).
- Algoritmos heurísticos e meta-heurísticos (ver Puckinger [2004]).
- Cortes.

✓ Variante de Dank

Este autor propõe que a variável a ser escolhida para a divisão em um certo nível da árvore seja a que possuir o maior *resíduo* em relação à solução inteira, ou seja, considerando que \tilde{x}_j seja a variável que escolheremos para a separação proposta pela equação (5.2), então:

$$\tilde{x}_j = \text{Máximo } \{(x_j - \lfloor x_j \rfloor)\} \text{ e } \{(x_j - \lceil x_j \rceil)\} \quad (5.2)$$

Se a solução contínua do nó a ser expandido se apresenta como na Figura 5.8 (a), então, pela proposta de Dank, o nó \tilde{x}_j será x_2 , uma vez que possui um resíduo igual a 0,75, maior que o de x_1 .

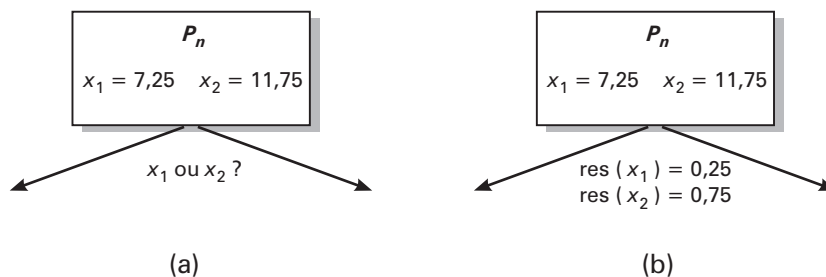


FIGURA 5.8 Estratégias de divisão de Dank.

✓ Variante de Land e Doig

Esta abordagem sugere que a árvore de enumeração seja expandida em vários valores simultâneos para a variável de divisão, como mostra a Figura 5.9.

✓ Variante de Spielberg

Desenvolve o nó com maior valor de z^* e o mais recentemente calculado, usando o critério de Land e Doig para aumentar a retirada de espaço contínuo no entorno do nó pesquisado. Trata-se de uma busca em profundidade associada ao critério de Land e Doig.

✓ Método das Penalidades

Esta estratégia pode ser descrita nos seguintes passos:

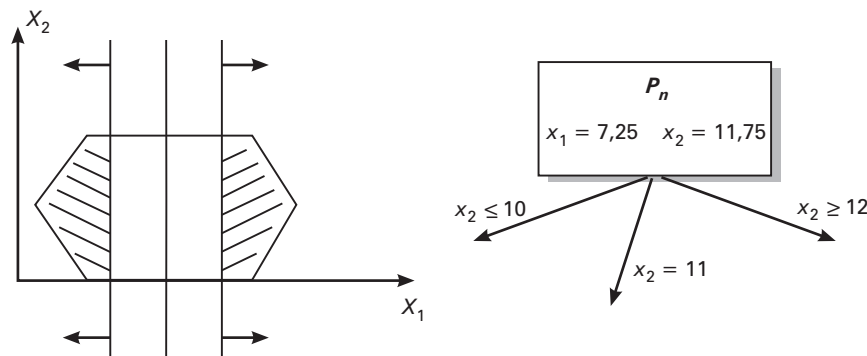


FIGURA 5.9 Estratégias de divisão de Land e Doig.

- Determina uma estimativa do decréscimo da função objetivo quando obrigamos as variáveis a assumir valores inteiros.
- Escolhe o nó na lista dos nós em aberto com a menor estimativa de decréscimo (utiliza um pivoteamento associado a variável \tilde{x}_j).
- Aproveita as reduções possíveis quando a estimativa de decréscimo ultrapassa o limite inferior corrente.

5.3 – PROGRAMAÇÃO DINÂMICA

A programação dinâmica é uma técnica utilizada para a otimização de processos de decisão multietágios. Denominamos um *processo de decisão multietágios* aquele que pode ser desdobrado segundo um certo número de etapas sequenciais ou estágios. As alternativas incluídas na conclusão de um estágio são denominadas *decisões*. A condição do processo dentro de cada estágio é denominada *estado*. Cada estágio inclui a tomada de uma decisão que pode ou não alterar o estado do processo, mas que, obrigatoriamente, representa uma transição entre o estado corrente e o estado futuro do processo. Um processo de decisão multietágios é denominado *finito* quando existem apenas um *número finito de estágios* no processo e um *número finito de estados* possíveis associado a cada estágio. Dentro do processo multietágios, o objetivo do tomador de decisão é encontrar uma política ótima (também chamada de trajetória ótima) em relação ao retorno auferido com as decisões. Um processo de decisão multietágios é determinístico se o resultado de cada decisão for conhecido exatamente. A Figura 5.10 mostra o quadro da tomada de decisão multietágio:

A determinação de uma política ótima para um processo de decisão multietágio está teoricamente embasada no princípio da otimalidade de Bellman, que pode ser assim enunciado:

Uma política ótima apresenta a propriedade segundo a qual, a despeito das decisões tomadas para assumir um estado particular num certo estágio, as decisões restantes a partir deste estado devem constituir uma política ótima.

O princípio de Bellman guarda uma semelhança com a estratégia gulosa ou míope, que determina que a decisão a adotar em cada estágio é sempre aquela que produz o maior acréscimo no critério de otimização. A estratégia gulosa trata todos os processos como se fossem multietágio e considera a tomada de decisão apenas dentro das condições vigentes para as variáveis no estágio. No caso de o problema abordado ser realmente passível de decomposição multietágio e o processo de decisão ser, face

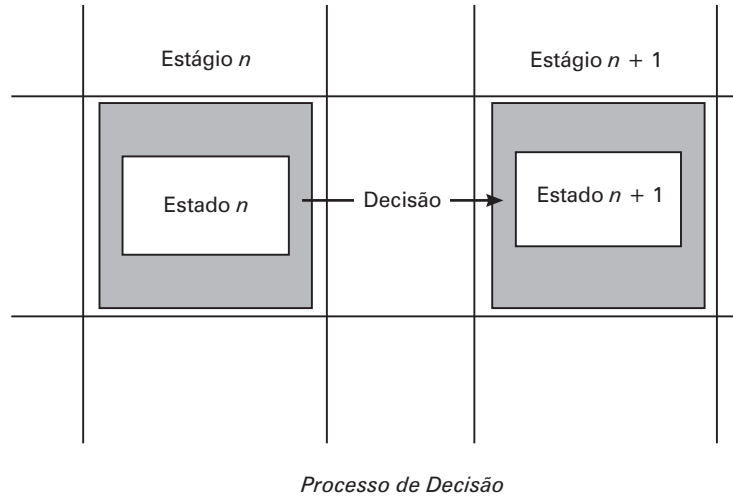


FIGURA 5.10 Um processo de decisão multiestágio.

as características da função de avaliação, completamente independente a cada estágio, a estratégia gulosa confunde-se com a da programação dinâmica e a solução via algoritmo guloso conduz ao ótimo. Isso acontece quando o problema está assentado em um matroide (uma estrutura de independência – ver anexo). Esse caso particular resume toda a possibilidade de semelhança entre o princípio de Bellman e a estratégia gulosa.

Para implementar-se o princípio de Bellman, na prática parte-se do último estágio de um processo com n estágios e determina-se a melhor política para se deixar aquele estágio e completar o processo. Desloca-se, então, do fim para o início do processo, estágio após estágio, repetindo-se o raciocínio. Em cada estágio, para cada estado, determina-se a melhor política para deixar o estágio e completar o processo, supondo-se sempre que os estágios anteriores foram completados de forma ótima. Os cálculos são sempre aproveitados de um estágio para o outro. Os elementos correspondentes ao último estágio do processo são, geralmente, obtidos diretamente. Os demais elementos são calculados de forma recursiva. A expressão de recorrência depende do problema e deve ser obtida para cada tipo de processo multiestágio. Se desejamos solucionar o problema DIM:

$$\begin{aligned} \text{(DIM) Otimizar } z &= f_1(x_1) + f_2(x_2) + \dots + f_n(x_n) \\ \text{sujeito a:} \\ x_1 + x_2 + \dots + x_n &\leq b \\ x_i &\in \mathfrak{S}^+ \end{aligned}$$

no qual as funções possivelmente não lineares $f_1(x_1), f_2(x_2), \dots, f_n(x_n)$ são conhecidas de uma variável, b é um valor não negativo e inteiro conhecido e o problema pode ser modelado como um processo multiestágio. O estágio 1 envolve a decisão sobre a variável x_1 , com uma contribuição resultante de $f_1(x_1)$. Os estados são $1, 2, 3, \dots, b$, representando os valores possíveis para o número de unidades disponíveis para alocação. Considerando:

- $u \equiv$ variável de estado cujos valores especificam os estados.
- $m_j(u) \equiv$ retorno ótimo para se completar o processo começando-se no estágio j com o estado u .
- $d_j(u) \equiv$ decisão tomada no estágio j que obtém $m_j(u)$.

Para o modelo DIM os valores de $m_j(u)$ serão dados pela seguinte expressão:

$$m_n(u) = \underset{0 \leq x \leq u}{\text{ótimo}} \{f_n(x)\}$$

que leva à equação de recorrência:

$$m_j(u) = \underset{0 \leq x \leq u}{\text{ótimo}} \{f_j(x) + m_{j+1}(u - x)\}$$

Considerando o vetor da decisão ótima $x^* = x_1^*, x_2^*, \dots, x_n^*$ que leva o critério de decisão a assumir o valor z^* , as componentes do vetor poderão ser encontradas sequencialmente com a seguinte recorrência:

$$\begin{aligned} x_1^* &= d_1(b); \\ x_2^* &= d_2(b - x_1^*) \\ &\dots = \dots \\ x_n^* &= d_n(b - x_1^* - x_2^* - \dots - x_{n-1}^*) \end{aligned}$$

Para que possamos compreender a aplicação do princípio de Bellman apresentaremos agora alguns exemplos práticos:

✓ Exemplo 1: Caminho mais curto

Apesar da eficiência do algoritmo de Dijkstra para o caminho mais curto (ver Capítulo 6), esse problema possui uma solução bastante interessante quando abordado pela técnica de programação dinâmica. No grafo da Figura 5.11, o problema proposto é o de encontrar o caminho mais curto entre os vértices A e J.

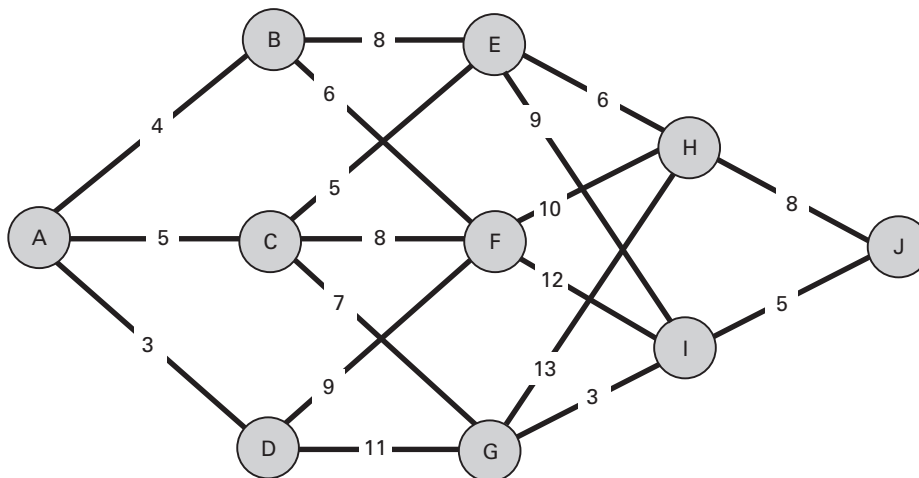


FIGURA 5.11 Grafo do exemplo 1.

Como o grafo da Figura 5.11 é um grafo em camadas (qualquer grafo pode ser transformado em um grafo em camadas – ver Hu [1982]), podemos decompor o problema de encontrar o caminho mais curto entre os vértices A e J em um processo de tomada de decisão multietápio. As fases da decisão do problema estão mostradas na Figura 5.12.

Realmente, vários problemas de decisão sequencial admitem um grafo de tomada de decisão semelhante ao grafo do caminho mais curto. Nesse sentido, estudar a aplicação da programação dinâmica ao caminho mais curto é estudar o caso de muitos outros modelos. Aplicando diretamente o princípio de Bellman ao problema, vamos estabelecer as condições para a tomada de decisão da primeira fase (fase final do caminho) do problema multietápio. A Figura 5.13 ressalta as possibilidades para a política ótima de chegada aos vértices H e I, a partir de J, considerando que o caminho até os vértices em pauta será realizado também de forma ótima:

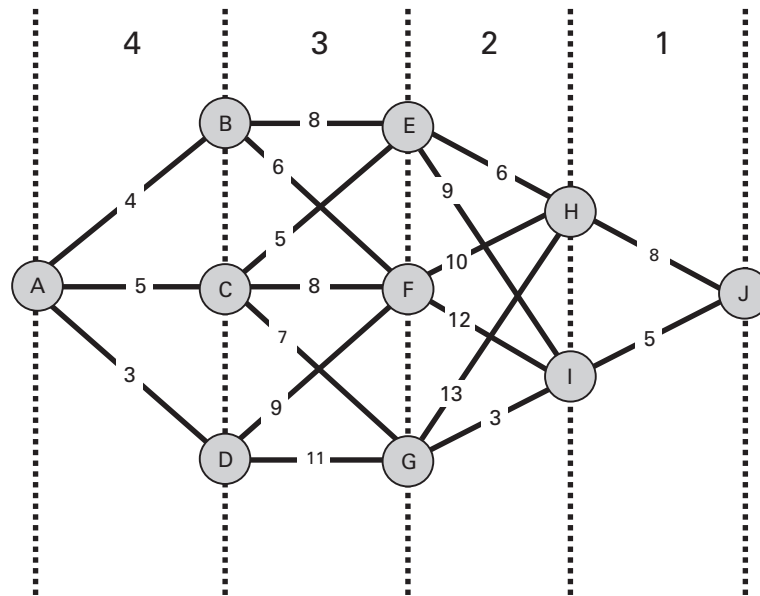


FIGURA 5.12 Fases de decisão do caminho mais curto.

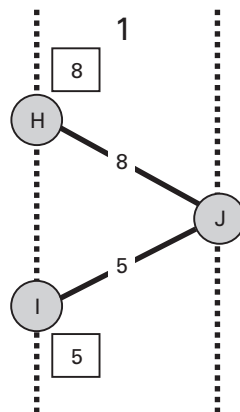


FIGURA 5.13 Primeira fase de decisão – política possível para os vértices H e I.

Com a tomada de decisão realizada na fase 1 podemos, a partir da política ótima para H e I, calcular a política ótima para E, F e G, como mostra a Figura 5.14.

A partir dos vértices E, F e G podemos desenvolver a terceira fase de decisão. A Figura 5.15 mostra que as informações constantes da primeira fase (vértices H e I) não são mais necessárias para isso.

Finalmente, podemos concluir os cálculos determinando o caminho mais curto e seu valor (o rótulo de A), como mostra a Figura 5.16.

✓ Exemplo 2: O problema da seleção de projetos

As fases de tomada de decisão mais comuns estão associadas a uma só variável. Nesse sentido cada estágio permite analisar a política ótima para uma variável. Seja o problema a seguir:

Um investidor dispõe de 3 unidades de capital que podem ser investidas em 3 distintas atividades produtivas. Para cada atividade produtiva é esperado um certo retorno de capital que é uma função do montante investido na própria atividade. As quantidades de capital são referentes a lotes de ações e podem, portanto, ser consideradas inteiras. As receitas auferidas em retorno são independentes entre si. A Tabela 5.4 mostra o quadro de investimento \times retorno dentro de cada projeto produtivo.

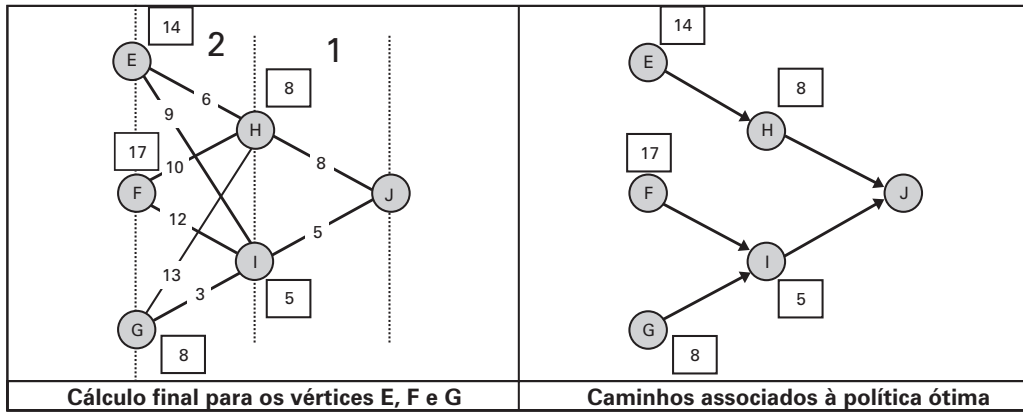


FIGURA 5.14 Segunda fase de decisão – política para os vértices E, F e G.

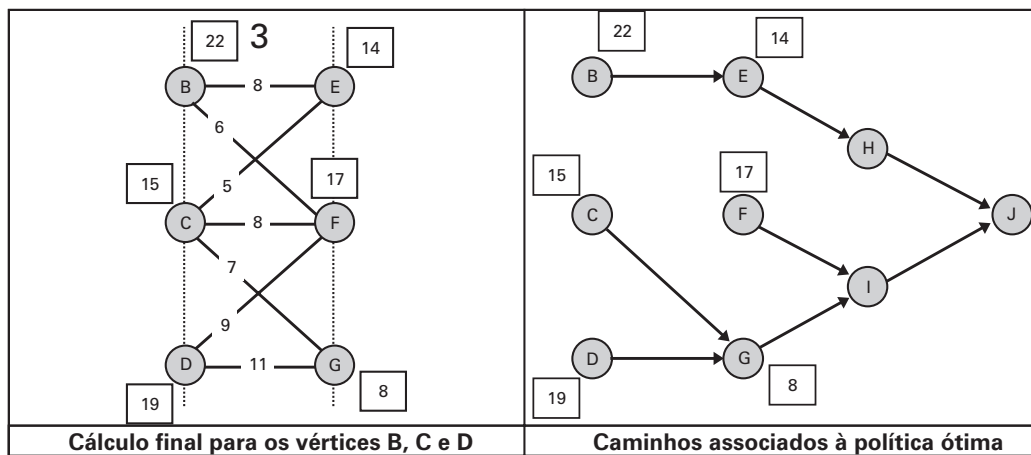


FIGURA 5.15 Terceira fase de decisão – política para os vértices B, C e D.

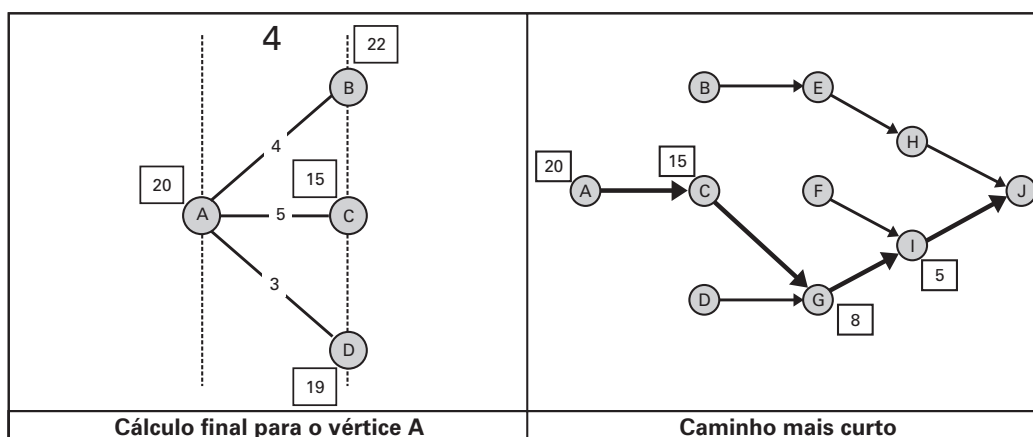


FIGURA 5.16 Fase final de decisão.

TABELA 5.4 SITUAÇÃO INVESTIMENTO \times RETORNO

<i>Quantidade de recursos aplicada (10^3 reais)</i>	<i>Receitas</i>		
	<i>Projeto 1</i>	<i>Projeto 2</i>	<i>Projeto 3</i>
0	0	0	0
1	2	1	3
2	4	5	5
3	6	6	6

Para esse problema definiremos a variável de decisão x_j , $j = 1, 2, 3$, como a quantidade de recurso aplicada no projeto produtivo j , $x_j \in \{0, 1, 2, 3\}$.

Temos uma restrição para esse problema: a quantidade total de capital disponível que não poderá exceder a 3 unidades. A função objetivo é descontínua e não linear e pode ser representada por:

$$z = f_1(x_1) + f_2(x_2) + f_3(x_3)$$

com os valores constantes da tabela.

Desse modo, podemos formular o problema de seleção e alocação em projetos da seguinte forma:

$$(\text{Projeto}_1) z = \sum_{j=1}^3 f_j(x_j)$$

sujeito a:

$$\begin{aligned} x_1 + x_2 + x_3 &\leq 3 \\ x_j &\geq 0 \text{ e inteiro } j = 1, 2, 3. \end{aligned}$$

Decisões:

$x_j \equiv$ quanto vai ser alocado ao projeto j .

Estágios:

$k = 0$, antes de qualquer decisão.

$k = 1$, após decidir o valor de x_1 .

$k = 2$, após decidir o valor de x_2 (e x_1).

$k = 3$, após decidir o valor de x_3 (e x_1 e x_2).

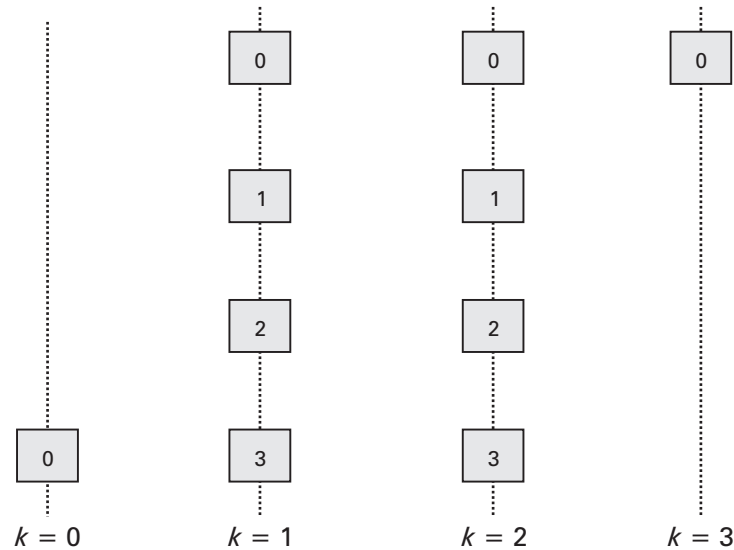
Estados:

São definidos a cada estágio pelo capital ainda disponível para aplicação.

Processo:

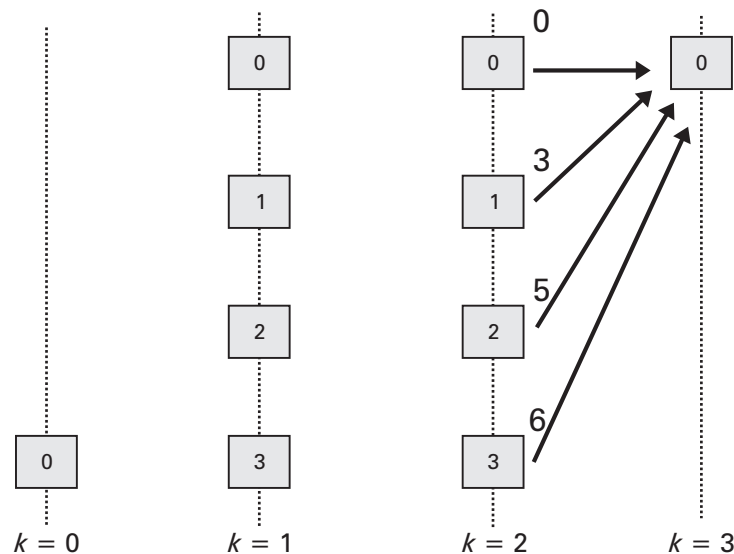
O investidor, o capital disponível e os projetos.

Semelhante ao exemplo 1, podemos construir um diagrama que nos permita acompanhar graficamente o comportamento estado \times estágio do processo. Vamos representar os valores para o capital dentro de retângulos. A política ótima será representada por uma trajetória ou caminho no diagrama


 FIGURA 5.17 Diagrama estado \times estágio.

capaz de rotular o nó final com o maior valor possível. Os cálculos serão realizados estágio a estágio, como no grafo do caminho mais curto, aproveitando-se todas as conclusões obtidas em cada estágio para a tomada de decisão em estágios anteriores.

Colocaremos junto ao canto superior direito de cada retângulo o valor de retorno correspondente à tomada de decisão a ele associada. Inicia-se a solução no último estágio decidindo sobre a variável x_3 . O retorno obtido com um investimento no projeto 3 (variável x_3) é a coluna do projeto 3 da Tabela 5.4. A transição do estágio 2 ($k=2$) para o 3 ($k=3$) decide a variável 3. As possibilidades dessa decisão podem ser resumidas na Figura 5.18.


 FIGURA 5.18 Primeira fase de decisão – política para a variável x_3 .

As opções da variável x_2 podem ser analisadas na Figura 5.19, a partir da primeira fase de decisão. Chegando finalmente à análise de x_1 , encontramos a política (trajetória) ótima da Figura 5.20.

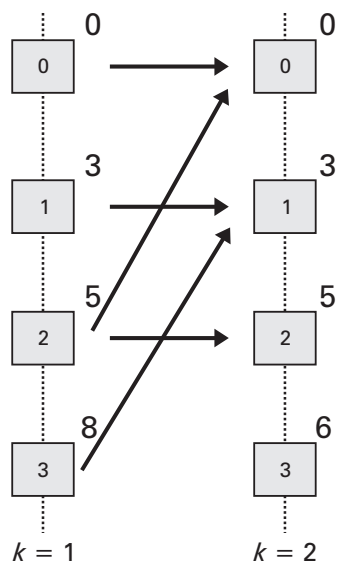
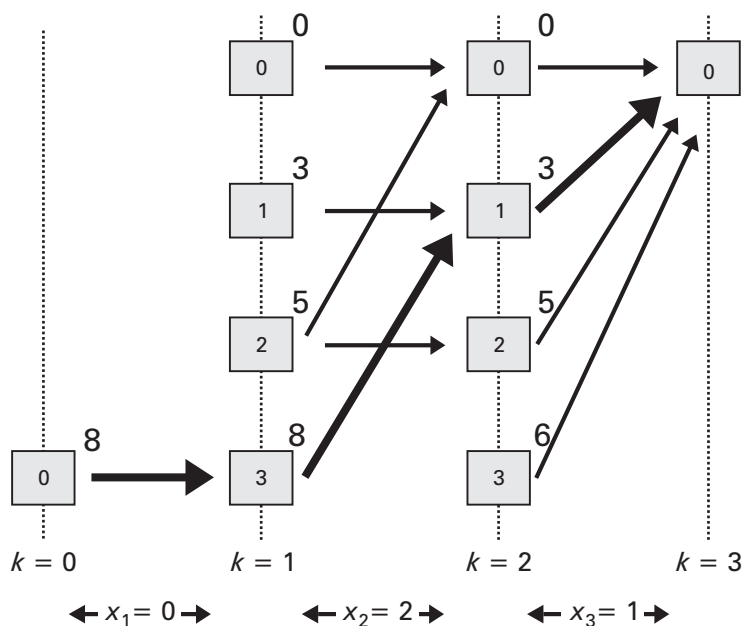
FIGURA 5.19 Política ótima para x_2 

FIGURA 5.20 Diagrama final do exemplo 2.

✓ Exemplo 3: O problema da produção e estocagem

Um fabricante de vagões ferroviários assina um contrato com uma empresa de transporte ferroviário para o fornecimento de 100 vagões frigoríficos em um período de cinco anos. Em janeiro de cada ano é prevista a entrega de 20 vagões. O custo de produção anual por lote de 10 vagões, segundo as projeções do departamento de engenharia, deve variar como especificado na Tabela 5.5.

TABELA 5.5 CUSTOS DE PRODUÇÃO DOS VAGÕES

<i>Produção anual</i>	0	1	2	3	4	5
<i>Custo (unidade monetária)</i>	5	12	16	19	21	22

A capacidade de armazenagem do fabricante é de 4 lotes (40 vagões), sendo que a estocagem consome em manutenção, mobilização da área e segurança, uma unidade monetária por lote armazenado por ano.

O fabricante começa a produção em 10 de janeiro e termina em 31 de dezembro, sendo que os custos de armazenamento do lote ao longo do ano de produção já estão computados na Tabela 5.5. No início do contrato não existe qualquer vagão frigorífico em estoque e não existe qualquer interesse em que permaneçam vagões após o término do contrato. Estabeleça a política ótima de produção, entrega e armazenagem para a indústria contratada.

Decisões:

$x_j \equiv$ número de lotes a produzir no ano j , $j = 1, 2, 3, 4, 5$

Estágios:

$k = 0$, antes de qualquer lote ser produzido.

$k = j$, após decidir o valor de x_j .

Estados:

Número de lotes em estoque no final de cada estágio, conforme a Tabela 5.6.

TABELA 5.6 ESTOQUE DE VAGÕES

k	<i>Estados Viáveis</i>
0	0
1	0 1 2 3
2	0 1 2 3 4
3	0 1 2 3 4
4	0 1 2
5	0

Variável de decisão auxiliar

$u_j \equiv$ número de lotes estocados no início do ano j , $j = 1, 2, 3, 4, 5$

Restrições

$$\sum_{j=1}^5 x_j = 10$$

$$0 \leq x_j \leq 5 \text{ e inteiro}$$

$$0 \leq u_j \leq 4 \text{ e inteiro}$$

$$u_1 = 0$$

$$u_6 = 0$$

Processo:

O fabricante, o cliente, a capacidade de fabricação e o estoque.

Podemos formular o problema de planejamento da produção e estocagem da seguinte forma:

$$\text{Min } z = \sum_{j=1}^5 f_j(x_j, u_j)$$

sujeito a:

$$\sum_{j=1}^5 x_j = 10$$

$$0 \leq x_j \leq 5, \quad j = 1, \dots, 5$$

$$u_{j+1} = x_j + u_j - 2$$

$$0 \leq u_j \leq 4, \quad j = 1, \dots, 6$$

$$u_1 = 0$$

$$u_6 = 0$$

$$x_j \geq 0 \text{ e inteiro}, \quad j = 1, \dots, 5$$

$$u_j \geq 0 \text{ e inteiro}, \quad j = 1, \dots, 6$$

O problema pode ser representado por meio do diagrama da Figura 5.21. A representação está em função da variável estoque. As variáveis $u_1 = u_6 = 0$. Do diagrama temos que $u_2 = 3$, ou seja, no primeiro ano serão produzidas cinco unidades, entregues duas e mantidas três em estoque. $u_3 = 1$ significa que no segundo ano não haverá produção. $u_4 = 1$ implica que no terceiro ano serão produzidas mais cinco unidades que, somadas com uma em estoque, acumularão seis lotes. Como serão entregues dois lotes

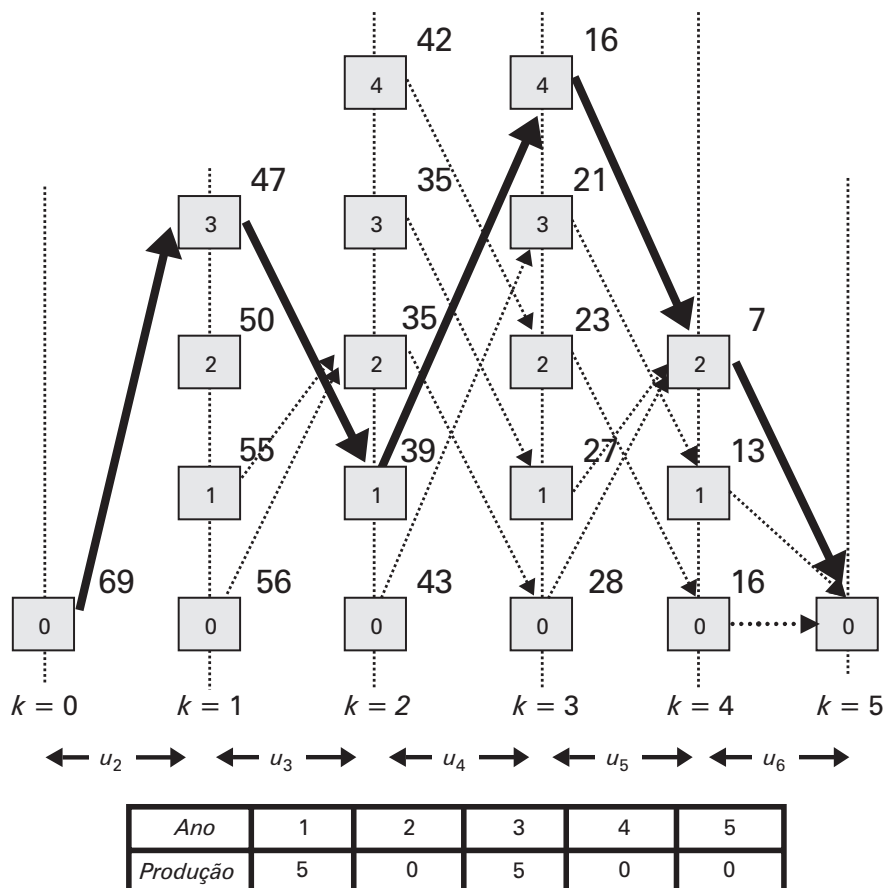


FIGURA 5.21 Solução do exemplo 3.

ao final do terceiro ano, permanecerão quatro em estoque. Daí para diante as duas próximas entregas anuais serão atendidas pelo estoque. O quadro ano/produção resume o valor das variáveis x . A programação dinâmica é uma técnica utilizada intensamente para a solução de modelos combinatórios (ver Haims e Freeman [1970], Psaraftis [1980], Carraway *et al.* (1989), Potts *et al.* (1995), Faaland e Brigs [1984]).

5.4 – SOLUÇÕES HEURÍSTICAS PARA OS PLI

Enquanto os problemas lineares contínuos possuem no simplex um algoritmo muito eficiente para a solução exata, os problemas lineares discretos, salvo alguns casos particulares, normalmente carecem da mesma sorte. O estudo de Garey e Johnson (Garey e Johnson [1979]) é um marco para o entendimento dos obstáculos que se interpõem entre uma possível solução teórica para um problema de programação discreta e sua implementação prática através dos instrumentos computacionais da atualidade. O cerne da dificuldade da abordagem exata dos problemas denominados NP-árduos, que, por sinal, representam uma grande parte dos problemas de PLI realmente interessantes, está na explosão combinatoria dos métodos enumerativos. É fácil perceber que uma árvore de B&B pode envolver um número da ordem de nós, onde n representa a altura (distância entre o nó raiz e o nó mais extremo, também denominado folha da árvore) da árvore de enumeração. Para que possamos entender melhor essa dificuldade vamos solucionar o problema (PAT) que se segue:

$$\begin{aligned}
 &(\text{PAT}) \text{ Maximizar } z = x_1 \\
 &\text{sujeito a:} \\
 &\quad 2x_1 + 2x_2 + 2x_3 = 3 \\
 &\quad x_1, x_2, x_3 \in \{0, 1\}
 \end{aligned}$$

Como vimos no item 5.1, PAT é um modelo que pertence à classe dos problemas da mochila. Se decidirmos solucionar PAT através da utilização de uma estratégia B&B básica, a árvore de enumeração seria a representada na Figura 5.22.

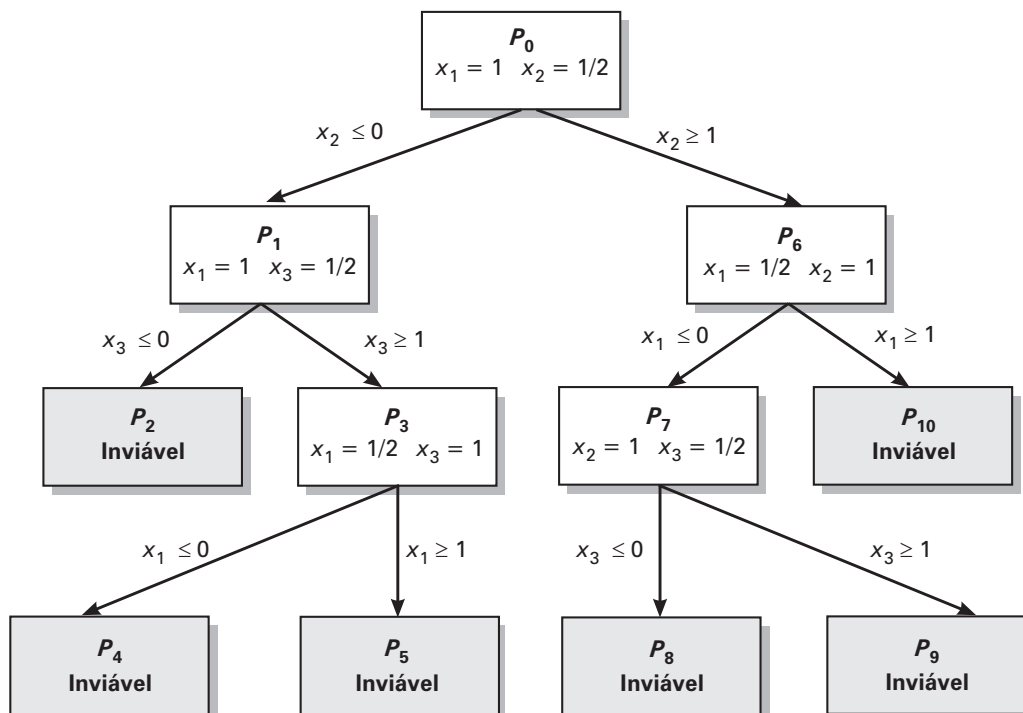


FIGURA 5.22 Árvore B&B de PAT.

Para concluir a inviabilidade de PAT foi necessário desenvolver 11 problemas. É fácil provar que esse algoritmo aplicado a um problema com n variáveis desenvolverá, no mínimo, cerca de $2^{\frac{n+1}{2}}$ nós. Se o problema possuir 201 variáveis (os problemas reais podem possuir da ordem de milhares ou milhões de variáveis), então a árvore será da ordem de 2^{100} nós. Um computador capaz de examinar 1,5 trilhão de nós por segundo dessa árvore levaria cerca de 537 milhões de anos para esgotar todas as possibilidades existentes!! Apesar de todo o avanço das técnicas atuais que, obviamente, fariam melhor que desenvolver a árvore de enumeração completa de PAT para determinar a inviabilidade do problema, ainda permanece a dificuldade imposta pela explosão combinatória desse tipo de problema. Em virtude dessa realidade, nos últimos anos tem se verificado o surgimento de um significativo conjunto de técnicas e algoritmos computacionalmente muito eficientes mas que não garantem a solução ótima do problema de programação linear inteira. Esses algoritmos são denominados de *heurísticos* ou *aproximativos* (para detalhes complementares ver, no anexo, complexidade de algoritmos).

O termo heurística é derivado do grego *heuriskein*, que significa descobrir ou achar. Mas o significado da palavra em pesquisa operacional vai um pouco além de sua raiz etimológica. Podemos dizer que uma heurística, no sentido dado ao termo, refere-se a um método de busca de soluções em que não existe qualquer garantia de sucesso. O sucesso do método pode ser expresso quantitativa ou qualitativamente. Em um problema de otimização o sucesso pode ser representado pela obtenção da solução ótima. Alguns autores entendem que a possibilidade de fracasso seja extensiva até para a obtenção de uma solução viável. Não comungamos dessa opinião. Definiremos o termo da seguinte forma:

Uma heurística é uma técnica que busca alcançar uma boa solução utilizando um esforço computacional considerado razoável, sendo capaz de garantir a viabilidade ou a otimalidade da solução encontrada ou, ainda, em muitos casos, ambas, especialmente nas ocasiões em que essa busca partir de uma solução viável próxima ao ótimo.

As heurísticas iniciaram historicamente seu desenvolvimento a partir de problemas específicos. As primeiras heurísticas relatadas na literatura pretendiam a solução de problemas específicos e não eram, via de regra, passíveis de serem utilizadas em outros problemas. As heurísticas clássicas de roteamento são um bom exemplo disso.

Podemos também considerar que os primeiros métodos gerais clássicos estavam associados a estratégias de enumeração incompleta, solução parcial ou relaxações. Dentre as muitas classificações possíveis para as heurísticas, sugerimos a constante da Figura 5.23.

Utilizando-se um diagrama semelhante ao sugerido no item 5.2, poderemos examinar o comportamento típico de uma busca heurística míope. A Figura 5.24 representa, dentro dos círculos, o valor final da solução obtida no estágio i da busca. Nesse caso, podemos visualizar o esforço de busca na medida em que as possíveis combinações entre as variáveis n e $n - 1$ são exploradas. No gráfico é facilmente perceptível que, na medida do crescimento do número de opções, representado pelo comprimento das colunas do diagrama, o esforço cresce exponencialmente. No caso da Figura 5.24, apenas a vizinhança imediata de uma solução (ou configuração) é explorada (a de maior valor no estágio i da busca). Entendemos por vizinhança de uma configuração as configurações que diferem da configuração base pelo valor atribuído a uma variável.

É claro que podemos melhorar o exame das soluções vizinhas e reduzir o grau de miopia da tomada de decisão. A Figura 5.24 mostra essa possibilidade. Nesse caso a exploração estende-se agora para a vizinhança de duas configurações (as duas melhores segundo o critério do valor de $f(s)$). A penalidade que devemos pagar pela remoção da miopia é o crescimento do número de configurações admitidas como candidatas ao exame. Obviamente se examinássemos todas as configurações encontraríamos a solução ótima, mas é exatamente esse enorme trabalho que desejamos evitar. Existem dois procedimentos que permitem a gerência do risco de não se encontrar a solução ótima:

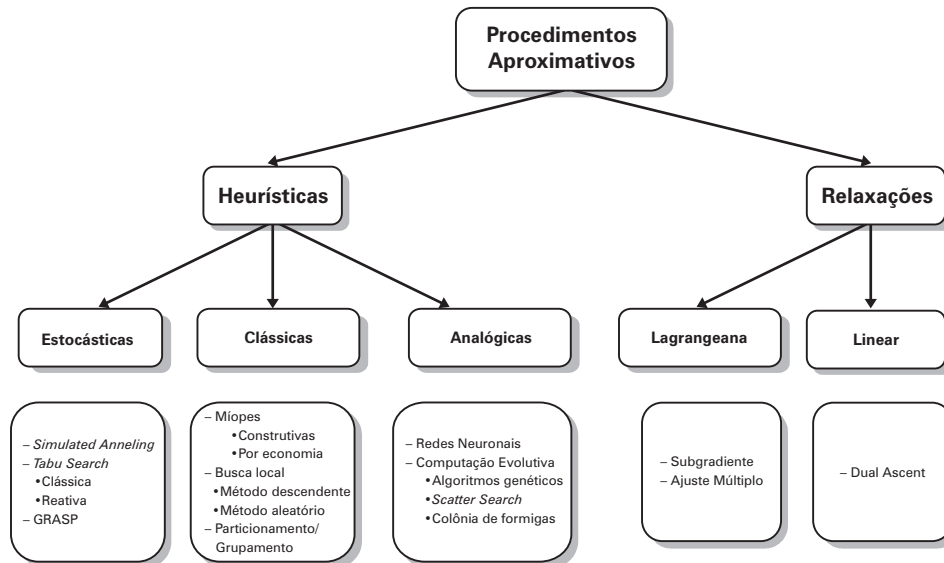


FIGURA 5.23 Classificação das abordagens aproximativas.

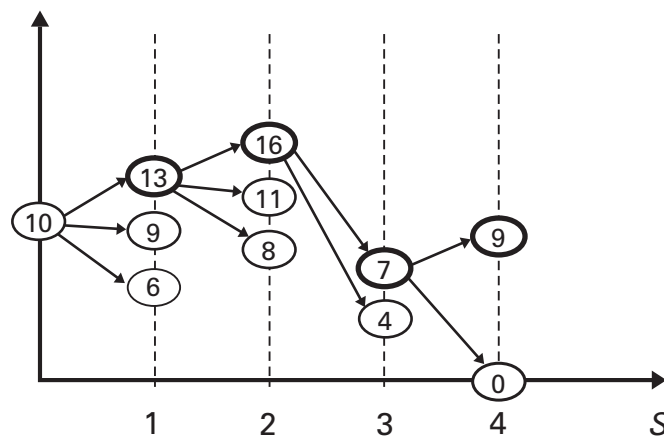


FIGURA 5.24 Tomada de decisão míope.

- Buscar examinar um número crescente de combinações entre as variáveis selecionadas em um determinado estágio da decisão e as variáveis dos estágios seguintes (intensificação ou melhoria na qualidade da solução).
- Buscar considerar um número cada vez maior de variáveis em cada nível (diversificação ou aumento do alcance [diâmetro] da busca).

Podemos perceber visualmente como a inclusão da segunda configuração parcial mais valorada em cada nível (diversificação) aumenta com o número de combinações para a definição da configuração S (intensificação) representada pelas setas. O aumento do esforço de busca traz seus resultados: as decisões 1 e 3 da Figura 5.25 não podem ser mais consideradas apenas gulosas. Se o procedimento adotado for suficientemente inteligente podemos encontrar o equilíbrio entre a diversificação e a intensificação, caminhando economicamente em relação ao esforço de busca, através dos máximos (ou mínimos) locais até o máximo (ou mínimo) global, como mostra a Figura 5.26. Nesse caso a inteligência foi expressa pela estratégia de explorar as três soluções mais interessantes em cada estágio. Infelizmente, não podemos deduzir *a priori* qual seria essa estratégia inteligente, uma vez que ela depende da instância de cada problema.

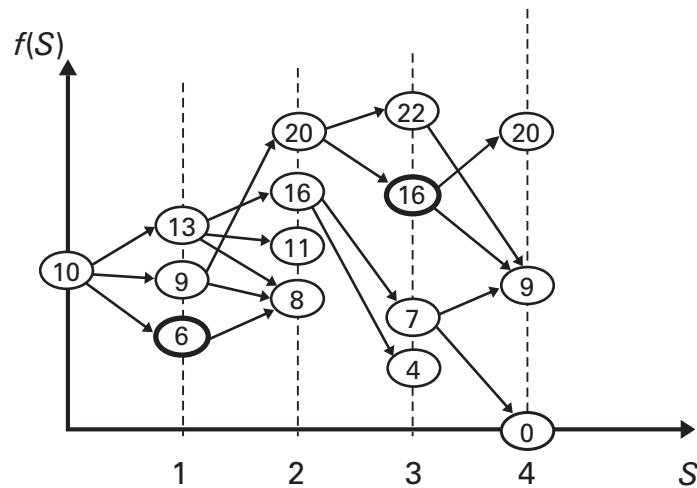


FIGURA 5.25 Incrementando a decisão míope.

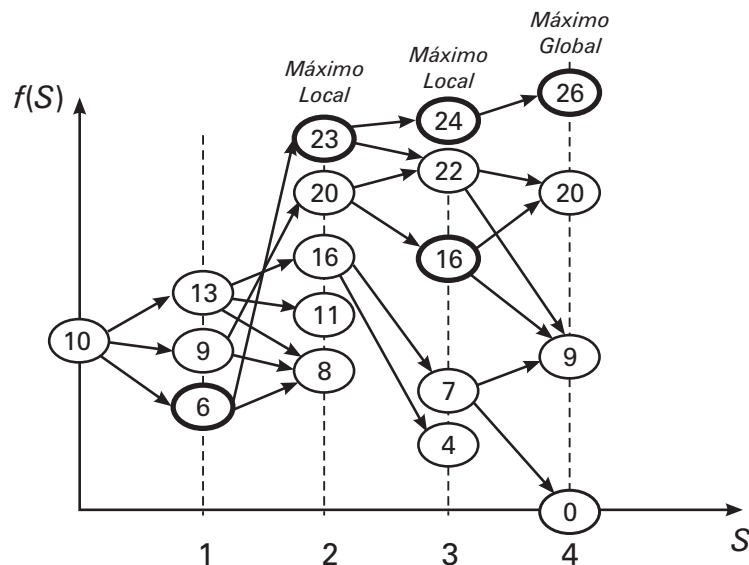


FIGURA 5.26 Esforço ótimo para a busca heurística.

A Figura 5.23 procura resumir as várias abordagens conhecidas para os procedimentos heurísticos de solução. Na linha clássica é bastante comum, como anteriormente ressaltamos, que as heurísticas explorem casuisticamente a estrutura do problema, sem que, normalmente, se possa definir claramente uma estratégia universal de solução. No caso das chamadas meta-heurísticas, que envolvem as chamadas heurísticas estocásticas e analógicas, existe invariavelmente uma estratégia geral de solução, cabendo apenas adaptá-la ao caso específico. As heurísticas modernas têm despertado crescente interesse da comunidade científica, tanto pelo bom retorno dado ao compromisso do objetivo, como em virtude da qualidade das soluções encontradas, tornando-se uma alternativa cada vez mais interessante para a solução de grande parte das aplicações reais dos modelos combinatórios. Dentre os mais recentes e abrangentes trabalhos na área destacamos Aarts e Lenstra (1996), Osman e Kelly (1996), Osman e Lapport (1996), Rayward-Smith *et al.* (1996). Dentre as heurísticas modernas destacam-se:

✓ Algoritmos Genéticos

Os algoritmos genéticos são detalhados no Capítulo 8, todavia a Tabela 5.7 relaciona uma série de recentes trabalhos aplicados aos problemas de otimização combinatória.

TABELA 5.7 APLICAÇÕES DE ALGORITMOS GENÉTICOS EM OTIMIZAÇÃO COMBINATÓRIA

Ano	Autores	Abordando
1985	Grefenstette	AG aplicados à solução do Caixeiro Viajante
1990	Karr e Goldberg	AGs aplicados à separação de minério
1991	Davidor	AG aplicados à robótica
1996	Arslan <i>et al.</i>	AGs na otimização de circuitos VSLI
1996	Notredame e Higgins	AGs na solução do alinhamento genético
1996	Jakobs	AGs na solução de acomodação de polígonos
1996	Anderson e Gerbert	AGs no projeto de asas de aviões
1997	Mavridou e Pardalos	AG e Simulated Annealing – problema de layout
1997	Zhou e Gen	AGs na solução de árvore geradora k-restrita
1997	Ingleston e Smith	AGs no auxílio ao combate aéreo
1998	Kim <i>et al.</i>	AGs no equilíbrio de linhas de montagem
1998	Pierrot e Hinterding	AGs na solução de programação inteira mista
1999	Azadivar e Tompkins	AGs na simulação de processo de otimização
2000	Dellaert <i>et al.</i>	AGs na otimização de lotes de produção
2001	Preston e Kozan	AGs na distribuição de containers em portos
2001	Vico <i>et al.</i>	AGs no planejamento de cirurgia
2002	Gantovnik <i>et al.</i>	AGs na otimização de laminados
2002	Ray e Mahajan	AGs na configuração ótima de sensores
2002	Lambert	AGs na desmontagem de equipamento eletrônico
2002	Sato <i>et al.</i>	AGs no projeto de arquitetura
2002	Carter e Ragsdale	AGs no planejamento de jornais
2002	Aytug e Saydam	AGs na otimização de problema de localização
2003	Hageman <i>et al.</i>	AGs e tabu na otimização de pintura
2003	Zhou <i>et al.</i>	AGs no planejamento de alocação bicritério
2003	Hassini e Vickson	AGs na otimização de alocação carrocel
2003	Valdés <i>et al.</i>	AGs na otimização térmica de turbinas de gás
2003	Topcuoglu <i>et al.</i>	AGs na localização de roteadores de redes
2003	Park <i>et al.</i>	AGs na solução do planejamento de tarefas
2003	Ho e Ji	AGs na otimização de chips eletrônicos
2004	Lim <i>et al.</i>	AGs na otimização da mistura para concreto
2004	Liew	AGs na otimização de problemas geométricos
2004	Chung e Jung	AGs na otimização de acessórios de segurança
2004	Kasprzyk e Jaskua	AGs híbridos (Simplex) para análise química

TABELA 5.7 CONTINUAÇÃO

Ano	Autores	Abordando
2004	Mohamed Jenkins	AGs na configuração de membranas
2004	Fühner e Jung	AGs na otimização do crescimento cristalino
2004	Kulkarni <i>et al.</i>	AGs na otimização microestrutural de ligas
2004	Ozturk <i>et al.</i>	AGs na estimação da “exergy” de residências
2004	Cordoba <i>et al.</i>	AGs na otimização de sistemas contra fogo
2004	Aydin e Yildirim	AGs no controle de um sistema laser
2004	Ge e Chen	AGs no controle de espaçonaves
2004	Shin <i>et al.</i>	AGs na reconfiguração de redes de energia
2004	Shokooh-Saremi <i>et al.</i>	AGs na otimização de polarizadores

Dentre os mais recentes e abrangentes trabalhos na área destacam-se Goldberg (1989), Davis (1991), Whitley (1994), Aarts e Lenstra (1996), Osman e Kelly (1996), Osman e Lapport (1996), Rayward-Smith *et al.* (1996), Reeves (1996), Bäck *et al.* (1997), Fogel (1998), Vose (1998), Mitchell (1999), Coley (1999), Spears (2000), Spears *et al.* (2000), Mühlenbein e Mahnig (2001), Lance (2001), Blum e Roli (2003). A Tabela 5.8 apresenta trabalhos que introduzem temas inovadores para a abordagem.

TABELA 5.8 ALGORITMOS GENÉTICOS

Ano	Autores	Abordando
1986	Grefenstette	Otimização de parâmetros de controle AG
1987	Davis	Algoritmos Genéticos e Simulated Annealing
1990	De Jong	AGs aplicados à aprendizagem
1993	Juliff	Técnica de cromossomos múltiplos
1996	Wineberg e Oppacher	AGs com introns
1997	Ronald <i>et al.</i>	Técnica de cromossomos múltiplos
1997	Hinterding	Autoadaptação e cromossomos múltiplos
2000	Cantu-Paz	AGs paralelos
2001	Beyer	Teorias para estratégias de evolução
2001	Jansen e Wegener	Utilidade da população dos AGs
2002	Beyer <i>et al.</i>	Metodologia de análise para AGs
2003	Aickelin e Bull	AGs Coevolucionários
2003	Ghosh <i>et al.</i>	Estudo da influência das gerações na evolução

A Tabela 5.9 apresenta trabalhos em aplicações originais e que guardam correlação com a otimização combinatória

TABELA 5.9 ALGORITMOS GENÉTICOS APLICADOS A PROBLEMAS ESPECIAIS

Ano	Autores	Abordando
1991	Caldwell e Johnston	AGs na identificação de rostos humanos
1991	Gibson e Byrne	AG aplicados à composição musical

TABELA 5.9 CONTINUAÇÃO

Ano	Autores	Abordando
1996	Hanebeck e Schmidt	AGs no treinamento de redes neurais
1998	Nayebzadeh <i>et al.</i>	AGs na determinação de arritmia fetal
1998	Reid	AGs na construção de Redes de Petri
2003	Ng e Perera,	AGs no controle de qualidade da água
2004	Zhang <i>et al.</i>	AGs na estimação da transferência de calor
2004	Ozturk <i>et al.</i>	AGs na estimação da "exergy" de residências
2004	Roychowdhury	AGs no diagnóstico de doenças
2004	Sarkar e Modak	AGs na otimização da operação de biorreatores
2004	Kim <i>et al.</i>	AGs na estimativa de emissão de radiação
2004	Mwembeshi <i>et al.</i>	AGs no controle de pH de reatores
2004	Imai <i>et al.</i>	AGs na operação de terminais portuários

✓ Genéticos Híbridos

Os algoritmos genéticos são promissora fonte de hibridização – mistura de técnicas aproximativas. Os trabalhos que seguem relacionados na Tabela 5.10 exploram tal possibilidade.

TABELA 5.10

Ano	Autores	Abordando
1990	Ulder <i>et al.</i>	Genético com busca local
1992	Schaffer <i>et al.</i>	Genéticos e redes neurais
1994	Fleurent e Ferland	Genético híbrido p/ quadrático de alocação
1996	Kim e Lee	Genético com "reinforcement learning"
1997	Tsutsui e Ghosh	Genéticos com busca local
1998	Cotta <i>et al.</i>	Genético híbrido para a mochila 0-1
2000	He e Mort	Híbrido na solução de redes de comunicação
2000	Hiroyasu <i>et al.</i>	Genético com simulated <i>annealing</i>
2000	Matsumura <i>et al.</i>	Genético com busca tabu
2001	Luo <i>et al.</i>	Genético com programação linear
2001	Savchenko e Schmitt	Genético com simulated <i>annealing</i>
2001	French <i>et al.</i>	Genético e B&B
2002	Lim e Yuan	Genético híbrido aplicado ao quadrático de alocação
2003	Ting <i>et al.</i>	Genético com busca tabu

Links Úteis:

<http://garage.cps.msu.edu/papers/papers-index.html>

<http://www-illigal.ge.uiuc.edu/index.php3>

<http://www.shef.ac.uk/~gaipp/links/>

✓ **Algoritmos Meméticos (Genéticos com Busca local, Genéticos Lamarckianos, Genéticos Baldwinianos).**

Algoritmos Meméticos utilizam o conceito de “evolução cultural”. Nesse paradigma a adaptabilidade de um indivíduo pode ser modificada por ações não programadas em seu repositório genético. Para simular o efeito da ação “cultural” os algoritmos meméticos se permitem empregar principalmente técnicas de busca local para aperfeiçoar o cromossomo. A alteração dita memética do genoma ocorre fora do clássico mecanismo de reprodução/mutação dos algoritmos genéticos. As melhorias dos cromossomos devidas à busca local são preservadas, caracterizando-se o que se convencionou denominar de evolução “Lamarckiana” – evolução que herda os sucessos obtidos pelos indivíduos. Sob vários pontos de vista os algoritmos meméticos podem ser considerados uma forma híbrida de algoritmos genéticos.

A Tabela 5.11 resume alguns trabalhos de aplicação em Otimização Combinatória em que a proposta da abordagem memética/lamarckiana é empregada.

TABELA 5.11 APLICAÇÕES DE ALGORITMOS MEMÉTICOS NA OTIMIZAÇÃO COMBINATÓRIA

Ano	Autores	Abordando
1990	Ulder <i>et al.</i>	Aplicado à solução do PCV
1992	Moscato e Norman	Aplicado à solução do PCV
1997	Cotta e Troya	Aplicação ao problema da mochila
1997	Dengiz <i>et al.</i>	Aplicado a confiabilidade em redes
1998	Charon e Hudry	Na solução de implicações lógicas
1998	Morris <i>et al.</i>	Modelagem molecular
1999	Burke e Smith	Aplicado ao problema de plano de manutenção
1999	Tesfaldet e Hermosilla	Aplicado à solução do PCV
2000	Knowles e Corne	Otimização multiobjetivo
2001	Burke <i>et al.</i>	Planejamento do trabalho de enfermagem
2002	Lim e Yuan	Solução do quadrático de alocação
2002	Kersting <i>et al.</i>	Solução do problema de conectividade
2002	Velenzuela e Smith	Aplicação ao setor de distribuição de energia
2003	Quintero e Pierre	Designação de células em redes móveis
2003	Ljubic e Raidl	Algoritmo em grafos
2004	Lima <i>et al.</i>	Roteamento de veículos com frota heterogenia
2004	Krasnogor e Gustafson,	Experimentação de “self-generation”
2004	Berreta e Rodrigues	Otimização de lotes de produção

A Tabela 5.12 apresenta trabalhos que desenvolvem os fundamentos da abordagem.

TABELA 5.12 TRABALHOS ASSOCIADOS AOS FUNDAMENTOS DA ABORDAGEM

Ano	Autores	Abordando
1987	Hinton e Nowlan	Evolução guiada
1988	Muhlenbein <i>et al.</i>	Conceitos em evolução artificial
1989	Moscato	Fundamentos da metáfora

TABELA 5.12 CONTINUAÇÃO

Ano	Autores	Abordando
1989	Jog <i>et al.</i>	Melhoria local
1994	Merz e Freisleben	Descrição formal de um algoritmo memético
1994	Radcliffe e Surry	Descrição formal de um algoritmo memético
1998	Ku e Mak	Estudo do efeito Baldwin
1999	Merz e Freisleben,	Estudo algorítmico
1999	Moscato	Fundamentos da metáfora
2000	Merz	Aprofundamento no tema

✓ Simulated Annealing

A metáfora é apoiada na distribuição de Boltzman e pretende simular um processo de resfriamento típico dos metais. A distribuição de Boltzman especifica a probabilidade de se alcançar um nível de energia $E=f(s)$, onde s é um estado do sistema, para uma dada temperatura t . A Tabela 5.13 resume alguns trabalhos no tema e aplicados a problemas de Otimização Combinatória.

TABELA 5.13 APLICAÇÕES DE SIMULATED ANNEALING À OTIMIZAÇÃO COMBINATÓRIA

Ano	Autores	Abordando
1953	Metropolis <i>et al.</i>	Trabalho pioneiro
1982	Kirkpatrick <i>et al.</i>	Trabalho pioneiro
1985	Cerny <i>et al.</i>	Trabalho pioneiro
1984	Bonomi e Lutton	Aplicado na solução do PCV
1986	Lundy e Mess	Estudo de convergência
1986	Bohachevsky <i>et al.</i>	Aplicado a otimização de funções
1986	Mitra <i>et al.</i>	Estudo de convergência
1987	Laarhoven e Aarts	Várias aplicações
1987	Wilhelm <i>et al.</i>	Aplicado ao quadrático de alocação
1988	Goldstein e Waterman	Estudo do tamanho da vizinhança
1989	Ingber	Técnicas de melhoria de desempenho
1989	Aarts e Krost	Várias aplicações
1989	Johnson <i>et al.</i>	Estudo algorítmico em várias aplicações
1989	Osman e Potts	Aplicação ao flow shop de permutação
1990	Dueck e Scheuer	Utilização de “theershold accepting”
1990	Eglese	Várias aplicações
1991	Cheh <i>et al.</i>	Vizinhanças eficientes
1991	Yao	Vizinhanças estendidas
1993	Downsland	Estudo algorítmico
1993	Osman	Aplicação ao roteamento de veículos
1994	Moccelin	Solução do “Shop Sequencing”

TABELA 5.13 CONTINUAÇÃO

Ano	Autores	Abordando
1996	Bölte e Thonemann	Hibridização com programação genética
1996	Andricioaei <i>et al.</i>	Generalização da abordagem
1996	Desai e Patil	Simulated annealing e busca local
1996	Ingber	Simulated annealing adaptativo
1997	Mavridou e Pardalos	Simulated annealing e algoritmos genéticos
1998	Chiang e Russel	Aplicação ao roteamento de veículos
1998	Tracey <i>et al.</i>	Teste de software
1999	Brooke <i>et al.</i>	Quantum Annealing
2000	Kliewer e Tschöke	Aplicações na indústria aérea
2000	Locatelli	Condições de convergência
2000	Hamam e Hindi	Distribuição de módulos de programação
2000	Hiroyasu <i>et al.</i>	Simulated annealing em arquitetura paralela
2001	Hindi e Mhlanga	Solução de problema de "Scheduling"
2002	Yarkhan e Dongarra	Solução de problema de "Scheduling"
2003	Serrurier <i>et al.</i>	Solução de programação linear inteira
2003	Schwarzfischer	Solução de problema de "Scheduling"
2003	Bouleimen e Lecocq	Solução de problema com restrição de recursos
2003	Gaetan e Yao	Multi-implementação
2003	Wrighton e DeHon	Solução assistida
2004	Neumann e Muncill	Estimação de propriedades oceânicas

Links Úteis:

<http://www.taygeta.com/annealing/simanneal.html>

✓ Nuvens de Partículas (Particles Swarm) e Colônia de Formigas

Nuvens de Partículas é uma técnica estocástica de otimização baseada em populações desenvolvida por Kennedy e Eberhart (1995) inspirado no comportamento social de pássaros e cardumes de peixes. Colônias de formigas é outra técnica estocástica que segue na mesma linha de biomimética social para o caso do comportamento de formigas e cupins, introduzida por Dorigo *et al.* (1991), Dorigo (1992). A Tabela 5.14 resume alguns trabalhos no tema e aplicados a problemas de Otimização Combinatória.

TABELA 5.14 APLICAÇÕES DE BIOMIMÉTICA SOCIAL

Ano	Autores	Abordando
1992a	Colorni <i>et al.</i>	Fundamentos de colônia de formigas
1992b	Colorni <i>et al.</i>	Propriedades de colônia de formigas
1994	Maniezzo <i>et al.</i>	Colônia de formigas para quadrático de alocação
1994	Artur	Colônia de formigas na solução do PCV
1996	Dorigo <i>et al.</i>	Fundamentos de colônia de formigas

TABELA 5.14 CONTINUAÇÃO

Ano	Autores	Abordando
1997	Costa e Hertz	Colônia de formigas na coloração de grafos
1997	Kennedy e Eberhart	Fundamentos de nuvens de partículas
1998	Angeline	Nuvens de partículas com seleção
1999	Gambardella <i>et al.</i>	Colônia de formigas p/ o quadrático de alocação
1999	Taillard	Fundamentos de colônia de formigas
1999	Clerc	Nuvens de partículas determinística e adaptativa
1999	Dorigo e Di Caro	Vários exemplos
1999	Bullnheimer <i>et al.</i>	Estudo computacional
1999	Yoshida <i>et al.</i>	Nuvens de partículas no controle de voltagem
2000	Dorigo <i>et al.</i>	Conceitos em colônia de formigas
2000	Stützle e Hoos	Colônia de formigas p/ caso Max-Min
2001	Eberhart e Shi	Aplicações para nuvens de partículas
2002	Cordon <i>et al.</i>	Revisão da abordagem em colônia de formigas
2002	Ray e Liew	Nuvens de partículas p/ multiobjetivo
2002a	Parsopoulos e Vrahatis	Revisão da abordagem nuvens de partículas
2002b	Parsopoulos e Vrahatis	Nuvens de partículas p/ multiobjetivo
2002	Ciuprina <i>et al.</i>	Nuvens de partículas p/ eletromagnética
2002	Salman <i>et al.</i>	Nuvens de partículas p/ alocação de tarefas
2002	Laskari <i>et al.</i>	Nuvens de partículas p/ programação inteira
2002	Middendorf <i>et al.</i>	Multi colônia de formigas
2003	Abd-El-Barr	Colônia de formigas p/ circuitos digitais
2003	Bonabeau <i>et al.</i>	Conceitos em nuvens de partículas
2003	Coello <i>et al.</i>	Colônia de formigas p/ circuitos lógicos
2003	Carbonato e Maniezzo	Conceitos em colônia de formigas
2003	Ting <i>et al.</i>	Nuvem de partículas p/ geração de energia
2004	Maniezzo <i>et al.</i>	Texto didático
2004	Onwubolu e Clerc	Nuvem de partículas p/ perfuração automática
2004	Chang <i>et al.</i>	Nuvem de partículas p/ bioquímica
2004	Coello <i>et al.</i>	Nuvem de partículas p/ multiobjetivo

Links Úteis:

<http://www.particleswarm.net/>
<http://www.swarm-bots.org/>

✓ Estratégias Multistart e GRASP (Greedy Randomized Adaptive Search Procedure)

As técnicas Multistart são compostas de procedimentos que, a partir de uma determinada solução para o problema, encontram uma nova solução viável. Então, a vizinhança da nova solução gerada é examinada por um procedimento de busca local de modo a intensificar o resultado. O GRASP é uma variante

Multistart na qual as soluções viáveis são encontradas através de um procedimento quase guloso – estocástico com viés guloso. A Tabela 5.15 relaciona trabalhos que, ou desenvolvem a metáfora, ou representam aplicações à solução de problemas de otimização combinatória.

TABELA 5.15 APLICAÇÕES DO GRASP

Ano	Autores	Abordando
1989	Feo e Resende	Trabalho pioneiro
1992	Betrò e Schoen	Estudo sobre regras de parada p/ multistart
1994	Boese <i>et al.</i>	Multistar adaptativo
1994	Li <i>et al.</i>	GRASP p/ quadrático de alocação
1995	Battiti e Protasi	Multistar reativo
1997	Fleurent e Glover	Multistar construtivo p/ quadrático de alocação
1997	Hickernell e Yuan	Multistar p/ otimização global
1998	Atkinson	GRASP p/ programação de veículos
1998	Hart	Estudo das regras de parada p/ multistart
1998	Deshpande e Triantaphyllou	GRASP p/ problemas de inferência
1998	Resende	GRASP p/ o problema de cobertura máxima
1999	Fleurent e Glover	Multistart construtivo p/ quadrático de alocação
1999	Fernandez	GRASP p/ traçado aerofotogramétrico
1999	Pardalos <i>et al.</i>	GRASP p/ “Feedback Vertex Set Problem”
2000	Melián-Batista <i>et al.</i>	Multistar de aplicação genérica
2001	Robertson	GRASP p/ alocação multidimensional
2001	Brunato e Battiti	GRASP p/ problema de controle de tráfego
2001	Hammer e Rader	GRASP p/ problema de recobrimento
2001	Jagota e Sanchis	GRASP p/ problema da clique máxima
2002	Schoen	Visão geral dos métodos multistart
2002	Corberán <i>et al.</i>	GRASP p/ carteiro chinês em grafo misto
2003	Martí	Visão geral dos métodos multistart
2003	Chaovalitwongse <i>et al.</i>	GRASP – roteamento de veículos c/ janela tempo
2003	Osman <i>et al.</i> (2003)	GRASP p/ problema de planaridade em grafos
2003	Ahuja <i>et al.</i>	GRASP p/ árvore geradora mínima capacitada

✓ *Busca Tabu*

É uma meta-heurística baseada em procedimentos de busca local enriquecidos por estratégias de manipulação de memória de modo a evitar que configurações já examinadas sejam reexaminadas durante o desenvolvimento do algoritmo. A Tabela 5.16 relaciona trabalhos que, ou desenvolvem a metáfora, ou representam aplicações à solução de problemas de otimização combinatória.

TABELA 5.16 APLICAÇÕES DE BUSCA TABU

Ano	Autores	Abordando
1968	Glover	Trabalho pioneiro
1987	Glover	Trabalho pioneiro
1989	Glover	Texto didático
1989	De Werra e Hertz	Texto didático
1990a	Glover	Várias aplicações
1990b	Glover	Várias aplicações
1990	Hansen e Jaumard	Tabu na solução do problema da satisfabilidade
1991	Glover e Laguna	Tabu na solução do problema de empacotamento
1991	Taillard	Tabu p/ quadrático de alocação
1992	Faigle e Kern	Estudo de convergência
1992	Cook e Han	Tabu p/ o problema de empacotamento múltiplo
1993	Glover <i>et al.</i>	Aprofundamento da abordagem
1994b	Glover	Tabu p/ programação não linear
1994	Battiti e Tecchiolli	Tabu reativa
1995	Laguna <i>et al.</i>	Tabu para o problema de alocação generalizado
1995	Battiti e Çela	Busca tabu reativa
1996	Zachariasen e Dam	Tabu p/ PCV geométrico
1996	Kolohan e Liang	Tabu p/ solução de operações de perfuração
1997	Cordeau <i>et al.</i>	Tabu p/ roteamento periódico de veículos
1997	Sharaiha <i>et al.</i>	Tabu p/ solução da árvore geradora capacitada
1997	Consiglio e Zenios	Tabu p/ modelos de licitação
1998	Thesen	Tabu em aproximação via caminhos mais curto
1998	Hao <i>et al.</i>	Tabu p/ alocação de frequência
1998	Woodruff	Tabu com informações selecionadas
1998	Løkketangen e Glover	Tabu p/ programação mista
1999	Gendreau <i>et al.</i>	Tabu p/ árvore de Steiner
1999	Nowicki	Tabu p/ <i>flow shop</i> de permutação
2000	Yin	Tabu p/ aproximação de curvas digitais
2000	Cullenbine	Tabu p/ alocação arma × alvo
2000	Berger <i>et al.</i>	Tabu p/ problema em redes
2000	Morley e Grover	Tabu p/ problema em redes
2002	Crainic e Gendreau	Tabu p/ problema de redes capacitadas
2002	Ware <i>et al.</i>	Tabu p/ geração automática de mapas
2002	Zhang e Sun	Tabu p/ seleção de características
2002	Yamamoto <i>et al.</i>	Tabu p/ cartografia
2002	Klau <i>et al.</i>	Busca tabu guiada por operador humano

TABELA 5.16 CONTINUAÇÃO

Ano	Autores	Abordando
2002	Sait e Zahra	Tabu p/ otimização de circuitos
2003	Kulturel-Konak <i>et al.</i>	Tabu p/ solucionar redundância de alocação
2003	Blum e Roli	Visão geral
2003	Greisthofer	Tabu e Scatter Search
2004	Riaz <i>et al.</i>	Tabu p/ problema de alinhamento

✓ *Busca em Vizinhança Variável (BVV) e de Decomposição em Vizinhança Variável (DVV)*

As presentes meta-heurísticas objetivam o exame sistemático de estruturas de vizinhança. A Tabela 5.17 relaciona trabalhos que, ou desenvolvem a metáfora, ou representam aplicações à solução de problemas de otimização combinatória.

TABELA 5.17 APLICAÇÕES DA BUSCA EM VIZINHANÇA VARIÁVEL

Ano	Autores	Abordando
1996	Vahrenkamp	Busca aleatória para o problema de cortes
1997a	Hansen e Mladenovic	Trabalho pioneiro
1997b	Hansen e Mladenovic	Trabalho conceitual
1997c	Hansen e Mladenovic	BVV p/ PCV com vizinhança 2-opt
1999	Caporossi e Hansen	BVV p/ grafos em modelo de energia
1999	Hansen e Mladenovi	Trabalho didático
2001	González e Pérez-Brito	BVV p/ o problema de ordenação linear
2001	Hansen e Mladenovi	Aplicações industriais da BVV
2001	Hansen <i>et al.</i>	Proposta da DVV
2002	Lalonde e Gagnon	BVV p/ padrões geométricos deformáveis
2002	Costa <i>et al.</i>	BVV p/ o problema de localização de cabos
2002	García-López <i>et al.</i>	BVV p/ o problema das p-medianas
2003	Avanthay <i>et al.</i>	BVV p/ problema de coloração de vértices
2003	Bräysy	BVV reativa p/ roteamento de veículos
2003	Burke <i>et al.</i>	BVV p/ programação de tarefas de enfermagem
2003	Hansen e Mladenovi	Trabalho abrangente
2004	Fleszar e Hindi	BVV p/ problema da classe “scheduling”

✓ *Scatter Search e Path Relinking*

As meta-heurísticas Scatter Search e Path Relinking são inspiradas em um mesmo objetivo: com base em boas soluções conhecidas encontrar novas e melhores soluções. A Scatter Search propõe fazer uma “mistura” de boas soluções utilizando a abordagem dos algoritmos genéticos. Assim a Scatter Search sugere que mais de dois cromossomos sejam considerados como doadores de material genético (pais) para formar um cromossomo filho. No caso do Path Relinking a técnica indica que, para um dado par de soluções, seja encontrada uma sequência de soluções alteradas – soluções intermediárias ou mistas – que representem um caminho de transformação da solução inicial na solução alvo. A Tabela 5.18 rela-

ciona trabalhos que, ou desenvolvem a metáfora, ou representam aplicações à solução de problemas de otimização combinatória.

TABELA 5.18 APLICAÇÕES DE SCATTER SEARCH E PATH RELINKING

Ano	Autores	Abordando
1994a	Glover	Trabalho pioneiro em scatter search
1997	Glover	Abordagem geral Scatter search e path relinking
1996	Kelly <i>et al.</i>	Scatter search p/ treinamento de redes neurais
1996	Cung <i>et al.</i>	Scatter search p/ quadrático de alocação
1996	Fleurent <i>et al.</i>	Scatter search p/ otimização irrestrita
1998	Reeves	Algoritmo genético c/ path relinking
2000	Laguna e Martí	GRASP e path relinking
2000	Glover <i>et al.</i>	Fundamentos de scatter search e path relinking
2001	Beausoleil	Scatter search multicritério
2001	Campos <i>et al.</i>	Scatter search p/ o problema de ordenação linear
2003	Laguna e Martí	Texto didático p/ scatter search
2003	Glover <i>et al.</i>	Texto didático p/ scatter search
2003	Greistofe	Tabu e scatter search p/ roteamento em arcos
2003	Oliveira <i>et al.</i>	GRASP e path relinking p/ quadrático de alocação
2004	Dell'Amico <i>et al.</i>	Scatter search p/ P Cmax

✓ Algoritmos Culturais

Os algoritmos culturais (ACs) são baseados na noção de que em sociedades avançadas o desenvolvimento do indivíduo pode ser realizado tanto por seleção natural como regras de seleção cultural. Cultura, no sentido dos algoritmos culturais, é vista como o conjunto de informações acumuladas pela experiência da sociedade de indivíduos. Os algoritmos culturais propõem utilizar o domínio do conhecimento contido na “cultura” para guiar sua busca estocástica. Os ACs trabalham sobre dois espaços de decisão a saber: a população e o conjunto de crenças. A Tabela 5.19 relaciona trabalhos que, ou desenvolvem a metáfora, ou representam aplicações à solução de problemas de otimização combinatória.

TABELA 5.19 APLICAÇÕES DE ALGORITMOS CULTURAIS

Ano	Autores	Abordando
1994a	Reynolds	Trabalho pioneiro
1994b	Reynolds	Trabalho pioneiro
1995	Reynolds <i>et al.</i>	Aplicação à manipulação de restrições
2002	Coello e Becerra	Aplicação à otimização com restrições
2002	Xidong e Reynolds	Aplicação à mineração de dados
2004	Coello e Becerra	Técnica de mapeamento do espaço das crenças

✓ Algoritmos Transgenéticos

A Transgenética Computacional (TC) replica o paradigma da simbiogênese, em que criaturas de diferentes espécies trocam informações, eventualmente até material genético, de modo a facilitar a adaptação ao meio ambiente. O processo evolucionário da TC é realizado com base na troca de informações entre populações de indivíduos de diferente natureza. A população de cromossomos alvo constitui o primeiro nível de evolução, o nível que modela a solução do problema. Uma população de vetores de manipulação constitui o segundo nível. Finalmente um terceiro nível contém as regras que coordenam e alimentam a simbiogênese – a troca de informação entre as populações. Os vetores de manipulação são também denominados de vetores transgenéticos e podem transportar informações obtidas *a priori* ou durante a evolução. Observe que na TC os cromossomos não trocam material genético entre si – não existe o processo clássico de reprodução ou recombinação. A informação *a priori* pode ser originada de várias fontes, tais como conhecimento teórico e heurístico. A Tabela 5.20 relaciona trabalhos que, ou desenvolvem a metáfora, ou representam aplicações à solução de problemas de otimização combinatória.

TABELA 5.20 APLICAÇÕES DA TRANSGENÉTICA COMPUTACIONAL

Ano	Autores	Abordando
2001	Gouvêa	Trabalho pioneiro
2001	Goldbarg <i>et al.</i>	Aplicação ao problema de coloração de vértices
2001	Gouvêa e Goldbarg	Aplicação ao quadrático de alocação
2002	Goldbarg e Goldbarg	Aplicação ao quadrático de alocação
2002	Goldbarg <i>et al.</i>	Aplicação à exploração de petróleo
2002	Gouvêa <i>et al.</i>	Aplicação à programação de sondas de produção
2002	Goldbarg <i>et al.</i>	Aplicação à localização de poços de petróleo
2002	Castro <i>et al.</i>	Aplicação ao gás lift
2003	Goldbarg <i>et al.</i>	Aplicação ao PCV
2003	Ramos <i>et al.</i>	Aplicação ao PCV
2004a	Goldbarg <i>et al.</i>	Aplicação a sistemas de cogeração
2004b	Goldbarg <i>et al.</i>	Aplicação à exploração de petróleo
2004c	Goldbarg <i>et al.</i>	Aplicação ao flow shop de permutação
2004d	Goldbarg <i>et al.</i>	Aplicação à distribuição de gás natural
2004	Leite <i>et al.</i>	Aplicação à distribuição de vídeo

✓ O Paradigma Hiper-Heurístico

Tendo em vista a inexorável limitação de qualquer método heurístico isolado, em termos de solução eficiente dos problemas combinatórios NP-Árduos e outros, o paradigma hiper-heurístico propõe uma arquitetura que permita a seleção de algoritmos.

Basicamente o paradigma é uma arquitetura de tomada de decisão em que heurísticas são selecionadas e organizadas por uma camada superior composta por outras heurísticas. A abordagem prevê meta-heurísticas trabalhando em conjunto com outras meta-heurísticas, todavia em vez de se complementarem dentro de um processo de busca, como praticado tradicionalmente nas heurísticas híbridas, elas se orientam. Antes da denominação de hiper-heurística a proposta recebeu outros nomes como “combinação de heurísticas” (Hart e Ross [1998]), “direcionamento heurístico evolutivo” (Hart *et al.* [1998]), “estratégia de evolução de restrições de satisfação” (Terashima-Marin *et al.* [1999] ou “algoritmo evolucionário de múltiplos estágios” (Burke e Newall [1999]). A Figura 5.27 resume a estrutura geral da proposta do paradigma hiper-heurístico. A abordagem foi recentemente denominada de *hi-per-heurística* (Burke *et al.* [2003a] e [2003b]).

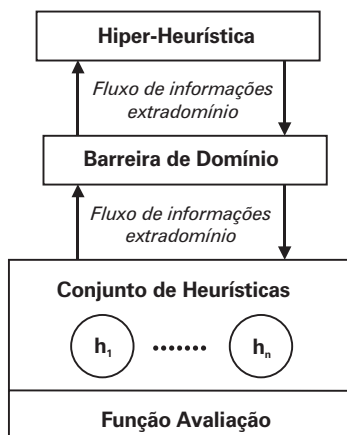


FIGURA 5.27 Arquitetura para Hiper-heurísticas [01].

A Figura 5.27 mostra que a arquitetura do paradigma separa o domínio do problema do domínio da hiper-heurística. A hiper-heurística não possui conhecimento sobre o domínio do problema, cabendo-lhe a tarefa de gerenciar as heurísticas do nível de solução. O processo de gerência se dá independentemente do problema tratado no nível das heurísticas (Cowling *et al* [2002]). As únicas informações que emergem no contexto da hiper-heurística são os dados que são comuns aos vários tipos de problemas e os dados que a hiper-heurística decide registrar como parte de seu estado interno. Por exemplo, uma hiper-heurística pode armazenar quanto tempo uma determinada heurística usou na sua última chamada, se houve mudança na função avaliação quando a heurística foi chamada ou quanto tempo de processamento se passou desde a última chamada de certa heurística. A hiper-heurística desconhece até mesmo qual problema está sendo solucionado por suas heurísticas escravas.

São relatados experimentos com a busca tabu (Burke *et al.* [2003a]) e algoritmos genéticos (Hart e Ross [1998]).

✓ Reinforcement Learning – Aprendizagem através de Reforço

Trata-se de um método estocástico e construtivo em que as melhores soluções são bonificadas. As bonificações são utilizadas para guiar a obtenção de novas soluções. Os fundamentos da abordagem estão descritos em Barto *et al.* (1981). A estratégia é bastante empregada em robótica (Stone *et al.* [2001]). A Tabela 5.21 resume as aplicações relatadas na solução de problemas de otimização combinatória.

TABELA 5.21 APLICAÇÕES DA APRENDIZAGEM ATRAVÉS DE REFORÇO

Ano	Autores	Abordando
1992	Barto	Conceitos
1995	Dietterich <i>et al.</i>	Aplicação ao “job-shop scheduling”
1996	Kaelbling <i>et al.</i>	Revisão da literatura
1997	Singh e Bertsekas	Alocação dinâmica de canais de telefone celular
1998	Moody <i>et al.</i>	Aplicação à carteira de investimento
1998	Sutton e Barto	Trabalho didático
1999a	Miagkikh e Punch	Solução p/ otimização combinatória
1999b	Miagkikh e Punch	Solução p/ otimização combinatória
2002	Gosavi <i>et al.</i>	Aplicação à alocação de assentos em avião
2002	Pontrandolfo <i>et al.</i>	Gestão de cadeia de suprimento
2004	Hong e Prabhu	Controle de produção e estoque

As relaxações são úteis tanto para a geração de soluções aproximadas, como especialmente para a disponibilização de limites que são utilizados em algoritmos exatos de solução do tipo B&B e outros. As relaxações mais utilizadas são a linear e a lagrangeana.

As heurísticas clássicas são desenvolvidas de forma específica para um certo tipo de modelo, explorando, via de regra, suas peculiaridades. Para cada problema de otimização, portanto, é possível elaborar estratégias aproximativas que levem em conta as particularidades do modelo. No presente trabalho abordaremos heurísticas clássicas principalmente para problemas de roteamento, recobrimento, localização e conexão. Com um exemplo complementar da utilização dessas técnicas em problemas de otimização, relacionaremos, a seguir, alguns trabalhos para problemas de corte, empacotamento e carregamento, problemas esses que, apesar de importantes, não serão abordados no texto:

✓ Heurísticas para o Problema de Corte, Empacotamento e Carregamento

Johnson *et al.* (1974), Chistofides e Whitlock (1977), Steudel (1979), George e Robinson (1980), Coffman *et al.* (1980), Sarin (1983), Wang (1983), Baker e Schwartz (1983), Chanzelle (1983), Roberts (1984), Marcotte (1986), Voight (1987), Dagli e Tatoglu (1987), Farley (1988), Tsai *et al.* (1988), Schneider (1988), Haessler (1988), Han *et al.* (1989), Daniels e Ghandforoush (1990), Yanasse *et al.* (1990), Haessler e Talbot (1990), Gehring *et al.* (1990), Coffman e Shor (1990), Yanasse *et al.* (1990), Sweeney e Haessler (1990), George (1992), Mohanty *et al.* (1994), Bischoff e Ratcliff (1995), Bischoff *et al.* (1995), Azar e Epstein (1997), Coffman *et al.* (1997), Hifi (1997), Fayard *et al.* (1998), de Werra (1997), Csirik *et al.* (1999), Vanderbeck (1999), Gradisar *et al.* (1999), Chu e Antonio (1999), Csirik *et al.* (2001), Hopper e Turton (2001), Liang *et al.* (2002), Alvarez-Valdés *et al.* (2002), Valério de Carvalho (2002), Zhang *et al.* (2002), Gradisar *et al.* (2002), Onwubolu e Mutingi (2003), Martello *et al.* (2003), Ragsdale e Zobel (2004), Shahin e Salem (2004).

Na atualidade, os procedimentos heurísticos são também muito empregados em conjunto com os métodos exatos, especialmente para produzirem limites e soluções viáveis de boa qualidade. Com o objetivo de exemplificar um algoritmo heurístico para a solução do problema da mochila apresentamos o denominado método míope ou guloso.

✓ Heurística Gulosa para a Solução do (PK)

Algoritmo Guloso para PK

INÍCIO

Ler vetor de pesos w_j , vetor de custos c_j e termo independente b .

Inicializar variáveis $z \leftarrow 0$, $x_j \leftarrow 0$, $j = 1, \dots, n$

Ordenar as variáveis de forma que: $\frac{c_j}{w_j} \geq \frac{c_{j+1}}{w_{j+1}}$, $j = 1, \dots, n-1$

Para $j = 1$ até n enquanto $b \neq 0$, fazer:

Início

$$\left[\begin{array}{l} x_j \leftarrow \left\lfloor \frac{b}{w_j} \right\rfloor \\ b \leftarrow b - (w_j \times x_j); \\ z \leftarrow z + c_j x_j \end{array} \right.$$

Fim

Escrever $\{z, x_j, j = 1, \dots, n\}$

FIM

Essa é uma heurística simples e de complexidade dominada pelo procedimento de ordenação, ou seja, $O(n \log_2 n)$. Sua razão de performance é $\frac{1}{2}$ (Fisher [1980]). A ideia do algoritmo é basicamente carregar a mochila com os itens de maior valor. Vamos aplicar o procedimento para encontrar uma solução para a instância da mochila representada pela Tabela 5.22, com o valor do termo independente igual a 14.

1. Leitura:

$$b = 14$$

TABELA 5.22 RELAÇÃO DE ITENS COM PESOS E VALORES

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
c	7	10	3	2	4	12	3	5	8	9	7	2	1	11	12	5
w	4	3	2	2	7	3	4	3	6	7	9	4	10	1	5	2

2. Ordenação:

c_j/w_j	7/4	10/3	3/2	1	4/7	4	3/4	5/3	8/6	9/7	7/9	2/4	1/10	1/11	12/5	5/2
-----------	-----	------	-----	---	-----	---	-----	-----	-----	-----	-----	-----	------	------	------	-----

j	6	2	16	15	1	8	9	3	10	4	11	7	5	12	13	14
c_j/w_j	4	10/3	5/2	12/5	7/4	5/3	8/6	3/2	9/7	1	7/9	3/4	4/7	2/4	1/10	1/11

Para $j = 1, \dots, n$

$$b = 14; x_6 = \lfloor 14/3 \rfloor = 4; z = 0 + 7 \times 4 = 28$$

$$b = 14 - 12 = 2; x_2 = \lfloor 2/3 \rfloor = 0; z = 28 + 0 = 28;$$

$$b = 2; x_{16} = \lfloor 2/2 \rfloor = 1; z = 28 + 1 \times 3 = 31.$$

5.5 – PROBLEMAS PROPOSTOS

1 – O Problema do Arredondamento de Soluções Inteiras



Seja o programa que se segue:

$$\text{Maximizar } z = 2x_1 + 3x_2$$

sujeito a:

$$x_1 \leq 4,2$$

$$x_2 \leq 5,8$$

$$19x_1 + 20x_2 \leq 156$$

$$x_1, x_2 \in \mathbb{Z}^+$$

que pode ser representado graficamente pela Figura 5.28.

Estude a adequação do uso da técnica de arredondamento da solução inteira para esse problema.

2 – O Problema do Socorro Aéreo



Após um período de intensas chuvas, uma vasta região da Amazônia está alagada muito além dos níveis normalmente admissíveis. Uma série de localidades estão completamente ilhadas no acesso por terra. As ligações por barco estão perigosas face ao extravazamento dos rios. O governo resolveu providenciar uma ponte aérea de suprimentos envolvendo Manaus e três outras cidades que atuarão como pólos de distribuição de remédios e gêneros de primeira necessidade e combustível. A Figura 5.29 mostra o mapa da operação. Para efetuar o transporte, o governo dispõe de basicamente dois tipos de aeronave: o helicóptero Valente e o avião de carga Sucuri. O helicóptero só pode operar com uma escala, pois só abastece em Manaus. O Sucuri possui maior autonomia e só opera economicamente com duas escalas. Os custos (em reais) para o quilômetro de voo estão resumidos na Tabela 5.23.

Sabendo-se que Altamira necessita de 10 ton de suprimento diário, Tabatinga 21 ton e Humaitá 7 ton, e que, em virtude do carregamento em Manaus, o suprimento máximo que poderá ser enviado por helicóptero não ultrapassa 15 ton. Formular o atendimento da demanda minimizando os custos globais da operação.

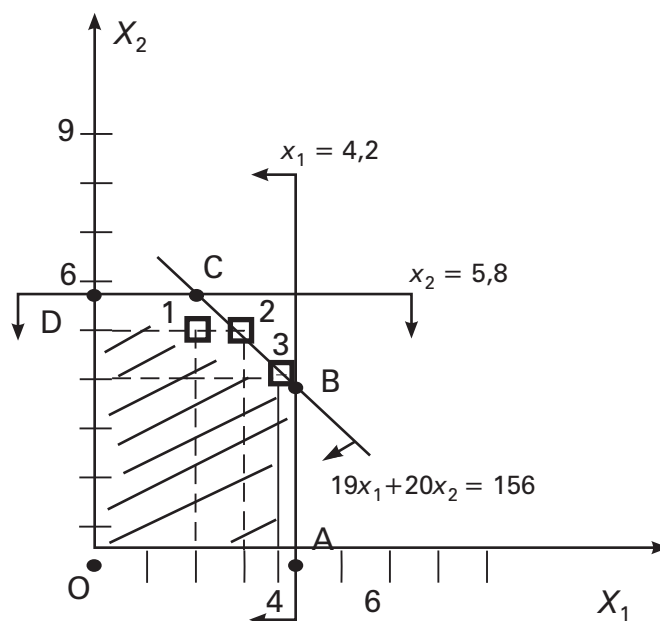


FIGURA 5.28 PPL Inteiro.

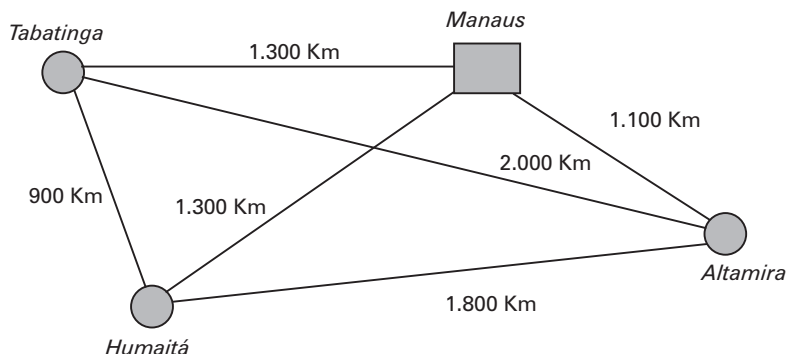


FIGURA 5.29 O Mapa da Operação.

TABELA 5.23

Trajeto	Valente		Sucuri	
	Vazio	Carregado	Vazio	Carregado
Manaus ⇔ Tabatinga	2, 5	3	2	4
Manaus ⇔ Humaitá	2	3	1, 5	3, 5
Manaus ⇔ Altamira	2	3, 5	2	3, 5
Altamira ⇔ Tabatinga	–	–	–	3
Altamira ⇔ Humaitá	–	–	–	4
Humaitá ⇔ Tabatinga	–	–	–	2, 5

3 – O Problema das Tarifas de Hospedagem



Um certo hotel com capacidade para 180 hóspedes distribuídos em 30 quartos simples, 30 duplos e 30 triplos pretende otimizar sua política de preços. O hotel trabalha com quatro tipos de clientes: excursões (e), convenções (c), empresas (s) e avulsos (a). Uma consultoria providenciou o levantamento das equações de demanda em função dos preços cobrados em relação aos clientes de excursões e de empresas da seguinte forma:

$$e_p \leq 100 - \left(\frac{p_S}{2} + \frac{p_P}{3} + \frac{p_T}{4} \right) \text{ e } s_p \leq 100 - \left(p_S + \frac{p_P}{4} + \frac{p_T}{2} \right)$$

Onde e_p é o número de pessoas que comporão uma excursão em função dos p_i ou preço dos quartos simples, duplos e triplos. Os preços podem variar (em unidades monetárias) dentro das seguintes faixas:

$$\begin{aligned} 20 &\leq p_s \leq 50 \\ 30 &\leq p_d \leq 80 \\ 40 &\leq p_T \leq 90 \end{aligned}$$

As convenções trazem uma demanda fixa de 100 pessoas e exigem os custos mínimos. Ocupam 20 quartos simples obrigatoriamente. As convenções utilizam o salão de convenções com um retorno de V unidades monetárias. Quando o salão não está sendo utilizado, o hotel gasta R unidades monetárias em sua manutenção.

Os convênios com empresas garantem a ocupação de 10 quartos simples ao preço de mercado ($S_p \geq 10$).

Sabe-se que as despesas rateadas correspondem aos custos mínimos dos preços dos quartos, que é possível captar 20 pessoas avulsas diariamente pagando o preço máximo da tabela e ocupando quartos duplos e, que, no mínimo, 30% das pessoas das excursões ficam em quartos simples. Sabendo-se que não são viáveis excursões com menos de cinco pessoas e que os quartos duplos ou triplos podem ser alugados como simples (mantendo os custos mínimos rateados). Estabelecer o programa de programação matemática para otimizar a ocupação do hotel.

4 – O Problema das Damas do Jogo de Xadrez



Em um tabuleiro de xadrez (padrão 8×8) vazio, sabe-se que uma alocação de uma peça do jogo em uma casa preta vale o dobro do que em casa branca. Determine a localização ótima para 8 (oito) damas de modo que nenhuma delas seja ameaçada pelas demais. Formule esse problema como um PPL e o solucione com o auxílio do simplex.

5 – O Problema das Peças de Xadrez



1 – Um jogador resolveu distribuir as torres, bispos e cavalos de um jogo de xadrez em um tabuleiro vazio de forma a minimizar o número de casas não atacadas e não permitir que as peças alocadas se atacassem diretamente (considerado apenas o ataque entre peças de cores diferentes). Formule esse problema como um PPL e o solucione com o auxílio do simplex.

2 – Formule e solucione o caso 1, considerando que o jogador pode dispor do número que desejar de torres e bispos, mas as peças são independentes, ou seja, passíveis de serem atacadas por peças de mesma cor.

3 – Formular e solucionar o problema 2 considerando a casa de alocação de uma peça como casa não atacada.

6 – O Problema do Atendimento Bancário de Aposentados



Um certo banco de uma cidade resolveu redefinir seu sistema de atendimento de aposentados pelas agências de modo a maximizar o conforto no atendimento dessas pessoas idosas, uma vez que as medidas de automação e acesso remoto não se mostraram suficientes para aliviar as dificuldades no atendimento de balcão. A demanda de atendimento d_i , $i = 1, \dots, n$ foi discretizada dentro de n blocos geográficos quadrados com cerca de 1.500 m de lado, conforme Figura 5.30. Cada bloco foi considerado concentrado em seu centro de gravidade e representado por um nó em um grafo. Todos os nós desse grafo foram ligados por arestas aos seus blocos vizinhos (distância de 1.500 metros entre cada nó do grafo de discretização). Os nós (blocos marcados), que englobaram agências bancárias já existentes, foram anotados como nós de oferta, $j = 1, \dots, m$, contabilizando-se como oferta disponível a oferta total da agência menos a demanda gerada em seu bloco de localização.

Sabendo-se que a direção do banco deseja que:

1. Nenhum aposentado tenha que se deslocar mais de 4.500 m para ser atendido.
2. Nenhum aposentado gaste mais de R unidades monetárias com deslocamento.
3. Que a demanda prevista pela área de captação alocada a uma agência jamais ultrapasse sua oferta nominal.

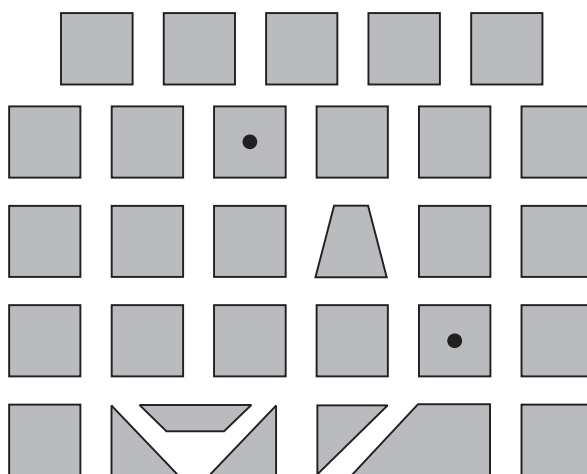


FIGURA 5.30 Discretização da demanda e do atendimento.

Sabendo-se, ainda, que para solucionar o problema da demanda pode-se ampliar uma agência j já existente a um custo de h_j unidades monetárias por usuário, ou implantar um novo posto que custará uma parcela fixa de f_j unidades monetárias, e cada usuário realocado a esses postos de atendimento custará ao sistema mais c_j unidades monetárias.

Conhecendo-se que o deslocamento entre os nós i e j do grafo de discretização custa, em média P_{ij} unidades monetárias, formular o problema de minimizar os gastos com a redefinição do sistema de atendimento, englobando a ampliação de agências e a criação de postos de atendimento no banco.

7 – O Problema da Distribuição de Água



Uma companhia de água e esgoto deseja implementar a rede de abastecimento de água de um bairro recentemente ocupado e em processo de urbanização. A Figura 5.31 mostra que junto ao bairro passa uma adutora de grande porte que deve ser ligada a uma rede de distribuição local que percorra as ruas do bairro. Os custos da rede local são diretamente proporcionais ao comprimento linear dos canos. A rede de água não deve possuir circuitos (canos ligados de modo a formarem um caminho fechado), pois essa configuração, além de dificultar a manutenção, ocasiona problemas no fluxo de água que acabam prejudicando os encanamentos e válvulas de pressão. Formular o problema de distribuir a água em todos os quarteirões do bairro minimizando o custo total do sistema.

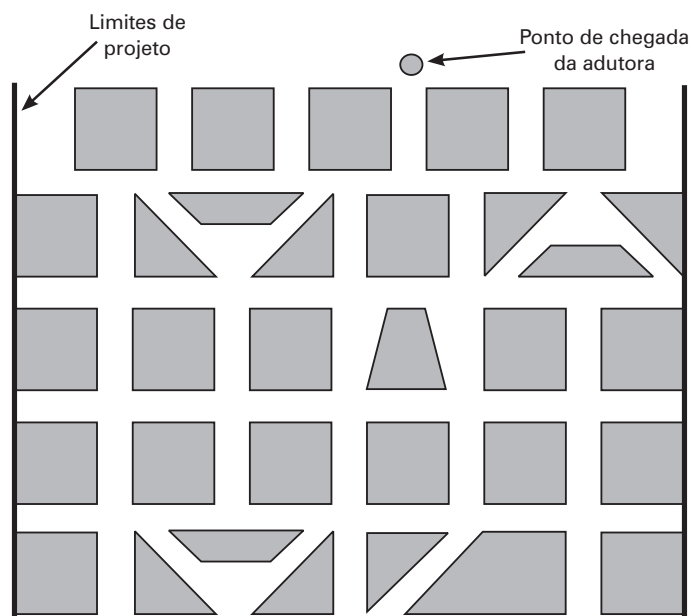


FIGURA 5.31 Mapa da rede a ser implantada.

8 – O Problema do Incêndio Florestal – Operação Prolongada



Considere as seguintes condições para o problema do exemplo 7 da página 57:

Fase 2: Operação Prolongada

Quando o programa de combate a incêndio da fase 1 estava para ser implementado e o dia amanhecia, as condições climáticas na área do incêndio foram alteradas drasticamente. O fogo começou a se alastrar a uma taxa nada desprezível alimentado por ventos de até 50 km por hora. Agora o tempo passou a ser crucial e as condições econômicas do desastre ecológico tornaram-se preponderantes. Os engenheiros florestais calcularam o crescimento global da área a ser varrida pela operação em uma taxa constan-

te de 25%/hora. Os prejuízos também se caracterizam diretamente proporcionais a essa área e são contabilizados em R\$10,00 por metro quadrado da área que será atingida a partir da expansão do fogo. Todos os meios solicitados pelos técnicos no dia anterior haviam chegado e, além disso, uma força reserva não prevista. Um grupo de bombeiros terrestres estará disponível em cerca de 2 horas. Esse grupo pode crescer a capacidade de combate ao fogo em cerca de 20.000 m²/hora ou ser empregado em acero e reduzir a expansão do fogo em cerca de 0,5% por hora de trabalho. O custo operacional do grupo é desprezível no trabalho de acero e de R\$30.000 por hora no combate direto devido aos riscos pessoais envolvidos. Por um erro de comunicação, chegaram com os aviões mais 10 pilotos de helicóptero. Com tantos pilotos de helicópteros descobriu-se que eles poderiam também pilotar os aviões, sendo que os pilotos de avião não podiam fazer o inverso. A Tabela 5.24 de disponibilidade é:

TABELA 5.24

<i>Aparelho</i>	<i>Aparelhos prontos</i>	<i>Pilotos</i>	<i>Operadores</i>
<i>Helicóptero AH-1</i>	7	20	22
<i>Avião Tanque</i>	5		
<i>Avião B67</i>	5	14	

Reformule o problema objetivando minimizar os custos da operação.

9 – O Problema do Professor Atarefado



Um professor recebeu oferta de trabalho em três escolas que são equivalentes em atributos como qualidade de ensino e oportunidade de aperfeiçoamento. Cada uma delas gostaria, se possível, de contratá-lo em regime de tempo integral (dedicação exclusiva). Existem, contudo, outras opções de comprometimento em carga horária. Os módulos de trabalho são de oito horas. A Tabela 5.25 resume as várias propostas que o professor está examinando:

TABELA 5.25 PAGAMENTO EM CADA ESCOLA

<i># de Módulos</i>	<i>Escola 1</i>	<i>Escola 2</i>	<i>Escola 3</i>
1	105	135	150
2	240	250	310
3	400	380	390
4	500	490	560
5	590	630	660

Qual o esquema de trabalho que maximiza o retorno financeiro de seu trabalho?

10 – O Problema do Entregador de Pizza



Um serviço de entrega rápida de pizza recebeu um pedido urgente e especial. A encomenda foi feita por um faminto e generoso milionário. Se eles forem capazes de entregar uma quente e succulenta pizza gigante em apenas cinco minutos terão um substancial bônus. A malha rodoviária que separa o ponto em que a pizza é feita e seu destino está representada no esquema da Figura 5.32. Sobre as ruas aparece a probabilidade percentual que ocorra um problema ao entregador naquele trecho, inviabilizando o cumprimento do prazo. A Figura 5.32 exhibe a malha rodoviária do problema.

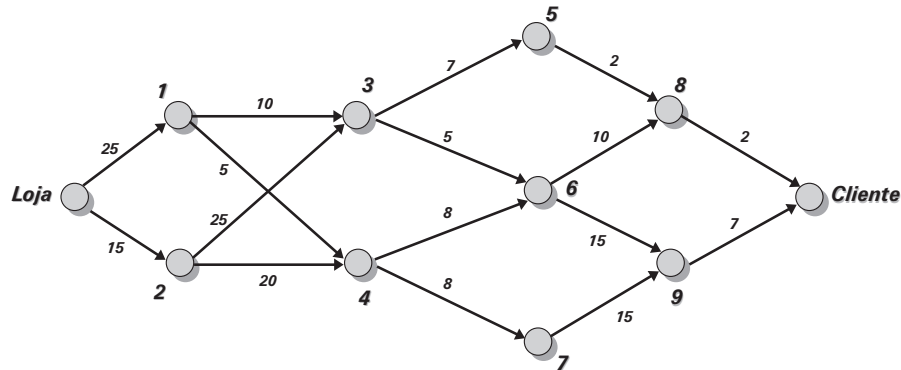


FIGURA 5.32 Malha de entrega de pizza.

Determine a rota que maximiza a chance do entregador de pizza entregá-la no prazo desejado.

11 – O Problema da Venda de Sorvete



Uma rede de lanchonetes possui quatro lojas em uma cidade. Para fazer frente a demanda do verão o superintendente regional de compras da empresa adquiriu sete *containers* de sorvete. Devido à diferente localização de cada uma das filiais existem diferentes potenciais para a venda do sorvete. Face à experiência de venda, o superintendente estimou o lucro líquido (em unidades monetárias) que seria possível obter, dentro do horizonte de venda usual, em cada filial como a Tabela 5.26 sugere:

TABELA 5.26 RETORNO DE INVESTIMENTO.

Lanchonetes				
# de Containers	Ponta Negra	Pirangi	Neópolis	Petrópolis
1	4	2	6	1
2	6	6	12	4
3	10	8	14	8
4	12	10	18	16
5	14	12	20	20
6	16	16	22	21
7	18	20	22	24

Formule o problema de otimizar o lucro total obtido. Determine a distribuição ótima dos *containers*.

12 – O Problema da Marcenaria



Em uma pequena marcenaria existem n tarefas de corte a serem realizadas. Cada tarefa requer p_i minutos para ser processada em uma máquina de serra, $i = 1, \dots, n$. As tarefas estarão prontas para ser entregues a uma máquina a partir do tempo a_i e devem terminar até o tempo b_i (tempos contados a partir do início das atividades diárias da marcenaria). Formule o problema de minimizar o número de máquinas necessárias para completar as n tarefas.

13 – O Problema da Produção de Turbinas



Uma indústria fabrica três tipos de turbina de grande potência (tipos I, II e III). Cada tipo de turbina pode ser vendida por 150, 170 e 240 unidades monetárias, respectivamente. As turbinas possuem três componentes críticos que são importados de empresas especializadas, a saber: o conjunto das pás, rolamentos e injetores de combustível. A indústria possui quatro fornecedores desses equipamentos, conforme mostram as Tabelas 5.27 e 5.28:

TABELA 5.27

<i>Fornecedor \ Item</i>	<i>Ejetor</i>	<i>Conjunto de Pás</i>	<i>Rolamento</i>
<i>A</i>	6 – (18 U.M.)	40 – (1 U.M.)	280 – (1 U.M.)
<i>B</i>	4 – (20 U.M.)	60 – (1, 5 U.M.)	150 – (0, 8 U.M.)
<i>C</i>	6 – (17 U.M.)	100 – (2 U.M.)	120 – (1, 7 U.M.)
<i>D</i>	10 – (19 U. M.)	150 – (1, 7 U. M.)	170 – (1, 6 U. M.)

Cada turbina exige os seguintes insumos:

TABELA 5.28

<i>Insumo Tipo de Turbina</i>	<i>Ejetor</i>	<i>Conjunto de Pás</i>	<i>Rolamento</i>	<i>Mão de Obra (0,01 U.M / H x h)</i>
<i>I</i>	1	6	40	500 Homem × hora
<i>II</i>	1	8	60	600 Homem × hora
<i>III</i>	2	10	85	1200 Homem × hora

A indústria possui $1.800 H \times h$ para a montagem das turbinas. Sabe-se também que uma empresa só vende para a indústria se a sua encomenda total (gastos com ejetores, conjuntos de pás e rolamentos) ultrapassar 400 unidades monetárias. O capital de giro da indústria é de 1.300 unidades monetárias, o que limita as compras a esse teto. Pede-se:

1. Formular o problema de maximizar o lucro com a produção de turbinas.
2. Uma empresa está oferecendo mão de obra para montagem da turbina a 0,02 U.M. por $H \times h$. A indústria deve comprar esse serviço para maximizar seus lucros?
3. Se uma fornecedora concorrente oferecesse no mercado ejetores a 21 U.M. por unidade, a indústria deveria comprá-los? Em caso positivo, quantos? Em caso negativo, qual seria o menor valor que a indústria deveria aceitar pagar?

14 – O Problema da Produção de Ferragens para Janelas de Madeira



Uma metalúrgica produz puxadores, rolamentos e guias para janelas de madeira. Essas ferragens podem ser bronzeadas, pintadas ou anodizadas. A Tabela 5.29 resume o número máximo de lotes passíveis de produção, em uma semana, nas máquinas da empresa, bem como o lucro unitário de venda por lote.

TABELA 5.29 MÁXIMO DE LOTES PASSÍVEIS DE PRODUÇÃO

	<i>Bronzeado</i>	<i>Pintado</i>	<i>Anodizado</i>
<i>Puxador</i>	25 – (2 U.M.)	30 – (1,3 U.M.)	15 – (3 U.M.)
<i>Rolamento</i>	45 – (0,3 U.M.)	20 – (0,5 U.M.)	10 – (1 U.M.)
<i>Guia</i>	10 – (1,2 U.M.)	30 – (1 U.M.)	5 – (2 U.M.)

A empresa possui 800 H x h por semana de mão de obra especializada. A Tabela 5.30 resume a utilização de mão de obra por lote das diversas ferragens produzidas.

TABELA 5.30 MÃO DE OBRA POR LOTE

<i>Tipo</i>	<i>Mão de obra (H × h) por lote</i>		
	<i>Bronzeado</i>	<i>Pintado</i>	<i>Anodizado</i>
<i>Puxador</i>	1	2	4
<i>Rolamento</i>	2	3	3
<i>Guia</i>	1	2	4

Sabe-se que, para bem utilizar o maquinário, para cada 5 lotes de puxadores produzidos deve-se fabricar entre 2 e 8 lotes de rolamentos e entre 1 e 9 lotes de guias.

Formular o problema de maximizar o lucro da metalúrgica.

15 – O Problema do Tratamento Químico de Peças Metálicas



Em uma indústria metalúrgica após a fabricação, todas as peças são submetidas a um tratamento químico contra a corrosão e abrasão. A empresa produz vários tipos de peças em sua linha de fundição e moldagem. Para realizar o tratamento existem quatro tanques de imersão. As peças metálicas são fixadas em um varal e submersas em cada tanque durante um certo tempo. Os tanques contêm elementos químicos diferentes e são dispostos em uma sequência que produz, após as imersões, o efeito desejado. Durante todo o tempo de imersão as peças sofrem a ação do processo associado ao respectivo tanque. Cada peça $i, i = 1, \dots, n$, possui um tempo mínimo $t_{ij}, j = 1, 2, 3, 4$, e um tempo máximo r_{ij} de permanência no tanque j . Os custos do processo são diretamente proporcionais ao tempo de permanência da peça i no tanque j e a constante de absorção associada ao efeito do composto químico do tanque j sobre a peça i . O peso p_i da peça i é conhecido. Cada varal tem quatro posições (que não necessitam ser obrigatoriamente preenchidas) para fixar as peças e deve ser carregado com, no mínimo, v quilos de modo que não exista problemas de flutuação no momento da imersão e, no máximo, w quilos para que não se rompa.

Sabendo-se que existe uma certa disponibilidade de h horas para completar o tratamento químico do lote de n peças, formular o problema de programar a sequência de carregamento dos varais de modo a minimizar os custos globais do processo.

16 – O Problema do Balanceamento das Rodas de um Veículo Blindado



Um certo tipo de veículo de combate blindado deve ter suas rodas balanceadas através da possível colocação de calços em quatro pontos específicos de seus aros. Um calço $i, i = 1, \dots, 16$, quando fixado na posição $j, j = 1, \dots, 4$, da roda $k, k = 1, \dots, 4$, produz um deslocamento horizontal na dinâmica do giro da roda igual a h_{ij}^k , e um deslocamento vertical v_{ij}^k . Esses valores podem ser positivos ou negativos. Cada

roda possui um desequilíbrio dinâmico de constituição igual a h_k e v_k , nos eixos horizontal e vertical, respectivamente. Nem todas as posições próprias para a fixação de calços necessitam ser ocupadas obrigatoriamente. Em uma operação de campanha, para aumentar a eficiência do processo, no sentido de diminuir os tempos operacionais e eliminar o procedimento da correção artesanal, foram realizados estudos estatísticos que resultaram na elaboração de um *kit* de balanceamento padrão com 16 calços específicos.

Sabendo-se que a soma dos desequilíbrios horizontais residuais das rodas dianteiras do veículo devem ser menores ou iguais que as somas dos desequilíbrios verticais das rodas traseiras, formular o problema de programação matemática capaz de planejar a utilização do *kit* padrão de modo a minimizar o desequilíbrio residual total das rodas do blindado.

17 – O Problema da Bibliotecária



Uma bibliotecária deseja organizar um novo lote de livros (classificá-los) e para isso será utilizado uma nova área da biblioteca. Existem dois fatores limitantes para o acondicionamento dos livros nas estantes: a altura do livro e o comprimento da prateleira. Um livro i possui uma altura h_i e uma espessura w_i . Obviamente um livro só pode ser guardado em uma prateleira que possua uma altura igual ou maior que a sua. Os livros não podem ser colocados uns sobre os outros. Não existe limitação no espaço físico, de modo que podem ser construídas tantas prateleiras quanto forem necessárias, porém a verba que compra os livros é a mesma que paga as prateleiras, sugerindo-se que os gastos com as prateleiras devem ser minimizados. A Figura 5.33 exemplifica a configuração das prateleiras em estantes.

As prateleiras são fixadas de forma a que produzam espaços de altura variável mas com um comprimento padrão igual a L metros. A criação de uma nova prateleira custa R unidades monetárias (o valor inclui os montantes [paredes laterais] da estante). A altura máxima da estante (um conjunto de prateleiras) é de S metros, sendo desprezada a espessura do metal da prateleira. O espaço não ocupado em uma prateleira é considerado desperdiçado.

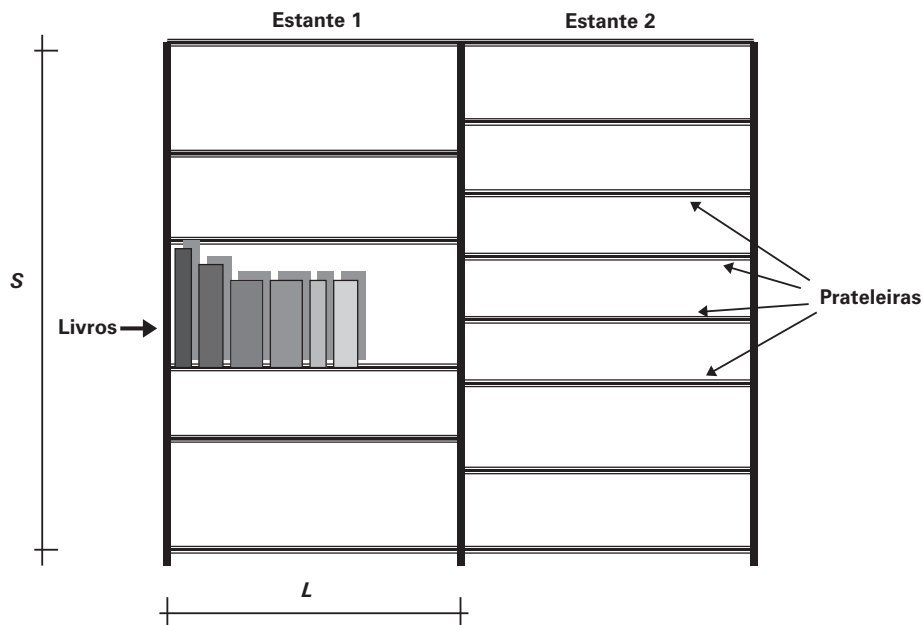


FIGURA 5.33 Esquema de prateleira.

1. Formular o problema da classificação dos livros com o objetivo de minimizar o número de prateleiras montadas.
2. Considerando que livros iguais (cópias de um mesmo livro) devem ser classificados sempre juntos (na mesma prateleira), reformular o problema anterior. Para o caso específico, consideraremos que não existe caso em que o grupo de livros repetidos exceda o tamanho L de uma prateleira.
3. Em uma certa partida do lote, os livros são muito pesados, de modo que as prateleiras podem sofrer deformações desagradáveis se esse aspecto não for levado em conta. Considerando p_i o peso de cada livro e V a capacidade máxima de uma prateleira, reformular o problema 1 levando em conta a capacidade nas prateleiras.
4. Reformular o problema 2, incluindo as restrições de peso com a seguinte adaptação: permitir que livros iguais sejam classificados em no máximo duas prateleiras.

18 – O Problema do Muro de Arrimo



Um certo talude será estabilizado com um muro de arrimo composto de 33 placas pré-moldadas em concreto com 1 m^2 de área e 20 cm de espessura, encaixadas em uma estrutura metálica de sustentação. O terreno do talude é bastante heterogêneo e encontra-se sujeito a diversas infiltrações. Nos estudos preliminares para o projeto, o levantamento de carga foi avaliado através de um corte transversal no terreno do talude, optando-se por discretizar, dentro do padrão das placas pré-moldadas, as cargas que esse talude acarretaria sobre o muro de arrimo, conforme a figura que se segue. O muro é fixado na posição vertical em relação ao terreno, sendo que sua estrutura metálica de sustentação é fundada na rocha, cujo perfil a figura destaca em uma vista frontal.

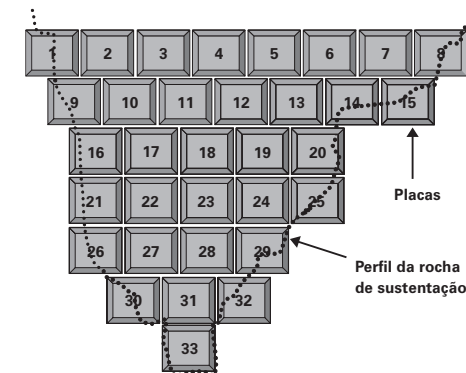
Os esforços de tração despertados pelo talude no muro de arrimo serão absorvidos pela estrutura metálica de sustentação e por uma conveniente armação nas placas de concreto. Os esforços de compressão serão neutralizados pelo concreto das placas, cabendo dimensioná-lo para tal. A resistência à compressão de uma placa de concreto está associada a diversos fatores, dentre os quais a percentagem do cimento utilizado no “traço” (na mistura cimento, areia e brita que constituirá o concreto). Considerando-se que o consumo de concreto (em uma aproximação linear justificada pelo regime de deformação que a estrutura será submetida) está diretamente associado à resistência final do pré-moldado e que, devido ao efeito conjunto da estrutura, a resistência ao esforço de solicitação (cargas da figura) de certa placa é auxiliada pela resistência das placas vizinhas segundo a seguinte fórmula:

$$p_{ij} = r_{ij} + 0,1 \left(\sum_{s \neq k} r_{sk} \right)$$

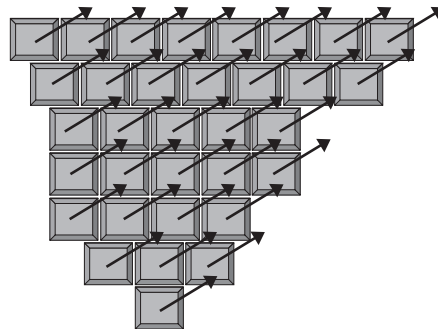
em que s e k representam os índices das células vizinhas a célula $i-j$, p_{ij} representa a tensão nominal máxima admissível para a célula $i-j$ (dentro do modelo de discretização adotado) e r_{ij} a resistência a ser obtida na fabricação da placa $i-j$. Considerando-se ainda que o custo do cimento a ser utilizado na confecção do concreto das placas corresponde a c unidades monetárias para cada unidade de resistência alcançada (em kg/cm^2 , conforme a norma brasileira¹):

Formular o problema de fabricar o conjunto das placas do muro de arrimo minimizando o consumo de cimento.

1. Resistência obtida no ensaio de tração aos 28 dias.



Configuração do muro de arrimo – vista frontal



Esquema dos esforços atuantes sobre as placas

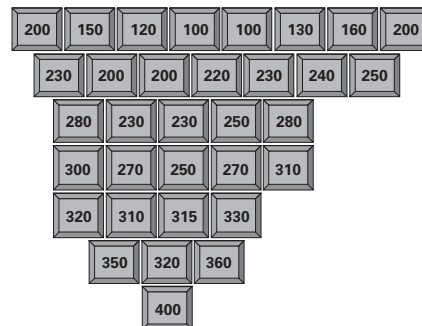
Valor discretizado dos esforços resultantes de compressão nas placas (kg/cm^2)

FIGURA 5.34 Vista frontal do muro de arrimo

19 – Escolha de Projetos Alternativos



Uma operadora de turismo associada a um banco de investimento regional resolveu investir na região Nordeste do Brasil, estabelecendo parcerias com hotéis e pousadas da região. Os projetos estão divididos em três categorias: investimento em construção e infraestrutura, eventos religiosos e cidades históricas. Para projetos de investimento em construção e infraestrutura existem as propostas A, B e C da rede Sol, Mar e Recife, objetivando a modernização de vários de seus hotéis, a um custo de US\$7, US\$5 e US\$4 milhões, respectivamente, o Hotel Duna Branca com os projetos D e E a um custo de US\$7 e 8 milhões, e o Hotel Pirangi com o projeto F a um custo de 4 milhões. Para os eventos religiosos estão disponíveis os projetos G, H e I das Pousadas Natal, avaliados em US\$6, 2 e 3 milhões, respectivamente. Finalmente, para promoção e sustentação de programações em cidades históricas, existem os projetos J e K do Hotel do Virgulino e os projetos L e M, do Hotel da Volante, orçados em

US\$6, 10, 12 e 4 milhões, respectivamente. Desejando maximizar o retorno do investimento, os projetos foram examinados por uma comissão de especialistas que atribuiu, comparativamente, o seguinte valor em pontos aos projetos:

A = 3; B = 5; C = 8; D = 11; E = 6; F = 13; G = 15; H = 2; I = 7; J = 8, K = 7; L = 10; M = 9 pontos. A operadora, interessada em criar laços diversificados na área, estabeleceu que pelo menos um projeto em cada área seria apoiado. Sabendo-se que:

1. Os projetos de modernização são julgados prioritários, de modo que, pelo menos três, serão apoiados.
2. Face a problemas políticos, o Hotel do Virgulino e o Hotel Pirangi devem ter o mesmo número de projetos aprovados.
3. Existe US\$20 milhões para investir.

- ✓ Planejar o melhor investimento possível do recurso disponível.
- ✓ Após o planejamento ter sido realizado, uma informação deu conta de que os custos dos projetos do Hotel da Volante tinham sido avaliados com imprecisão, impondo-se uma correção de 30% a mais para seus custos finais. Isso altera o quadro de distribuição dos projetos? De que forma?
- ✓ Caso esse problema se repita com o Hotel Sol, Mar e Recife (cujas propostas são vistas com desconfiança pela operadora de turismo), até que ponto os custos podem aumentar sem alterar a solução?
- ✓ O Hotel Piscinas do Sertão resolveu encaminhar, tardiamente, um projeto para construção e infraestrutura. Seu custo total foi de US\$5 milhões e obteve uma avaliação de 8 pontos. O departamento de análise do banco de investimento recusou o projeto pois, segundo suas palavras: “Não era competitivo face aos demais projetos vencedores.” Isso é verdade?

20 – O Problema dos Anúncios na Rede de TV



Etapas1: O ponto de vista dos anunciantes

Uma rede de televisão resolveu estabelecer preços competitivos para o tempo de comercial em certos horários. Existem 3 horários para a propaganda em promoção na rede: horário nobre noturno (horário 1), horário da tarde em fins de semana (horário 2), horário da tarde em dias da semana (horário 3). O preço de um módulo mínimo de propaganda em cada horário é p_1 , p_2 e p_3 , respectivamente. A rede vende grandes espaços de tempo dentro da programação, denominados pacotes promocionais. Os anunciantes desejam entrar no programa de pacotes de propaganda promocional porque podem obter, com isso, melhores preços unitários. Um pacote representa um esquema de desconto baseado na homogeneização de demanda e economia de escala. Existem quatro faixas de preços sendo praticadas nos pacotes. Faixa 1, ou de desconto zero, faixa 2, ou de desconto de 10%, faixa 3 com um desconto de 20% e faixa 4 com desconto de 10 %. Os descontos incidem sobre o somatório total dos tempos adquiridos, desde que o anunciante cumpra as condições de habilitação. O desconto da faixa 4 pode ser somado aos descontos obtidos nas faixas 2 e 3. Os descontos das faixas 2 e 3 não são cumulativos entre si.

As condições de habilitação dizem respeito a restrições de tempo total mínimo de propaganda adquirido e distribuição da programação do anunciante dentro dos horários (1, 2 e 3) da emissora. A Tabela 5.31 resume as regras da promoção:

TABELA 5.31

<i>Descontos</i>	<i>Somatório de tempo</i>	<i>Condições de distribuição da programação</i>	<i>Valor do desconto</i>
Faixa 1	sem exigência	sem exigência	0%
Faixa 2	sem exigência	comprar, no mínimo, tempo em dois horários	10%
Faixa 3	sem exigências	comprar, no mínimo, tempo em três horários	20%
Faixa 4	comprar, no mínimo, f unidades de tempo	sem exigências	10%

Os anunciantes que competem na disputa do uso da mídia são n e dispõem, cada um deles, de s_n unidades monetárias para investir em propaganda.

Os anunciantes, reunidos em uma associação, desejam estabelecer a melhor estratégia de negociação com a rede, objetivando maximizar, dentro das disponibilidades orçamentárias, o tempo global de utilização da mídia.

Etapa 2: O ponto de vista da rede de TV

Como existem tempos disponíveis para a propaganda ainda não utilizados, a rede resolveu incentivar a demanda. Ela sabe que o investimento dos anunciantes é sensível à estratégia de preço. Através de uma pesquisa de mercado descobriu que, na medida em que reduz os preços básicos (em todos os horários), o investimento do anunciante n tende a crescer segundo a equação:

$$s_n^* = s_n + 10d$$

onde s_n^* representa o montante final de investimento disponível e d o desconto em unidades monetárias. Sabe também que, se os investimentos forem aumentados pela redução dos preços, o cliente procurará minimizar seus custos finais através da estratégia descrita na etapa 1.

Formular o problema de estabelecer a política de preços que maximize o retorno total, mantidos os pacotes de promoção, sabendo-se que o tempo que não for comprado pelos anunciantes pode ser utilizado pela própria rede de TV, valendo r_i unidades monetárias por segundo, $i = 1, 2, 3$, dentro de cada horário possível.

Sugestões para a solução da etapa 1

Sob o ponto de vista dos anunciantes o desejável é maximizar o desconto, atendidas as restrições próprias de cada anunciante.

1 – Escolha da variável de decisão

$t_i \equiv$ tempo disponível para propaganda no horário i , $i = 1, 2, 3$.

$x_{ik} \equiv$ tempo adquirido no horário i , $i = 1, 2, 3$, pelo anunciante k , $k = 1, \dots, n$.

$y_{ik} \equiv \begin{cases} 1, & \text{caso o anunciante } k \text{ tenha adquirido tempo no horário } i \\ 0, & \text{caso contrário} \end{cases}$

$w_{ik} \equiv \begin{cases} 1, & \text{caso o anunciante } k \text{ tenha adquirido mais do que } f \text{ módulos de tempo} \\ 0, & \text{caso contrário} \end{cases}$

$s_k \equiv$ montante máximo de recursos disponibilizado pelo anunciante k .

$p_i \equiv$ preço do tempo de anúncio no horário I .

$c_k \equiv$ custo total que será pago pelo anunciante k , face ao total de desconto que lhe cabe.

2 – Formulação das restrições tecnológicas (abordagem não linear)

a) Restrições relativas ao montante pago pelo anunciante, face aos possíveis descontos

$$pg_k = \sum_{i=1}^3 x_{ik} p_i, \quad k = 1, \dots, n$$

$$c_k \leq (3 - \sum_{i=1}^3 y_{ij}) 0,1 pg_k + 0,8 pg_k - 0,1 pg_k (1 - w_{ij}), \quad k = 1, \dots, n$$

$$s_k \geq c_k, \quad k = 1, \dots, n$$

Observamos que, se o anunciante k faz qualquer anúncio, o somatório dos y_{ik} na segunda restrição é pelo menos 1. A variável pg_k acumula o gasto sem desconto.

21 – O Problema da Operação da Frota de Táxis



Uma operadora de táxis controla cerca de 120 associados em uma grande área metropolitana. A operadora foi constituída para centralizar a operação dos veículos. A central recebe as chamadas de atendimento e designa os carros para as corridas. Ela possui um sistema que permite o controle do posicionamento dos veículos, de modo que pode tomar decisões com uma visão global do atendimento. Existem três tipos de chamadas: programadas, normais e urgentes. Na chamada normal, sob a ótica do cliente, deseja-se minimizar o tempo de atendimento, respeitadas as restrições operacionais. A chamada urgente designa o carro mais próximo ao ponto do pedido e dá uma expectativa de chegada ao cliente. As chamadas urgentes são 50% mais caras. Se o tempo de chegada da chamada urgente for maior que três minutos então essa chamada é cobrada como normal, apesar do carro designado ser o mais próximo. Em todos os casos, (chamadas normais ou urgentes) deve ser considerada a possibilidade de atendimento face um tempo máximo r_j , estabelecido pelo cliente j . As chamadas programadas determinam a presença de um táxi no destino exatamente na hora acordada. Para tal, as chamadas programadas são feitas com, no mínimo, uma hora de antecedência.

Os taxistas, na verdade trabalhadores autônomos associados em cooperativa, exigem que a central realize uma designação equânime sob alguns critérios. A cidade é dividida em quatro áreas, como mostra a Tabela 5.32.

TABELA 5.32

<i>Desempenho</i> \ <i>Área</i>	<i>Urbana</i>	<i>Suburbana</i>
<i>Alto</i>	a_1	a_3
<i>Baixo</i>	a_2	a_4

Os profissionais desejam que exista um equilíbrio no número de chamadas de cada táxi em cada área, de modo que fatores como engarrafamento, riscos de segurança, “corridas longas” etc., sejam aproximadamente os mesmos para todos. Outro ponto a ser considerado é que os carros não tenham

que se deslocar muito para atender “corridas curtas”. Formule o problema de designação dos táxis a chamadas como um programa de programação linear.

Informação adicional: o objetivo do sistema computacional de designação deverá ser otimizar os ganhos com os atendimentos. Os ganhos são diretamente proporcionais ao número de quilômetros rodados e pagos a R\$1,0, sendo que a bandeirada é de R\$1,5. Considere a operação em horário comercial com todos os 120 carros em disponibilidade.

22 – O Problema da Reforma da Pirâmide



Uma pirâmide construída há milhares de anos com blocos cúbicos de calcário de 1 m de lado está necessitando ser reformada. Vários blocos estão apresentando trincas e podem entrar em colapso, o que levaria à destruição do monumento histórico. Uma prospeção sonora classificou os blocos segundo seu risco de ruptura (r_i), $i = 1, \dots, 380$. Operacionalmente, para se retirar um bloco na pirâmide, é necessário remover, caso existam, até nove outros blocos no nível superior, como mostra a Figura 5.35:

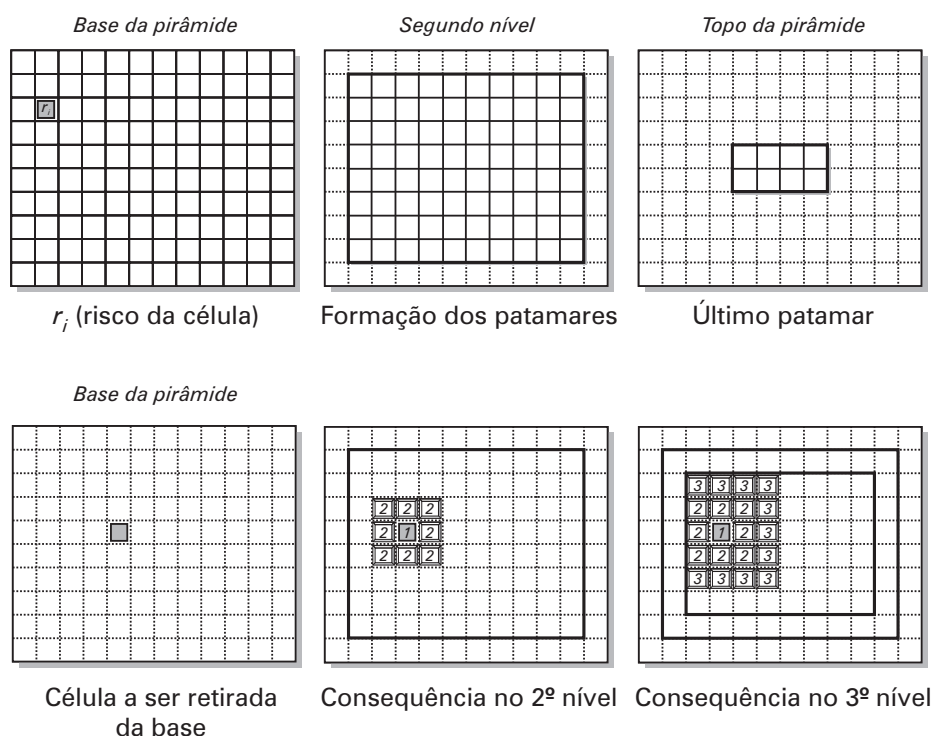


FIGURA 5.35 Esquema da retirada de blocos

O efeito da retirada de um bloco se propaga na pirâmide em virtude da imposição da estabilidade estrutural. Para a retirada do bloco da base da pirâmide marcado em negrito pelo esquema da figura, é necessário remover outros 34 blocos. Se por um lado existe o interesse em substituir os blocos de maior risco, por outro deve-se evitar remover um grande número de blocos, sob pena de desfigurar o monumento e de ocasionar novas fraturas. Dos 380 blocos originais, fixou-se em 80 o número máximo dos que poderão ser removidos.

Visando à substituição dos blocos, formular o problema de determinar um movimento de blocos que seja estável e que maximize o somatório dos riscos de ruptura de seus blocos componentes.

6

PROBLEMAS DE CONEXÃO: ÁRVORES, CAMINHOS E EMPARELHAMENTO

6.1 – INTRODUÇÃO

6.1.1 – As Origens do Problema – A Otimização da Conexão

Em inúmeras situações, o problema de otimização estará associado às alternativas de conexão entre facilidades, ou entre os clientes e suas facilidades. Os problemas de localização caracterizam várias situações em que existe a necessidade de se examinar a distribuição do atendimento. Uma outra espécie de situação exige o estudo mais aproximado das ligações entre as demandas e seus pontos de atendimento. Nesse caso, a ênfase do modelo se desloca para a conexão ou a forma de acesso ou contato. Para esses tipos de problemas, o tomador de decisão necessita ser capaz de exibir uma topologia adequada ao modelo, formas de organizar as configurações desejáveis dentro dessa topologia e critérios de escolha dessas configurações. Nesse tipo de problemas em que a tomada de decisão é polarizada pela arquitetura de ligação, três estruturas de representação são extremamente importantes:

✓ Caminhos

Quando o foco está em uma trajetória.

✓ Árvores

Quando o foco está na continuidade da conexão e no estabelecimento de uma espinha dorsal sobre um conjunto de pontos demandantes.

✓ Emparelhamentos (*matching*)

Quando o problema da conexão envolve pequenos grupamentos.

O critério mais comum para esses tipos de problemas é o de minimização da estrutura conectora ou do processo de ligação, daí a importância do *caminho mais curto* (a ligação mais barata entre dois pontos), das diversas *árvores minimais* (a estrutura de conexão global mais barata) e do *1-matching mínimo* (emparelhamento de pontos dois a dois a custo mínimo). Eventualmente, essas estruturas podem estar associadas a objetivos de maximização como o caminho mais longo e a árvore geradora máxima. Metodologicamente, existe uma forte ligação entre essas duas últimas estruturas (ver Fredman e Willard [1994]), o que torna seu estudo conjunto conveniente sob o aspecto algorítmico.

6.2 – O PROBLEMA DA CONEXÃO SIMPLES

O problema mais simples dessa classe é exatamente aquele em que o objetivo é simplesmente verificar (problema de decisão) ou estabelecer (problema de localização) a continuidade da ligação entre os elementos da estrutura em análise ou projeto. Uma interpretação para essa chamada “continuidade” pode ser, por exemplo, passar de um vértice para outro em um grafo através de uma aresta ou arco. A Figura 6.1 mostra duas formas de conexão em um grafo. Lembrando que as arestas significam a possibilidade de ligação em ambos os sentidos, na Figura 6.1 (a) vemos que os vértices 3 e 1 se ligam apenas ao nó 2. Na Figura 6.1 (b), 3 pode se ligar diretamente a 1.

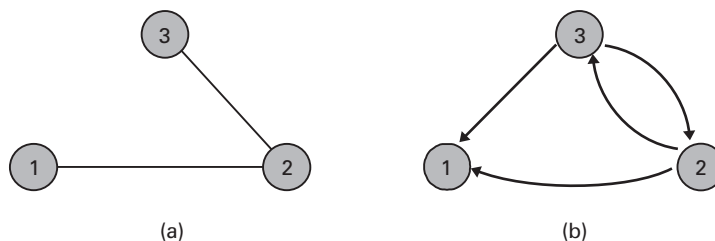


FIGURA 6.1 Formas de conexão.

É evidente que a intensidade da possibilidade de ligações é um fator importante. Denominaremos essa característica de conexidade. Ela pode variar desde o estado dito desconexo, até o chamado fortemente conexo. Os grafos *desconexos* possuem um ou mais vértices que não se ligam a qualquer outro. Nos grafos fortemente conexos, de qualquer vértice de G é possível alcançar a todos os outros através de um passeio legal sobre suas arestas ou arcos. Os grafos não direcionados conexos são, consequentemente, fortemente conexos. Mesmo que o grafo seja desconexo é extremamente provável que possua subgrafos conexos. Cada subgrafo conexo de G é chamado de componente conexa. Para a conexidade, de uma forma geral, o fato de um grafo ser ou não direcionado pode significar importante distinção. Por essa razão, denominaremos os grafos direcionados por $G = (V, E)$, onde V representa o conjunto de n vértices e E o conjunto de m arcos.

A verificação da desconexidade de um grafo é um problema trivial e pode ser solucionada pelo simples exame das estruturas de representação de G . De um modo geral, podemos dizer que o desenho de bons algoritmos para a determinação de elementos associados à conexidade em grafos está associado ao domínio de boas técnicas de busca em grafos. Podemos descrever uma busca em um grafo no seguinte algoritmo geral:

Algoritmo Busca Geral em Grafos Conexos

INÍCIO

Ler os dados de G , {grafo direcionado ou não}

Escolher e marcar um vértice x_i inicial;

Enquanto existir $x_j \in N$, $j = 1, \dots, n$ marcado e com uma aresta (x_j, x_k) não explorada **efetuar**

 Início

 Escolher o vértice x_j e explorar a aresta (x_j, x_k) {*condição variável em conformidade com o tipo de busca *}

 Se x_k é não marcado **então** marcar x_k

 Fim

FIM

O algoritmo busca geral descrito examina pelo menos duas vezes as arestas e os vértices de G , possuindo, portanto, complexidade $O(nm)$. À medida que os critérios de escolha dos vértices podem exigir um esforço computacional maior, a complexidade da busca também crescerá. Uma busca é denominada *em profundidade* se, para o critério de seleção de vértices, exigirmos que a *escolha seja feita sobre o mais recentemente alcançado na busca* e incidente em alguma aresta não marcada.

A seguir descrevemos um algoritmo clássico para a determinação das componentes fortemente conexas em um grafo G .

Algoritmo de Roy (Componentes Fortemente Conexas de G)

INÍCIO

Ler os dados de $G = (V, E)$ {direcionado}

$i \leftarrow 0$

Enquanto $V \neq \emptyset$ efetuar

 Início

 Escolher e marcar um vértice qualquer $x_i, x_i \in V$, com (+) e (-);

 Marcar com (+) todo vértice não marcado com (+) e
 que tenha como sucessor um vértice (+)

 Marcar com (-) todo vértice não marcado com (-) e
 que tenha como antecessor um vértice (-)

 Se todos os vértices já não podem ter suas marcas
 alteradas então

 Início

$i \leftarrow i + 1$

$S_i \leftarrow$ os vértices que estão marcados com
 (+) e (-) simultaneamente

$V \leftarrow V \setminus \{S_i\}$

 Fim

 Fim

FIM

Vamos aplicar o algoritmo de Roy para a determinação das componentes fortemente conexas do grafo $G = (V, E)$ da Figura 6.2.

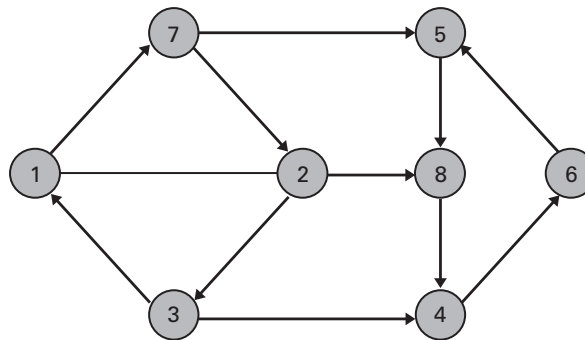


FIGURA 6.2 Grafo exemplo para determinação de componentes conexas.

Decidindo sempre escolher o vértice de menor índice para dar sequência à busca, o desenvolvimento da rotulação gerada pelo algoritmo pode ser acompanhado na Figura 6.3. O algoritmo pode ser utilizado em grafos direcionados ou não.

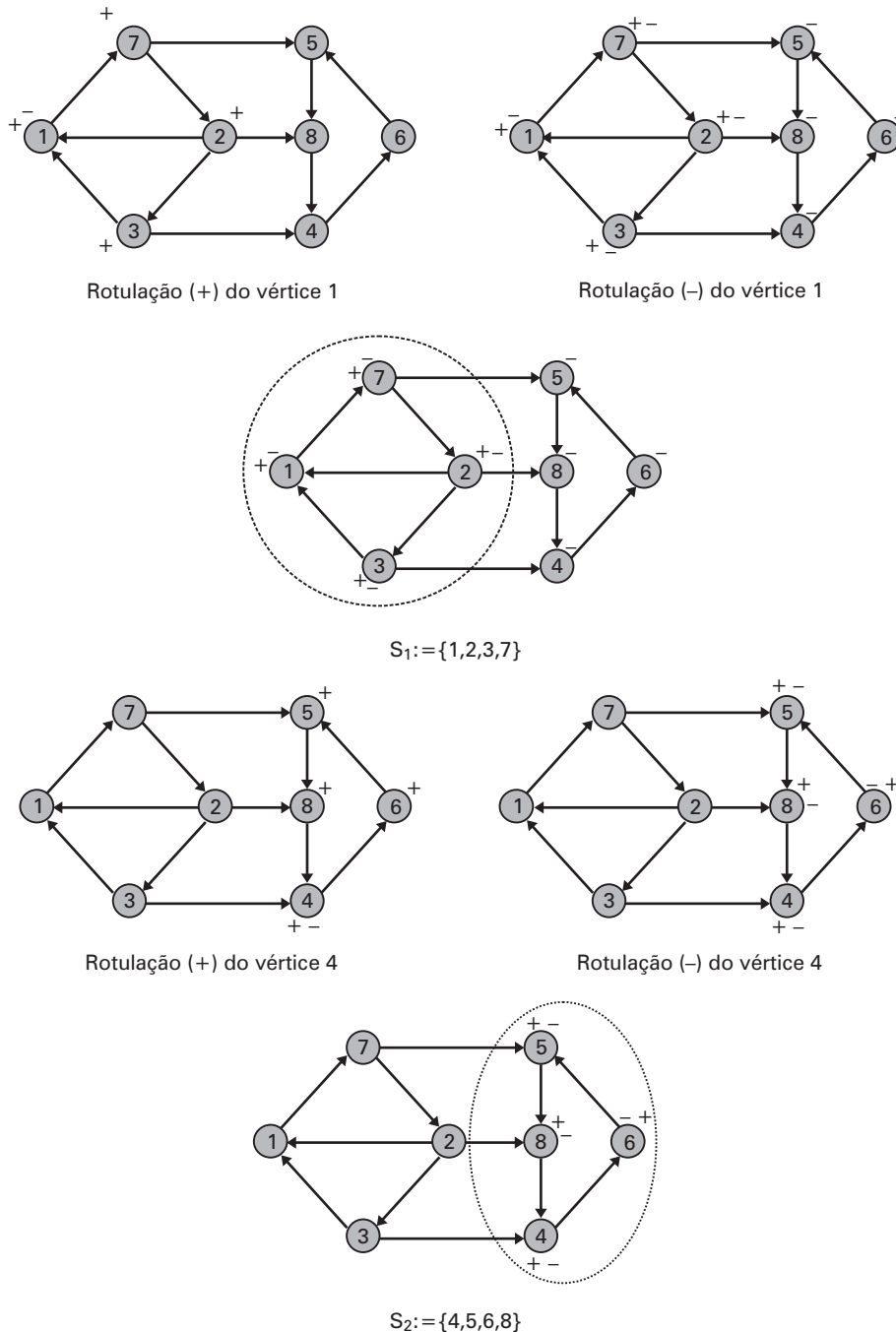


FIGURA 6.3 Solução via algoritmo de Roy.

6.3 – O PROBLEMA DO CAMINHO MAIS CURTO (PCMC)

Sendo u e v dois vértices do grafo $G = (N, A)$, o *caminho mais curto* entre u e v é uma sequência de arestas que, passando por vértices distintos, liga u a v de forma a acumular o menor comprimento, ou distância. Para que possa haver um caminho mais curto entre os vértices u e v é indispensável que exista pri-

meiramente uma conexão entre u e v , ou seja, se existe um caminho de u para v , significa que v é um sucessor de u em algum passeio legal sobre o conjunto N . O problema do caminho mais curto está contextualizado dentro do problema maior de percursos em grafos, que envolve:

- Caminhos mais curtos.
- Caminhos mais longos.
- Percursos Hamiltonianos e Eulerianos.

O problema de caminho mais curto está intimamente relacionado à solução de vários problemas combinatórios como os de roteamento, programação e sequenciamento de tarefas etc.

O problema de percursos ou de passeios é um dos mais conhecidos e antigos na pesquisa operacional e sua solução foi sugerida até mesmo em lendas mitológicas da Grécia antiga. O problema mais elementar na otimização de percursos é o chamado problema do labirinto, em que o objetivo é apenas não repetir trajetos já realizados. Nessa linha de raciocínio, encontramos o chamado algoritmo de Trémaux (ver Boaventura [1996]). A característica desse algoritmo é que a estrutura do grafo (labirinto) não precisa ser conhecida e, na medida da própria leitura dos dados, o exame pode ser realizado. Algoritmos nessa linha de solução são hoje bastante desejáveis, na medida em que a robótica exige cada vez mais bons algoritmos para a navegação (Bar-Eli *et al.* [1994] e Papadimitriou e Yannakakis [1989]) em contextos com vários elementos de incerteza. Como a solução encontrada por Teseu na lenda do Minotauro, o algoritmo de Trémaux marca cada aresta percorrida pela busca para evitar que o expedicionário se perca no trajeto. Os algoritmos de caminho mais curto, contudo, dispendo de todas as informações possíveis sobre o grafo, objetivam minimizar os esforços de busca.

6.3.1 – Formulação

Podemos formular o problema do caminho mais curto como um problema de programação matemática da seguinte forma:

$$\begin{aligned}
 & \text{(CMC) Minimizar } z = \sum_{(i,j) \in A} c_{ij} x_{ij} \\
 & \text{sujeito a:} \\
 & \sum_{(i,j) \in A} x_{ij} - \sum_{(k,i) \in A} x_{ki} = \begin{cases} -1 & \text{se } i = o \\ 0 & \text{se } i \neq o \text{ e } i \neq d \\ +1 & \text{se } i = d \end{cases} \\
 & x_{ij} \in \{0, 1\} \quad (i, j) \in A
 \end{aligned}$$

Onde os vértices o e d representam os vértices de início e término do caminho. É interessante observar que, para essa formulação, a matriz de incidência é totalmente unimodular, o que permite, caso utilizemos a regra de Cramer para obter a solução desse sistema, que a exigência de integralidade seja relaxada sem qualquer prejuízo para a solução inteira.

6.3.2 – Problemas Correlatos e Aplicações

O problema do caminho mais curto está associado a um enorme número de outros problemas na área de roteamento, manufatura, fluxo em redes etc. Destacaremos dois modelos mais recentes.

✓ Caminho mais Curto com Janelas de Tempo (PCMC-JT)

Trata-se da versão do caminho mais curto restrita em um intervalo de chegada aos nós, semelhante aos problemas de roteamento com janela de tempo. Desrochers e Saumis (1988) estudam a adaptação dos algoritmos de rotulação para esse caso.

✓ *Caminho mais Curto em Grafos Estocásticos (PCMC-GE)*

Nesse caso existem algumas linhas de trabalho que incluem:

- Falha nos vértices (Jaillet [1992]).
- Custos estocásticos (Loui [1983]).
- Ligações estocásticas (Psaraftis e Tsitsiklis [1993]).

As estratégias de abordagens para a solução definem o “caminho ótimo sob incerteza” como:

- Aquele que não excede o comprimento do caminho em valores prefixados (Henig [1990]).
- Aquele que mais provavelmente vai se aproximar do caminho mais curto (Sigal *et al.* [1980]).
- Aquele que *a priori* minimiza a distância esperada (Jaillet [1992]).

Murthy e Sarkar (1997) sugerem três algoritmos exatos para o PCMC-GE na linha da maximização da utilidade esperada.

6.3.3 – Algoritmos de Solução

Existem várias abordagens possíveis para a solução do PCMC. Na área de programação matemática encontramos a possibilidade de:

- Algoritmos com especialização do simplex.
- Algoritmos de fluxo.

Entretanto, os mais eficientes algoritmos hoje estão disponíveis através da abordagem em grafos. Nessa linha, dentre várias, duas grandes vertentes são dominantes (Frederickson [1987] e Chen [1990]):

- Algoritmos de ajustes sucessivos.
- Algoritmos por indução e ajuste.

Em relação ao problema em si, três situações são distintas:

- O caminho mais curto entre um par de nós.
- O caminho mais curto entre todos os nós do grafo.
- O caminho mais curto entre um nó origem e vários nós destino.

✓ *Algoritmo de Dijkstra*

Em 1959 Dijkstra sugeriu um algoritmo de rotulação para caminhos em grafos com arcos positivos, utilizando indução e ajuste, eficiente e de fácil implementação computacional. Chamando de:

Lista *F* (lista dos nós fechados) \equiv o conjunto dos vértices para o qual já se conhece um caminho mínimo.

Lista *A* (lista dos nós abertos) \equiv o conjunto dos nós para o qual ainda não se conhece um caminho mínimo.

t \equiv contador de iterações.

$V \equiv$ representando o conjunto dos nós rotulados e abertos em G .

$r \equiv$ índice do vértice a ser fechado na iteração t .

$C = [c_{ij}] \equiv$ matriz de custos representando as distâncias entre vértices ligados diretamente.

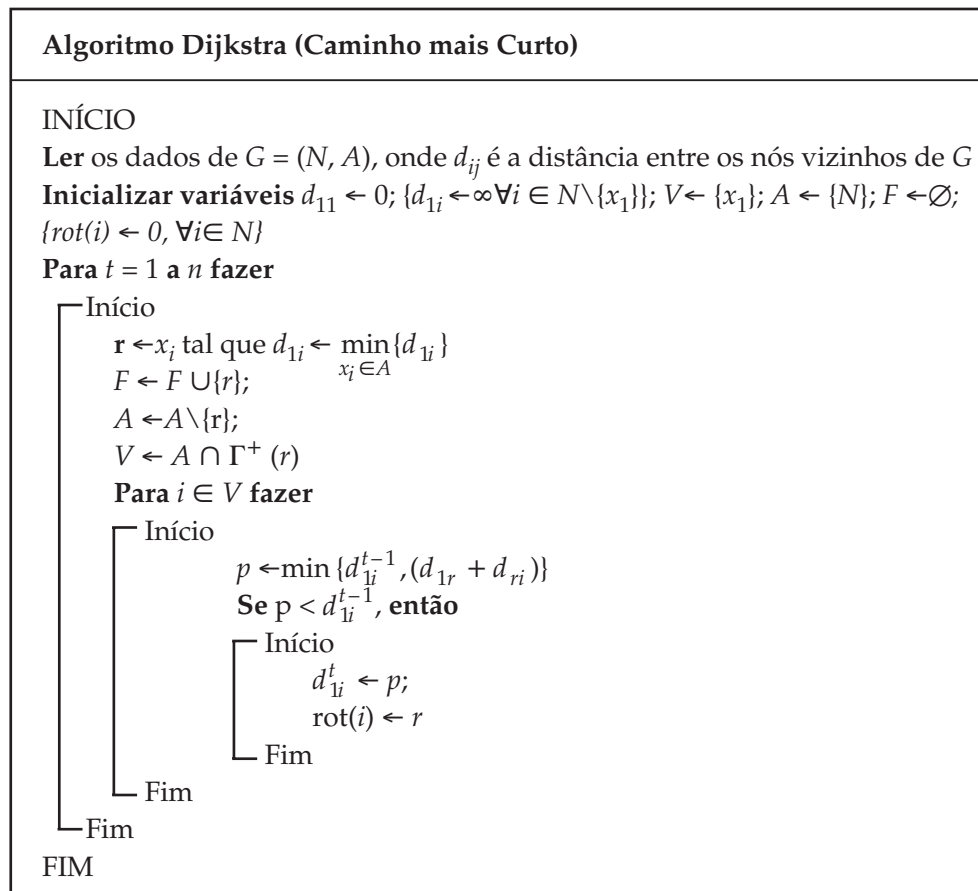
$d_{ij} \equiv$ a distância entre os vértices x_i e x_j .

$d_{ij}^t \equiv$ a distância calculada entre os vértices x_i e x_j na iteração t .

$\text{rot}(i) \equiv$ vetor que guarda o vértice que deu origem à distância calculada para o vértice de índice i .

$\Gamma^+(r) \equiv$ conjunto de vizinhos do vértice de índice r .

Podemos descrever o algoritmo como o quadro a seguir resume:



A Figura 6.4 mostra a sequência de rotulação e fechamento dos nós do algoritmo de Dijkstra quando aplicado no grafo da Figura 6.4 (a). O rótulo que está colocado na proximidade de cada vértice em sua primeira posição guarda a origem da rotulação. A segunda posição anota o valor do caminho acumulado até o vértice. A cada iteração o nó examinado que acumula a menor distância é fechado. O foco de exame da busca é sempre deslocado para o último nó fechado (na figura simbolizado pelo asterisco). A partir de um nó fechado, os vizinhos ($\Gamma^+(r)$) são examinados. O exame atualiza

obrigatoriamente a marcação ∞ que é $(\text{rot}(i))$ atribuída inicialmente a todos os nós com exceção do nó 1 e, eventualmente, quando uma distância menor é encontrada. Um exemplo dessa atualização por uma distância menor ocorre na Figura 6.4 (c) quando a distância do vértice 4 passa de 5 para 4.

A complexidade do algoritmo de Dijkstra é $O(n^2)$. Esse algoritmo não é capaz de encontrar caminhos mais curtos em presença de arcos negativos, uma vez que a cada iteração, o vértice examinado com menor distância acumulada é fechado. O algoritmo calcula as menores distâncias de um vértice inicial a todos os outros.

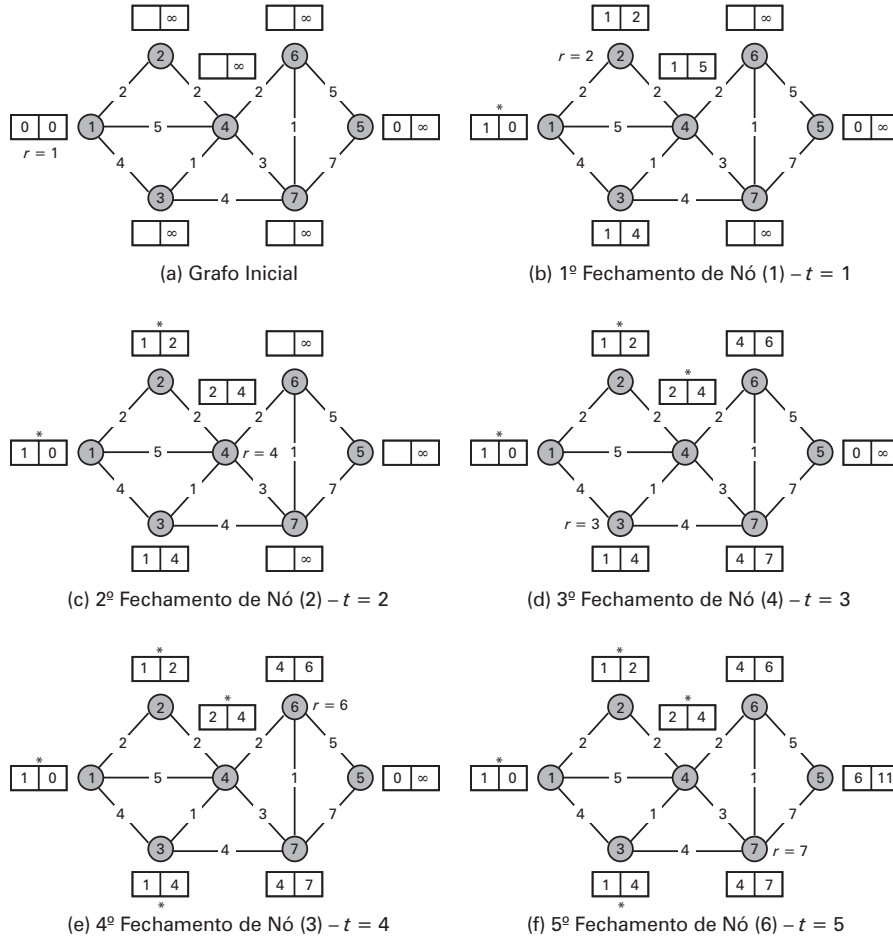


FIGURA 6.4 Exemplo da evolução do algoritmo de Dijkstra.

✓ Algoritmo de Ford-Moore-Bellman

O algoritmo de Ford-Moore-Bellman, assim denominado em homenagem ao trabalho simultâneo desses pesquisadores (ver Bellman [1958]), mas publicado em épocas diferentes, abre mão do fechamento de um nó a cada iteração, e examina todos os nós até que não seja mais possível melhorias, podendo, com isso, aceitar arestas negativas. O critério de parada está associado à não modificação de todos os rótulos em uma iteração. A ideia básica da iteração é de que, se um caminho de um vértice s para um j contém k arestas, um caminho melhor de s para j conterá, no máximo, $k + 1$ arestas.

Denominado de:

$l(s, j) \equiv$ comprimento de um caminho entre s e j .

$l_s \equiv$ comprimento do caminho associado ao nó s .

l_j^k comprimento do menor caminho P_{sj}^k usando no máximo k arcos, em que $P_{sj}^k \subset A$. O algoritmo pode ser assim descrito:

Algoritmo Ford-Moore-Bellman

INÍCIO

Ler os dados de $G = (N, A)$, onde $l(i, j)$ é o comprimento da aresta $(i-j)$ para quaisquer que sejam i, j vizinhos em G e S o nó inicial da busca.

Inicializar variáveis $k \leftarrow 1; \{l_s \leftarrow l_s^0 \leftarrow 0\} \{l_j \leftarrow 0, j = 1, \dots, n-1\}; \{l_s^1 \leftarrow l(s, j) \mid j = 1, \dots, n-1\}; \{l(i, j) \leftarrow \infty \text{ se } (i, j) \notin A\};$

Enquanto $l_j^{k-1} \neq l_j^k, \forall j \in N$ fazer

Início
 $l_j^{k+1} \leftarrow \min \{l_j^k, \min_i [l_i^k + l(i, j)]\} \mid i \neq j, s; j = 1, \dots, n-1$
 $k \leftarrow k+1$
Fim
FIM

Vamos aplicar o algoritmo no exemplo da Figura 6.5 para encontrar o caminho entre os vértices S e 5 . Inicialização:

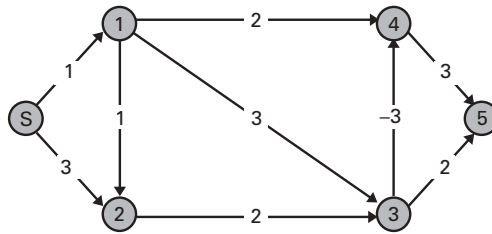


FIGURA 6.5 Grafo exemplo para o algoritmo de Ford-Moore-Bellman.

$$l_1^1 = 1; l_2^1 = 3$$

$$k = 1$$

$$l_1^2 = \min \{l_1^1, \dots\} = 1$$

$$l_2^2 = \min \{l_2^1; \lfloor l_1^1 + l(1, 2) \rfloor\} = 2 \text{ (melhorou)}$$

$$l_3^2 = \min \{l_3^1; \lfloor l_1^1 + l(1, 3); l_2^1 + l(2, 3) \rfloor\} = 4 \text{ (melhorou)}$$

$$l_4^2 = \min \{l_4^1; \lfloor l_1^1 + l(1, 4); l_3^1 + l(3, 4) \rfloor\} = 3 \text{ (melhorou)}$$

$$l_5^2 = \min \{l_5^1; \lfloor l_4^1 + l(4, 5); l_3^1 + l(3, 5) \rfloor\} = \infty$$

$$k = 2$$

$$l_1^3 = 1$$

$$l_2^3 = \min \{l_2^2; \lfloor l_1^2 + l(1, 2) \rfloor\} = 2$$

$$l_3^3 = \min \{l_3^2; \lfloor l_1^2 + l(1, 3); l_2^2 + l(2, 3) \rfloor\} = 4$$

$$l_4^3 = \min \{l_4^2; \lfloor l_1^2 + l(1, 4); l_3^2 + l(3, 4) \rfloor\} = 1 \text{ (melhorou)}$$

$$l_5^3 = \min \{l_5^2; \lfloor l_4^2 + l(4, 5); l_3^2 + l(3, 5) \rfloor\} = 6 \text{ (melhorou)}$$

$$k = 3$$

$$l_1^4 = 1$$

$$l_2^4 = \min \{l_2^3; \lfloor l_1^3 + l(1, 2) \rfloor\} = 2$$

$$l_3^4 = \min \{l_3^3; \lfloor l_1^3 + l(1, 3); l_2^3 + l(2, 3) \rfloor\} = 4$$

$$l_4^4 = \min \{l_4^3; \lfloor l_1^3 + l(1, 4); l_3^3 + l(3, 4) \rfloor\} = 1$$

$$l_5^4 = \min \{l_5^3; \lfloor l_4^3 + l(4, 5); l_3^3 + l(3, 5) \rfloor\} = 4 \text{ (melhorou)}$$

$$k = 4$$

Não haverá mais melhoria, e o algoritmo chega ao fim.

O caminho mais curto pode ser recuperado a partir do nó 5, da seguinte forma: na linha que se calculou o caminho mais curto para 5 a parcela determinante foi $l_4^3 + l(4, 5) = 4$, tornando claro que o vértice antecessor a 5 pelo caminho mais curto é 4 [veio através de $l(4, 5)$]. No cálculo da distância até 4 a parcela determinante é $l_4^3 = 1$, o que determina que o vértice 3 é o antecedente do vértice 4 uma vez que l_4^3 é obtido da expressão $l_3^2 + l(3, 4)$ [através de $l(3, 4)$], e assim por diante. O algoritmo calcula a menor distância entre todos os pares de vértices de um grupo G . A Tabela 6.1 resume alguns dos mais conhecidos algoritmos para o caminho mais curto.

TABELA 6.1 ALGORITMOS EXATOS DE SOLUÇÃO PARA O CAMINHO MAIS CURTO

Ano	Autores	Descrição	Complexidade
1959	Dijkstra	Seleciona o nó de menor potencial	$O(n^2)$
1956	Ford-Moore-Bellman	Técnica de rotulação FIFO	$O(mn)$
1962	Ford-Fulkerson	Técnica de rotulação FIFO	$O(mn)$
1962	Floyd-Warshall	Técnica da “operação tríplice”	$O(n^3)$
1967	Hu	Algoritmo matricial	$O(mn)$
1969	Dijkstra por Dial	“Buckets e FIFO”	$O(m + Cn)$
1974	Pape e Levit	Incremental sobre um conjunto restrito de nós	$\Theta(n2^n)$
1984	Pallottino	Incremental sobre um conjunto restrito de nós	$O(n^2m)$
1984	Glover & Glover – Klingman	Combinação das abordagens Dijkstra e F-M-B	$O(mn)$
1988	Klein e Reif	Algoritmos paralelos	$O(n^{2/3} \log^{7/3} n (\log n + \log D))^*$
1993	Goldberg-Radzik	Manipulação de conjuntos de rotulação	$O(nm)$
1993	Cohen	Algoritmos paralelos	$O(\log^4 n)$
1996	Träff e Zaroliagis	Algoritmos paralelos	$O((n^{2\varepsilon} + n^{1-\varepsilon}) \log n)^{**}$

* D é a soma dos pesos das arestas

** $0 < \varepsilon < 1/2$

6.4 – ÁRVORE GERADORA MÍNIMA (AGM)

Uma árvore é um grafo conexo e acíclico. Uma árvore de um grafo G é um subgrafo de $G = (N, A)$ conexo e acíclico. Uma *árvore geradora* de G é uma árvore que contém todos os nós do grafo. Uma *árvore geradora mínima* é uma árvore geradora cuja soma do comprimento de suas arestas é minimal em G . A

importância dessa estrutura é enorme tanto na prática dos problemas de otimização, como parte da solução de vários outros modelos. Suas maiores aplicações na pesquisa operacional estão associadas aos problemas de comunicação e conexões. Na computação, existem várias aplicações na área de estrutura de dados. Uma resenha dos mais recentes resultados alcançados na determinação da AGM pode ser obtida em Bazzlamaçci e Hindi (2001).

6.4.1 – Problemas Correlatos e Aplicações

As conexões em árvores ainda comportam dois modelos de otimização além da AGM, a saber:

✓ *Árvore Min-Max*

Nesse problema também chamado de otimização de “gargalo” o objetivo é determinar a árvore que possua a menor aresta máxima. O grafo da Figura 6.6 apresenta um gargalo igual a 3, uma vez que as menores arestas que incidem sobre os vértices 7 e 8 possuem esse valor. A árvore min-max é uma relaxação da AGM, uma vez que toda AGM é min-max (devido à estrutura matroide), contudo a recíproca não é verdadeira.

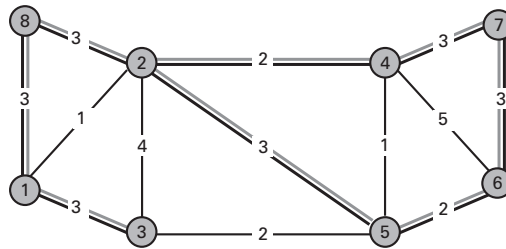


FIGURA 6.6 Exemplo de árvore geradora MinMax.

✓ *Árvore Min-Max-Min-Sum (M³S)*

O problema MinMax-Min-Sum será abordado no Capítulo 10. Dentre os problemas de otimização combinatória, dois dos mais conhecidos, dado o número de aplicações que possuem, são: os problemas de *bottleneck* ou *gargalo*, em que o objetivo é minimizar o custo máximo, dentre todas as variáveis que compõem a solução, e os problemas de custo linear, nos quais o objetivo é minimizar a soma dos custos dessas variáveis. O (M³S) é uma composição desses dois problemas, em que a função objetivo a ser minimizada é a soma algébrica das funções objetivo dos problemas mencionados.

Dados p elementos de um conjunto $E = \{e_1, e_2, \dots, e_p\}$ e $F \subset P(E)$, onde $P(E)$ é o conjunto potência de E , uma família finita de subconjuntos de E perfeitamente definidos. As instâncias do conjunto F , ou famílias de E , poderão ser, por exemplo, as subárvores de um grafo, os caminhos entre dois vértices de um grafo ou conjuntos de arestas de um *matching* máximo.

A cada elemento e_i , $i = 1, 2, \dots, p$, são associados dois números reais, um representado o custo C_i e outro o peso P_i .

Sendo $S \in F$, onde F é o conjunto das soluções viáveis de um certo problema, define-se a função:

$$Z(S) = \max_{e_i \in S} \{P_i\} + \sum_{e_i \in S} C_i$$

O problema (M³S) é da forma:

$$\min_{S \in F} Z(S)$$

Dado um grafo não direcionado $G = (N, A)$, onde $|N| = n$, e $|A| = m$, considerando uma árvore $T_k = \{u_j\}$, $j = 1, \dots, n - 1$, e T o conjunto de todas as árvores de G , podemos definir a AG-M³S como:

$$Z(t_k) = \max_{u_i \in t_k} \{P_i\} + \sum_{u_i \in t_k} C_i$$

$$\min_{t_k \in T} \{Z(t_k)\}$$

Ou ainda como (M³S-MS)

$$\min_{1 \leq k \leq p} \left\{ P_k + \min_{S \in F_i^k} \left\{ \sum_{e \in S} C_i \right\} \right\}$$

de modo que solucionar o M³S-MS pode ser equivalente a solucionar uma sequência de p subproblemas Min_Sum. O algoritmo de Minoux (1989) fundamenta-se exatamente nesse fato.

✓ Aplicações do Problema M³S-MS

O modelo M³S-MS pode representar uma série de situações em que existem dois critérios de valor associados às variáveis de decisão. Alguns autores denominam esses tipos de problemas de “com ligações balanceadas”. O modelo dito balanceado possui várias aplicações (Duin e Volgenant [1991]) práticas que abrangem:

- Roteamento de meios terrestres na defesa de costa (Moreira [1995]).
- Projeto de redes de comunicações (Punnen e Nair [1996]).
- Roteamento de veículos com função multiobjetivo (pedágio, condições de risco etc.).

✓ Árvore Geradora Mínima de Grau Restrito – (AGM-GR)

A Árvore Geradora Mínima de Grau Restrito (AGM-GR) é uma árvore geradora mínima cujos vértices possuem um determinado grau k , onde $k > 2$. Papadimitriou (1978) demonstrou que AGM-GR é NP-Completo. O trabalho de Narula e Ho (1980) apresenta um estudo pioneiro em algoritmos de solução para o problema. Várias abordagens de solução são possíveis, tais como Savelsbergh e Volgenant (1985) – Combinação de arestas, Fekete *et al.* (1997) – técnicas de fluxo em rede, Zhou e Gen (1997) e Rothlauf e Goldberg (1999) – Algoritmos genéticos, Raidl (2000) – computação evolucionária.

✓ Árvore Geradora Mínima Capacitada

No problema da Árvore Geradora Mínima Capacitada (AGM-C) uma central serve a um conjunto de vértices N , $N = \{1, \dots, n\}$, de um determinada rede $R(N, A, F)$, onde A representa o conjunto dos arcos de ligação $A = \{1, \dots, m\}$ e F o fluxo que percorre cada arco de R . Cada vértice cliente i está associado a uma demanda de fluxo $d_i > 0$. O fluxo deve se originar na central e chegar a cada cliente i . Entre cada par de nós i - j uma ligação pode ser estabelecida com o custo $C_{ij} > 0$ e segundo uma capacidade máxima $m_{ij} > 0$. O problema objetiva determinar a mais econômica que atenda a demanda exigida. Papadimitriou (1978) demonstrou que AGM-C é NP-Completo. Recentemente destacam-se os trabalhos de Domschke e Voß (1996) – busca local, Krishnamoorthy *et al.* (1996) – várias heurísticas, Deo e Kumar (1998) – algoritmos paralelos, Patterson (1999) – memória adaptativa, Raidl e Drexel (2000) – computação evolucionária, Ribeiro e Souza (2002) – Busca em vizinhança variável, Mao e Lang (2002), algoritmos paralelos e Fukasawa *et al.* (2003) – branch-and-cut-and-price.

✓ Árvore Geradora Mínima Robusta

Este problema corresponde a um modelo de otimização com incerteza. No problema da árvore Geradora Mínima Robusta (AGM-R) os custos dos arcos se encontram dentro de intervalos, em vez de serem números fixos e conhecidos. No caso da AGM-R deseja-se encontrar uma árvore geradora que minimize o máximo desvio padrão seu custo em relação a uma árvore geradora mínima que leve em conta todas as possibilidades de custos. No tema destacam-se os seguintes trabalhos: Andrade *et al.* (1999), Yaman *et al.* (2001), Aron e Van Hentenryck (2003), Montemanni e Gambardella.

✓ *Árvore Geradora de Máximo Número de Folhas*

Árvore geradora que maximiza o número de folhas. O trabalho de Galbiati *et al.* (1994) demonstra que o problema proposto presentemente é APX-Completo. Lu e Ravi (1992) apresentam algoritmos heurísticos para sua solução, e Fujie (2003) descreve um algoritmo exato para o problema.

6.4.2 – Algoritmos de Solução

Como uma AGM atende aos requisitos para ser definida dentro dos matroides (ver conceito no anexo) temos a expectativa de solucioná-la de forma bastante eficiente e, inclusive, através de algoritmos gulosos. Tarjan (1981) apresenta um método geral que inclui todos os métodos baseados em abordagens míopes. Nesse sentido, existem três algoritmos clássicos que permitem uma rápida e eficiente solução para a AGM, sendo eles os algoritmos de Kruskal (1956), de Prim (1957) e de Borůvka (Wai-Kai Chen, 1990). O algoritmo de Kruskal e o de Prim podem ser considerados dois casos particulares da abordagem gulosa. No caso de Prim a árvore geradora é construída a partir de uma aresta pelo acréscimo de novas arestas, aumentando-se a arborescência inicial até que todos os nós sejam incluídos. No caso de Kruskal podem ser desenvolvidas várias arborescências simultâneas até que uma só árvore inclua todos os nós. Em qualquer hipótese, o critério de inclusão de arestas nas arborescências é guloso. Descreveremos os dois algoritmos citados e uma variação para o Prim nos próximos itens.

✓ *Algoritmo Prim*

Considerando os conjuntos S , T e V , onde $S \subseteq A$, $T \subseteq N$, $V \subseteq N$, podemos escrever o algoritmo Prim da seguinte forma:

Algoritmo Prim (Árvore Geradora Mínima)	
INÍCIO Ler $G = (N, A)$ e $D = [d_{ij}]$ a matriz distância entre os nós vizinhos de G . Escolha qualquer vértice $i \in N$ $T \leftarrow \{i\}$ $V \leftarrow N \setminus \{i\}$ Enquanto $T \neq N$ Para todo $j \in T$, Faça : Início Encontrar a menor aresta $(j, k) \in A$ tal que $j \in T, k \in V$ $T \leftarrow T \cup \{k\}$ $V \leftarrow V \setminus \{k\}$ $S \leftarrow S \cup (j, k)$ Fim Escrever S {arestas da árvore geradora mínima} FIM	

A versão do algoritmo Prim exibida é $O(n^3)$, uma vez que o comando de encontrar a menor aresta $(j, k) \in A$ exige até $O(m)$ operações e m é $O(n^2)$. Tomando um cuidado especial para não obrigar o algoritmo a computar todas as arestas incidentes sobre a árvore que vai sendo expandida, é possível baixar a complexidade de Prim. Chamaremos a esse algoritmo refinado de Prim_Colorido, sendo formalizado no quadro de mesmo título. Vamos aplicar as duas versões do algoritmo Prim no grafo da Figura 6.7. O algoritmo Prim_Colorido é $O(n^2)$. Denominando de:

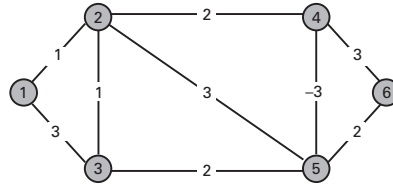


FIGURA 6.7 Grafo exemplo para as versões Prim.

$T_s \equiv$ uma árvore em G .

$l(i, j) \equiv$ a aresta de mínimo comprimento incidente sobre o vértice j , $i, j \in T_s$.

Podemos escrever Prim_Colorido da seguinte forma:

Algoritmo Prim_Colorido (Árvore Geradora Mínima)

INÍCIO

Ler $G = (N, A)$ e $D = [d_{ij}]$ a matriz distância entre os nós vizinhos de G .

escolha qualquer vértice $j \in N$;

$i \leftarrow 0$;

$T_i \leftarrow \{j\}$

Colorir com verde a menor aresta incidente a cada vértice j extremo de T_0

Enquanto $i < n - 1$ **Fazer**

Início

Selecionar a menor aresta verde $(j, k) \in A, j \in T_i$ e colori-la de azul

Faça $T_{i+1} \leftarrow T_i \cup (j, k)$

Para cada aresta $(k, z), z \notin T_{i+1}, k \in T_{i+1}$

Início

Se não existe aresta verde incidente em z , **colorir** (k, z) com verde

Caso Contrário

Início

Se existe aresta $(w, z), w \in T_{i+1}$, tal que $l(w, z) > l(k, z)$ colorir com vermelho (w, z) e verde (k, z)

Caso Contrário não faça nada;

Fim

Fim

$i \leftarrow i + 1$

Fim

Escrever T_i {arestas da árvore geradora mínima}

FIM

Escolhendo-se o vértice 1 para iniciar a árvore, a Figura 6.8 mostra, da esquerda para direita e de cima para baixo, a sequência de inclusão de arestas que é gerada pela versão clássica do Prim.

A Figura 6.9 mostra as escolhas e marcações do algoritmo Prim_Colorido (V representa arestas *verdes*, as arestas *ressaltadas* são as que o algoritmo denomina por *azuis*, e as que aparecem marcadas com um X, representam as *vermelhas*) no caso da inicialização ser realizada com o nó número 1.

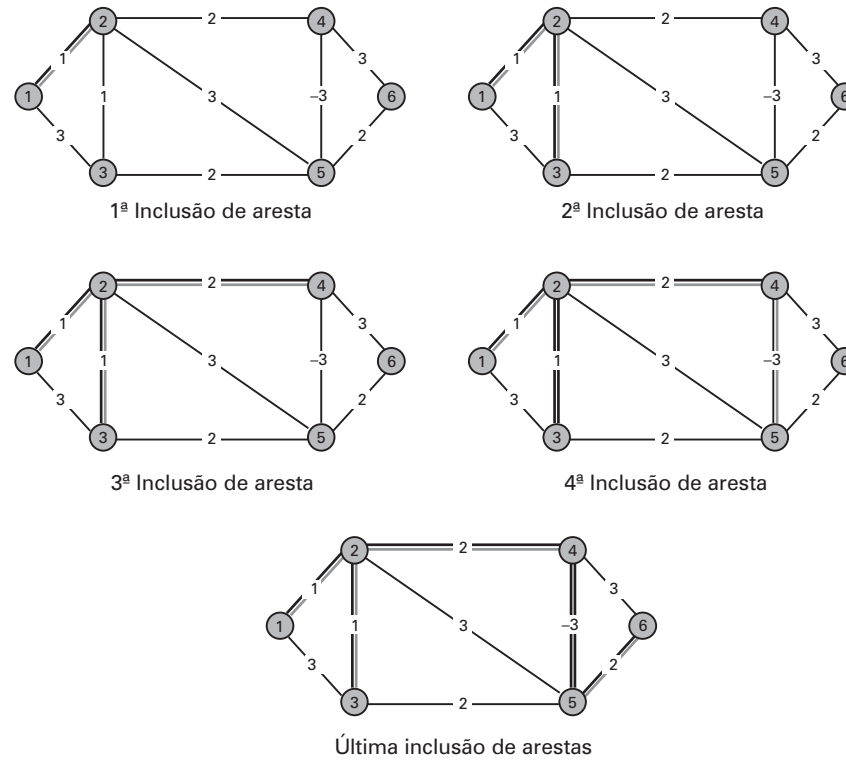


FIGURA 6.8 Exemplo da evolução do algoritmo de Prim.

✓ Algoritmo Kruskal

O segundo algoritmo clássico para a AGM é o de Kruskal, que pode ser formalizado como:

Algoritmo Kruskal (Árvore Geradora Mínima)

INÍCIO

Ler $G = (N, A)$ e $D = [d_{ij}]$ a matriz distância entre os nós vizinhos de G .

Ordene as arestas de G (conjunto A) em ordem não crescente das distâncias d_{ij} no vetor $H = [h_i], i = 1, 2, \dots, m$

$T \leftarrow h_1$

$i \leftarrow 2$

Enquanto $|T| < n$ **Tome** $h_i \in H$ e **Faça**

Início

Se $T \cup h_i$ é um grafo acíclico (árvore) **então**

Início

$T \leftarrow T \cup h_i$

$i \leftarrow i + 1$

Fim

Caso Contrário

$i \leftarrow i + 1$

Fim

Escrever T {arestas da árvore geradora mínima}

FIM

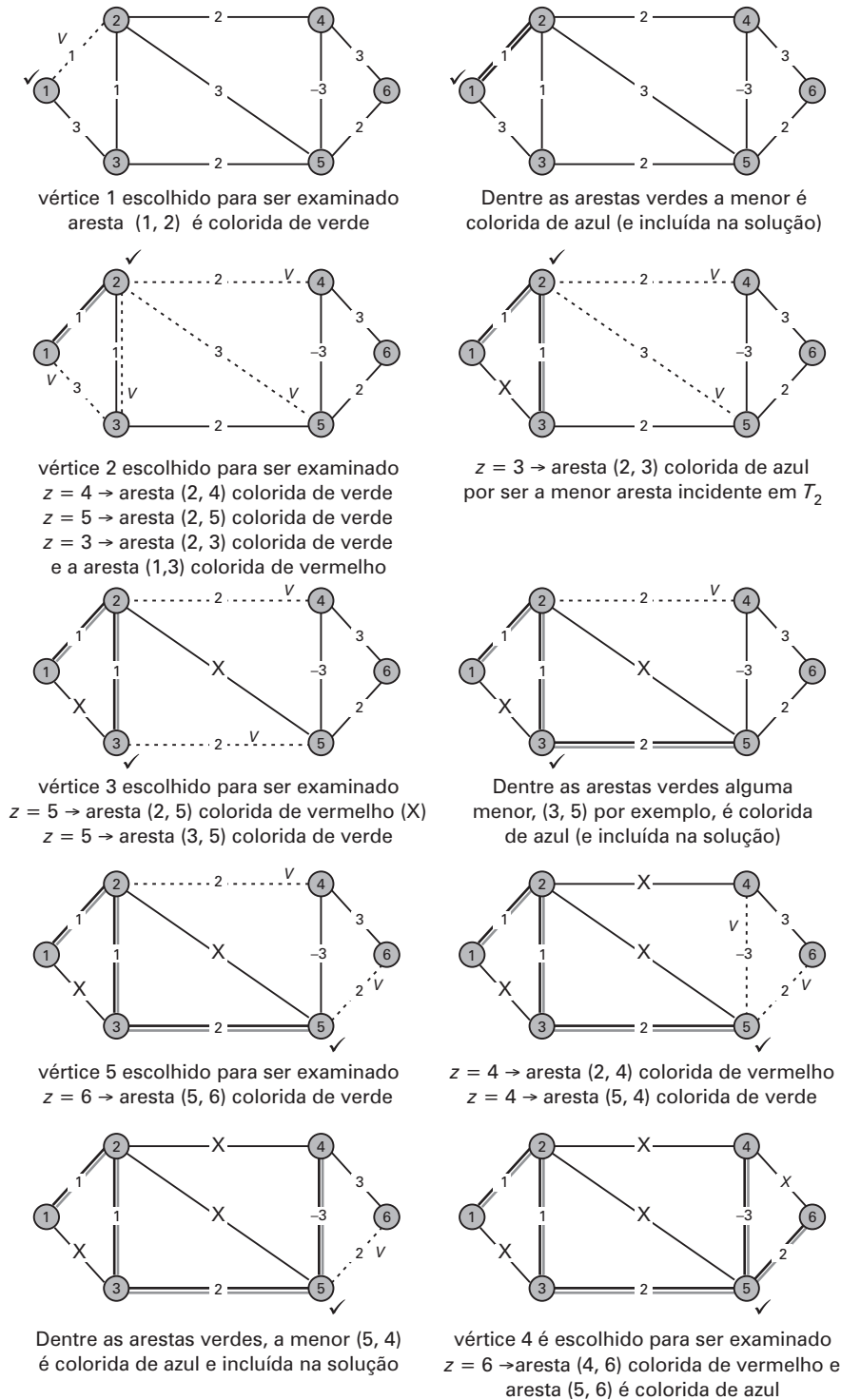


FIGURA 6.9 Exemplo da evolução do algoritmo Prim_Corolado.

A sequência de inclusão de arestas do algoritmo de Kruskal é o vetor $H = \{(4, 5); (1, 2); (2, 3); (2, 4); (3, 5); (5, 6); (1, 3); (2, 5); (4, 6)\}$. A Figura 6.10 mostra a execução do algoritmo.

A complexidade do procedimento de Kruskal é dominada pela etapa de ordenação das arestas, uma vez que a escolha das arestas será realizada em $O(m)$ preparações, e a verificação da formação de um ciclo pelo acréscimo de uma aresta em um grafo acíclico é também $O(n)$. Esse procedimento, se as arestas não possuem qualquer propriedade específica, é $O(m \log n)$.

✓ Algoritmo de Borůvka

Finalmente o algoritmo de Borůvka abre uma outra linha de abordagem para o problema, desenvolvendo simultaneamente uma floresta minimal no grafo G . Chamando por T_j uma subárvore de G , e por F_i uma floresta em G , podemos formalizar o algoritmo da seguinte forma:

Algoritmo Borůvka (Árvore Geradora Mínima)

INÍCIO

Ler $G = (N, A)$ e $D = [d_{ij}]$ a matriz distância entre os nós vizinhos de G .

F_0 é uma floresta inicial com n subárvores $T_j, j = 1, \dots, n$ de G (todos os nós isolados)

$i \leftarrow 0$

Enquanto F_i não for uma árvore **Faça** para cada $T_j \in F_i$

 Início

Determine a menor aresta (x_α, y_α) incidente em T_j onde

$x_\alpha \in T_j$ e $y_\alpha \notin T_j$

Faça $F_{i+1} \leftarrow F_i \cup \left[\bigcup_{\alpha} (x_\alpha, y_\alpha) \right]$

$i \leftarrow i + 1$

 Fim

Escrever F_i {arestas da árvore geradora mínima}

FIM

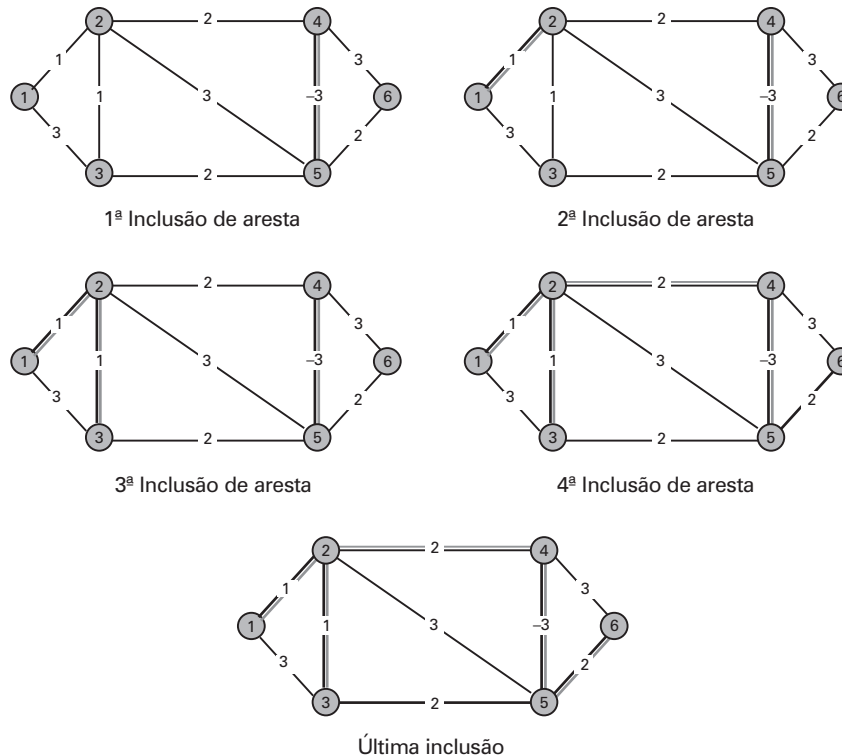


FIGURA 6.10 Exemplo da evolução do algoritmo de Kruskal

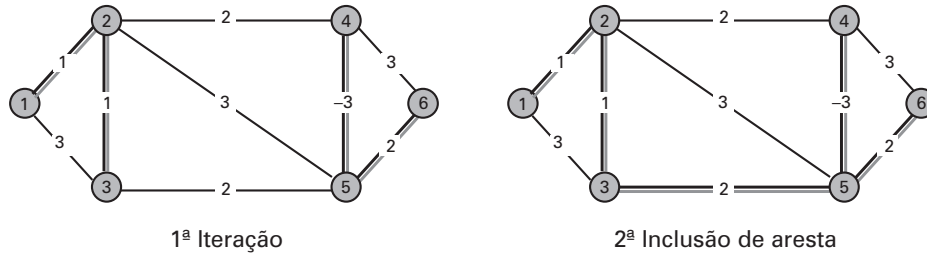


FIGURA 6.11 Exemplo da evolução do algoritmo de Borůvka.

O algoritmo, da forma descrita anteriormente, só funciona corretamente se as arestas de G tiverem custos distintos. Para evitar que possam ocorrer inclusões indevidas por ocasião das uniões das subárvores é indispensável ordenar lexicograficamente as arestas de G antes de aplicar o algoritmo. Nesse caso a complexidade do procedimento será $O(n \log n)$.

Salehi-Fathabadi e Ahrabian (1995) sugerem um algoritmo que em certos casos pode funcionar em $O(m)$.

✓ O Algoritmo de Minoux para o problema M^3S

O algoritmo de Minoux (1989) consiste basicamente em buscar balancear os subproblemas MinMax e Min-Sum denotado respectivamente por:

$$(MM) \quad \min_{S \in F} \left\{ \max_{e_i \in S} \{P_i\} \right\}$$

$$(MS) \quad \min_{S \in F} \left\{ \sum_{e_i \in S} C_i \right\}$$

Minoux observa que, para qualquer $S \in F$, existem p valores distintos para $\max_{e_i \in S} \{P_i\}$, e que para cada P_i existe associada uma solução Min_Sum. O algoritmo polinomial proposto por Minoux fundamenta-se nos seguintes pontos:

- Existe um algoritmo eficiente que resolve o problema Min_Sum.
- $\max_{e_i \in S} \{P_i\}$ para qualquer $S \in F$ pode assumir no máximo $p = |E|$ valores distintos.
- Existe um limiar (*threshold*) P que supostamente está associado à solução ótima de $\max_{e_i \in S} \{P_i\}$.

A estratégia geral do algoritmo de Minoux é examinar as p soluções ótimas do Min_Sum associadas aos valores de cada um dos valores de $P = \{P_1, P_2, \dots, P_p\}$, $P_i \in P$. Nesse caso, a complexidade do algoritmo será $O(p \cdot O(MS))$, representando p vezes o exame das soluções Min_Sum.

Goldbarg e Gouvêa (1997) sugerem algoritmos heurísticos de bom desempenho para problemas M^3S -MS de grande porte, um deles, denominado *Busca*, está transcrito no quadro respectivo. Considerando $Cl_r = \{u_i \dots u_j \dots u_i\}$ um ciclo r em G , temos:

Algoritmo M³S Busca

INÍCIO

Ler os dados de $G = (N, A)$, onde d_e é o custo da aresta $e \in A$, e P_e o seu peso.

Procedimento Busca (u_i)

Marcar o vértice u_i

Enquanto houver aresta (u_i, u_j) não examinada faça

Seja e a aresta (u_i, u_j)

Se u_j não é marcado **então**

Se $P_e > P_{cor}$, **então**

$P_{cor} \leftarrow P_e$

$Z \leftarrow Z - P_e + P_{cor} + d_e$

Caso Contrário

Se $P_e = P_{cor}$ **então** $Z \leftarrow Z + d_e$

Caso Contrário Ciclo (u_i, u_j)

Procedimento Ciclo (u_i, u_j)

Para toda aresta $e' \in Cl_r$, calcule a contribuição de e' em Z

Retire de Cl_r a aresta de maior contribuição calculada.

Em caso de empate escolha a aresta de maior peso.

Se $P_e = P_{cor}$

Se e' é a única aresta que possui o limite máximo na árvore **então**

Encontrar o novo limite máximo

$T \leftarrow 0, Z \leftarrow 0;$

FIM

O procedimento *Ciclo* (u_i, u_j) retira do ciclo formado em $O(n)$ a aresta de maior contribuição dentro do ciclo. Caso a aresta a ser retirada tenha peso igual ao peso corrente e seja a única com essa propriedade, um novo peso corrente deverá ser encontrado. Esse passo de busca pode ser realizado em $O(n)$ sem o auxílio de qualquer estrutura de dados especial. O procedimento *Ciclo* (u_i, u_j) será chamado $m-n+1$ vezes. Portanto, o procedimento *Busca* (u_i) terá complexidade $O(mn)$.

Punnen e Nair (1996) apresentam um algoritmo $O(m \log n)$ para o problema. A dificuldade do algoritmo é a necessidade do uso de uma estrutura de dados especial para sua implementação.

6.5 – O PROBLEMA DE EMPARELHAMENTO (PE)

Este problema está associado a vários modelos de otimização. Nos próximos capítulos verificaremos sua ligação com outros problemas em teoria dos grafos. No Capítulo 7 mostraremos a possibilidade de sua solução através de algoritmos especializados na determinação de fluxos em redes. Nesse ponto, vamos examinar sua relação com os problemas de conexão. De fato, um *matching* ou emparelhamento nada mais é que uma forma de reunião ou ligação entre dois elementos ou, no caso dos grafos, dois vértices. O chamado 1-*Matching*, ou o problema de emparelhamento, em que a união válida é realizada apenas por uma aresta, é extremamente útil na determinação de agrupamentos de afinidades e designações de associações.

6.5.1 – Uma Visão Geral

A literatura que aborda a solução do problema de emparelhamento é rica. Alguns dos mais importantes trabalhos nessa área estão resumidos na Tabela 6.2:

TABELA 6.2 ALGORITMOS EXATOS DE SOLUÇÃO PARA O PE

<i>Ano</i>	<i>Pesquisador</i>	<i>Algoritmos Exatos</i>
1965	Edmonds	$O(n^4)$
1973	Hopcroft e Karp	$O(n^{2.5})$
1974	Kameda e Munro	$O(mn)$
1975	Even e Kariv	$O(n^{2.5})$
1975	Even e Tarjan	$O(n^{2.5})$ – grafos bipartidos
1976	Gabow	$O(n^3)$
1976	Bondy e Murty	$O(n^3)$
1978	Mengido e Tamir	$O(n \log n)$: – certas classes de grafos
1980	Micali e Vazirani	$O(mn^{1/2})$
1982	Papadimitriou e Steiglitz	$O(mn^{1/2})$ – grafos bipartidos
1982	Papadimitriou e Steiglitz	$O(n^4)$ e $O(n^3)$
1988	Derigs	$O(mn^{1/2})$

Apesar da existência de algoritmos exatos em ordem polinomial para a solução do problema, algumas instâncias podem possuir da ordem de 10^6 ou 10^7 pontos. Com um algoritmo semelhante ao de Derigs seriam necessárias da ordem de 10^{16} operações para uma solução exata! Nesse caso, justifica-se a busca de algoritmos aproximativos mais eficientes. As técnicas usadas nessas heurísticas são:

- Curvas de Sierpinski e Hilbert (Bertholdi e Platzman [1982]).
- Curvas de Hilbert (Iri *et al.* [1983]).
- Enquadramento Espiral (Iri *et al.* [1983]).

A Tabela 6.3 resume alguns desses esforços:

TABELA 6.3 ALGORITMOS HEURÍSTICOS DE SOLUÇÃO PARA O PE

<i>Ano</i>	<i>Pequisador</i>	<i>Algoritmo Heurístico</i>
1981	Iri <i>et al.</i>	$O(n)$
1982	Bertholdi e Platzman	$O(n \log n)$
1983	Avis	(vários)
1983	Iri <i>et al.</i>	$O(n \log n)$

6.5.2 – Problemas Correlatos e Aplicações

✓ O Problema do Emparelhamento Estável

Em relação ao emparelhamento o mais conhecido problema correlato é o denominado *stable matching problema* ou problema do emparelhamento estável. Esse problema visa a determinar uma associação entre agentes em que cada um possui uma lista de prioridades independente. A solução desse problema consiste em obter um conjunto de associações (emparelhamento ou “casamento”) tais que não exista um par de agentes unidos em que ambos prefeririam outros parceiros para a associação. Seja a lista da Tabela 6.4

TABELA 6.4 LISTA DE PRIORIDADES DO EMPARELHAMENTO ESTÁVEL

<i>Pessoa</i>	<i>Lista de Preferência</i>
1	2 – 3 – 4 – 5 – 6
2	3 – 4 – 5 – 6 – 1
3	4 – 5 – 6 – 1 – 2
4	5 – 6 – 1 – 2 – 3
5	6 – 1 – 2 – 3 – 4
6	1 – 2 – 3 – 4 – 5

A associação 1-2; 3-4; 5-6; os indivíduos 1, 3 e 5 estão associados a seu parceiro preferido, atendendo a definição de emparelhamentos estáveis. A associação 3-6, por exemplo, não é estável.

Uma outra situação para o problema do emparelhamento estável é admitir compromisso. Entendemos como situação de compromisso uma exigência estrita de um parceiro. Nesse caso, o emparelhamento estável é definido como aquele em que não existam dois indivíduos que violem a exigência de compromisso em suas associações.

Dentre os problemas dessa classe destacamos:

✓ Problema do Casamento

Nesse caso, o problema associa H homens e M mulheres em um emparelhamento estável, ou seja, aquele em que cada um dos parceiros atende à maior prioridade de seu cônjuge ou, no segundo caso, que ambos são a maior prioridade do outro. Obviamente, para qualquer instância do problema do casamento não estrito existe pelo menos um casamento estável. Esse casamento pode ser obtido em $O(n^2)$, onde n representa o número de homens ou de mulheres (pressuposto da possibilidade de solução monogâmica viável). Gale e Shapley (1962) solucionam esse problema para o caso do número de homens ser igual ao de mulheres. Knut (1976) analisa o problema e os algoritmos de solução.

✓ Problema da Escolha do Comparte

Este problema envolve um conjunto par de n elementos, cada um com uma lista de preferências com $n - 1$ agentes. O emparelhamento estável organiza o grupo em um conjunto de $n/2$ pares em que nunca dois elementos já associados em duplas estranhas tenham mútua preferência. Esse problema com exigências restritas é *NP-Completo*. A Tabela 6.5 mostra uma situação em que não é possível encontrar um amigo para o indivíduo 4. Seja a solução 1-2 e 3-4. Nesse caso, 3 prefere 2 e 2 prefere 3 em relação a seus próprios pares. Em todas as demais situações, o mesmo acontecerá impedindo a existência de uma solução estável nessa instância (exemplo em Ronn [1990]).

TABELA 6.5 LISTA DE PRIORIDADES PARA A ESCOLHA DO COMPORTE

<i>Pessoa</i>	<i>Lista de Preferência</i>
1	2 – 3 – 4
2	3 – 1 – 4
3	1 – 2 – 4
4	qualquer um

✓ Problema de Alocação Interna

Trata-se de uma versão generalizada do problema do casamento, ou seja, uma versão poligâmica em que os indivíduos podem ser associados a mais de um outro indivíduo. Um caso prático de aplicação dessa forma de distribuição ocorre na distribuição de médicos internos em vários hospitais. Cada hospital possui uma certa quantidade de posições a serem preenchidas com certa prioridade associada. Por outro lado, os médicos possuem preferências e prioridades em relação aos hospitais.

Uma alocação é considerada não estável nesse caso quando:

1. O médico ou o hospital preferem não ser alocados a participarem de uma associação.
2. Tanto o hospital como o médico possuem uma outra pretensão com maior prioridade.

Roth (1984) define essa aplicação e apresenta um algoritmo de solução para o problema.

✓ Problema de Alocação Interna com Casais

O problema anteriormente descrito sofre uma transformação profunda se for permitido que os médicos possam se juntar em duplas e unir suas listas de pretensões. Esse caso corresponderá à possibilidade de os médicos internos serem casados e desejarem ser alocados em regiões geograficamente próximas. Esse problema é *NP-Árduo* (Ronn ([1990])).

✓ Problema do Emparelhamento com Gargalo

Trata-se da versão MinMax do emparelhamento. O interessante nesse caso do emparelhamento é a existência de um algoritmo bastante eficiente e de fácil implementação com complexidade $O((n \log n)^{1/2} m)$ (Gabow e Tarjan [1988]).

6.5.3 – Abordagens de Solução

Existem algumas abordagens clássicas de solução para esse problema, a saber:

• Abordagem de Edmonds – Solução Construtiva Baseada em Enumeração de Caminhos

Essa abordagem está associada ao desenvolvimento de caminhos dentro do grafo que permitam aumentar o número de emparelhamentos obtidos. Na linha dos algoritmos de Edmonds alguns conceitos são fundamentais:

✓ Caminho Alternante

Se desenvolvermos um caminho qualquer iniciado no vértice 1 podemos construir um emparelhamento entre os vértices do caminho os juntando dois a dois. Para isso, podemos imaginar um processo de rotulação sobre os nós, alternado rótulos (+ e –, por exemplo). Um emparelhamento válido unirá vértices de sinais contrários. Ao caminho de i para j assim rotulado, denominamos caminho alternante. A Figura 6.12 mostra o processo de emparelhamento utilizando um caminho qualquer em G .

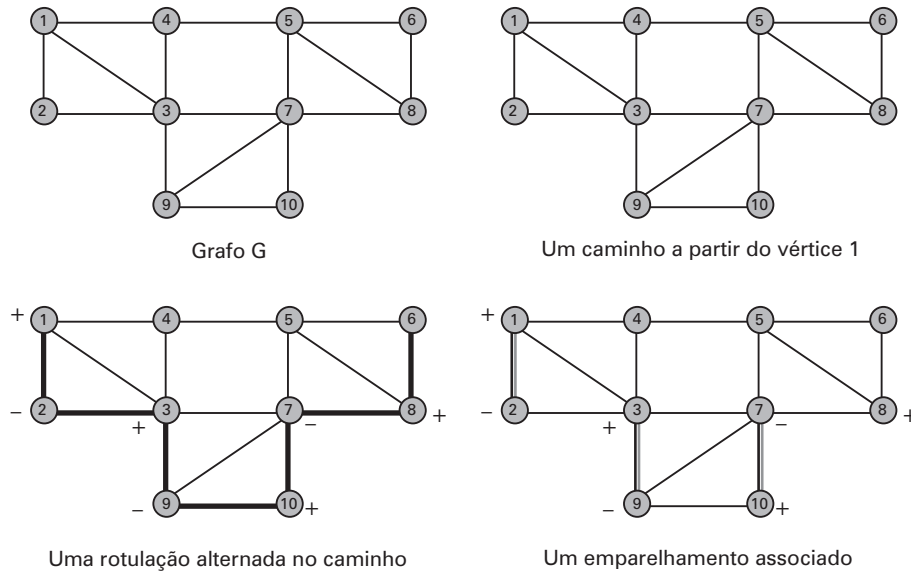


FIGURA 6.12 Um emparelhamento a partir de um caminho em G .

Em um grafo G , no qual estamos considerando um emparelhamento de vértices $M \subseteq A$, um caminho é dito M -alternante se suas arestas pertencem alternadamente a M e a $A \setminus M$. No caso da Figura 6.12, $M = \{(1, 2); (3, 9); (7, 10)\}$.

✓ Caminho Aumentante

Se no exame de um grafo que já foi atribuído um emparelhamento (não ótimo – não máximo), se obtivermos um caminho que ligue dois nós desemparelhados (não saturados – de sinais contrários), então poderemos aumentar o número de emparelhamentos associados ao caminho alternante apenas pelo deslocamento do esquema de emparelhamento (de positivo para negativo ou vice-versa).

A Figura 6.13 exemplifica um caminho aumentante e mostra o seu efeito no crescimento do número de nós emparelhados em G .

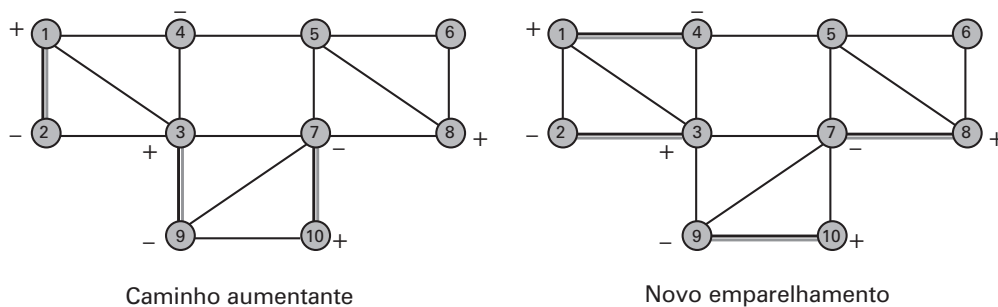


FIGURA 6.13 Caminho Aumentante.

Imediatamente podemos retirar das estruturas anteriormente definidas uma condição necessária e suficiente para que um emparelhamento M seja maximal: um emparelhamento M em G é maximal se e somente se não existir um caminho aumentante em G . Os possíveis caminhos aumentantes podem ser pesquisados de modo sistemático em G , através de uma árvore enraizada em um certo nó. A essa árvore denominaremos árvore alternante. A Figura 6.14 mostra o uso da árvore alternante para uma busca de caminhos aumentantes, sendo enraizada no vértice 5 do grafo da Figura 6.12. Formalmente, podemos definir uma árvore alternante $T_a = (V, L)$ em relação a um emparelhamento M como aquela que:

1. Contém um único vértice isolado x_1 (ou raiz de T_a).
2. Todo caminho que une $x_1 \in V$ a qualquer outro vértice $x_j \in V$ é um caminho alternante;
3. Contém todas as arestas de M incidentes em vértices de V .

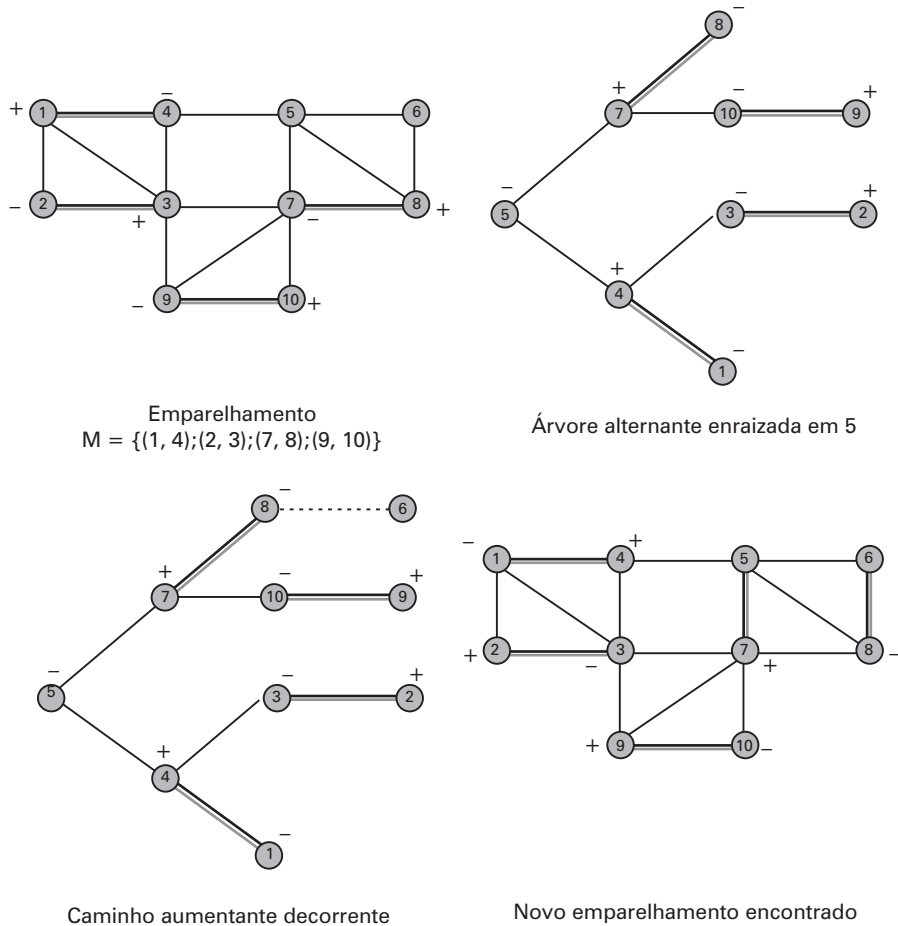


FIGURA 6.14 Árvore alternante enraizada em 5.

Edmonds verificou que os problemas de permutações na rotulação dos caminhos estavam associados à enumeração de ciclos de comprimento ímpar em G com vértices não emparelhados (chamados não saturados). A esses conjuntos de vértices pertencentes a um ciclo de comprimento ímpar na árvore de enumeração, Edmonds denominou de *blossoms* (alguns autores traduzem o termo por *broto*). Os vértices 5, 6 e 8 formam um *blossom* de comprimento 3. Os *blossoms* permitem que uma aresta possa ser enumerada mais de uma vez em um caminho aumentante, trazendo problemas para o desdobramento da árvore alternante. Para evitar esse problema, e baseado em um teorema (Papadimitriou e Steiglitz [1982]), os *blossoms* devem ser identificados e contraídos a um nó, após seus nós serem emparelhados.

A abordagem de Edmonds permite o desenvolvimento de uma série de algoritmos que exploram a possibilidade de limitar a enumeração dos caminhos aumentantes, identificar eficazmente os *blossoms*, etc.

Hopcroft e Karp trabalham exatamente em casos em que o caminho aumentante ocorre sobre um caminho mais curto. Nesse caso a complexidade do emparelhamento mínimo pode ser reduzida para $O(n^{2.5})$.

• Abordagem por Fluxo em Redes

Será desenvolvida no Capítulo 7, item 7.4.3.

6.6 – O PROBLEMA DA ÁRVORE DE STEINER

O problema de Steiner em grafos é derivado do bem conhecido problema euclidiano de Steiner, proposto na verdade por Fermat no século XVII. O problema em sua versão simplificada consistia em, dados três pontos no plano (daí o termo euclidiano), encontrar um quarto ponto tal que a soma das distâncias desse quarto ponto aos três originais fosse mínima. Atribui-se a Torricelli (por volta de 1640) a solução do problema em sua primeira versão. Apesar de não ter sido Steiner o formulador do problema (ou mesmo seu solucionador) acabou sendo responsável por sua divulgação no meio científico, o que lhe valeu o nome no problema. O caso geral consiste em determinar a minimização do peso de ligação entre n pontos do plano. Na verdade, a solução do problema é constituída por uma árvore que engloba os pontos a serem ligados e os pontos de passagem que serão determinados (ou pontos de Steiner). O problema de Steiner em grafos não direcionados é o problema de conectar, a custo mínimo, um conjunto de nós de G . Para caracterizar o problema denominaremos Λ e X os conjuntos resultantes de uma partição do conjunto N , ou seja, $\Lambda \subseteq N$ e $X \subseteq N$, sendo que $\Lambda \cup X = N$ e $\Lambda \cap X = \emptyset$, $|X| = r$, $|\Lambda| = s$. O problema será determinar um subconjunto de arestas $B \subseteq A$ que ligue todos os vértices $x_i \in X$ em um grafo conexo, $i = 1, \dots, r$, de modo a minimizar $\sum_{j \in B} c_j$, onde c_j é o custo ou comprimento das arestas de B . O

problema não exige que somente vértices de X sejam utilizados pela estrutura de ligação, ou que o número das arestas de B seja mínimo, a única restrição é a conexidade da solução. Os vértices de Λ que forem utilizados na solução são denominados *nós de Steiner*. Na Figura 6.15 temos $\Lambda = \{2, 4, 5, 6, 8, 9, 11, 12\}$ e $X = \{1, 3, 7, 10\}$.

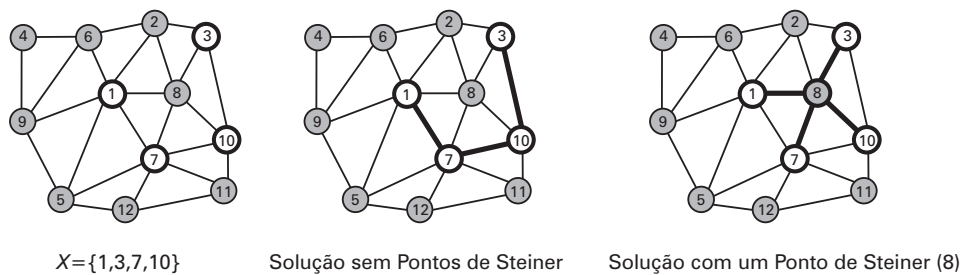


FIGURA 6.15 Árvore de Steiner em grafos.

Examinando a Figura 6.15 verificamos que se $|X| = 2$, o problema se reduz ao caminho mais curto entre os dois nós de X . Se $\Lambda = \{\emptyset\}$, a solução é uma árvore geradora mínima, uma vez que, nesse caso, $X = N$. Apesar de, como vimos no presente capítulo, existirem algoritmos eficientes para solucionar esses dois casos específicos, a situação em que $\Lambda \neq \emptyset$ e $|X| > 2$ conduz a um problema NP-árduo como mostra Karp (1972). Um trabalho bastante completo sobre as formulações desse problema é encontrado em Goemans e Myung (1993).

6.6.1 – Uma Visão Geral

A literatura que aborda a solução do problema de Steiner em grafos é numerosa. Alguns dos mais importantes trabalhos na área exata estão resumidos na Tabela 6.6:

TABELA 6.6 ALGORITMOS EXATOS DE SOLUÇÃO PARA O PROBLEMA DE STEINER EM GRAFOS

Ano	Pesquisador	Algoritmos Exatos
		Algoritmo de Enumeração baseado na Árvore Geradora (STEA)
1971	Hakimi	Primeira formulação e apresentação do STEA
1976	Lawer	Modificações no STEA
1985	Balakrishnan e Patel	Adição de nós artificiais e solução da árvore geradora por fluxo

TABELA 6.6 CONTINUAÇÃO

<i>Ano</i>	<i>Pesquisador</i>	<i>Algoritmos Exatos</i>
		Algoritmo de Enumeração Topológica (TEA)
1971	Hakimi	Modificação de um algoritmo para o problema de Steiner euclideano
		Algoritmos de Programação Dinâmica
1971	Dreyfus e Wagner	Desenvolvimento de subárvores de Steiner
		Algoritmos Baseados no Problema de Recobrimento
1980	Aneja	Formulação via PR, solução via algoritmo de Bellmore e Ratliff (1971)
		Algoritmos Baseados em Enumeração Implícita
1982	Shore <i>et al.</i>	Uso de limites inferiores e superiores para limitar a enumeração
		Algoritmos Baseados em Relaxação Lagrangeana
1984	Beasley <i>et al.</i>	Limites obtidos da RL para reduzir o número de subárvores de Steiner
		Algoritmos Duais (Dual Crescente)
1984	Wong	Limites obtidos de formulação dual para reduzir a enumeração
		Outras Abordagens
1989	Duin e Volgenant	Técnicas de Redução
1992	Chopra <i>et al.</i>	"Branch-and-cut"
1998	Koch e Martin	"Branch-and-cut"
1998	Lucena e Beasley	"Branch-and-cut"
2001a	Polzin e Daneshmand	Técnicas de redução

O trabalho de Hwang e Richards (1992) apresenta uma revisão bibliográfica bastante atualizada. O problema é também rico em abordagens heurísticas, como mostra a Tabela 6.7.

TABELA 6.7 ALGORITMOS HEURÍSTICOS PARA A SOLUÇÃO DO PROBLEMA DE STEINER EM GRAFOS

<i>Ano</i>	<i>Pesquisador</i>	<i>Algoritmos Heurísticos</i>
1980	Takahashi e Matsuyama	Heurísticas Baseadas no Caminho mais Curto $O(r.n^2)$
1980	Takahashi e Matsuyama	Heurísticas Baseadas na Árvore Geradora Mínima $O(n^2)$
1980	Aneja	Heurísticas Baseadas no Problema de Recobrimento $O(n^3)$
1981	Kou <i>et al.</i>	Heurísticas Baseadas no Grafo de Distâncias Induzido $O(r.n^2)$
1981	Plesnik	Heurísticas Baseadas no Grafo de Distâncias Induzido $O((n-r)^q \cdot r^2 + n^3)$ onde $0 \leq q \leq r-2$
1981	Plesnik	Heurísticas Baseadas em Contração de Nós $O(n^3)$
1981	Smith	Heurísticas para o Problema em Norma Euclidiana $O(n \log n)$
1983	Rayward-Smith	Heurísticas Baseadas em Distâncias Médias $O(n^3)$
1984	Wong	Heurísticas Baseadas na Relaxação do Método Dual Crescente
1991	Diané e Plesník	Heurísticas híbrida $O(n^3)$

TABELA 6.7 CONTINUAÇÃO

Ano	Pesquisador	Algoritmos Heurísticos
		Heurísticas Baseadas em Árvores k-restritas
1993	Zelikovsky	Razão de Aproximação de 1,834
1994	Berman e Ramaiyer	Razão de Aproximação de 1,734
1996	Zelikovsky	Razão de Aproximação de 1,694
1997	Promel e Steger	Razão de Aproximação de 1,667
1997	Karpinsky e Zelikovsky	Razão de Aproximação de 1,644
1999	Hougardy e Promel	Razão de Aproximação de 1,598
2000	Robins e Zelikovsky	Razão de Aproximação de 1,550
1989	Hesser <i>et al.</i>	Algoritmos Genéticos
1991	Dowsland	Hill-climbing e Simulated Annealing
1993	Williamson <i>et al.</i>	Método primal-dual
1995	Esbensen	Algoritmos Genéticos
1995	Agrawal <i>et al.</i>	Heurística baseada na união de subárvores
1996	Verhoeven <i>et al.</i>	Busca local
1997	Wade e Rayward-Smith	Busca local
1998	Charikar <i>et al.</i>	Algoritmos para o problema de Steiner em grafos direcionados
1999	Gendreau <i>et al.</i>	Busca Tabu
1999	Duin e Voss	Método guloso com estratégia de memória – Branch-and-Greedy
2000	Gröpl <i>et al.</i>	Resenha de Algoritmos Heurísticos
2000	Martins <i>et al.</i>	GRASP
2001b	Polzin e Daneshmand	Algoritmo heurístico
2002	Julstrom	Algoritmo Genético para o PSG
2002	Satoh <i>et al.</i>	Algoritmo heurístico

6.6.2 – Aplicações e Problemas Correlatos

Dentre várias aplicações destacamos:

- Projeto de circuitos eletrônicos (Smith *et al.* [1980] e Grötschel e Weismantel [1996]).
- Redes de comunicação (Hanan *et al.* [1975], Goel e Munagala [1999]).
- Planejamento de redes externas de comunicação (Luna *et al.* [1987]).
- Redes de comunicação e de tráfego (Frank e Frisch [1976]).
- Tubulações de gás e óleo (Rothfarb [1970]).
- Projetos de instalações elétricas e mecânicas (Smith e Liebman [1979]).

- Distribuição de água para irrigação e redes de drenagem (Lee [1976]).
- Modelos para confecção de circuitos VLSI (Grötschel e Weismantel [1997]).
- Edificações (Smith e Liebman [1979]).
- Várias aplicações (Hwang e Richards ([1992])).
- Aplicações na indústria (Cheng e Du [2001]).

Uma série de problemas e situações caracterizam casos correlatos ao problema de Steiner em grafos. Alguns deles são:

✓ *O Problema de Steiner em Norma Linear (PSNL)*

Trata-se de um caso particular do problema de Steiner no plano. Nesse problema, as coordenadas dos pontos a serem ligados, além de pertencerem a um plano e atender à norma euclidiana geral, atendem à norma euclidiana chamada *linear*, ou seja, $L_x = |x_1 - x_2| + |y_1 - y_2|$. Essa norma é aplicável em circuitos elétricos, no arruamento de cidades, em sistemas mecânicos etc. O problema foi formulado pela primeira vez por Hanan (1966) e várias aplicações práticas sugeridas por Smith e Liebman (1979) bem como uma heurística $O(r^4)$ para sua solução. Smith *et al.* (1980) apresentam uma heurística $O(r \log r)$ baseada nos diagramas de Voronoi e em grafos duais para a solução do PSNL. Para esse problema, temos ainda uma heurística proposta por Richards (1989) que, baseada na heurística de Hanan (1966), originalmente $O(n^2)$, utiliza a geometria da norma linear para encontrar uma implementação $O(n \log n)$ para essa mesma heurística. Komlós e Shing (1985) desenvolvem duas heurísticas em abordagem probabilística, dividindo os pontos de passagem obrigatórios em pequenos conjuntos para aplicar, nessas instâncias reduzidas, o procedimento exato de Dreyfus e Wagner (1971). Posteriormente os subgrafos obtidos são combinados e ampliados para chegar a uma solução global. Beasley (1992) sugere uma heurística para o problema em \mathbb{R}^2 .

✓ *O Problema de Steiner com Conexão Estocástica (PSCE)*

Os problemas em redes de comunicações possuem muitas situações em que a manutenção da efetividade de uma ligação é uma variável aleatória. As arestas possuem uma certa probabilidade de não estarem aptas a manterem uma conexão julgada legal para o sistema. Valiant (1979) demonstra que esse problema de otimização é NP-árduo. Richey *et al.* (1985) apresenta um algoritmo eficiente para um caso particular de grafo de comunicação.

✓ *O Problema do Coletor de Prêmios em Árvore de Steiner - Prize-collecting Steiner tree problem (PCPAS)*

O Problema do Coletor de Prêmios em Árvore de Steiner em um grafo ponderado G com ganhos associados em seus nós, consiste em determinar uma subárvore em G que minimize a soma dos pesos das arestas da subárvore somado ao total de ganhos acumulado nos vértices de G não pertencentes à subárvore. O presente problema é descrito em Johnson *et al.* (2000) e solucionado em Canuto *et al.* (2001) e Klaus *et al.* (2004). O modelo pode representar situações de implantação de redes como, por exemplo, de distribuição de gás em que é necessário além de atender aos pontos de demanda através de uma rede de distribuição, escolher quais pontos são mais atrativos para suportar o investimento da construção da rede de distribuição. Aos pontos de demanda são associados valores de atração que irão penalizar a função objetivo de minimização em caso de não atendimento. Na aplicação descrita o custo de implantação é considerado no valor das arestas.

✓ *O Problema da Árvore de Steiner Terminal (PST)*

O Problema da Árvore de Steiner Terminal, também denominado na literatura como o *Full Steiner Tree Problem* ou *Terminal Steiner Tree Problem*, representa um caso particular do Problema de Steiner

em que todos os nós terminais X são folhas da árvore. Os trabalhos de Lin, e Xue (2002) e Lu *et al.* (2003) abordam o problema.

✓ O Problema da Árvore de Steiner em Grupamentos (PSG)

Também denominado na literatura de “Group Steiner Tree”, consiste em determinar em um grafo conexo e ponderado e contendo grupos disjuntos de nós, uma árvore de comprimento mínimo e pelo menos um nó de cada grupo. Os nós escolhidos nos grupos são denominados de nós terminais. O problema foi abordado por Reich e Widmayer (1989) e pode ser considerado como uma generalização do problema de cobertura mínima (*minimum set covering problem*). Bateman *et al.* (1997) abordam uma aplicação do problema na operação de terminais portuários e sugerem um algoritmo aproximativo bastante eficiente. Even e Kortsarz (2002) desenvolvem um algoritmo com razão polilogarítmica de aproximação igual a $O(\log S \cdot \log m / \log \log S)$, onde m representa o número de grupos e S o número de terminais.

✓ O Problema de Steiner Generalizado (PEG)

Este caso é de ampla aplicação em redes de suprimento de eletricidade, água ou comunicações. Nesses casos o modelo, via de regra, procura o atendimento de dois objetivos conflitantes:

- Minimizar os custos totais de operação e instalação.
- Maximizar a confiabilidade da operação.

A confiabilidade é frequentemente expressa através de graus de conectividade alcançado pela configuração em análise. Através das árvores de Steiner é possível modelar o problema dos custos das conexões. É visível que a solução de conexão baseada em uma estrutura de árvore tem grande vulnerabilidade face a uma desconexão estocástica. A cada falha ocorrida nas ligações ou nos vértices da estrutura de custo mínimo o serviço será descontinuado para pelo menos uma parte da rede. Nesse caso, a solução de equilíbrio vai buscar um grau k de conectividade. O problema pode ser formulado (Winter [1987]) da seguinte forma:

$$(PEG) \text{ Minimizar } z = \sum_{i,j \in A} c_{ij} x_{ij}$$

sujeito a:

$$x_{ij} \geq y_{ij}^{kl} + y_{ji}^{kl} \quad \forall (i,j) \in A, \forall k, l \in X, k \neq l \quad (6.1)$$

$$\sum_{(k,j) \in A} y_{kj}^{kl} \geq s_{kl} \quad \forall k, l \in X, k \neq l \quad (6.2)$$

$$\sum_{(i,l) \in A} y_{il}^{kl} \geq s_{kl} \quad \forall k, l \in X, k \neq l \quad (6.3)$$

$$\sum_{(p,j) \in A} y_{pj}^{kl} - \sum_{(i,p) \in A} y_{ip}^{kl} = 0 \quad \forall k, l \in X, \forall p \in N \setminus \{k, l\} \quad (6.4)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i,j) \in A, \quad (6.5)$$

$$y_{ij}^{kl} \geq 0 \quad \forall i, j : (i,j) \in N, \forall k, l \in X, k \neq l \quad (6.6)$$

Em que:

$y_{ij}^{kl} \equiv$ a quantidade da comodidade a ser deslocada de k para l ao longo da aresta (i, j) na direção de i para j .

$S = [s_{ij}] \equiv$ a matriz de requisitos de conectividade entre os pares de vértices $i, j \in X$, ou seja, entre os vértices i e j existe uma s_{ij} conectividade (s_{ij} arestas de ligação).

$x_{ij} \equiv$ variável binária em que 1 significa que a aresta (i, j) pertence à solução e 0 caso contrário.

A restrição (6.1) garante que o produto associado a cada par origem-destino (k, l) só pode usar um arco (i, j) que seja efetivamente ativado pelo modelo.

A restrição (6.2) indica que o fluxo do produto (k, l) deve escoar a partir do nó k , através de um número de nós sucessores pelo menos igual ao grau de conectividade requisitado para o produto (k, l) .

A restrição (6.4) garante que as conexões do produto (k, l) que deixem os vértices de Steiner sejam pelo menos iguais ao número de conexões que chegam.

✓ O Problema de Network Design e as Formulações Multiproduto para o Problema de Steiner (PSM)

A área de fluxo tem sofrido, nos últimos anos, um grande desenvolvimento na direção da modelagem e solução de problemas de comunicação em rede. Existem, nessa linha de problemas práticos, inúmeros casos, que vão desde o estágio da definição das comunicações em cenários restritos, até os modelos para a política global de definição dos equipamentos e do tráfego de comunicações, considerando todos os níveis necessários e levando em conta vários cenários para a interpretação econômica dos sistemas. Apresentaremos duas formulações bastante interessantes para essa importante aplicação da pesquisa operacional no mundo atual. Consideremos o grafo $G = (V, E)$ da Figura 6.16, em que as arestas modelam a possibilidade de comunicação direta entre seus nós. Vamos supor que existe um nó o centro de comunicação da rede e origem do fluxo que circulará pela rede e que existe a necessidade de ligar um número $k, k = 1, \dots, |K|$ de nós que possuem uma demanda definida como d_k , a esse centro de comunicação.

Considerando que existe um conjunto de custos associados à implantação e operação das linhas de comunicação, então esse problema pode ser resumido à necessidade de determinar uma arborescência em G que minimize os custos de operação e de implantação, para um conveniente horizonte operacional do sistema. Esse tipo de problema pode ser modelado através das formulações de fluxo multiproduto que apresentaremos a seguir.

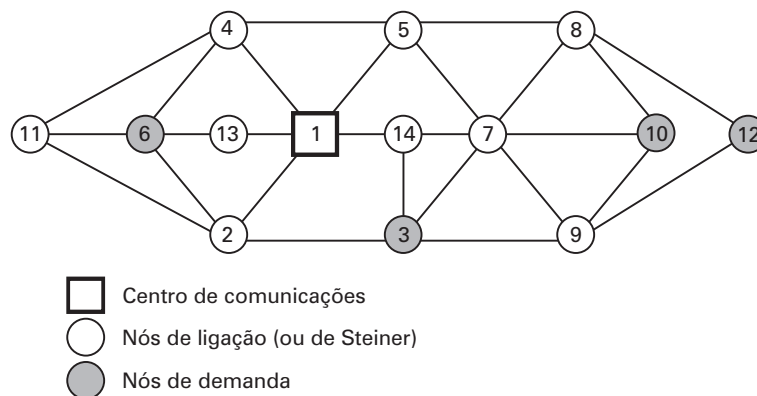


FIGURA 6.16 Grafo do problema de projeto de redes – visão local.

O interesse de formulações multiproduto para o problema de Steiner e extensões está no fato de que relaxações de programação linear a esses problemas podem resolver de forma exata diversas instâncias significativas (Luna *et al.* [1998]).

• Formulação de Produto Único (FPU)

Considerando a definição de adequados custos estruturais e operacionais, o problema proposto consiste em determinar uma arborescência de mínimo custo que ligue o centro de comutação a todos os pontos de demanda, ressaltando-se que, no modelo, todo fluxo será originado no centro de comunicação.

Definindo-se sobre um grafo $G = (V, E)$:

$$x_{ij} \equiv \begin{cases} 1, & \text{caso uma ligação direta utilize o arco } (i, j) \\ 0, & \text{caso contrário} \end{cases}$$

$b_{ij} \equiv$ custo estrutural fixo para instalar uma linha de transmissão direta no arco (i, j)

$b_{ij} = \beta d_{ij}$, onde d_{ij} é a distância (em metros) entre os vértices i e j , e β o custo unitário (por metro) de ligação.

$c_{ij} \equiv$ custo operacional variável relativo à transmissão de uma unidade de fluxo através do arco (i, j)

$x_{ij} \equiv \gamma d_{ij}$, onde d_{ij} é a distância (em metros) entre os vértices i e j , e γ é o custo por metro para transmitir uma unidade de fluxo.

$f_{ij} \equiv$ fluxo passando através do arco (i, j) .

$$\begin{aligned} \text{(FPU) Minimizar } & \sum_{(i,j) \in E} (c_{ij}f_{ij} + b_{ij}x_{ij}) \\ \text{sujeito a:} & \\ & - \sum_{(o,j) \in E} f_{oj} = - \sum_{k \in K} d_k \end{aligned} \quad (6.7)$$

$$\sum_{(i,k) \in E} f_{ik} - \sum_{(k,j) \in E} f_{kj} = d_k \quad \forall k \in K \quad (6.8)$$

$$\sum_{(i,j) \in E} f_{ij} - \sum_{(j,l) \in E} f_{jl} = 0 \quad \forall j \in V \setminus K \setminus \{o\} \quad (6.9)$$

$$f_{ij} \leq \left(\sum_{k \in K} d_k \right) x_{ij}, \quad \forall (i,j) \in E \quad (6.10)$$

$$\sum_{(i,j) \in E} x_{ij} \geq |K| \quad (6.11)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in E \quad (6.12)$$

A função objetivo possui dois termos. O primeiro está associado ao custo total de enviar fluxo através dos arcos, o segundo acumula o custo fixo relativo à utilização dos arcos. A restrição (6.7) assegura que a capacidade do vértice de comutação é suficiente para atender à demanda dos demais vértices. A restrição (6.8) modela as imposições de demanda, a (6.9) garante a conservação de fluxo nos vértices de Steiner. A restrição (6.10) assegura que não haverá fluxo nos arcos que não forem escolhidos pela solução, e a (6.11) estabelece que pelo menos $|K|$ arcos serão selecionados, assegurando a conexão de todos os nós através de uma árvore. O modelo possui uma restrição do tipo (6.7), $|K|$ restrições do tipo (6.8), $|V| - |K| - 1$ restrições do tipo (6.9), $|E|$ restrições do tipo (6.10), uma restrição do tipo (6.11) e $|E|$ variáveis do tipo (6.12).

• **Formulação Multiproduto (FPM)**

Considerando a definição de adequados custos estruturais e operacionais, o problema proposto consiste em determinar uma arborescência. Consideraremos as seguintes variáveis adicionais ao modelo (FPU):

$c_{ij}^k \equiv$ variável operacional que quantifica o custo de transitar uma unidade do produto k através do arco (i, j) .

$c_{ij}^k \equiv \delta_k d_{ij}$, $\forall k \in K$, onde δ_k é o custo por metro para transmitir uma unidade de fluxo do produto k .

$f_{ij}^k \equiv$ Fluxo passando pelo arco (i, j) e destinado ao ponto de demanda k .

Esse modelo permite que diferentes custos variáveis sejam associados aos produtos e aos arcos.

$$(FPM) \text{ Minimizar } z = \sum_{(i,j) \in E} \left(b_{ij} x_{ij} + \sum_{k \in K} c_{ij}^k f_{ij}^k \right)$$

sujeito a:

$$- \sum_{(i,j) \in E} f_{oj}^k = -d_k \quad \text{para todo nó } o \text{ e } \forall k \in K \quad (6.13)$$

$$\sum_{(i,k) \in E} f_{ik}^k = d_k \quad \forall k \in K \quad (6.14)$$

$$\sum_{(i,j) \in E} f_{ij}^k - \sum_{(j,l) \in E} f_{jl}^k = 0 \quad \forall j \in V \setminus K \setminus \{o\} \quad (6.15)$$

$$f_{ij}^k \leq d_k x_{ij}, \quad \forall (i, j) \in E \text{ e } \forall k \in K \quad (6.16)$$

$$\sum_{(o,j) \in E} x_{oj} \geq 1 \quad \text{para o nó } o \quad (6.17)$$

$$\sum_{(i,k) \in E} x_{ik} = 1 \quad \forall k \in K \quad (6.18)$$

$$\sum_{(l,j) \in E} x_{lj} - \sum_{(i,l) \in E} x_{il} \geq 0 \quad \forall l \in V \setminus K \quad (6.19)$$

$$\sum_{(i,l) \in E} x_{il} \geq \frac{\sum_{(l,j) \in E} x_{lj}}{\sum_{(l,j) \in E} 1} \quad \forall l \in V \setminus K \setminus \{o\} \quad (6.20)$$

$$f_{ij}^k \geq 0 \quad \forall (i, j) \in E \text{ e } \forall k \in K \quad (6.21)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in E \quad (6.22)$$

A função objetivo leva em conta não só os custos fixos, mas os relativos ao fluxo de cada produto. A restrição (6.13) assegura que a demanda total de um produto k é atendida pelo fluxo originado no vértice comutação. A restrição (6.14) impõe que a demanda de um produto k é igual ao fluxo desse produto que chega ao vértice k . A restrição (6.15) garante a conservação de fluxo através dos vértices de passagem. A restrição (6.16) obriga o fluxo nulo através dos arcos não escolhidos para compor a solução. A restrição (6.17) força que pelo menos um arco ativo ligue o vértice de comutação à rede. A restrição (6.18) garante que exatamente um arco ativo ligue cada vértice de demanda. A restrição (6.19) assegura que a ativação de qualquer arco que chegue a qualquer nó de Steiner implica a ativação de um ou mais arcos que saiam do nó. A restrição (6.20) garante que a ativação de arcos que saem de um nó de Steiner implicam a ativação de exatamente um arco incidente ao nó. As restrição (6.21) assegura a não negatividade do fluxo.

Os modelos FPU e FPM são bastante úteis para a formulação de problemas de projeto de redes de comunicações (Randazzo e Luna [2001]), permitindo sucesso na solução de problemas de grande porte (ainda Geoffrion e Graves [1974], Boorstyn, e Frank [1977], França e Luna [1982], Gavish [1982], Christofides e Beasley [1982], Cho [1983a] e [1983b], Magnanti e Wong [1984], e Balakrishnan e Altinkemer [1992]).

✓ O Problema de Steiner com Vértices Ponderados (PSVP)

Consideremos que, nas mesmas condições do problema de Steiner em grafos a cada vértice do grafo $G = (N, A)$ esteja associado um valor ou peso, w_i . Para um dado subconjunto de vértices $X \subseteq N$ uma solução ótima do (PSVP) é uma árvore em G , $T_S = (T, A_r)$, $T \supseteq X$, $A_r \subseteq A$ que liga o conjunto de vértices atendendo à seguinte função objetivo:

$$\text{Minimizar } z = \sum_{i \in T} w_i + \sum_{(i,j) \in A_r} c_{ij}$$

Sem dúvida trata-se de uma generalização do PEG, uma vez que se o vetor de peso w_i for nulo o PSVP transforma-se no PEG. Situações de localização de facilidades em plantas industriais podem ser formuladas via PSVP, (Wong [1984]). Nesses casos, os vértices podem representar clientes que demandam serviço. Os demais nós do grafo podem ser associados a possíveis localizações dos postos de atendimento. O vetor c_{ij} pode representar os custos de transporte entre os nós i e j . O vetor w_i pode quantificar os custos fixos de localizar na planta uma facilidade no vértice i .

✓ O Problema da Arborescência de Steiner (PAS)

Esse caso é uma generalização do PSVP. O problema é apresentado por Wong (1984) e detalhado em Duin e Volgenant (1987). Consideremos O PSVP em um grafo $G = (V, E)$. O PSVP objetiva determinar uma arborescência em T_S em G , enraizada fora de um conjunto $x_0 \in X$. Segev (1987) apresenta uma transformação de custos que o reduz ao PSVP.

✓ O Problema da Floresta de Steiner (PFS)

Algumas situações de modelagem podem não exigir que as estruturas de conexão formem um só corpo. Esse caso aborda problemas em que existem várias estações de distribuição de água ou óleo, ou a possibilidade de constituição de redes locais de informação em que os maiores custos são associados a configurações das redes locais, sendo o custo de conexão com o “exterior” (estação central, Internet etc.) ou interligação das redes, não significativo face ao custo local. Obviamente a solução de conexão de todos os pontos via essa ligação exterior não seria uma solução razoável por motivos de segurança, tráfego e confiabilidade. Nesse caso, a exigência recai sobre a configuração mínima válida, ou seja, sobre o número de nós mínimo que um ponto de atendimento vai suprir ou um número máximo de sub-redes de atendimento.

Podemos enunciar esse problema, então, da seguinte forma: determinar no máximo m subgrafos conexos em $G = (N, A)$, ($m < |X|$), onde cada um desses subgrafos contenha pelo menos um vértice de um certo conjunto de vértices X , e que todos os vértices de X estejam contidos nesses subgrafos, minimizado o custo total das arestas utilizadas na solução.

✓ O Problema de Agrupamento de Árvores de Steiner (PAAS)

Este problema, denominado na literatura internacional de *packing steiner tree*, está associado especialmente ao projeto e fabricação de circuitos eletrônicos (ver Grötschel e Weismantel [1996]). Consideremos a Figura 6.17 que representa um modelo de circuito integrado.

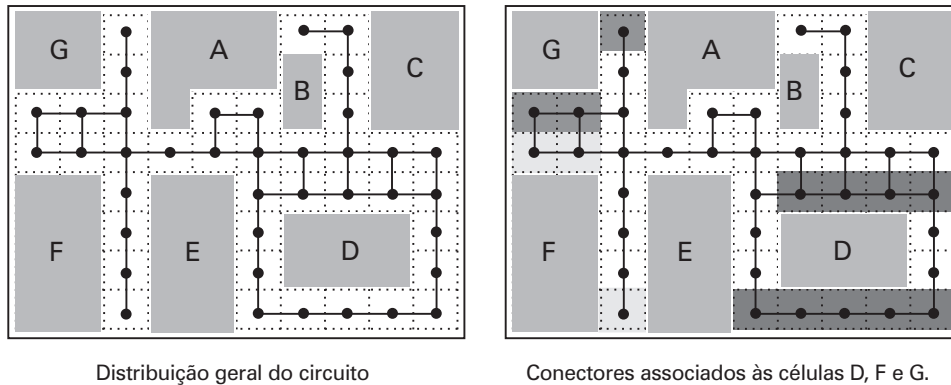


FIGURA 6.17 Um circuito integrado.

Normalmente, um projeto de um circuito integrado, em relação à disposição dos componentes na placa, envolve pelo menos duas etapas:

- O projeto lógico, que corresponde ao esclarecimento de como as operações lógicas necessárias são associadas às unidades eletrônicas (chamadas células) e como essas unidades podem ser ligadas entre si pelas vias (barramentos condutores que correspondem aos “fios” das ligações clássicas). Os pontos em que as vias fazem contato com as células são denominados terminais. Na Figura 6.17 os terminais são representados pelos pontos pretos e as vias pelos traços que unem os pontos. As células são os retângulos escurecidos. Um conjunto de terminais são chamados de rede. Os terminais conectores são designados às células.
- O projeto físico que corresponde à designação das células dentro de uma certa área e o roteamento das redes.

As ligações entre células estão exemplificadas na Figura 6.18, em que verificamos que existe a possibilidade de que algumas vias sejam utilizadas mais de uma vez para a conexão (arestas marcadas com a barra grossa).

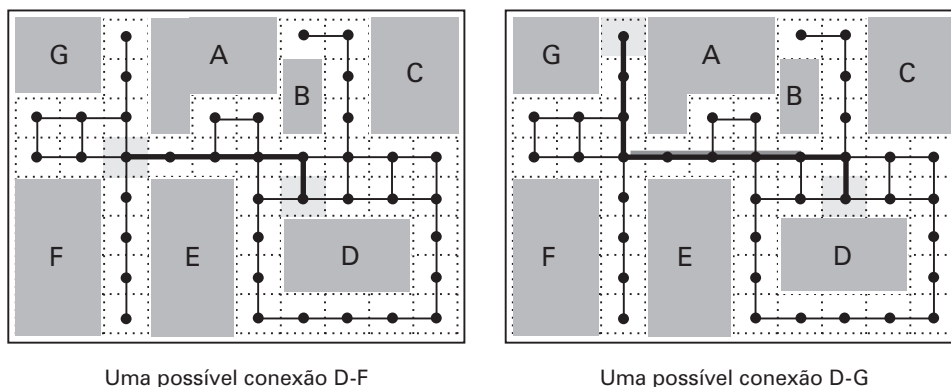


FIGURA 6.18 Possíveis conexões entre células.

Tipicamente, o problema combinatório que está definido nesse circuito é o de desenvolvermos ligações entre as células da placa com o objetivo de minimizar, por exemplo, o comprimento da via de ligação, não sobrecarregando o tráfego nos trechos comuns.

Chamando de $\bar{N} = \{T_1, \dots, T_n\}$, $n \geq 1$, com $T_k \subseteq N$, $k = 1, \dots, n$, uma lista de conjuntos de vértices de $G = (N, A)$, e para cada aresta $(i, j) \in A$ associando-se um custo c_{ij} e uma capacidade de utilização w_{ij} , o Problema de Agrupamento de Árvores de Steiner (PAAS) é determinar um grupo de conjunto de arestas de G , $S_1, \dots, S_n \subseteq A$ tais que:

1. $\forall S_n \subseteq A$ seja uma árvore de Steiner em G para as listas de vértices T_k associados, $k = 1, \dots, n$.
2. $\sum_{k=1}^n |S_k \cap (i, j)| \leq w_{ij}$ para toda aresta $(i, j) \in A$.
3. $\sum_{k=1}^n \sum_{(i, j) \in S_k} c_{ij}$ seja minimal.

✓ O Problema da Árvore de Steiner MinMax (Gargalo) (PASM)

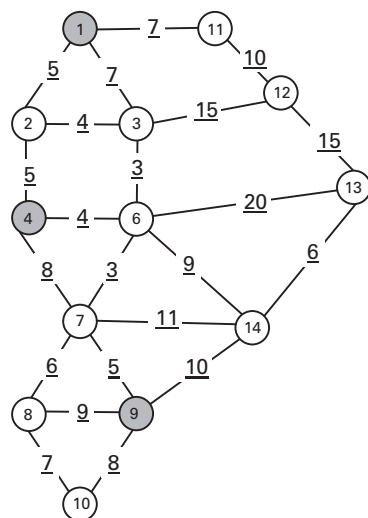
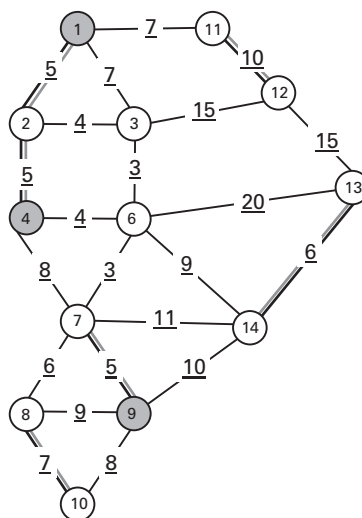
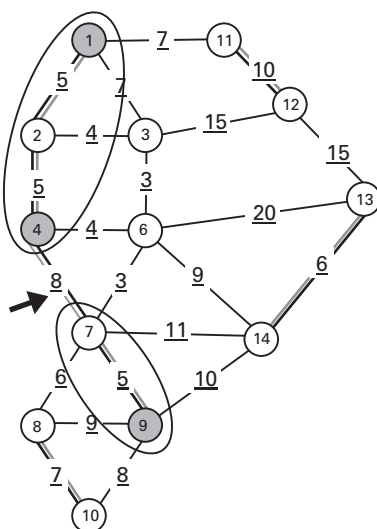
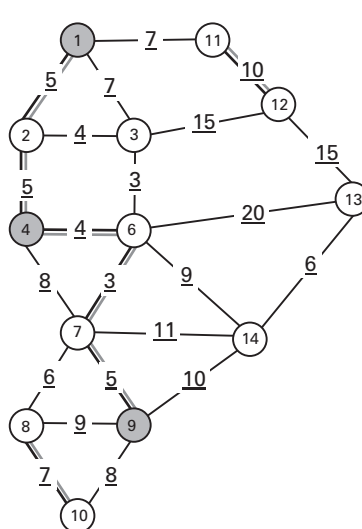
Trata-se do caso MinMax para a árvore de Steiner. Duin e Volgenant (1997) apresentam um algoritmo $O(n)$ para a solução desse caso.

6.6.3 – Algoritmo Heurístico de Solução

Aneja (1980) sugere a utilização da solução do problema de recobrimento para acelerar a convergência de um procedimento B&B para a solução do PAS em grafos. Baseado em seu procedimento podemos organizar o seguinte procedimento heurístico:

Algoritmo Recobrimento em Arestas
<p>INÍCIO</p> <p>Ler os dados de $G = (N, A)$, onde d_{ij} é a distância entre os nós vizinhos de G, e X o conjunto dos nós a serem ligados</p> <p>Determinar uma cobertura de arestas minimal para G</p> <p>Identificar os subgrafos conexos que contêm os nós de X</p> <p>Ligá-los entre si utilizando a menor distância possível</p> <p>Remover as arestas desnecessárias à árvore de Steiner</p> <p>FIM</p>

Um exemplo de aplicação da heurística de recobrimento encontra-se na Figura 6.19.

Grafo inicial. Conjunto $X = \{1, 4, 9\}$ Solução do *Edge Covering*Ligação entre os grupamentos
com nós X
 $S = \{1, 2, 4, 7, 9\} = 23$ Solução ótima (não encontrada
pelo algoritmo)
 $S = \{1, 2, 4, 6, 7, 9\} = 22$ **FIGURA 6.19** Solução do algoritmo recobrimento de arestas.

6.7 – O PROBLEMA DO CAMINHO CRÍTICO OU PERT (*CRITICAL-PATH METHOD*)

Esse modelo é extremamente útil para a solução de problemas que possuem um número muito grande de atividades que ocorrem paralelamente e com duração variada e, além disso, apresentem pontos de concorrência e interdependência. As maiores aplicações ocorrem na construção civil, e na gerência de grandes projetos, especialmente na fase de planejamento e como ferramenta de acompanhamento. Examinemos um exemplo do uso da rede PERT para o planejamento de uma simulação de um protótipo industrial. As etapas necessárias para a realização da simulação do protótipo estão descritas na tabela da Figura 6.20. A título de esclarecimento, informamos que os tempos necessários para a execução de cada etapa do planejamento deverão ser motivo de um estudo específico. De um modo geral, é possível considerar tempos médios.

Número	Atividades	Atividade de Pré-Requisito	Duração	Início mais Cedo
0	Início do Trabalho	–	0	–
1	Projeto de Simulação	0	2	t_1
2	Treinamento de Pessoal	1	9	t_2
3	Construção de Instalações Especiais	1	4	t_2
4	Certificação das Instalações	3,6	1	t_3
5	Aquisição de Material	1	1	t_2
6	Aferição dos Instrumentos	5	2	t_4
7	Teste do Material Adquirido	2,4	3	t_5
8	Montagem da Cabine de Simulação	7	1	t_6
9	Execução da Simulação	8	2	t_7
10	Fim	9	0	t_8

FIGURA 6.20 Tabela de descrição das atividades.

A tabela de atividades pode ser representada por um gráfico de setas como mostra a Figura 6.21. Nesse gráfico, cada atividade é simbolizada por um nó numerado. Os arcos (ou setas) representam as restrições de precedência. Sobre cada um dos nós podemos representar a duração da atividade a ele associada. Esse gráfico é também conhecido por *rede orientada por tarefa*, ou ainda por *diagrama de setas* (uma das ferramentas da qualidade).

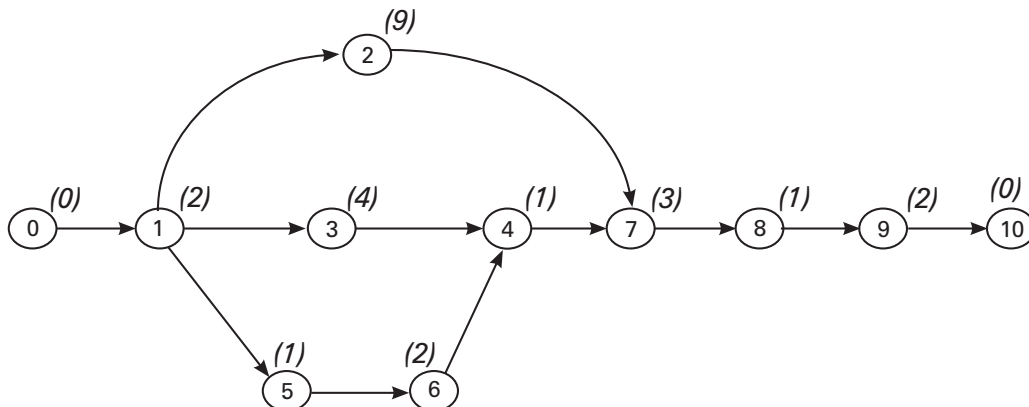


FIGURA 6.21 Rede orientada por tarefas.

Denominando por t_{10} o menor tempo a que está associada a obra como um todo (tempo mais cedo para a última atividade), então podemos definir a função objetivo desse programa como:

$$\text{Minimizar } z = t_{10} - t_1$$

sujeito às restrições de precedência das atividades que serão impostas pelos arcos dos diversos caminhos que ligam o nó 0 ao nó 10. Considerando a atividade 4, certificação das instalações, por exemplo, o tempo menor para a ocorrência dessa atividade deve atender:

$$t_3 \geq t_2 + 4 \text{ (pelo caminho da atividade 3)}$$

$$t_3 \geq t_4 + 2 \text{ (pelo caminho da atividade 6)}$$

De uma forma geral, se t_j é o menor tempo associado a um antecessor de t_i , e d_{ij} é a duração da atividade $i-j$, o tempo t_i é calculado como:

$$t_i \geq t_j + d_{ij}$$

Calculando as restrições associadas aos t_i (matriz T) temos o quadro da Figura 6.22 para o modelo de programação linear:

	<i>min</i>	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
	<i>Valor de z</i>	-1							+1
<i>Nós</i>	<i>Duração da atividade</i>								
2,3,5	2	-1	+1						
4	4		-1	+1					
6	1		-1		+1				
7	9		-1			+1			
7	1			-1		+1			
4	2			+1	-1				
8	3					-1	+1		
9	1						-1	+1	
10	2							-1	+1

FIGURA 6.22 Matriz de restrições do programa PERT orientado por tarefas.

O problema escrito da forma anterior não é um problema de fluxo em rede, mas seu dual o é. Notamos que existe apenas uma entrada positiva e uma negativa nas linhas da matriz de restrição do primal, o que implica que a matriz transposta de T é a matriz nó-aresta de uma rede. Considerando uma variável dual x_{ij} associada a cada linha do problema primal, ou seja, representando atividades, conforme a tabela da Figura 6.23:

Para o programa da Figura 6.23 podemos expressar o seu programa dual como mostra o quadro da Figura 6.24.

<i>Variável (Atividade)</i>	<i>Atividades</i>	<i>Atividade de Pré-Requisito</i>	<i>Duração</i>
x_{01}	Projeto de Simulação	—	2
x_{15}	Treinamento de Pessoal	x_{01}	9
x_{12}	Construção de Instalações Especiais	x_{01}	4
x_{23}	Certificação das Instalações	x_{12}	1
x_{14}	Aquisição de Material	x_{01}	1
x_{43}	Aferição dos Instrumentos	x_{14}	2
x_{35}	Teste do Material Adquirido	x_{23}, x_{43}	3
x_{56}	Montagem da Cabine de Simulação	x_{15}, x_{35}	1
x_{67}	Execução da Simulação	x_{56}	2

FIGURA 6.23 Exemplo de um programa PERT.

nós	max	x_{01}	x_{15}	x_{12}	x_{23}	x_{14}	x_{43}	x_{35}	x_{56}	x_{67}
	W	2	9	4	1	1	2	3	1	2
0	+1	+1								
1	0	-1	+1	+1		+1				
2	0			-1	+1					
3	0				-1		+1	-1		
4	0					-1	-1			
5	0		-1					+1	-1	
6	0								+1	-1
7	-1									+1

FIGURA 6.24 Matriz de restrições do problema PERT dual.

A formulação dual da rede PERT, orientada por tarefas, possui características interessantes e é denominada *rede orientada por eventos*. A Figura 6.25 exemplifica a rede orientada por eventos do caso examinado.

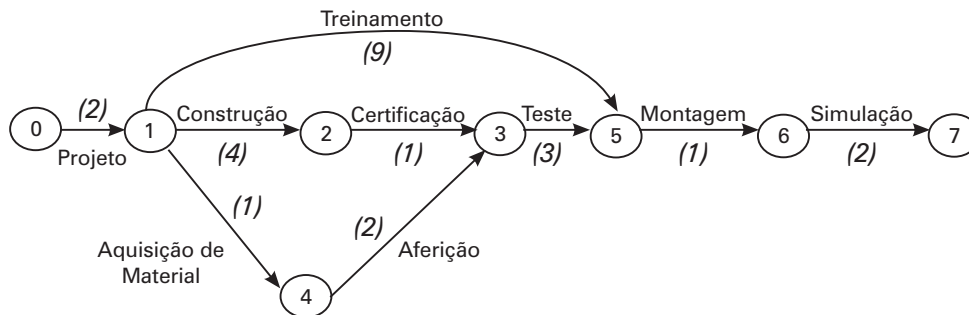


FIGURA 6.25 Rede orientada por eventos.

Nessa rede, um nó representa o início ou o término de uma atividade, um evento instantâneo. No modelo dual, os valores de duração das atividades estão nos arcos. A variável dual x_{ij} representa uma atividade definida pelos nós $i-j$. Notemos que o número de arcos da rede orientada por eventos é igual ao número de nós da rede orientada por tarefas (desconsiderando os nós início e fim). Os tempos do modelo primal correspondem ao número de nós da rede dual. Na rede orientada por eventos, o problema de determinar o maior tempo mais cedo reduz-se a encontrar o caminho mais longo entre os nós extremos da rede, considerando o comprimento de cada arco $i-j$ igual à duração da atividade a ele associada.

Existe ainda uma terceira representação para esse tipo de problema de planejamento. Essa rede é extremamente difundida, especialmente porque facilita a visualização das variáveis de folga da rede orientada por tarefas. Uma outra vantagem dessa variante é que ela facilita a distinção entre atividades paralelas e/ou logicamente correlacionadas. A rede PERT/CPM, como difundida na literatura, é uma rede orientada por eventos com uma correção na definição dos arcos. Nela as atividades paralelas devem possuir nós extremos distintos, mesmo que iniciem e terminem nos mesmos eventos. Para fazer essa distinção entre as atividades o modelo utiliza o conceito de atividade fantasma. Uma atividade fantasma é um arco na rede de eventos cuja duração é igual a zero (instantânea). No caso do exemplo proposto, a representação PERT/CPM está expressa na Figura 6.26.

Os arcos pontilhados representam as atividades fantasmas. Na solução da rede por eventos, o caminho mais longo vai determinar uma cadeia dominante em relação ao tempo. Esse caminho representa o gargalo para a duração de todo o programa. Obviamente poderão existir outros caminhos mais curtos dentro da rede, mas eles serão dominados em relação ao tempo do mais longo. A existência de caminhos de duração variada em um mesmo planejamento implica que, em relação ao tempo de conclusão, algumas atividades acabarão assumindo um papel destacado ou crítico, enquanto outras esta-

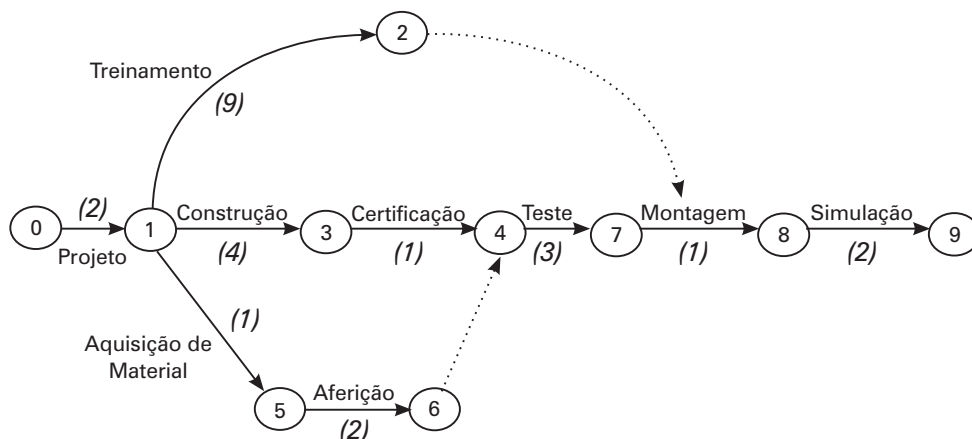


FIGURA 6.26 Rede PERT/CPM.

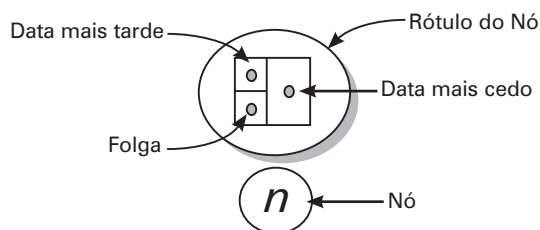


FIGURA 6.27 Rótulo do nó na rede PERT/CPM.

rão sujeitas a uma menor pressão. Essa informação poderá ser muito útil na redistribuição dos esforços e meios no planejamento. Podemos examinar visualmente esse fenômeno acrescentando a cada nó da rede da Figura 6.26 o rótulo representado na Figura 6.27.

A rede orientada por eventos e rotulada como proposto na Figura 6.27 poderá ser solucionada através da busca dos caminhos que ligam o evento inicial e o final. A Figura 6.28 mostra a rede rotulada e o início do cálculos dos possíveis caminhos que ligam o nó 0 ao nó 7. Ao longo de cada caminho possível os tempos vão se acumulando, o que fica claro na rede orientada por tarefas. Quando um nó é alcançado por mais de um caminho, possivelmente ocorrerá uma situação de conflito em relação ao tempo as-

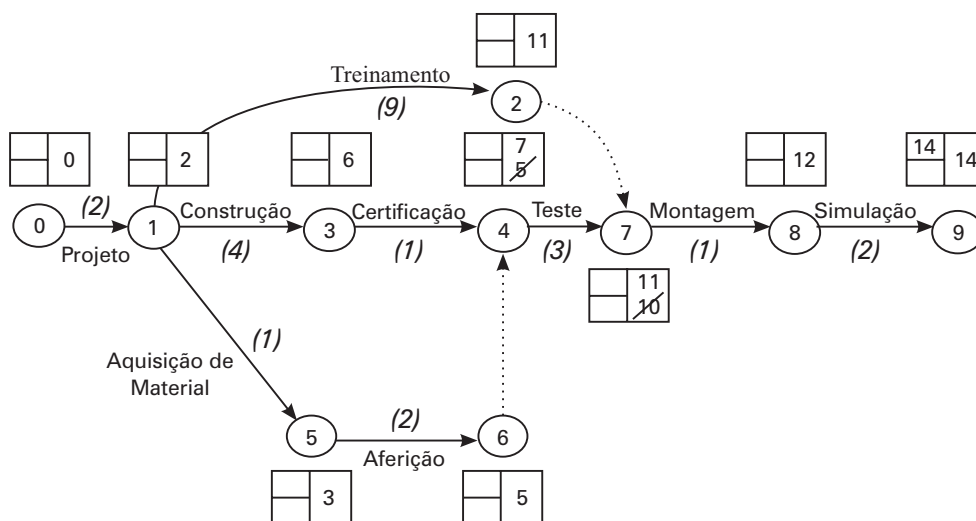


FIGURA 6.28 Cálculo do tempo mais cedo em situação de conflito.

sociado ao nó. Como desejamos encontrar a data mais cedo possível mas que leve em conta a duração de todas as atividades, essa data vai corresponder ao maior valor acumulado no tempo. Essa situação é exemplificada no nó 4 da Figura 6.28. Esse nó pode ser alcançado através do caminho 0-1-2-3, acumulando um tempo igual a 7 unidades $((2) + (4) + (1))$, e pelo caminho 0-1-5-6-4, acumulando um tempo de 5 unidades $((2) + (1) + (2))$. O tempo mais cedo do nó 4 será 7, o valor maior.

O mesmo raciocínio se aplica ao nó 7. Dessa forma, calculamos o tempo mais cedo possível para a rede completa, ou seja, 14 unidades. Para encontrarmos as folgas existentes nos diversos caminhos possíveis devemos, partindo do nó 9 (nó final), tentar alcançar o nó 0 (nó inicial) descontando os valores associados às atividades nos arcos. No caso de acontecer que um nó possa ser alcançado por mais de um caminho de retorno, então devemos preservar o menor tempo. Isso acontece no nó 2, onde, pelo caminho 9-8-7-2-1, o valor é 2, pelo caminho 9-8-7-4-3-1 é 3 e pelo caminho 9-8-7-4-6-5-1 é 5. O valor a ser registrado será 2. Após os cálculos encontraremos os valores constantes da Figura 6.29.

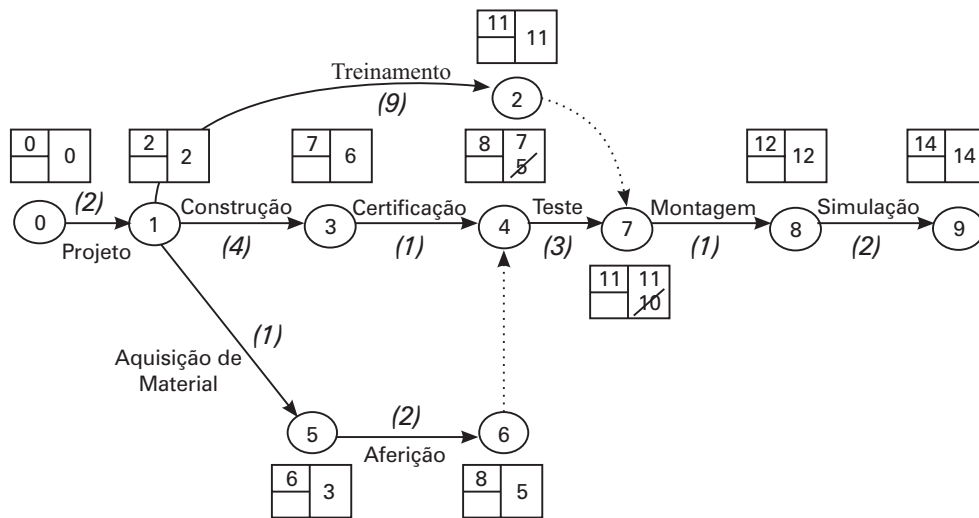


FIGURA 6.29 Cálculo do tempo mais cedo.

Finalmente, podemos preencher o campo da folga fazendo a diferença entre a data mais cedo e a mais tarde de cada nó. Os caminhos que possuem folga zero são denominados caminhos críticos. Na Figura 6.30, o caminho crítico está ressaltado.

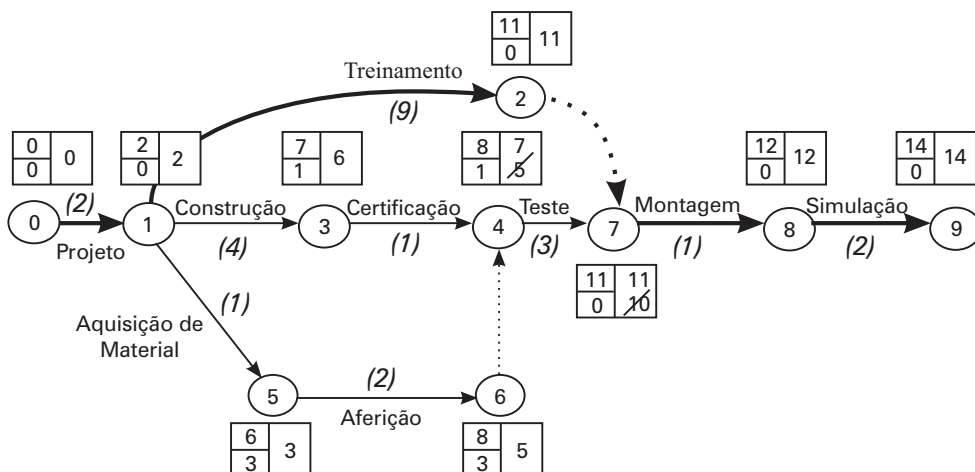


FIGURA 6.30 Cálculo do tempo mais cedo.

6.8 – PROBLEMAS PROPOSTOS

1 – O Problema das Obras de Saneamento



Em um determinado bairro, a prefeitura pretende instalar a infraestrutura necessária de água e esgoto. Um trecho do bairro está representado na Figura 6.31.

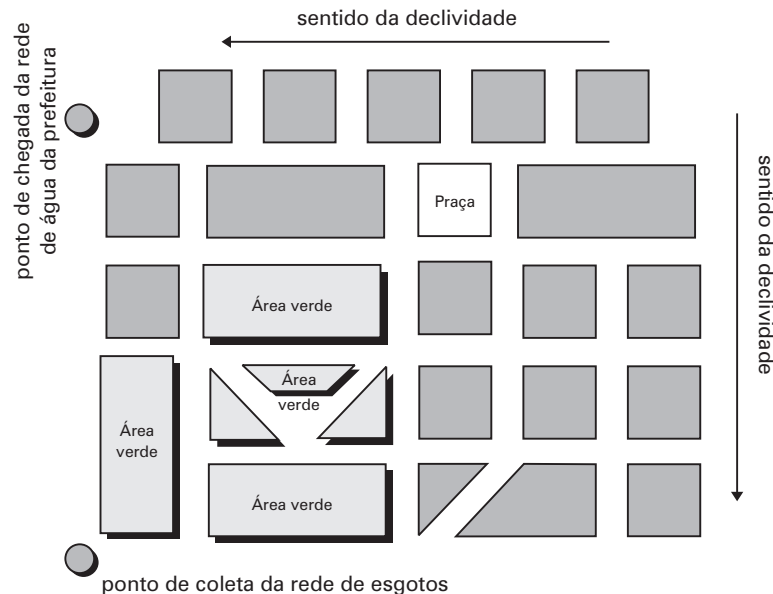


FIGURA 6.31 Bairro do problema.

As redes de água devem passar por pelo menos três lados de um quarteirão habitado e as de esgoto por todos. As escavações realizadas no sentido horizontal são quatro vezes mais caras do que as de sentido vertical. A praça só precisa ser abastecida por água em um de seus lados (qualquer um) e não precisa de rede de esgoto.

Para realizar o planejamento da obra e seu respectivo orçamento, o prefeito determinou ao departamento de engenharia que elaborasse um projeto inicial de distribuição física das redes que atendessem às seguintes situações:

1. Projetos independentes para as redes de água e esgoto.
2. Projetos das redes objetivando utilizar as mesmas escavações, sempre que possível.

Os tubos de esgoto devem ser assentados no sentido da declividade do terreno. Os tubos de água podem seguir qualquer trajetória, pois trabalham sob pressão. Para evitar problemas de perda de carga os ramais de abastecimento de água não devem possuir comprimento maior do que 10 lados dos quarteirões quadrados (sem levar em conta o comprimento das ruas). Os quarteirões retangulares e oblíquos são considerados equivalentes a dois quarteirões pequenos em suas faces mais longas.

2 – O Problema do Caminho mais Curto na Cidade Destra



Uma certa cidade possui um código de trânsito rígido e peculiar: em suas ruas jamais é permitido que um veículo faça uma curva à esquerda. Toda vez que um motorista desejar fazer uma mudança de direção ele terá de seguir sempre à direita. Todas as ruas da cidade são de mão dupla. Uma cooperativa de táxi resolveu desenvolver um *software* para auxiliar os motoristas a encontrarem sempre o caminho mais curto entre sua posição e as chamadas de atendimento. Modifique o algoritmo de **Dijkstra** (se necessário) para que esse possa funcionar corretamente nesse tipo de malha viária.

3 – O Problema do Reabastecimento dos Carros Elétricos



Uma cidade está ativando um novo sistema de transporte pessoal para a sua região central. O sistema é constituído por pequenos carros elétricos. O sistema visa a reduzir a poluição, o barulho e os engarrafamentos na região, minimizando a presença de veículos convencionais. Os pequenos e ágeis carros elétricos funcionam a partir de dois grandes estacionamentos articulados com as estações de ônibus e metrô. O principal problema desses carros é que sua autonomia é pequena e eles necessitam ser abastecidos a cada duas horas de operação contínua. No centro e nas cercanias foram criados vários eletropostos para providenciar esse serviço, mas, obviamente, eles possuem limitações de demanda. Para ajudar o motorista a programar seu trabalho, inclusive fora do centro da cidade, se necessário, o carro dispõe de um software que auxilia a encontrar sua melhor rota, face aos pontos de visita desejados. Desenvolva um algoritmo para esse tipo de roteamento, fazendo as considerações que julgar pertinente ao caso.

4 – O Problema do Labirinto Bifurcado



O famoso herói Teseu, após escapar do labirinto do Minotauro, foi preso por um rei que se julgava muito esperto. Esse rei havia criado um labirinto no qual todos os caminhos eram sempre bifurcados, ou seja, nunca havia mais de dois caminhos a seguir. Segundo ele, era possível sair de lá sem possuir um novelo de lã ou sem fazer qualquer marca nas paredes do labirinto. O rei propôs esse desafio a Teseu em troca da mão de sua filha, a princesa Bela. Se você fosse Teseu aceitaria o desafio? Justifique sua resposta.

5 – O Problema da Rota mais Barata sujeita a Pedágio



Um carro pretende cruzar um trecho de estradas administrado por certa empresa e sujeitas a pedágio, trafegando da cidade S para a cidade T . O custo da travessia é igual à soma do total da distância percorrida com o valor do pedágio. As regras de pedágio são simples: o carro paga a tarifa do pedágio correspondente ao trecho percorrido, descontado o valor pago anteriormente em qualquer trecho administrado pela mesma empresa. Com essa forma de tarifar, o veículo pagará, a título de pedágio em uma rota, o valor da maior tarifa correspondente a um trecho desta. A Figura 6.32 apresenta no campo (a) as distâncias entre as cidades S e T dentro das rotas possíveis, e no campo (b) as tarifas de pedágio de cada trecho passível de utilização.

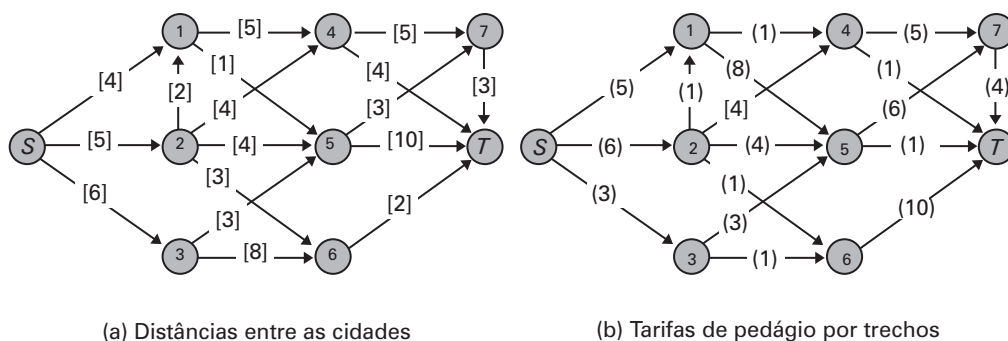


FIGURA 6.32 Distâncias e tarifas das estradas.

- ✓ Elabore um algoritmo polinomial para determinar de modo exato o caminho de menor custo entre S e T para um caso geral.
- ✓ Aplique esse algoritmo ao exemplo solucionando-o.
- ✓ Calcule a complexidade de pior caso do algoritmo elaborado.

6 – O Problema da Rede de Comunicação



A rede de computadores da Figura 6.33 é de alta segurança. Dessa forma, o desenho de sua configuração deve contemplar sempre a possibilidade da existência física de duas rotas (uma de uso corrente e outra de segurança) para ligar qualquer par de pontos da rede.

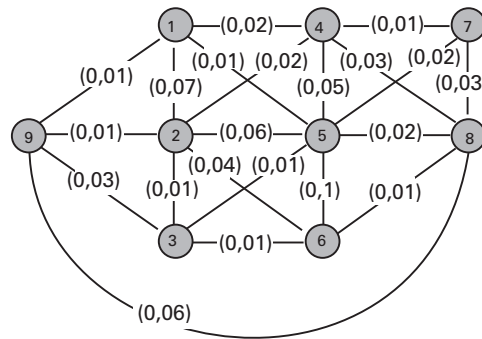


FIGURA 6.33 Rede de computadores.

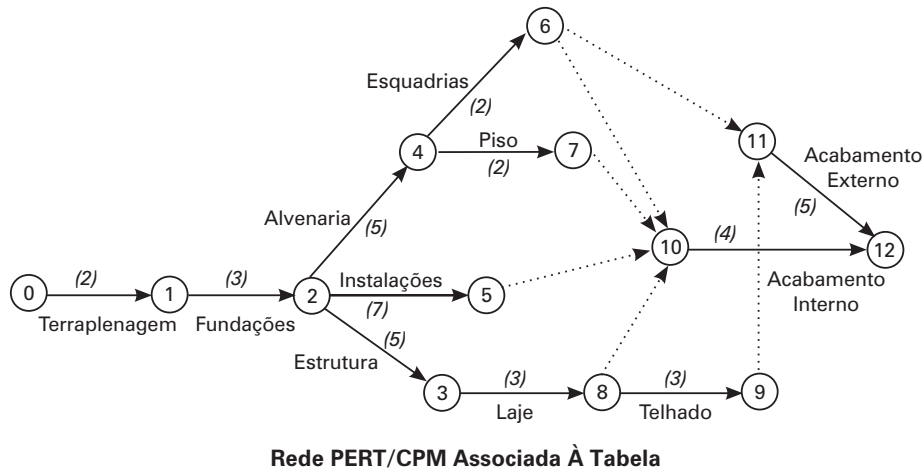
Desenvolva um algoritmo que seja capaz de determinar a solução procurada para qualquer instância desse tipo de problema. Aplique o algoritmo na rede do problema encontrando a solução desejada.

7 – Rede PERT/CPM



Desenvolva a rede PERT/CPM do problema descrito abaixo, calculando o caminho crítico e as folgas das atividades.

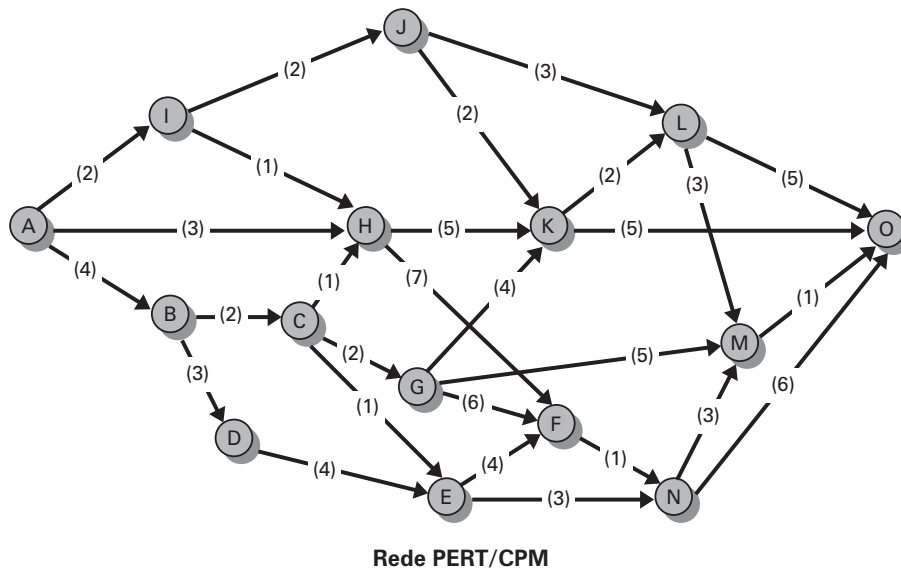
Número	Atividades	Atividade de Pré-Requisito	Duração	Arco	Tempo mais Cedo
0	Início da Obra	–	0	–	
1	Terraplenagem	0	2	0-1	t_1
2	Fundações	1	3	1-2	t_2
3	Estrutura	2	5	2-3	t_3
4	Alvenaria Não Estrutural	2	5	2-4	t_4
5	Instalações	2	7	2-5	t_5
6	Esquadrias	4	2	4-6	t_6
7	Piso	4	2	4-7	t_7
8	Laje	3	3	3-8	t_8
9	Telhado	8	3	8-9	t_9
10	Acabamento Interno	5,6,7,8	4	vários	t_{10}
11	Acabamento Externo	6,9	5	vários	t_{11}
12	Entrega da Obra	10,11	0	–	t_{12}



8 – Rede PERT/CPM



A rede PERT/CPM que se segue representa as atividades de desenvolvimento de um software. O desenvolvedor está interessado em reduzir o prazo final do projeto. Se uma atividade da rede for contratada de terceiros, seu tempo será reduzido a zero dentro do modelo.



Sabendo-se que os valores mostrados sobre as setas representam duração esperada para atividade pede-se:

1. Qual o número máximo de atividades paralelas dessa rede?
2. Qual o caminho crítico da rede?
3. Se o desenvolvedor decidiu contratar apenas uma atividade para melhorar o desempenho do desenvolvimento, qual será a escolhida? Supõe-se que a atividade contratada deve maximizar a folga final entre o tempo de projeto (o mais cedo possível) e o contratual (um certo tempo maior que o de projeto).

7

FLUXOS EM REDES E LOGÍSTICA DE DISTRIBUIÇÃO

7.1 – CARACTERÍSTICAS DOS MODELOS DE FLUXOS EM REDES

Vários e importantes modelos de programação linear possuem uma estrutura especial que permite o desenvolvimento de algoritmos eficientes e especializados em sua solução. Nessa família de problemas, um dos mais tradicionais é o *problema de transportes*. De uma maneira geral, os denominados modelos em redes permitem a solução de importantes problemas reais e são de extraordinária aplicação prática (ver casos em Boorstyn e Frank [1977], Biros e Boros [1984], Carraresi e Gallo, [1984], Langevin *et al.* [1993], Phillips e Westbrook [1993]). Na dimensão da solução, esses modelos permitem o aperfeiçoamento de conhecidas e tradicionais técnicas, de modo a alcançarem uma enorme eficiência no seu processo de resolução.

7.2 – UMA TAXONOMIA PARA OS PROBLEMAS DE FLUXOS EM REDES

7.2.1 – Uma Descrição do Problema de Fluxos em Redes

Os chamados problemas de fluxo abordam o processo de otimização da distribuição de produtos originados em pontos de oferta e consumidos em pontos de demanda dentro de uma rede de interligações possíveis. Esses problemas normalmente ocorrem dentro de plantas industriais, sistemas de comunicação e de transporte, de distribuição de água etc., contudo servem de modelo para inúmeras outras situações absolutamente diversas que lhe são assemelhadas por abstração. Normalmente, a oferta de cada produto, bem como a sua demanda, possui um valor conhecido. O processo de distribuição dos produtos não é realizado obrigatoriamente de um ponto de produção a um ponto de demanda, permitindo-se que pontos intermediários, tais como depósitos ou centros de concentração e distribuição, sejam utilizados. As interligações podem possuir restrições de capacidade de tráfego e custos variados.

7.2.2 – Formulação Matemática do Problema de Fluxos em Redes

Definimos uma *Rede* $R = (E, V, F)$ como um grafo direcionado $G = (E, V)$ atravessado por um *Fluxo* $F = \{f_1, f_2, \dots, f_m\}$ que circula em suas m arestas. Em uma rede normalmente dois nós são destacados: o *nó fonte* e o *nó sumidouro* (ver os detalhes da definição no anexo *Tópicos em Teoria dos Grafos*). Podemos representar através de equações matemáticas o problema de fluxo ilustrado na Figura B.12 do Anexo e

exemplificado na Figura B.13 do Anexo. Denominando por x_{ij} o fluxo que irá circular no arco $i-j$ podemos representar as restrições expressas nos arcos da seguinte forma:

✓ *Restrições de Capacidade dos Arcos*

Todos os arcos da rede são capacitados, mesmo quando o limite superior é $+\infty$, e o inferior $-\infty$. Nesse caso é indispensável representar os limites de fluxo dentro de cada arco. Representaremos o limite inferior ou mínimo de fluxo do arco $(i, j) \in E$ por l_{ij} e o superior ou máximo por L_{ij} . Dessa forma, as restrições de limite de fluxo serão escritas como se segue:

- ✓ Arco $i-j$
 $l_{ij} \leq x_{ij} \leq L_{ij}$
- ✓ Arco 1-2
 $0 \leq x_{12} \leq 8$
- ✓ Arco 2-5
 $1 \leq x_{25} \leq 10$ etc.

✓ *Restrições de Equilíbrio de Fluxo em Cada Nó*

Os nós da rede, com exceção dos nós fonte e sumidouro, são conservativos em relação ao fluxo, ou seja, o fluxo que chega ao nó é igual ao fluxo que deixa o nó. Os pontos de oferta e demanda podem ser também representados como pontos conservativos de fluxo, desde que associados artificialmente a um nó fonte (pontos de oferta) e a um nó sumidouro (nós de demanda). A equação geral de equilíbrio de fluxo é:

$$\left\{ \begin{array}{l} \text{Fluxo que chega} \\ \text{ao nó } i \end{array} \right\} = \left\{ \begin{array}{l} \text{Fluxo que sai} \\ \text{do nó } i \end{array} \right\}$$

Representando o fluxo que atravessa o arco $(i, j) \in E$ pela variável x_{ij} , podemos escrever as equações de equilíbrio de fluxo da seguinte forma:

✓ Equilíbrio de fluxo nos nós

$$\sum_{(k, i) \in E} x_{ki} = \sum_{(i, j) \in E} x_{ij}$$

✓ Equilíbrio do nó 2

$$x_{12} - x_{23} - x_{25} = 0$$

✓ Equilíbrio do nó 1 (arco 6-1 é o arco artificial de equilíbrio)

$$x_{61} - x_{12} - x_{14} = 0 \text{ etc.}$$

✓ *Funções Objetivo Possíveis*

No processo de otimização dos fluxos em uma rede podemos dizer que dois objetivos são muito comuns: maximizar o fluxo total que passa pela rede ou minimizar o custo de um certo fluxo. Assim:

✓ Fluxo máximo

$$\text{Maximizar } x_{61}$$

✓ Fluxo de custo mínimo

Considerando que existe um custo unitário c_{ij} associado ao trânsito de cada unidade de fluxo pelo arco x_{ij} então a função objetivo pode ser escrita como:

$$\text{Minimizar } \sum_{(i,j) \in E} c_{ij}x_{ij}$$

✓ Características da Matriz de Restrições de uma Rede

A matriz das restrições de fluxo, matriz A , possui uma configuração bastante peculiar, como podemos observar na Figura 7.1, com conferência à rede exemplificada na Figura B.13.

	<i>Arcos</i>							
<i>Nós</i>	x_{12}	x_{23}	x_{25}	x_{36}	x_{14}	x_{45}	x_{56}	x_{61}
1	+1				+1			+1
2	-1	+1	+1					
3		-1		+1				
4						+1		
5			-1			-1	+1	
6				-1	-1		-1	-1

FIGURA 7.1 Matriz das restrições de fluxo.

A configuração da matriz A , observada na Figura 7.1, lhe confere uma série de propriedades que, basicamente, serão responsáveis pelo ganho em eficiência na implementação de versões especializadas do algoritmo simplex para sua solução. Ao estruturarmos a matriz de incidência nó \times arco da Figura 7.1 adotamos a convenção de marcar o sentido de entrada do arco em um nó como negativo e o sentido de saída como positivo. Ao observarmos a coluna de x_{12} da matriz distinguimos que a direção do arco que liga os nós 1 e 2 é do nó 1 (sinal positivo) para o nó 2 (sinal negativo). A principal propriedade da matriz A é a total unimodularidade. Dizemos que uma matriz é totalmente unimodular quando seu determinante é sempre 0, 1 ou -1 para a matriz e para todas as suas submatrizes quadradas (ver teorema de Heller e Tompkins – Tópicos em Álgebra Linear e Convexidade – no anexo). Uma segunda propriedade decorrente diz respeito às soluções de um sistema $Ax = b$, onde A é totalmente unimodular. Nesse caso, as soluções do sistema obtidas pelo algoritmo simplex para tal sistema são inteiras desde que o termo independente b seja também inteiro.

✓ Formulação Geral de Fluxos em Redes

Em muitas ocasiões é interessante considerar o equilíbrio de fluxo associado a um vetor de oferta ou de demanda em cada nó. A formulação assim elaborada não necessita da consideração explícita de um nó fonte e um nó sumidouro, bem como libera a necessidade da consideração de arcos artificiais que liguem os nós de oferta a uma fonte e os de demanda a um sumidouro. De fato a essência da modelagem por fluxo em rede permanece preservada; contudo, para determinados fins, essa abordagem é mais eficiente. Nesse caso, a equação de equilíbrio de fluxos poderá ser reescrita como:

$$\left\{ \begin{array}{c} \text{Fluxo que chega} \\ \text{ao vértice} \end{array} \right\} + \left\{ \begin{array}{c} \text{Fluxo produzido} \\ \text{no vértice} \end{array} \right\} = \left\{ \begin{array}{c} \text{Fluxo que sai} \\ \text{do vértice} \end{array} \right\} + \left\{ \begin{array}{c} \text{Fluxo consumido} \\ \text{no vértice} \end{array} \right\}$$

Se denominarmos genericamente por d_i o balanço final entre o que é produzido e consumido dentro de cada nó, então podemos reescrever a formulação geral do problema de fluxo em rede da seguinte forma:

$$\begin{aligned} \text{Minimizar } z &= \sum_{(i,j) \in E} c_{ij} x_{ij} \\ \text{sujeito a:} \\ \sum_{(i,j) \in E} x_{ij} - \sum_{(k,i) \in E} x_{ki} &= d_i \quad i = 1, \dots, n \\ l_{ij} &\leq x_{ij} \leq L_{ij} \quad \forall (i,j) \in E \end{aligned}$$

A escolha da formulação mais adequada será determinada pelas circunstâncias do problema e do algoritmo de solução adotado.

✓ Formulação Restrita

De um modo geral, o problema de fluxo de custo mínimo em uma rede com $n = m + 2$ nós, onde m representa o número de nós reais e n o número de nós da rede equilibrada, poderá ser formulado também como:

$$\begin{aligned} \text{Minimizar } z &= \sum_{(i,j) \in E} c_{ij} x_{ij} \\ \text{sujeito a:} \\ \sum_{(i,j) \in E} x_{ij} - \sum_{(k,i) \in E} x_{ki} &= 0 \quad i = 1, \dots, n \\ l_{ij} &\leq x_{ij} \leq L_{ij} \quad \forall (i,j) \in E \end{aligned}$$

7.2.3 – Uma Taxonomia para o Problema de Fluxo em Redes

Os problemas em redes podem ser classificados, de um modo geral, dentro da seguinte hierarquia:

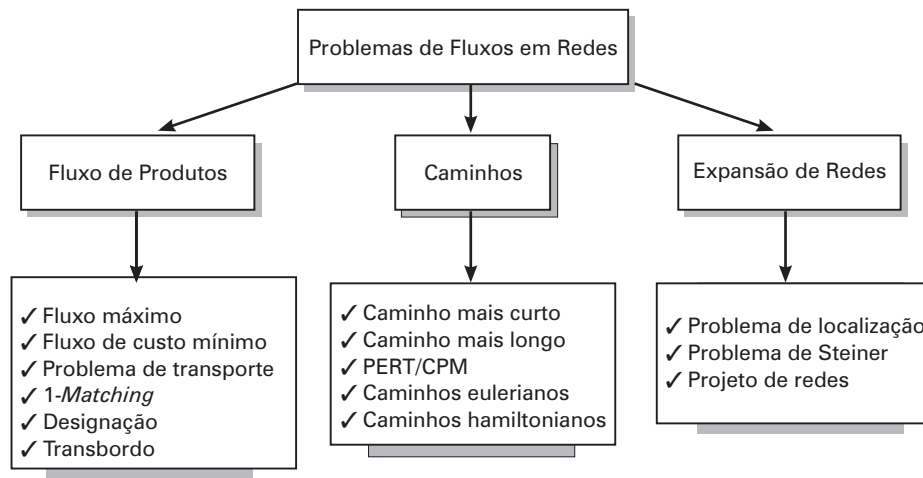


FIGURA 7.2 Taxonomia para problemas de fluxo.

7.3 – MODELOS ESPECIAIS EM REDES

7.3.1 – O Problema de Transporte

O problema de transporte é um problema de fluxo em grafo bipartido, de modo que não existem nós intermediários de transbordo ou transição para o fluxo. Na visão clássica deste problema, os arcos não possuem limite de capacidade para o fluxo. De uma forma geral a descrição gráfica do problema pode ser resumida na Figura 7.3:

Os vértices da rede da Figura 7.3 foram numerados de modo a destacar a condição de saída (vértices de oferta) e entrada de fluxo (vértices de demanda). O problema de transporte pode ser visto como um problema de fluxo em que o objetivo é minimizar globalmente os custos do fluxo através dos arcos de uma rede oferta \times demanda semelhante à da Figura 7.4.

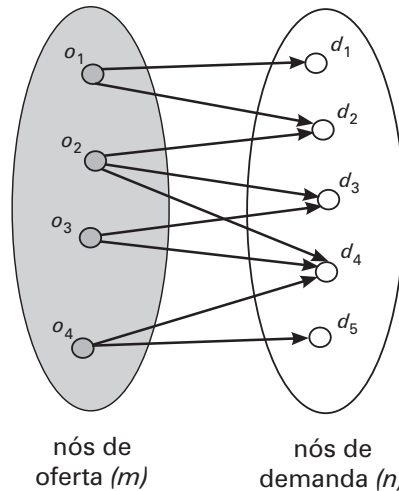


FIGURA 7.3 Característica do fluxo no problema de transporte.

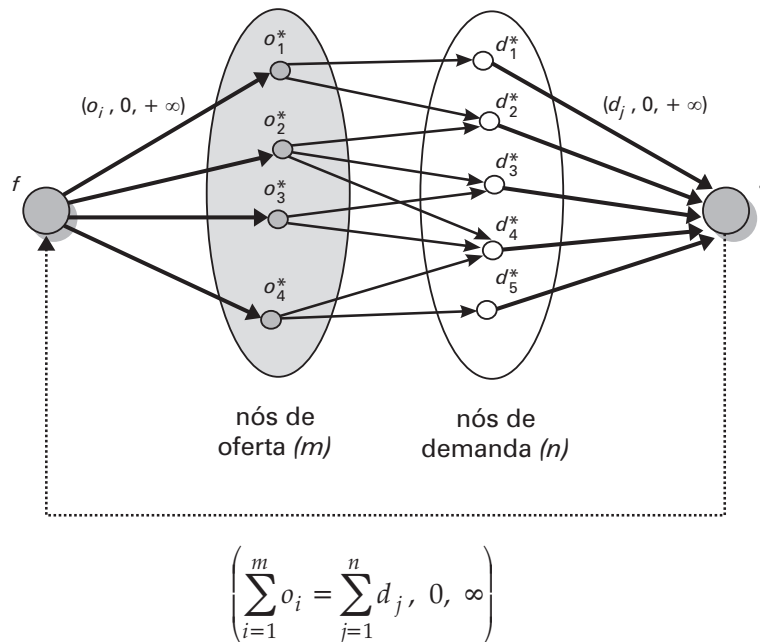


FIGURA 7.4 O Problema de transporte no formato de fluxo em redes.

✓ *Formulação Restrita para o Problema de Transporte*

Podemos também considerar o problema de transporte como ocorrendo em uma rede de fluxo conservativo. Adicionando um nó fonte (f) e um nó sumidouro (s), e considerando limites inferiores para a oferta e a demanda iguais a o_i e d_j respectivamente, podemos formular o problema de transporte como um caso particular do problema de fluxos na rede da Figura 7.4. Nessa figura, os nós de oferta e demanda estão sendo marcados com * para serem distinguidos dos valores de limite de fluxo.

Considerando um problema com $R = \{1, 2, \dots, r\}$ nós e $|R| = m + n + 2$. Denominando por $O, O = \{1, 2, \dots, m\}$ o conjunto dos nós de oferta, por $D, D = \{1, \dots, n\}$ o conjunto dos nós de demanda e por c_{ij} o custo de transporte de uma unidade de fluxo pelo arco $i-j$, temos:

$$\text{Minimizar } z = \sum_{i \in O} \sum_{j \in D} c_{ij} x_{ij}$$

sujeito a:

$$\sum_{(i,j) \in E} x_{ij} - \sum_{(k,i) \in E} x_{ki} = 0 \quad i \in V$$

$$x_{fi} = o_i \quad i \in O$$

$$x_{js} = d_j \quad j \in D$$

$$x_{sf} = \sum_{j \in D} d_j = \sum_{i \in O} o_i$$

$$x_{ij} \geq 0, \quad (i,j) \in E$$

✓ *Formulação Clássica para o Problema de Transporte*

Podemos formular o problema de transporte somente considerando as demandas e ofertas localizadas nos nós e sem os arcos e nós artificiais da seguinte forma:

$$\text{Minimizar } z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

sujeito a:

$$\sum_{j=1}^n x_{ij} = o_i \quad i = 1, \dots, m$$

$$\sum_{i=1}^m x_{ij} = d_j \quad j = 1, \dots, n$$

$$x_{ij} \geq 0 \quad i = 1, \dots, m; j = 1, \dots, n$$

Observamos que a matriz de restrições do problema de transporte na formulação clássica ainda mantém a propriedade da unimodularidade. Seja o exemplo numérico da Figura 7.5.

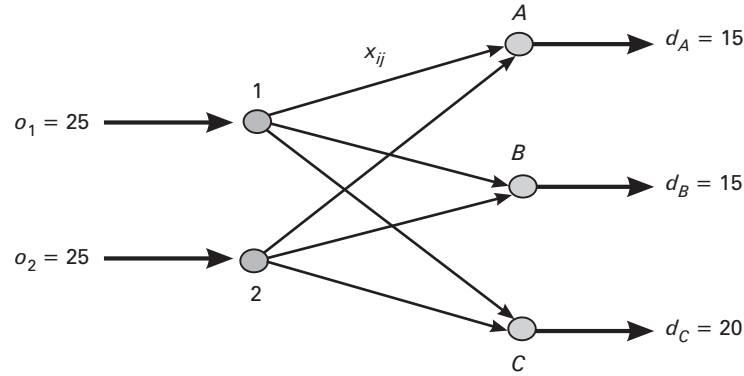


FIGURA 7.5 Exemplo 1 de problema de transportes.

A formulação geral do problema pode ser desdobrada da seguinte forma:

Minimize $z = c_{11}x_{11} + \dots + c_{1n}x_{1n} + c_{21}x_{21} + \dots + c_{2n}x_{2n} + \dots + c_{m1}x_{m1} + \dots + c_{mn}x_{mn}$
 sujeito a:

$$\begin{array}{ccccccc}
 x_{11} & \dots & x_{1n} & & & & = o_1 \\
 & & & x_{21} + \dots + x_{2n} & & & = o_2 \\
 \dots & & \dots & \dots & \dots & & \\
 & & & & & + x_{m1} + \dots + x_{mn} & = o_m \\
 x_{11} & & & + x_{21} \dots & & + x_{m1} & = d_1 \\
 & & \dots & & \dots & & \\
 & x_{1n} & \dots & + x_{2n} \dots & & \dots + x_{mn} & = d_n
 \end{array}$$

$$x_{ij} \geq 0, \quad i \in M, \quad j \in N$$

Que para o exemplo, considerando que o vetor de custos é $[c] = [2, 5, 3, 7, 4, 1]$ redundaria em:

$$\begin{array}{l}
 \text{Minimize } z = 2x_{1A} + 5x_{1B} + 3x_{1C} + 7x_{2A} + 4x_{2B} + 1x_{2C} \\
 \text{sujeito a:} \\
 \begin{array}{ccc}
 x_{1A} + x_{1B} + x_{1C} & & = 25 \\
 & x_{2A} + x_{2B} + x_{2C} & = 25 \\
 x_{1A} & + x_{2A} & = 15 \\
 x_{1B} & + x_{2B} & = 15 \\
 x_{1C} & + x_{2C} & = 20
 \end{array} \\
 x_{ij} \geq 0 \quad i \in \{1, 2\}, \quad j \in \{A, B, C\}
 \end{array}$$

As variáveis associadas ao fluxo nos arcos $i-j$ formam uma interessante matriz de restrições $A = \{a_{ij}\}$, totalmente unimodular e de $\text{rank}(A) = m + n - 1$, uma vez que existem m linhas no conjunto de restrições de oferta e n no conjunto de restrições de demanda, mas com apenas $m + n - 1$ graus de liberdade, já que $\sum_{i=1}^m o_i = \sum_{j=1}^n d_j$.

Abordagem Primal de Solução

O processo de solução desse sistema de equações pode ser bastante simplificado face a peculiaridade de sua matriz de restrições. Considerando a seguinte matriz A' , submatriz de $rank\ m + n - 1$ da matriz A do modelo decorrente do grafo da Figura 7.5, obtida pela eliminação da última restrição, bem como das duas primeiras colunas da matriz A :

$$A' = \begin{matrix} & x_{1C} & x_{2A} & x_{2B} & x_{2C} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

Nota-se que essa submatriz é uma matriz triangular superior (possui todas as posições abaixo da diagonal iguais a zero), podendo ser decomposta, portanto, em submatrizes a saber:

$$A' = \begin{bmatrix} I_m & R \\ 0 & I_{n-1} \end{bmatrix}$$

onde I representa uma matriz identidade.

Suponhamos agora que tenhamos obtido uma base B qualquer de A . Como o $rank(A) = m + n - 1$, pelo menos uma coluna de B deverá conter apenas uma única entrada positiva, ou $\det B = 0$. Nesse caso, permutando as linhas e as colunas de B de forma a colocar sua única entrada positiva na posição ocupada por I_m poderemos obter:

$$B = \begin{bmatrix} 1 & Q \\ 0 & B_{m+n-2} \end{bmatrix}$$

Aplicando novamente o mesmo raciocínio à matriz B_{m+n-2} podemos reescrever a matriz B , onde P é uma submatriz de B , da seguinte forma:

$$B = \begin{bmatrix} 1 & Q_1 & Q_2 \\ 0 & 1 & P \\ 0 & 0 & B_{m+n-3} \end{bmatrix}$$

O que demonstra que B é uma matriz triangular. Como B é uma matriz triangular existe uma certa permutação de B que soluciona o sistema $Bx_B = b$ apenas com substituições elementares de valores das variáveis. Vamos exemplificar o que apresentamos analiticamente no exemplo 1 (Figura 7.5). Podemos obter uma base de A adicionando uma variável artificial x_a ao conjunto formado pelas variáveis $x_{1C}, x_{2C}, x_{1A}, x_{1B}$

$$B^i = \begin{matrix} & x_{1C} & x_{2C} & x_{1A} & x_{1B} & x_a \\ \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix} \end{matrix} [x^B] = \begin{bmatrix} 25 \\ 25 \\ 15 \\ 15 \\ 20 \end{bmatrix}$$

que poderá ser triangularizada em:

$$B^i = \begin{matrix} & x_a & x_{1C} & x_{2C} & x_{1A} & x_{1B} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix} [x^B] = \begin{bmatrix} 25 \\ 25 \\ 15 \\ 15 \\ 20 \end{bmatrix}$$

Com essa estrutura favorável para a obtenção de uma solução para o problema da inversão de B , seria natural supor que o algoritmo simplex pudesse ser especializado para um melhor desempenho. Podemos resumir o problema no quadro da Figura 7.6, que denominaremos quadro de transporte, em que cada célula do quadro representa uma variável, ou seja, um arco de ligação, e o valor incluído no quadro menor significa o custo da célula (custo de uso do arco). Assim, para o exemplo, temos:

		Demanda					
		Índices	A	B	C	Valor	
Oferta	1	2		5		3	25
		x_{1A}		x_{1B}		x_{1C}	
	2	7		4		1	25
		x_{2A}		x_{2B}		x_{2C}	
Valor		15		15		20	

FIGURA 7.6 Quadro de transporte do exemplo 1.

Para que possamos especializar o simplex seria indispensável que pudéssemos explorar a peculiaridade do formato e das propriedades da matriz A dentro das seguintes etapas do método:

- ✓ Obtenção de uma solução inicial.
- ✓ Procedimento de melhoria da solução corrente.
- ✓ Regra de parada e de verificação da viabilidade.

Então, vamos aos pontos em questão:

Ponto 1: Obtenção da solução básica inicial

O quadro de transporte é conveniente para a determinação de uma base viável inicial para o simplex devido a uma interessante propriedade:

Proposição 7.1

A condição necessária e suficiente para que determinadas variáveis do problema de transporte sejam linearmente dependentes é que formem um ciclo no quadro de transportes.

A Proposição 7.1 pode ser verificada para o problema numérico do exemplo nos quadros abaixo:

Os traços que unem as casas hachuradas ressaltam a formação de ciclos dentro do quadro de transportes (ver grafo do ciclo – Figura 7.7 (c)). Facilmente verificamos que esses conjuntos de variáveis associados aos ciclos no quadro de transporte são linearmente dependentes (L. I.). Por outro lado, são linearmente independentes os conjuntos de variáveis que formam uma árvore (ver a definição de árvore

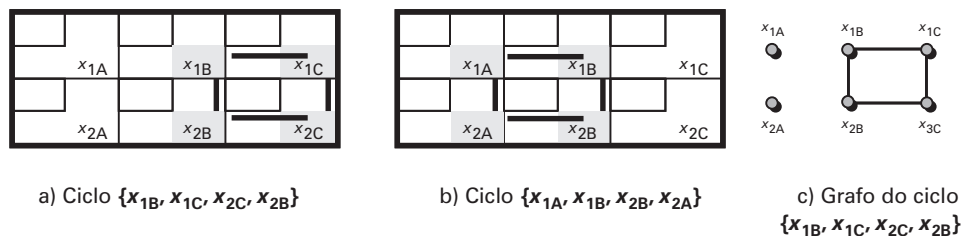


FIGURA 7.7 Representação dos ciclos no quadro de transportes.

no Anexo B) nesse mesmo quadro de transporte, quando consideramos cada célula como um nó de um grafo. Na Figura 7.8 exemplificamos dois casos de variáveis L.I. Como são L.I., tais variáveis podem ser utilizadas para compor uma base do algoritmo.

A literatura relata algumas regras para a obtenção de bases iniciais de boa qualidade, ou seja, que possuam valores próximos do valor ótimo procurado. Os dois métodos mais conhecidos são:

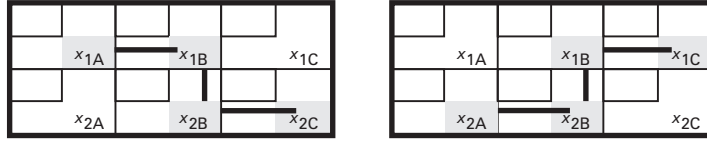


FIGURA 7.8 Exemplos de bases para o problema de transporte.

✓ Método do Canto Noroeste

1. Inicie pela célula (1, 1) do quadro de transporte.
2. Faça: $x_{11} = \min\{\tilde{o}_1, \tilde{d}_1\}$ e substitua \tilde{o}_1 por $\tilde{o}_1 - x_{11}$ e \tilde{d}_1 por $\tilde{d}_1 - x_{11}$.
 Se $\tilde{o}_1 > \tilde{d}_1$, vá para a célula (1, 2) e Faça: $x_{12} = \min\{\tilde{o}_1, \tilde{d}_2\}$ e substitua \tilde{o}_1 por $\tilde{o}_1 - x_{12}$ e \tilde{d}_2 por $\tilde{d}_2 - x_{12}$.
 Se $\tilde{o}_1 < \tilde{d}_1$ vá para a célula (2, 1) e Faça: $x_{21} = \min\{\tilde{o}_2, \tilde{d}_1\}$ e substitua \tilde{o}_2 por $\tilde{o}_2 - x_{21}$ e \tilde{d}_1 por $\tilde{d}_1 - x_{21}$.
 Se $\tilde{o}_1 = \tilde{d}_1$ (degeneração), Faça: $x_{11} = 0$ e vá para (2, 2) ou (1, 3).
3. Se uma solução já foi alcançada, pare. Caso contrário, desconsidere linhas e colunas satisfeitas e aplique as regras anteriores fazendo a célula corrente ocupar a posição da célula (1, 1).

Vamos exemplificar a obtenção da base inicial para o exemplo numérico da Figura 7.5.

2		5		3		
x_{1A}		x_{1B}		x_{1C}		25
7		4		1		
x_{2A}		x_{2B}		x_{2C}		25
15		15		20		

Quadro inicial associado à Figura 7.5

$$x_{11} = \min\{\tilde{o}_1, \tilde{d}_1\} = \min\{25, 15\} = 15 \Rightarrow \tilde{o}_1 - 15 = 25 - 15 = 10; \tilde{d}_1 - 15 = 15 - 15 = 0 \quad \tilde{o}_1 > \tilde{d}_1, \text{ então:}$$

$$x_{12} = \min\{\tilde{o}_1, \tilde{d}_2\} = \min\{10, 15\} = 10 \Rightarrow \tilde{o}_1 - 10 = 10 - 10 = 0; \tilde{d}_2 - 10 = 15 - 10 = 5 \quad \tilde{o}_2 > \tilde{d}_2$$

$$x_{22} = \min\{\tilde{o}_2, \tilde{d}_2\} = \min\{25, 5\} = 5 \Rightarrow \tilde{o}_2 - 5 = 25 - 5 = 20; \tilde{d}_2 - 5 = 5 - 5 = 0 \quad \tilde{o}_2 = \tilde{d}_3$$

$$x_{23} = 20$$

2		5		3		
	15	10		x_{1C}		25
7		4		1		
	x_{2A}	5		20		25
15		15		20		

Base inicial (árvore) associada à Figura 7.5

Que representa a seguinte base inicial:

✓ **Método de Vogel para Obtenção de uma Solução Inicial**

1. Construa o quadro de transporte de Vogel (ver Quadro 1 de Vogel). Se o total da oferta for diferente do total da demanda crie uma origem (ou destino) fictícia para que haja equilíbrio.
2. Determine a diferença entre os dois menores custos de cada linha e cada coluna.
3. Selecione a linha (ou coluna) de maior diferença. Caso ocorra empate, escolha aleatoriamente.
4. Na coluna (ou linha) escolhida, atribua, na célula de menor custo, a maior quantidade possível de suprimento.
5. Atualize as ofertas (e demandas), subtraindo a quantidade de suprimento atribuída. As linhas ou colunas em que a oferta ou demanda já foi satisfeita devem ser eliminadas nas próximas etapas.
6. Se não restar mais linhas ou colunas a serem preenchidas, pare. Caso contrário, volte ao ponto 2.

Utilizando o método de Vogel para obter a solução inicial de nosso exemplo numérico:

O problema já é equilibrado, restando-nos construir o quadro de cálculo. O quadro de Vogel é o mesmo quadro de transporte acrescido de áreas para o registro das diferenças entre os dois menores custos:

Iniciamos calculando as diferenças de custos entre as células (linhas e colunas)

Coluna A = 5; coluna B = 1; coluna C = 2; linha 1 = 3; linha 2 = 6 (ver o Quadro 1).

	A	B	C		Diferenças
1	2	5	3	25	1 ← (3 - 2)
2	7	4	1	25	3 ← (4 - 1)
	15	15	20		
Diferenças	5 (7 - 2)	1 (5 - 4)	2 (3 - 1)		

Quadro 1 de Vogel

Escolhemos a célula de menor custo da linha 2 (2, 3) para receber a maior quantidade de suprimento possível, atualizando-se as demandas e ofertas em virtude dessa ação (ver Quadro 2). A coluna 3 pode ser eliminada do quadro, recalculando-se as diferenças. A coluna A possui a maior diferença e a célula (1, 1) o menor custo.

coluna eliminada na primeira iteração

	A	B	C			Diferenças
1	2	5	3	25	1	3 (5 - 2)
2	7	4	1	5	3	3 (7 - 4)
	15	15	20			
Diferenças	5	1	2			
	5	1				

Quadro 2 de Vogel

Com as atualizações, a coluna A pode ser eliminada. O processo segue até o quadro final (Quadro 4).

coluna eliminada na segunda iteração

	A	B	C			Diferenças
1	2	5	3	10	1	3 0
2	7	4	1	5	3	3 0
	15	15	20			
Diferenças	5	1	2			
	5	1				

Quadro 3 de Vogel

	A	B	C			Diferenças
1	2	5	3	10	1	3 0
2	7	4	1	5	3	3 0
	15	10	20			
Diferenças	5	1	2			
	5	1				

Quadro 4 de Vogel

Ponto 2: Procedimento de melhoria da solução básica corrente

Sabemos que o método simplex calcula a contribuição gerada pela entrada na base de cada variável não básica, escolhendo a variável de maior contribuição para ingressar na base. Essa contribuição é obtida através do valor dos custos reduzidos, ou $z_j - c_j$ para $j \in J$, que são expressos, no caso geral do problema de programação linear, pela já conhecida relação:

$$z_j - c_j = c_B B^{-1} a_j - c_j$$

como

$$B^{-1} a_j = y_j$$

temos que

$$z_j - c_j = c_B y_j - c_j$$

Podemos reescrever a equação anterior em função do quadro de transporte, em que cada célula representa uma variável (o valor do fluxo em um arco). Nesse caso, deveremos associar um valor de contribuição pela utilização da célula como uma variável básica. Como os custos estão associados a duas dimensões, a equação dos custos reduzidos pode ser reescrita da seguinte forma:

$$z_{ij} - c_{ij} = c_B y_{ij} - c_{ij}$$

O interessante nesse caso é que a matriz de colunas transformadas y_{ij} pode ser obtida simplesmente por adições e subtrações das variáveis básicas, o que permite uma forte simplificação nos cálculos necessários. Podemos calcular a contribuição para a melhoria da solução corrente, em virtude da possível entrada de uma nova variável na base através do procedimento primal abaixo sugerido:

	j	k	s
u	+1 a_{uj}		-1 a_{us}
l		-1 a_{lk}	+1 a_{ls}
i		+1 a_{ik}	
	a_{ij}		

Cálculo $z_{ij} - c_{ij} = (c_{uj} - c_{us} + c_{ls} - c_{lk} + c_{ik}) - c_{ij}$

O cálculo é facilmente justificado uma vez que a entrada na base de uma variável, no contexto do quadro, implica a substituição de variáveis que têm valor não nulo na linha e na coluna da variável que entra. Utilizando-se essa propriedade a expressão $z_{ij} - c_{ij}$ pode ser calculada apenas através do balanço contábil. Note que a independência linear pode ser caracterizada pelo valor (+) e (-) associado às colunas das variáveis. Variáveis com entradas (-) são L.I. em relação à variável para a qual o exame foi estruturado. Vamos exemplificar o cálculo dos $z_{ij} - c_{ij}$ do quadro que se segue:

2		5		3
	x_{1A}	x_{1B}		x_{1C}
7		4		1
	x_{2A}	x_{2B}		x_{2C}

Quadro do cálculo $z_{ij} - c_{ij}$

Como existem duas variáveis fora da base x_{1C} e x_{2B} , temos de calcular $z_{1C} - c_{1C}$ e $z_{2B} - c_{2B}$ para verificar a possível contribuição decorrente da entrada na base dessas variáveis.

$$z_{1C} - c_{1C} = +2 - 7 + 1 - 3 = -7$$

$$z_{2B} - c_{2B} = +7 - 2 + 5 - 4 = +6$$

o que nos mostra que a solução básica não é ótima e x_{2B} pode entrar na base.

Resta-nos agora interpretar as condições de saída da base. No caso particular do quadro de transportes, uma variável que entra na base substitui totalmente a demanda coberta pela variável que sai. Para evitar o surgimento de inviabilidade na solução (valores negativos de fluxo) devemos considerar a substituição pelo menor valor (Δ) existente entre todas as variáveis que possuírem a atribuição +1 associada à variável que entra. No caso do exemplo:

$$(\Delta) = \text{mínimo} \{x_{2A}, x_{1B}\}.$$

Ponto 3: Regra de parada

A regra de parada será dada por $z_{ij} - c_{ij} \leq 0, \forall i, j$ do quadro.

Para concluir o nosso estudo da solução primal especializada para o problema de transporte, seria interessante confrontar o quadro do simplex com o quadro de transporte associado. Seja a base calculada pelo método do canto noroeste e seu quadro simplex associado como descritos a seguir:

2		5		3	
	15	10		x_{1C}	25
7		4		1	
	x_{2A}	5		20	25
15		15		20	

Quadro do Canto Noroeste

		x_{1A}	x_{1B}	x_{1C}	x_{2A}	x_{2B}	x_{2C}	x_a
	-120	0	0	-1	-6	0	0	0
x_{1A}	15	1	0	0	1	0	0	0
x_{1B}	10	0	1	1	-1	0	0	0
x_{2B}	5	0	0	-1	1	1	0	0
x_{2C}	20	0	0	1	0	0	1	0
x_a	0	0	0	0	0	0	0	1

Quadro simplex

Abordagem utilizando a variável dual

O método de resolução do problema de transporte é também denominado método do ciclo. De posse do quadro inicial obtido, por exemplo, pelo método do canto noroeste, o método calcula os $z_{ij} - c_{ij}$ associados às células não básicas e efetua a melhoria com a entrada de nova variável (célula ativa). O valor da variável que entra é imediatamente calculado em função da substituição da variável que sai e a distribuição do quadro atualizada.

É possível, contudo, um outro caminho para o cálculo dos $z_{ij} - c_{ij}$, da seguinte forma:

$$z_{ij} - c_{ij} = c_B B^{-1} a_{ij} - c_{ij} = c_B y_{ij} - c_{ij}$$

O elemento y_{ij} é determinado através da construção de um ciclo pela ocupação da célula $i-j$, porém sabemos que:

$$z_{ij} - c_{ij} = w a_{ij} - c_{ij}$$

Se fizermos w_i , $i = 1, \dots, m$ ser representado por u_i , e w_{m+j} , $j = 1, \dots, n$ por v_j , então o vetor dual w é dado por:

$$w = (u_1, \dots, u_m, v_1, \dots, v_n)$$

Como a_{ij} possui um 1 na i ésima e $(m + j)$ ésima posições, então $w a_{ij} = u_i + v_j$. Então:

$$z_{ij} - c_{ij} = u_i + v_j - c_{ij}$$

Esse método de calcular os custos reduzidos é denominado método das variáveis duais. Como $w = c_B B^{-1}$, então w é uma solução do sistema $wB = c_B$, lembrando que:

$$B = (a_{pq}, \dots, a_{st}, e_{m+n})$$

$$c_B = (c_{pq}, \dots, c_{st}, c_a)$$

Onde a_{pq}, \dots, a_{st} são as $m + n - 1$ colunas da base e c_{pq}, \dots, c_{st} são seus custos correspondentes. e_{m+n} é a coluna artificial de custo c_a , um vetor de zeros com uma atribuição igual a 1 na posição $m + n$. Lembrando que a variável artificial estará sempre atribuída a zero, o valor de c_a é irrelevante (normalmente é arbitrado igual a zero). Como B é uma matriz triangularizável, o sistema de equações associado às variáveis duais w é facilmente solucionável. O sistema:

$$(u_1, \dots, u_m, v_1, \dots, v_n) (a_{pq}, \dots, a_{st}, e_{m+n}) = (c_{pq}, \dots, c_{st}, 0)$$

é equivalente uma vez que $a_{ij} = e_i + e_{m+j}$. Logo:

$$\begin{aligned} u_p + v_q &= c_{pq} \\ &\vdots \\ u_s + v_t &= c_{st} \\ v_n &= 0 \end{aligned}$$

Vamos calcular os valores das variáveis duais para o quadro de nosso exemplo numérico: a variável dual v_3 é artificial e a ela será atribuído o valor zero. Daí:

$$v_c = 0$$

$$\text{para a célula } (2, C) \quad u_2 + v_c = 1 \Rightarrow u_2 = 1$$

$$\text{para a célula } (2, A) \quad u_2 + v_A = 7 \Rightarrow v_A = 6$$

$$\text{para a célula } (1, A) \quad u_1 + v_A = 2 \Rightarrow u_1 = -4$$

$$\text{para a célula } (1, B) \quad u_1 + v_B = 5 \Rightarrow v_B = 9$$

Com os valores duais podemos calcular os $z_{ij} - c_{ij}$:

$$z_{2A} - c_{2A} = u_2 + v_A - c_{2A} = 1 + 6 - 7 = 0 \text{ (variável básica)}$$

$$z_{2B} - c_{2B} = u_2 + v_B - c_{2B} = 1 + 9 - 4 = +6$$

$$z_{1C} - c_{1C} = u_1 + v_C - c_{1C} = -4 - 0 - 3 = -7$$

A Solução do Exemplo Numérico

Em nosso exemplo tanto o método do canto noroeste como o de Vogel chegaram à solução ótima. Suponhamos que a base inicial escolhida fosse outra:

✓ Obtenção de uma solução inicial (o método de escolher uma árvore qualquer no quadro de transporte):

2		5		3		
	x_{1A}		5	20		25
7		4		1		
	15	10			x_{2C}	25
15		15		20		

✓ Procedimento de melhoria da solução corrente:

De posse da base inicial obtemos \bar{x}_B e os $z_{ij} - c_{ij}$ (através do procedimento primal, por exemplo). Nesse caso:

$$z_{1A} - c_{1B} = +5 - 4 - 7 = 2 - 5 = +6$$

$$z_{2C} - c_{2C} = +4 - 5 + 3 - 1 = +1$$

✓ Regra de parada e de verificação da viabilidade:

Como existe $z_{ij} - c_{ij} \geq 0$ a base não é ótima e x_{1A} deve entrar na base. A variável que deve sair, dentre as que formam conjuntos L.D. com x_{1A} , é aquela que possui menor valor de atribuição, ou seja, $\Delta = \min\{15, 5\}$. $\Delta = 5$ implica a retirada da variável x_{1B} . O quadro evolui para:

2		5		3		
	5		x_{1B}	20		25
7		4		1		
	10	15			x_{2C}	25
15		15		20		

✓ Procedimento de melhoria da solução corrente:

Nesse caso:

$$z_{1B} - c_{1B} = +4 - 7 + 2 - 5 = -6$$

$$z_{2C} - c_{2C} = +7 - 2 + 3 - 1 = +7$$

✓ Regra de parada e de verificação da viabilidade:

Como existe $z_{ij} - c_{ij} \geq 0$ a base não é ótima e x_{2C} deve entrar na base, é aquela que possui menor valor de atribuição, ou seja, $\Delta = \min\{10, 20\}$. $\Delta = 10$ implica a retirada da variável x_{2A} . O quadro evolui para:

2		5		3		
	15		x_{1B}		10	25
7		4		1		
	x_{2A}		15		10	25
15		15		20		

✓ Procedimento de melhoria da solução corrente:

Nesse caso:

$$z_{1B} - c_{1B} = +3 - 1 + 4 - 5 = +1$$

$$z_{2A} - c_{2A} = +2 - 5 + 4 - 7 = -6$$

✓ Regra de parada e de verificação da viabilidade:

Como existe $z_{ij} - c_{ij} \geq 0$ a base não é ótima e x_{1B} deve entrar na base. A variável que deve sair, dentre as que formam conjuntos L.D. com x_{1B} , é aquela que possui menor valor de atribuição, ou seja, $\Delta = \min\{10, 15\}$. $\Delta = 10$ implica a retirada da variável x_{1C} . O quadro evolui para:

2		5		3		
	15		10		x_{1C}	25
7		4		1		
	x_{2A}		5		20	25
15		15		20		

Que já sabemos ser o quadro ótimo!

7.3.2 – O Problema de Designação (Alocação ou Atribuição)

Um dos casos mais importantes do problema de transporte se configura quando as ofertas e demandas são unitárias. Esse problema é denominado de problema de designação e pode ser assim formulado:

$$\text{Minimizar } z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

sujeito a:

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

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

$$x_{ij} \geq 0 \quad i = 1, \dots, n; j = 1, \dots, n$$

O problema de designação pode ser entendido como o problema de alocar n células de produção a n tarefas. Cada uma das células é capaz de atender à tarefa segundo um custo c_{ij} peculiar a cada uma das n células. A variável binária $x_{ij} = 1$ significa que a célula i recebeu a tarefa j como designação no esquema de trabalho, e 0 em caso contrário. A matriz A deste problema preserva as mesmas condições de unimodularidade total, sendo que a atribuição das variáveis é mais simples, sendo apenas 0 ou 1. O dual do problema de designação pode ser escrito como:

$$\text{Maximizar } w = \sum_{i=1}^n u_i + \sum_{j=1}^n v_j$$

sujeito a:

$$u_i + v_j \leq c_{ij} \quad i, j = 1, \dots, n$$

$$u_i, v_j \text{ irrestritas} \quad i, j = 1, \dots, n$$

Pelo teorema das folgas complementares sabemos que, no ótimo,

$$(c_{ij} - u_i - v_j)x_{ij} = 0, \quad i, j = 1, \dots, n$$

O que conduz às relações denominadas de “ortogonalidade”

$$\begin{cases} x_{ij} > 0 \Rightarrow u_i + v_j = c_{ij} \\ x_{ij} = 0 \Rightarrow u_i + v_j \leq c_{ij} \end{cases}$$

O que significa que, se encontrarmos um conjunto de variáveis u, v e x , satisfazendo completamente às condições de ortogonalidade, teremos encontrado o ótimo do problema. O interesse do modelo dual é tornar o mais simples possível a tarefa de identificar as condições de viabilidade. Nesse caso, a solução dual viável pode ser encontrada simplesmente por:

$$\tilde{u}_i = \min_{1 \leq j \leq n} \{c_{ij}\} \quad i = 1, \dots, n$$

$$\tilde{v}_j = \min_{1 \leq i \leq n} \{c_{ij} - \tilde{u}_i\} \quad j = 1, \dots, n$$

Como vimos para o problema de transporte, \tilde{u}_i pode ser obtido considerando-se o menor na linha i e \tilde{v}_j como o menor $c_{ij} - \tilde{u}_i$ dentro da coluna j .

Vamos agora acompanhar o planejamento de um algoritmo especializado que utilizará as informações das variáveis duais e o teorema das folgas complementares para a solução do problema de designação. A estratégia será obter uma solução dual viável (tarefa fácil) e, a cada iteração do algoritmo, reduzir as condições de ortogonalidade não satisfeitas (tarefa um pouco mais elaborada). Seja o seguinte exemplo numérico.

Desejamos alocar quatro tarefas a quatro máquinas que possuam uma matriz de custo conforme a Figura 7.9. Nessa matriz, cada célula representa o custo por alocarmos a máquina i (representada na linha i) à tarefa j (representada na coluna j).

		Tarefas (colunas)			
		1	2	3	4
Máquinas (linhas)	1	3	4	4	2
	2	7	3	5	1
	3	4	6	2	5
	4	1	2	1	1

FIGURA 7.9 Quadro de designação do exemplo numérico.

Como a viabilidade primal do problema de designação exige que toda linha e coluna tenha uma e somente uma entrada positiva ($x_{ij} = 1$), uma solução básica viável para esse problema representado na forma de um quadro de designação, como na Figura 7.10, pode ser ilustrada pelas soluções exemplificadas nos quadros da Figura 7.10, em que o símbolo • representa que a célula tem a variável a ela associada igual a 1 ($x_{ij} = 1$), permanecendo as demais células iguais a zero.

•			
	•		
		•	
			•

	•		
•			
		•	
			•

FIGURA 7.10 Soluções básicas viáveis (no primal) para o problema de designação.

As relações que envolvem as variáveis duais nos indicam que as posições candidatas à designação deveriam ser identificadas como pontos em que encontramos valores de custos mínimos, quer seja em relação às linhas (variáveis u_i) ou colunas (variáveis v_j). Para buscarmos esse tipo de configuração poderíamos desenvolver um procedimento que subtraia o menor valor dos custos dentro de cada linha e coluna. Cada valor mínimo encontrado, ao ser subtraído de toda a coluna ou linha, vai produzir pelo menos uma posição nula, posição essa que sinalizaria uma designação dual viável. Por exemplo, se o quadro de designação tivesse os valores abaixo descritos:

	1	2	3	4
1	1	3	4	5
2	4	2	4	7
3	6	7	8	3
4	5	4	2	1

Valor Mínimo na Linha

1
2
3
1

Nos conduziria, quando subtraídos de cada linha, ao quadro:

	1	2	3	4
1	0	2	3	4
2	2	0	2	5
3	3	4	5	0
4	4	3	1	0

Valor Mínimo na Coluna

0 0 1 0

que novamente subtraído do valor mínimo das colunas nos leva a:

	1	2	3	4
1	0	2	2	4
2	2	0	1	5
3	3	4	4	0
4	4	3	0	0

Podemos observar que a distribuição de células nulas (soluções duais viáveis) contém uma configuração primal viável e duas inviáveis, dentro das possíveis distribuições de quatro alocações dentro do quadro, conforme mostra a Figura 7.11.

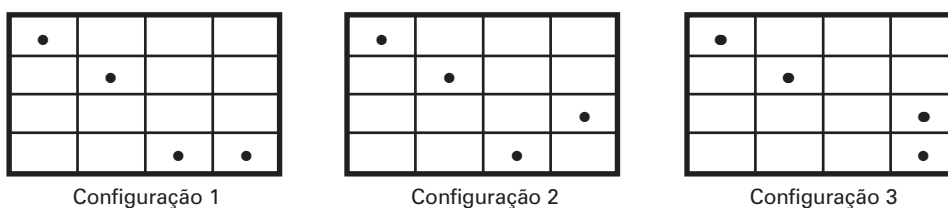


FIGURA 7.11 Alocações possíveis.

A configuração 2 é primal e dual viável, logo é ótima. A matriz resultante da subtração sucessiva dos valores dos mínimos custos pertencentes a cada linha e coluna no problema de designação é chamada de *matriz reduzida*. O número máximo de células com custo reduzido zero tal que não mais de duas ocupem a mesma linha é igual ao número mínimo de linhas necessárias para cobrir todas as células nulas da matriz. Essas células são chamadas *independentes*. Um teorema importante deriva do procedimento de obtenção de soluções duais viáveis no quadro do problema de designação:

Teorema 7.1:

O número máximo de células independentes em um quadro reduzido do problema de designação é igual ao número mínimo de linhas que podem cobrir todos os zeros da matriz.

Claramente, nem sempre a primeira solução dual viável obtida no quadro reduzido apresenta uma configuração de elementos nulos que permita a identificação de uma base primal viável. Seja o caso do quadro de transporte que se segue. Aplicando a técnica descrita para obter uma solução dual viável obtemos a matriz reduzida da seguinte forma:

	1	2	3	4	
1	3	4	4	2	2
2	7	3	5	1	1
3	4	6	2	5	2
4	1	2	1	1	1

Valor Mínimo na Linha

que leva ao quadro:

	1	2	3	4
1	1	2	2	0
2	6	2	4	0
3	2	4	0	3
4	0	1	0	0
Valor Mínimo na Coluna	0	1	0	0

que, quando subtraído do valor mínimo por colunas conduz à matriz reduzida:

1	1	2	0
6	1	4	0
2	3	0	3
0	0	0	0

Notamos que, no exemplo que estamos desenvolvendo, não existe uma solução primal viável para o quadro reduzido encontrado, uma vez que as duas primeiras linhas são cobertas por zeros localizados em uma mesma coluna. É necessário, por conseguinte, modificar essa distribuição de células alocadas dentro do quadro, no sentido de obtermos viabilidade primal. Ao desenvolvermos um algoritmo de solução com um tal propósito, estaremos interessados em encontrar uma melhoria nas condições de viabilidade primal violadas (representadas pelos zeros localizados sobre uma mesma linha ou coluna).

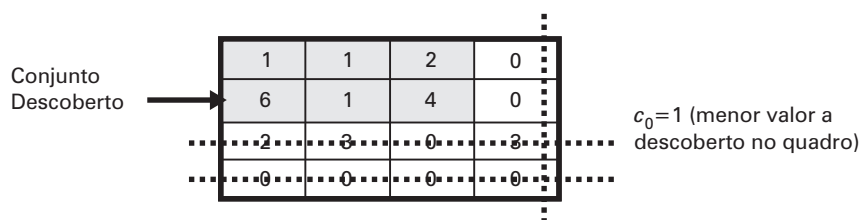
Consideremos a matriz reduzida do quadro de designação com zeros cobertos pelo menor número de linhas possível. Façamos k ser esse número de linhas necessárias. Façamos $S_r = \{i_1, i_2, \dots, p\}$ ser o conjunto das linhas não cobertas por zeros e $S_c = \{j_1, j_2, \dots, q\}$ o conjunto das colunas não cobertas. Definindo como $\tilde{S}_r = N \setminus S_r$ e $\tilde{S}_c = N \setminus S_c$, onde $N = \{1, \dots, n\}$. Finalmente, seja p o número de linhas em S_r e q o número de colunas em S_c , então $k = (m - p) + (m - q)$. Denominando por c_0 o menor elemento não coberto, ou seja:

$$c_0 = \min_{\substack{i \in S_r \\ j \in S_c}} \{ \tilde{c}_{ij} \} > 0$$

É fácil demonstrar que uma nova solução dual (modificação no quadro inviável) poderá ser obtida da seguinte forma:

$$\begin{aligned} \bar{u}_i &= \tilde{u}_i + c_0 & i \in S_r \\ \bar{u}_i &= \tilde{u}_i & i \in \tilde{S}_r \\ \bar{v}_j &= \tilde{v}_j & j \in S_c \\ \bar{v}_j &= \tilde{v}_j - c_0 & j \in \tilde{S}_c \end{aligned}$$

Resumindo: na matriz reduzida, sob o aspecto da manutenção da viabilidade dual, existe uma equivalência entre subtrair c_0 de cada linha descoberta e adicionar c_0 a cada coluna coberta. No caso em que o quadro possa possuir células que tenham dupla cobertura, é equivalente subtrair c_0 das células descobertas e acrescentá-lo às células cobertas duplamente. Como são subtraídos os valores mínimos pertencentes a S_r e S_c , o quadro permanece não negativo e cria-se pelo menos um zero em cada linha e coluna. No exemplo temos:



Utilizando essa técnica para melhorar as condições de ortogonalidade violadas devemos, portanto, subtrair 1 dos elementos descobertos e somar 1 aos elementos cobertos por dois traços, para obtermos uma nova solução dual viável. O resultado nos leva às soluções a seguir enumeradas, todas ótimas:

0	0	1	0
5	0	3	0
2	3	0	4
0	0	0	1

Solução 1

0	0	1	0
5	0	3	0
2	3	0	4
0	0	0	1

Solução 2

0	0	1	0
5	0	3	0
2	3	0	4
0	0	0	1

Solução 3

FIGURA 7.12 Alocações possíveis.

Podemos consolidar os resultados alcançados nas considerações das iterações duais em um algoritmo para o problema de designação, da seguinte forma:

Algoritmo Húngaro

1. Trace o menor número possível de linhas que cubram os zeros existentes nas colunas e linhas da matriz do quadro de designação. Se o número de linhas é n , a solução ótima foi encontrada.
2. Determine o menor elemento não coberto pelas linhas. Subtraia esse elemento de todos os demais não cobertos e o acrescente aos elementos duplamente cobertos. Volte ao passo anterior.

E para orientar o processo de cobrir as linhas e colunas de forma ótima, sugerimos o seguinte procedimento de rotulação:

Algoritmo de Rotulação

1. Marcar todas as linhas que não tenham alocações.
2. Marcar todas as colunas que tenham zeros em linhas marcadas.
3. Marcar as linhas que tenham alocações em colunas marcadas.
4. Repetir 1 e 2 até não ser mais possível marcar linhas ou colunas.
5. Traçar uma reta sobre cada linha não marcada e sobre cada coluna marcada.

Apresentaremos um exemplo completo desse algoritmo no item 7.3.4.

7.3.3 – O Problema de Transbordo

Um caso aparentado ao problema de transportes e de designação é o problema de alocação capacitado. Na verdade, esse problema une os dois anteriores ao modelar a possibilidade de um problema de

transporte ser desenvolvido com um conjunto de nós intermediários entre os pontos de oferta e demanda. A Figura 7.13 descreve esse problema de fazer o fluxo dos pontos de oferta (pontos o) chegarem aos pontos de demanda (pontos d) passando por pontos intermediários de transbordo, ou chamados de pontos de armazenagem.

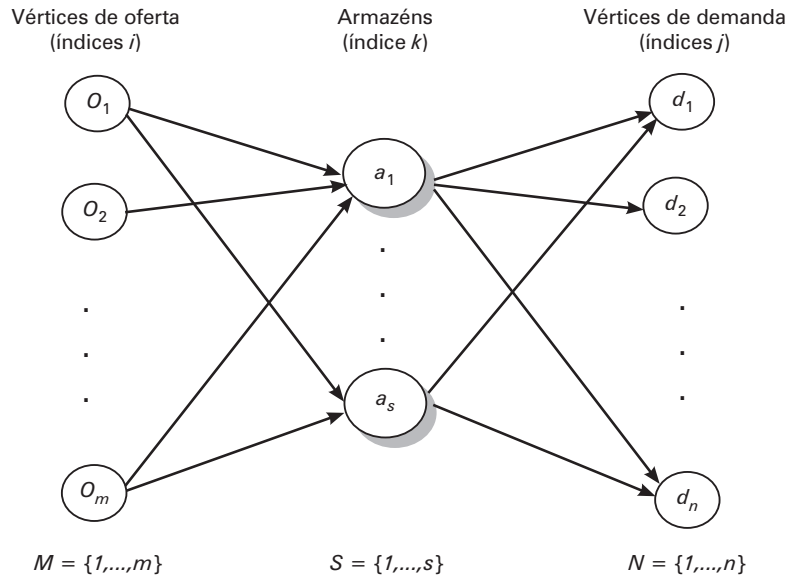


FIGURA 7.13 O fluxo do problema de transbordo não capacitado.

Considerando:

f_k \equiv custo fixo de se ativar um ponto de transbordo ou armazém k , $k \in S$, $S = \{1, \dots, s\}$, o conjunto dos pontos de oferta.

v_k \equiv variável binária que assume o valor 1 se o ponto de transbordo é ativado e 0 em caso contrário.

a_k \equiv capacidade de armazenagem disponibilizada pelo ponto de transbordo k .

x_{ik} \equiv fluxo que percorre um arco $i-k$.

y_{kj} \equiv fluxo que percorre um arco $k-j$.

c_{ik} \equiv custo de percorrer o arco $i-k$.

w_{kj} \equiv custo de percorrer o arco $k-j$.

Podemos formular o problema de transbordo não capacitado (PTNC):

$$(PTNC) \text{ Minimize } z = \sum_{i=1}^m \sum_{k=1}^s c_{ik} x_{ik} + \sum_{k=1}^s \sum_{j=1}^n w_{kj} y_{kj} + \sum_{k=1}^s f_k v_k$$

sujeito a:

$$\sum_{i=1}^s x_{ik} \leq a_k v_k \quad k = 1, \dots, s \quad (7.1)$$

$$\sum_{k=1}^s y_{kj} = d_j \quad j = 1, \dots, n \quad (7.2)$$

$$\sum_{i=1}^m x_{ik} = \sum_{j=1}^n x_{kj} \quad k = 1, \dots, s \quad (7.3)$$

$$x_{ik} \geq 0 \quad i \in M, k \in S \quad (7.4)$$

$$y_{kj} \geq 0 \quad k \in S, j \in N \quad (7.5)$$

$$v_k \in \{0, 1\} \quad k \in S \quad (7.6)$$

A restrição (7.1) descreve o processo de criação (ou abertura) de armazéns, considerando o aspecto do atendimento do fluxo demanda \times oferta. A restrição (7.2) garante o atendimento da demanda. A restrição (7.3) assegura a continuidade de fluxo. As demais restrições atendem às exigências de não negatividade e da natureza das variáveis.

A formulação anterior descreve um problema não capacitado, ou seja, sem restrições associadas aos fluxos nos arcos. Em muitas situações reais, o problema de transporte com transbordo possui limites de fluxo em seus arcos, uma vez que as rodovias, ferrovias ou mesmo aerovias possuem limitações de tráfego. Nesse caso, podemos reescrever (PTNC) utilizando a rede alternativa da Figura 7.14 que facilita a incorporação desse tipo de restrição.

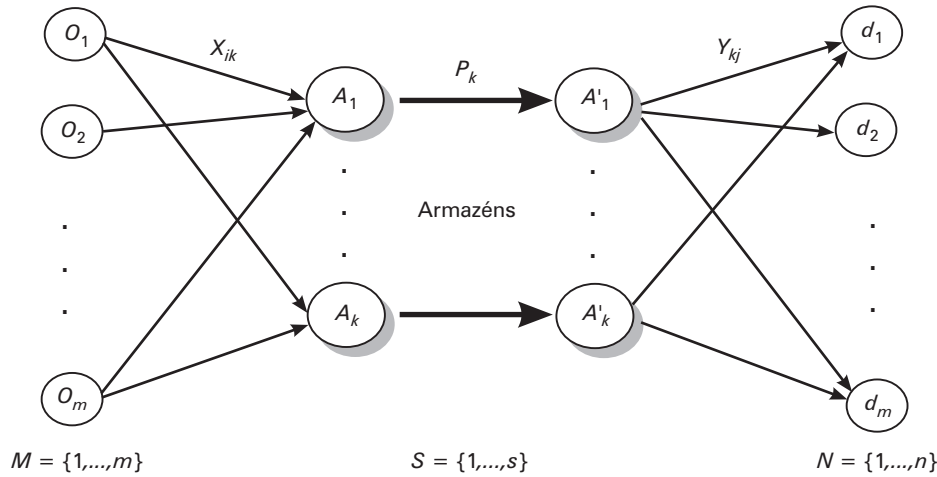


FIGURA 7.14 Rede para o problema de transbordo capacitado.

Considerando:

$g_k \equiv$ custo de armazenagem na unidade k por unidade de fluxo $k \in S, S = \{1, \dots, s\}$.

H_k e $h_k \equiv$ capacidades máximas e mínimas de armazenagem nas unidades de armazenamento k , que correspondem também a limites para o fluxo na rede.

$p_k \equiv$ fluxo circulando na unidade de armazenamento k .

$l_{ik} \equiv$ limite inferior de fluxo no arco $i-k$.

$L_{ik} \equiv$ limite superior de fluxo no arco $i-k$.

Podemos escrever o problema de transbordo capacitado (PTC) da seguinte forma:

$$(PTC) \text{ Minimizar } z = \sum_{i=1}^m \sum_{k=1}^s c_{ik} x_{ik} + \sum_{k=1}^s \sum_{j=1}^n v_{kj} y_{kj} + \sum_{k=1}^s f_k v_k + \sum_{k=1}^s g_k p_k$$

sujeito a:

$$\sum_{i=1}^m x_{ik} \leq a_k v_k \quad k = 1, \dots, s \quad (7.7)$$

$$\sum_{k=1}^s y_{kj} = d_j \quad j = 1, \dots, n \quad (7.8)$$

$$\sum_{i=1}^m x_{ik} = p_k = \sum_{j=1}^n x_{kj} \quad k = 1, \dots, s \quad (7.9)$$

$$h_k \leq p_k \leq H_k \quad k = 1, \dots, S \quad (7.10)$$

$$l_{ik} \leq x_{ik} \leq L_{ik} \quad i \in M, k \in S \quad (7.11)$$

$$l_{kj} \leq y_{kj} \leq L_{kj} \quad k \in S, j \in N \quad (7.12)$$

$$x_{ik} \geq 0 \quad i \in M, k \in S \quad (7.13)$$

$$y_{kj} \geq 0 \quad k \in S, j \in N \quad (7.14)$$

$$v_k \in \{0, 1\} \quad k \in S \quad (7.15)$$

Nesse modelo, encontramos as restrições de capacidade associadas aos arcos nas restrições (7.11) e (7.12). O fluxo nos pontos de armazenagem é controlado pelas restrições (7.10) e (7.9). A função objetivo inclui os custos de armazenagem do produto.

7.3.4 – O Problema de Designação e o 1-Matching Ponderado

O problema de 1-matching ponderado é o problema de obter, em um grafo G , um conjunto maximal independente de arestas a custo mínimo, como vimos no item 5 do Capítulo 6. Seja $G = (N, A)$, $N = \{1, \dots, 2n\}$, um grafo bipartido em dois conjuntos de nós com n componentes cada. Nesse caso o problema pode ser formulado do seguinte modo:

$$(1_MAT) \text{ Minimizar } z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

sujeito a:

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

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

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A$$

Em que:

$$x_{ij} \equiv \begin{cases} 1 & \text{caso a aresta } (i, j) \text{ seja incluída na solução} \\ 0 & \text{caso contrário} \end{cases}$$

$$c_{ij} \equiv \text{O custo da aresta } (i, j) \text{ ser escolhida para integrar a solução.}$$

Conforme o modelo desenvolvido no item 7.3.2, o modelo (1_MAT) representa também o problema de designação. No caso do grafo bipartido, em que os dois conjuntos de vértices possuem a mesma cardinalidade, podemos solucionar o problema de designação como um 1-matching ponderado, e o fazer, se assim o desejarmos, através de um modelo por fluxo em rede, segundo o esquema da Figura 7.15.

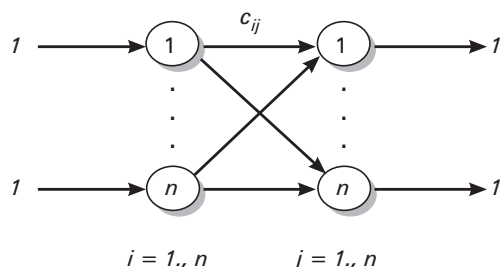


FIGURA 7.15 1-Matching ponderado modelado por fluxo em redes.

Esse modelo poderá ser também solucionado pelo algoritmo húngaro anteriormente desenvolvido. Como exercício solucionaremos o 1_matching ponderado do grafo da Figura 7.16, associado à matriz de custo da Figura 7.17, como um problema de designação, mostrando a associação dos passos do algoritmo à solução de fluxo sugerida pelo modelo da Figura 7.15.

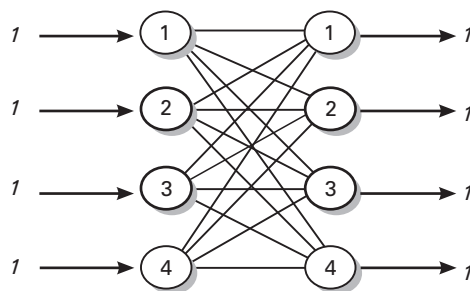


FIGURA 7.16 Quadro de cálculo do algoritmo húngaro.

Para um problema como esse, que possui uma representação em grafos de fácil visualização, podemos usar um diagrama que permitirá acompanharmos as soluções duais e primais a cada iteração do método. A matriz de custos $C = [c_{ij}]$ aparece representada à esquerda, com os vetores duais na lateral esquerda e abaixo. O fluxo associado é representado à direita, como na Figura 7.15.

	1	2	3	4	u_i	
1	2	3	5	8	2	1 → 1
2	7	5	1	4	1	1 → 2
3	3	2	5	6	2	1 → 3
4	7	3	2	1	1	1 → 4
v_j	0	0	0	0		

Quadro Inicial

Fluxo Associado

FIGURA 7.17 Exemplo numérico.

Nesse caso, já na primeira iteração do algoritmo húngaro alcançamos um fluxo primal e dual viável (ótimo) como mostra a Figura 7.18.

Seja agora o exemplo numérico sugerido pela nova matriz de custos da Figura 7.20.

Quando aplicamos o algoritmo húngaro para o caso do exemplo da Figura 7.20 não obtemos na primeira distribuição dos 0s, as condições necessárias de ortogonalidade como mostra a Figura 7.21.

Como não é possível obter uma solução primal viável, o fluxo gerado não corresponde a um emparelhamento (ver Figura 7.22).

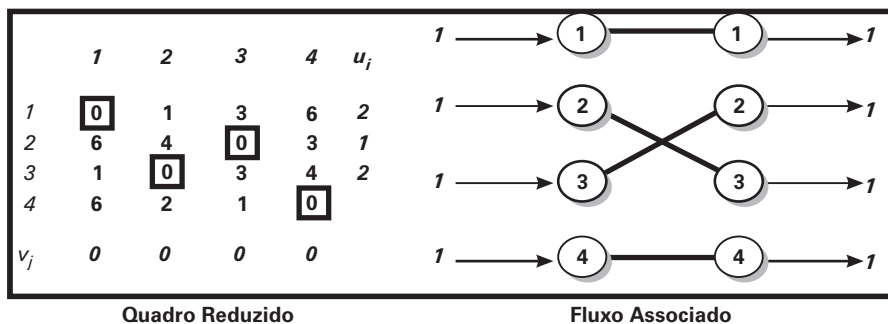


FIGURA 7.19 Fluxo viável.

	1	2	3	4	u_i
1	94	1	54	68	1
2	74	10	88	82	10
3	62	88	8	76	8
4	11	74	81	21	11
v_j	0	0	0	0	

(cálculo dos u_i)

FIGURA 7.20 Segundo exemplo numérico.

	1	2	3	4	u_i
1	93	0	53	67	1
2	64	0	78	72	10
3	54	80	0	68	8
4	0	63	70	10	11
v_j	0	0	0	0	

(cálculo dos v_j)
A quarta coluna viola condições de ortogonalidade

FIGURA 7.21 Primeira iteração do algoritmo húngaro.

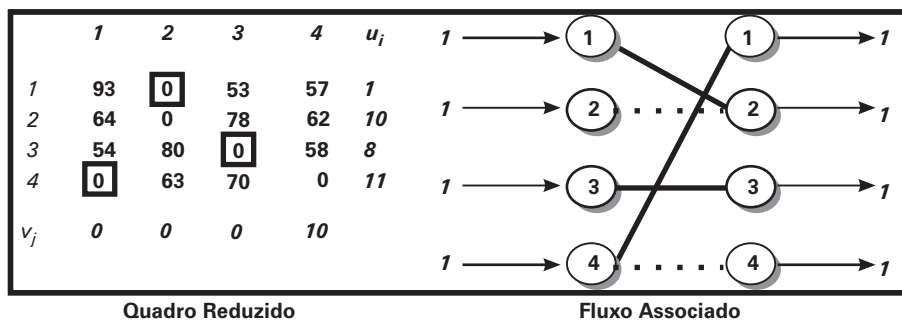


FIGURA 7.22 Fluxo associado à solução primal inviável.

Aplicando o procedimento de rotulação, necessitamos de três linhas para cobrir os zeros ($n = 3 \neq 4$), o que nos indica que o quadro não é ótimo. Daí encontramos:

	1	2	3	4	u_i
1	93	0	53	57	1
2	64	0	78	62	10
3	54	80	0	58	8
4	0	60	70	0	11
v_j	0	0	0	10	

53 é o menor valor descoberto

FIGURA 7.23 Primeira rotulação do algoritmo húngaro.

que nos leva ao quadro da Figura 7.24:

	1	2	3	4	u_i
1	40	0	0	4	0
2	11	0	25	9	0
3	54	133	0	58	0
4	0	116	70	10	0
v_j	0	0	0	0	

FIGURA 7.24 Quadro de cálculo após a primeira rotulação.

que é ainda inviável como podemos ver na Figura 7.25.

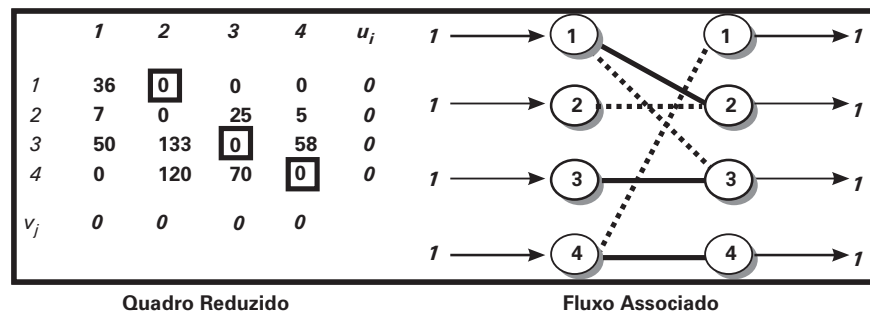


FIGURA 7.25 Segundo fluxo inviável.

Prosseguindo a aplicação do algoritmo temos:

	1	2	3	4	u_i
1	40	0	0	4	0
2	11	0	25	9	0
3	54	133	0	58	0
4	0	116	70	10	0
v_j	0	0	0	0	

4 é o menor valor descoberto

FIGURA 7.26 Segunda rotulação.

A segunda rotulação nos conduz, finalmente, a um fluxo viável, como mostra a Figura 7.27:

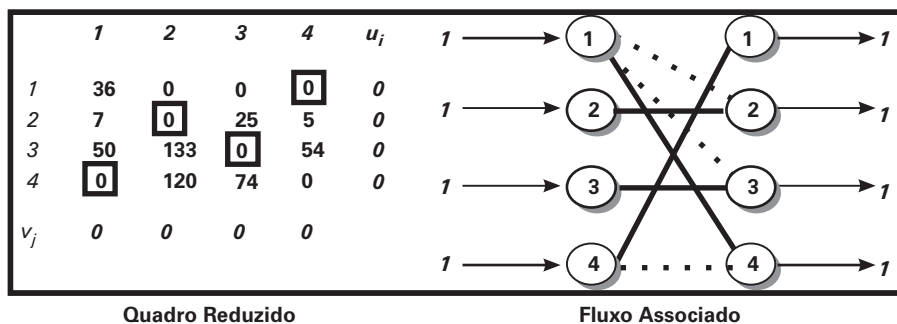


FIGURA 7.27 Fluxo viável.

A distribuição viável é identificada pelo procedimento de rotulação do algoritmo, como mostra a Figura 7.28:

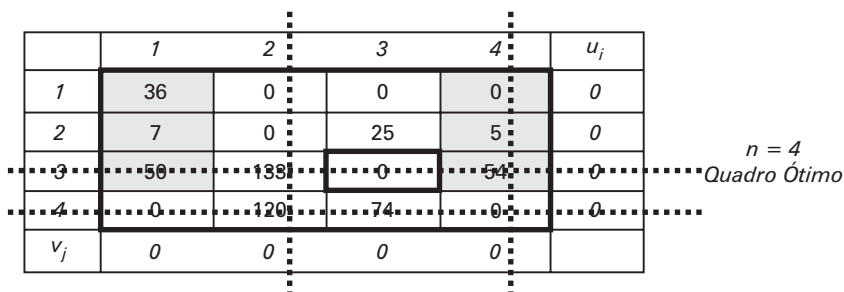


FIGURA 7.28 Quadro ótimo.

7.3.5 – O Problema do Caminho mais Curto s - t em Modelo de Redes

Este problema, quando consideramos os comprimentos das arestas ou arcos de um grafo como custos, pode ser visto como um caso do transbordo não capacitado, como mostra a Figura 7.29.

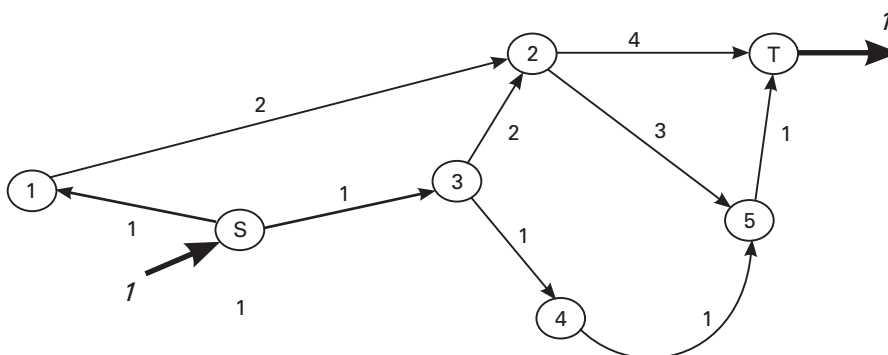


FIGURA 7.29 Caminho mais curto em modelo de redes.

Em uma rede com n nós, em que s é o nó sumidouro e t o nó fonte, a formulação matemática desse problema pode ser resumida da seguinte forma:

$$(CC) \text{ Minimizar } z = \sum_{(i,j) \in A} c_{ij} x_{ij}$$

sujeito a:

$$\sum_{(i,j) \in A} x_{ij} - \sum_{(i,j) \in A} x_{ji} = \begin{cases} 1 & \text{se } i = s \text{ (nó fonte)} \\ 0 & \text{demais casos} \\ -1 & \text{se } i = t \text{ (nó sumidouro)} \end{cases}$$

$$x_{ij} \geq 0 \quad \forall (i,j) \in A$$

onde x_{ij} representa o fluxo circulando no arco $(i,j) \in A$. No caso do exemplo da Figura 7.29 teremos associado o quadro de restrições da Figura 7.30.

Nó – Arco	x_{s1}	x_{s3}	x_{12}	x_{32}	x_{34}	x_{45}	x_{25}	x_{2t}	x_{5t}	Restrição	Fluxo
S	1	1								=	1
1	-1		1							=	0
2		-1	-1	-1			1	1		=	0
3				1	1					=	0
4					-1	1				=	0
5						-1	-1		1	=	0
T								-1	-1	=	-1
Custo	1	1	2	2	1	1	3	4	1		

FIGURA 7.30 Matriz de fluxo para o problema do caminho mais curto s-t.

A solução do caminho mais curto no grafo da Figura 7.29 é s-3-4-5-t.

7.4 – PROBLEMAS DE FLUXO

Os mais tradicionais problemas no contexto da otimização de fluxos em rede são:

- Problema de circulação.
- Problema de fluxo máximo.
- Problema de fluxo de custo mínimo.

Estudaremos alguns dos algoritmos baseados na teoria dos grafos para a solução dos problemas de fluxo máximo. Para o caso do fluxo de custo mínimo estudaremos a vertente dos algoritmos do tipo primal-dual baseados em especialização do algoritmo simplex.

7.4.1 – O Problema de Circulação

O problema de circulação é um problema básico na área de fluxos. Seja na hipótese da maximização do fluxo passante em uma rede, seja quando o objetivo for determinar um fluxo de custo mínimo, antes de tudo, é necessário dominar as condições de circulação desse fluxo na rede.

✓ Conceitos Básicos

O primeiro conceito a examinar dentro do problema de circulação diz respeito à sua forma viável. Um fluxo viável deve respeitar tanto as condições de capacidade dos arcos, ou não será capaz de fluir, como, no caso de modelos lineares e conservativos, as restrições de conservação de seu valor. A primeira lei de Kirchhoff é especialmente útil na determinação de fluxos viáveis em redes capacitadas.

Considerando uma rede em que somente os nós s e t possuem oferta e demanda de fluxo, a denominada primeira lei de Kirschhoff para o fluxo é verificada. Não existe qualquer perda de generalidade ao considerarmos que todos os nós da rede, com exceção de s e t , são conservativos, uma vez que possíveis nós de oferta ou demanda poderão ser representados por um par de nós conservativos em relação ao fluxo ligados aos nós s e t com um arco na estrita capacidade de sua demanda, como mostra a Figura 7.31. Na figura, os parênteses associados a alguns dos arcos artificiais representam o valor do limite inferior para o fluxo no arco. Em todo nó conservativo de uma rede $R = (E, V, F)$ aplica-se a primeira lei de Kirschhoff:

Primeira lei de Kirschhoff

A soma dos fluxos que entram em um nó da rede é igual à soma dos fluxos que saem.

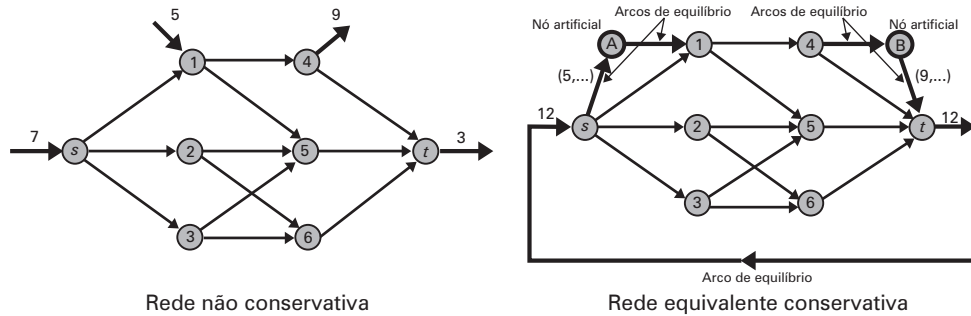


FIGURA 7.31 Transformação para nós de oferta e demanda.

Podemos aplicar a primeira lei de Kirschhoff para analisar o fluxo representado na Figura 7.32:

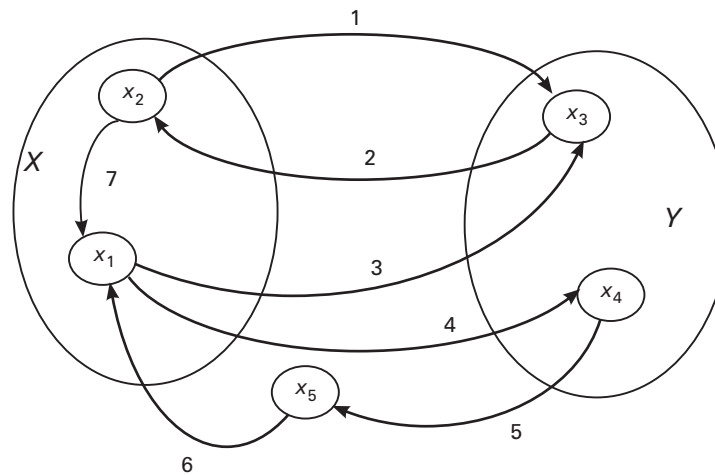


FIGURA 7.32 Fluxo entre conjuntos de nós.

Sabendo que o conjunto dos nós do grafo da Figura 7.32 é $V = \{x_1, x_2, x_3, x_4, x_5\}$ e representando por f um determinado fluxo, ou seja:

$$\begin{aligned} f(X, Y) &= f_1 + f_3 + f_4 \\ f(Y, X) &= f_2 + f_6 = f_2 + f_5 \end{aligned}$$

Podemos enunciar a lei de conservação de fluxo como:

$$f(x, V) = f(V, x), \quad \forall x \in V$$

De fato, exibir um fluxo viável em uma rede é uma tarefa elementar quando os arcos possuem limites mínimos de fluxo iguais a zero. Nessa hipótese, o fluxo nulo apresenta-se como solução trivial. Entretanto, nem sempre encontramos tal possibilidade na prática. Em inúmeros casos, os arcos da rede possuem imposições de fluxo diferentes de zero. Para a exibição de um fluxo que atenda às restrições de limites superiores e inferiores de fluxo nos arcos em qualquer rede, é necessário dispor-se de um algoritmo. O algoritmo que se segue está baseado no princípio da conservação de fluxo e será sempre capaz de exibir, se existir, um fluxo viável em uma rede capacitada.

Definindo a distância de $f(i, j)$ ao intervalo $[l_{ij}, L_{ij}]$ como sendo:

$$d(i, j) = \begin{cases} 0, & \text{se } l_{ij} \leq f(i, j) \leq L_{ij} \\ l_{ij} - f(i, j), & \text{se } f(i, j) < l_{ij} \\ f(i, j) - L_{ij}, & \text{se } f(i, j) > L_{ij} \end{cases}$$

$$\text{Seja } d(f) = \sum_{(i, j) \in A} d(i, j).$$

Então, se $d(f) = 0$, f é um fluxo viável. Caso contrário, existe pelo menos um arco $(i, j) \in A$ com $d(i, j) > 0$. Baseado nesse fato, o algoritmo abaixo vai buscar a viabilidade dos fluxos nos arcos ou provar que essa condição não existe.

✓ Algoritmo de Circulação

Passo 1: (inicialização)

- Construir uma circulação f que satisfaça à equação de conservação em todos os nós da rede, ou seja, $f(x, V) - f(V, x) = 0$ para $\forall x \in V$. Pode-se utilizar a circulação $f = 0$ em qualquer caso. Vá para o Passo 2.

Passo 2: (regra de parada)

- Se $d(f) = 0$, então f é uma circulação viável. Fim.

Caso contrário, selecione um arco $(x, y) \in E$, tal que $d(x, y) > 0$.

Passo 3: (regra de rotulação)

- Se $f(x, y) > L(x, y)$ então, partindo do nó i , alcançar o nó j , rotulando os nós pelo seguinte processo:
- Iniciar rotulando o nó i com $[i^-, \xi(i) = f(x, y) - L(x, y)]$. A seguir, selecione um nó rotulado ainda não examinado i , e rotular os nós não rotulados j utilizando as seguintes regras:
 - Se (i, j) é um arco com $f(i, j) < L(i, j)$ rotular j com:

$$[i^+, \xi(j) = \min \{ \xi(i), L(i, j) - f(i, j) \}]$$

- Se (i, j) é um arco com $f(i, j) > l(i, j)$ rotular j com:

$$[i^+, \xi(j) = \min \{ \xi(i), f(i, j) - l(i, j) \}]$$

Continuar a rotulação até que uma das situações ocorra:

- O nó y é alcançado e rotulado. Nesse caso, modificar o fluxo nos arcos do ciclo de x para y , formado pelos nós rotulados da seguinte forma:
 - Partindo de x , percorrer o ciclo subtraindo $\xi(y)$ em arcos reversos e somando em arcos diretos. Ao alcançar y , subtrair $\xi(y)$ de $f(x, y)$. Voltar ao Passo 2.
 - Se o nó y não puder ser alcançado e rotulado não existe circulação viável. Fim.
 - Se $f(x, y) < l(x, y)$ então rotular o nó y com:

$$[x^+, \xi(y) = l(x, y) - f(x, y)]$$

e tentar alcançar x partindo de y , aplicando as mesmas regras apresentadas para o caso anterior.

– Continuar a rotulação até que uma das seguintes condições ocorra:

- Se o nó x for rotulado e, portanto, um ciclo de y para x foi encontrado, modificar o fluxo somando $\xi(x)$ ao fluxo em arcos reversos e subtraindo $\xi(x)$ em arcos diretos deste ciclo e, finalmente, somar $\xi(x)$ a $f(x, y)$. Voltar ao Passo 2.
- Se o nó x não for rotulado não existe circulação viável. Pare.

✓ Exemplo Numérico de Aplicação do Algoritmo de Circulação

Seja a rede da Figura 7.33. Desejamos viabilizar, se possível, o fluxo entre os nós da rede. Os valores entre colchetes representam os limites do fluxo para cada arco [limite inferior, limite superior], e os valores entre parênteses representam o fluxo no arco.

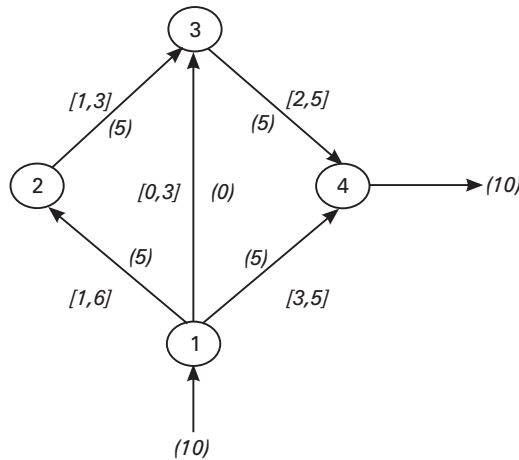


FIGURA 7.33 Rede com fluxo inviável.

O fluxo indicado para o arco (2, 3) viola o limite superior do arco, o que significa que o fluxo da Figura 7.33 não é viável. Aplicando o algoritmo de rotulação a partir do nó 1, considerando o caso do nó i , teríamos o observado na Figura 3.34.

Se o nó y é alcançado e rotulado é porque existe um ciclo que inclui o arco (x, y) e as alterações indicadas no fluxo produzem um novo fluxo conservativo que atende à equação:

$$[f'(x, y) = f(x, y) \pm \xi(y)]$$

Como $\xi(y) > 0$, a cada ciclo encontrado o fluxo aproxima-se em $\xi(y)$ unidades da situação de viabilidade nos arcos inviáveis, sendo mantidos ainda viáveis os fluxos em todos os arcos (i, j) do ciclo para os quais $l(i, j) \leq f(i, j) \leq L(i, j)$.

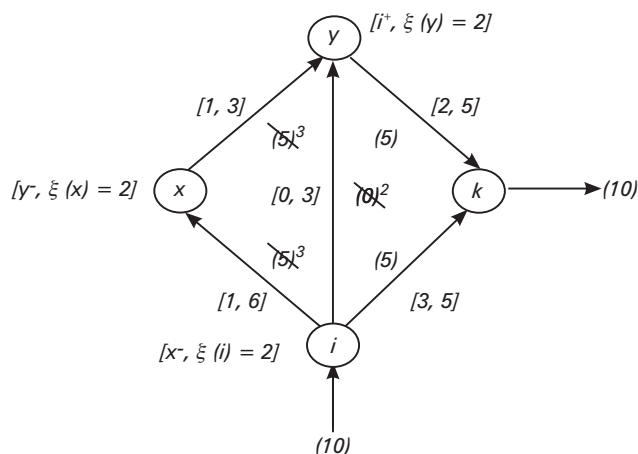


FIGURA 7.34 Rede com fluxo viável.

7.4.2 – O Problema de Fluxo Máximo

Este problema corresponde a forçar, por uma certa rede sujeita a limites de capacidade de circulação nos arcos, o maior fluxo possível. A Figura 7.35 exemplifica o problema, em que os colchetes expressam os valores mínimo e máximo para o fluxo em cada arco.

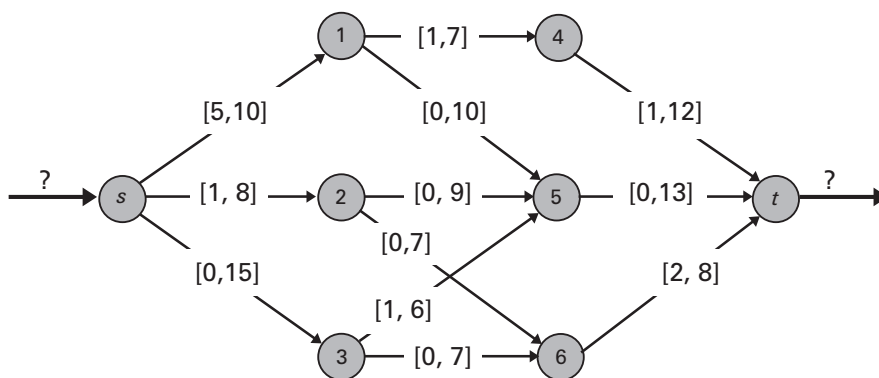


FIGURA 7.35 Fluxo máximo desconhecido.

Antes de apresentarmos alguns algoritmos de solução para o problema, cumpre abordar certos aspectos teóricos e de nomenclatura.

Conceitos Básicos

✓ Corte s-t:

Considerando a rede R , cujo grafo de substrato é $G = (V, E)$ e (X, \bar{X}) uma partição de V tal que $X \cup \bar{X} = V$ e $X \cap \bar{X} = \emptyset$, diremos que (X, \bar{X}) define um corte s-t em $G = (V, E)$. A Figura 7.36 exemplifica um corte s-t, em que $X = \{s, 1, 2, 3, 4\}$, $\bar{X} = \{2, t\}$.

Pelo teorema da circulação a existência de um fluxo implica:

$$L(X, \bar{X}) \geq l(X, \bar{X}), \quad \forall X, \bar{X} \subseteq V$$

onde $L(X, \bar{X})$ e $l(X, \bar{X})$ representam os limites inferior e superior do fluxo total entre o conjunto de nós X e \bar{X} . Nesse caso concluímos que:

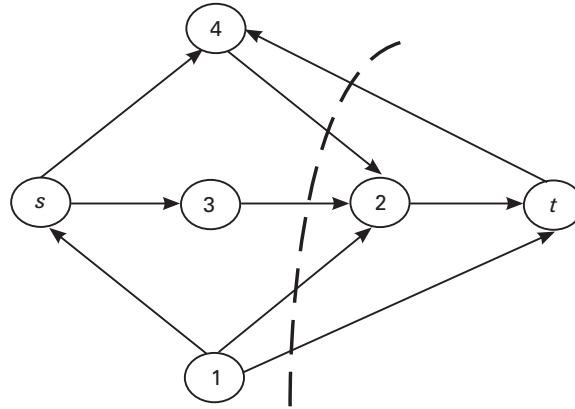


FIGURA 7.36 Um corte s-t.

1. O valor do fluxo máximo f_0^* na rede s - t deve ser menor ou igual à capacidade líquida de qualquer corte s - t ou seja:

$$f_0^* \leq L(X, \bar{X}) - l(\bar{X}, X) \quad \forall \text{ corte } s-t$$

2. Todo fluxo f^* na rede s - t , tal que: $f_0^* = \min_{\text{corte } s-t (X, \bar{X})} \{L(X, \bar{X}) - l(\bar{X}, X)\}$ é máximo.

✓ Grafo de Aumento de Fluxo

Denominaremos por grafo de aumento de fluxo ao grafo $G_f = (V, E_f)$ que, possuindo somente arcos simples, é construído da seguinte forma:

- $(x, y) \in E_f$ se $(x, y) \in E$ e $f(x, y) < L(x, y)$
- $(y, x) \in E_f$ se $(y, x) \in E$ e $f(x, y) > l(x, y)$

✓ Folga de um Arco

Denominamos folga do arco (x, y) aos seguintes valores:

- $\bar{\xi}(x, y) = L(x, y) - f(x, y)$ se $f(x, y) < L(x, y)$
- $\tilde{\xi}(x, y) = f(x, y) - l(x, y)$ se $f(x, y) > l(x, y)$

Um arco (x, y) pode ter tanto $\bar{\xi}$ como $\tilde{\xi}$.

✓ Caminho de Aumento de Fluxo

É um caminho em G_f que liga diretamente s a t . Observe que se existir um caminho ligando s a t em G_f é porque existe um caminho em G que, em relação ao fluxo corrente, possui uma folga que pode ser calculada por $\xi = \min \{\bar{\xi}, \tilde{\xi}\}$. A conservação dos fluxos nos vértices não será afetada se aumentarmos de ξ o fluxo nos arcos diretos e subtrairmos de ξ o fluxo nos arcos reversos do caminho de aumento de fluxo. A Figura 7.37 esclarece esse processo:

Teorema 7.2:

Dado um fluxo f , se existir um caminho direto de s para t , em um grafo G_f , então f não é fluxo máximo.

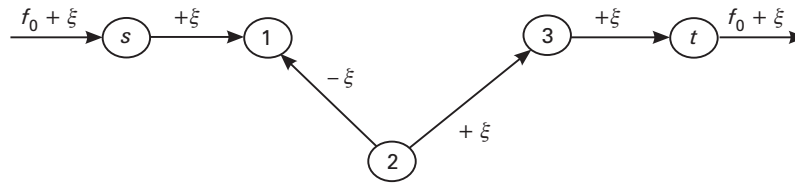
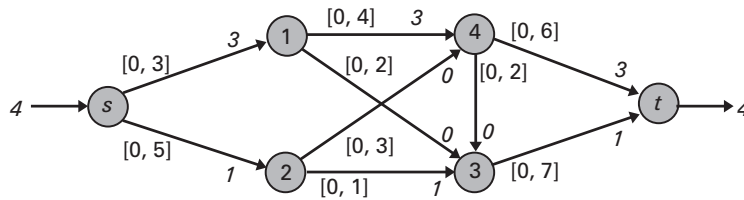
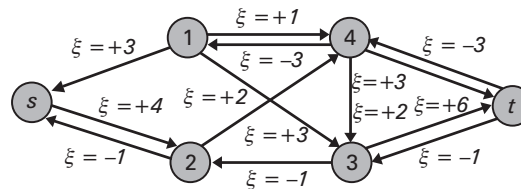


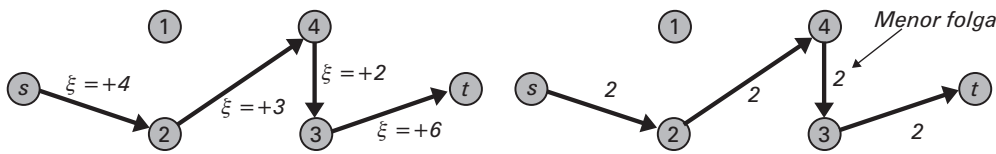
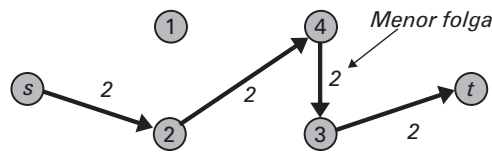
FIGURA 7.37 Aumento de Fluxo.

✓ **Algoritmo Ford-Fulkerson – Utilizando Caminhos Diretos em G_f**

Qualquer caminho existente no grafo de aumento de fluxo que ligue s a t pode incrementar o fluxo total de s para t no valor da menor folga existente nesse caminho. Um fluxo que circule entre s e t com esse valor é chamado um fluxo aumentante. Vamos exemplificar o processo de obtenção de uma sequência de fluxos aumentantes no grafo G_f da Figura 7.38. A Figura 7.38 (b) e 7.38 (c) mostra a identificação de um fluxo aumentante.

(a) Rede atravessada por um fluxo $f_1 = 4$ 

(b) Grafo de aumento de fluxo associado

(c) Caminho direto $s-t$ no grafo G_f 

(d) Aumento de fluxo

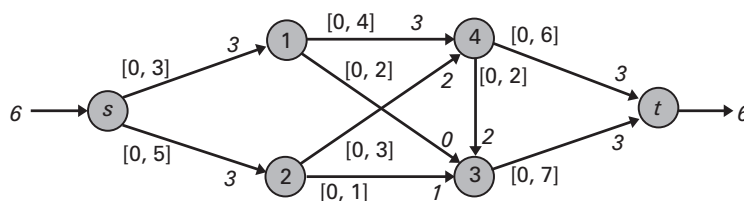
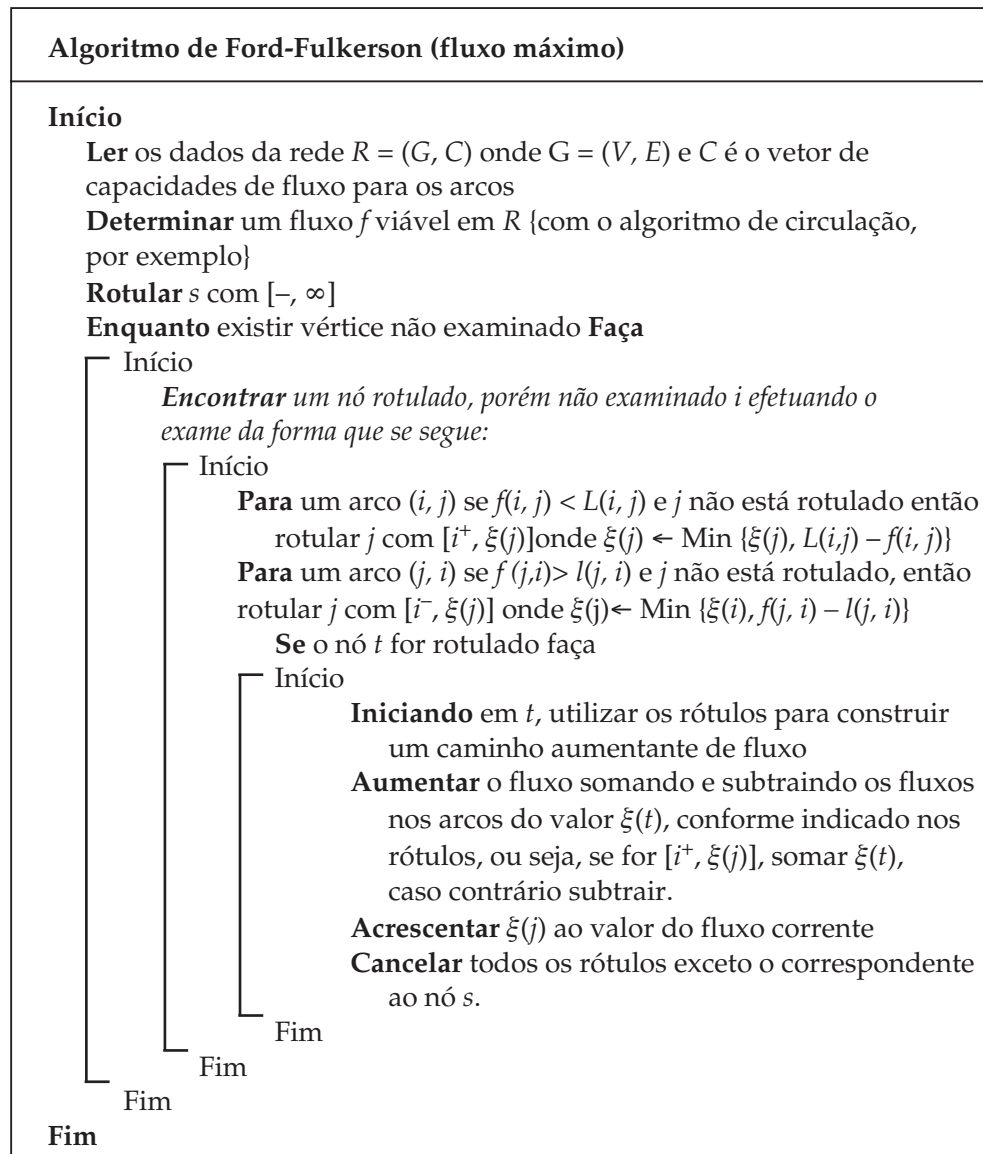
(e) Rede atravessada por um fluxo $f_2 = 6$

FIGURA 7.38 A identificação de um fluxo aumentante.

Podemos descrever um algoritmo baseado em uma sequência de fluxos aumentantes da seguinte forma:



O algoritmo Ford-Fulkerson ($F \& F$) termina quando não é mais possível rotular o nó t . Nesse caso, os nós rotulados e os não rotulados definem também um corte de fluxo mínimo em R . Esse algoritmo pode apresentar um comportamento bastante ineficiente ou mesmo não convergir no caso de as capacidades nos arcos possuírem valores irracionais. No caso de a capacidade dos arcos ser inteira então a complexidade de $F\&F$ será $O(mf_{\text{máx}})$. A rede da Figura 7.39 mostra um caso em que o algoritmo $F\&F$ torna-se extremamente ineficiente:

É possível que o algoritmo escolha alternadamente $(s, 1, 2, t)$ e $(s, 2, 1, t)$ como caminhos de aumento de fluxo. Nesse caso, serão necessários $2 \cdot 10^n$ operações de aumento de fluxo para encontrar o fluxo máximo. É claro que se a escolha do caminho for $(s, 2, t)$ e $(s, 1, t)$ serão necessárias apenas duas iterações, contudo nada no algoritmo $F\&F$ básico garante que o caso patológico não acontecerá. É fácil perceber que o problema do caso patológico reside no comprimento do caminho aumentante. Se o algoritmo realizar o aumento de fluxo no caminho que apresenta o menor número possível de arcos então a complexidade de $F\&F$ será $O(m^2n)$ (ver Dinic [1970]). Isso pode ser obtido modificando a sequência de rotulação e exame de modo que os nós sejam examinados na mesma ordem em que tiverem recebido seus r

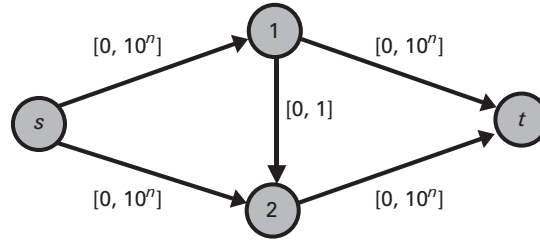


FIGURA 7.39 Caso patológico para o algoritmo de fluxo máximo F & F.

tulos. O teorema de Edmonds e Karp calcula a ordem do número de aumentos de fluxo necessário para o término do algoritmo F&F quando a modificação de rotulação e exame é introduzida.

Teorema (Edmonds e Karp)

Se cada aumento de fluxo é efetuado em um caminho se s para t de mínimo número de arcos, então um fluxo máximo é obtido após, no máximo, $mn/2 \leq (n^3 - n^2)/2$ aumentos de fluxo (ver Lawler [1976]).

Como para cada aumento de fluxo são necessárias $O(m)$ operações a complexidade do algoritmo F&F modificado é $O(m^2n)$. Um outro resultado bastante interessante diz respeito à complexidade do algoritmo ótimo para a solução do problema de fluxo máximo.

Teorema 7.3:

Para qualquer rede s - t existe uma sequência de aumentos de fluxo a partir da qual obtemos o fluxo máximo.

Suponhamos que sejam listados os caminhos diretos de s para t e que com eles formemos uma matriz de incidência $D = [d_{ij}^k]$ em que:

$d_{ij}^k = 1$ se (i, j) pertence ao caminho k e 0 em caso contrário.

$f^k \equiv$ o valor do fluxo através do caminho k .

$l = [l_{ij}]$ o vetor de capacidade.

Então, podemos formular o problema de fluxo máximo como:

$$\begin{aligned}
 \text{(PFM) Maximizar } F &= \sum_k f^k \\
 \text{sujeito a:} \\
 \sum_k d_{ij}^k f^k &\leq l_{ij} \quad (i, j) \in V \\
 f^k &\geq 0
 \end{aligned}$$

Como este problema possui m restrições, uma para cada arco, então existirá uma solução básica viável para a qual no máximo m dos f^k fluxos são estritamente positivos. As variáveis não nulas do problema representam aumentos de fluxo. Essa situação teórica sugere a possibilidade da existência de um algoritmo capaz de obter o fluxo máximo em $O(m^2)$.

✓ **Algoritmo de Malhotra, Pramodh-Kamar e Maheshwari (MPM) (1978)**

O raciocínio de forçar o fluxo através dos arcos, via caminhos que liguem s e t , pode ser estendido aos nós. Definindo como *potencial de fluxo* de um *vértice* ao maior fluxo capaz de ser forçado através desse vértice, é trivial verificar que em qualquer caminho que ligue s a t o máximo de fluxo passível de ser induzido será o referente ao nó de menor potencial pertencente ao caminho. A Figura 7.40 mostra como o potencial de um nó limita qualquer fluxo que ligue s a t através de p .

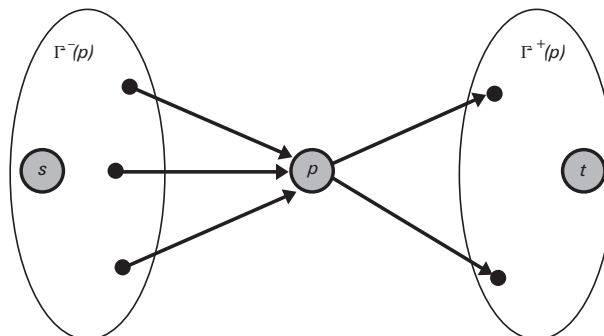


FIGURA 7.40 Limitação de fluxo gerada por um vértice da rede.

A ideia do algoritmo MPM é esgotar a capacidade dos nós da rede R , em vez da capacidade de suas arestas. Ao fazer circular o fluxo, o procedimento tenta esgotar os arcos que utilizar. À medida que os arcos vão sendo saturados, são eliminados das próximas iterações. O algoritmo para quando s ou t são eliminados da lista de nós potencialmente ativos, por terem todos os seus acessos saturados. O fluxograma da Figura 7.41 resume as ações do MPM:

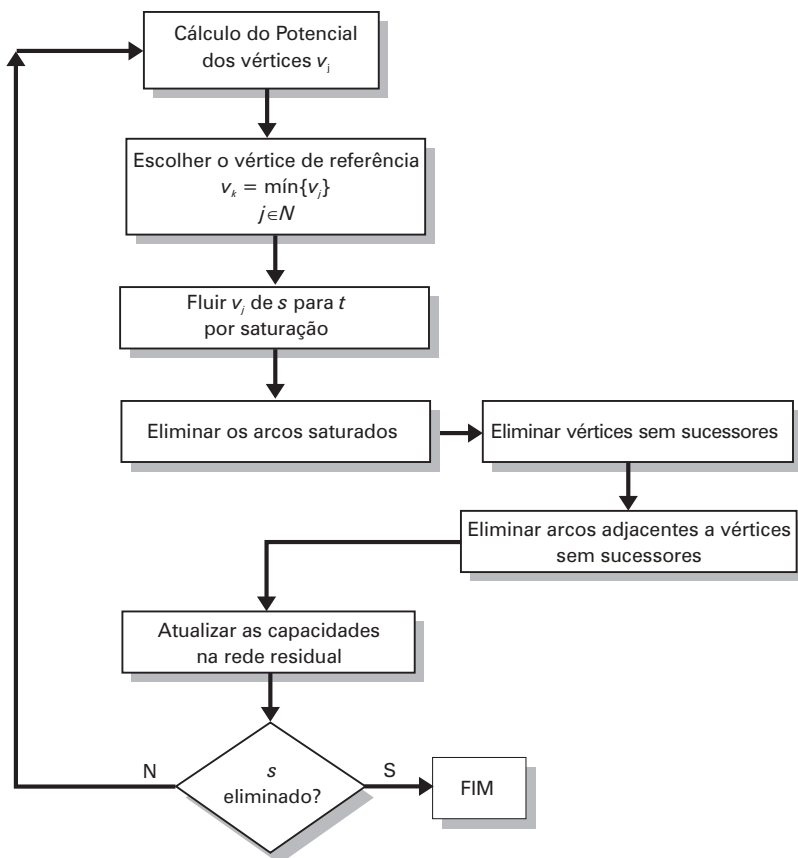


FIGURA 7.41 Fluxograma do algoritmo MPM.

Seja a rede da Figura 7.42. Aplicando o algoritmo MPM obteremos a sequência de fluxos e eliminações que se segue:

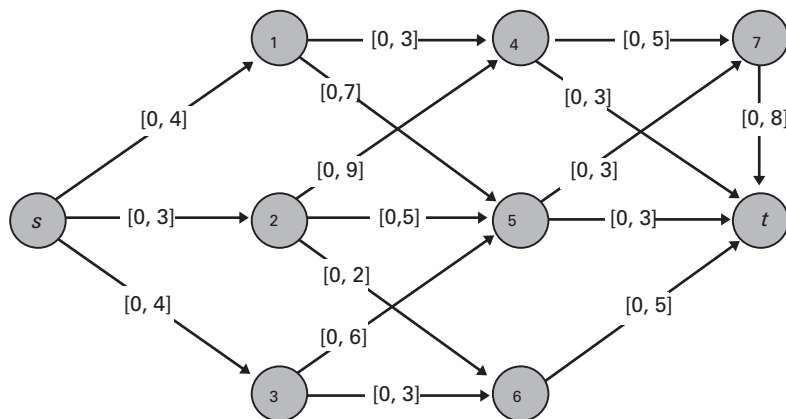


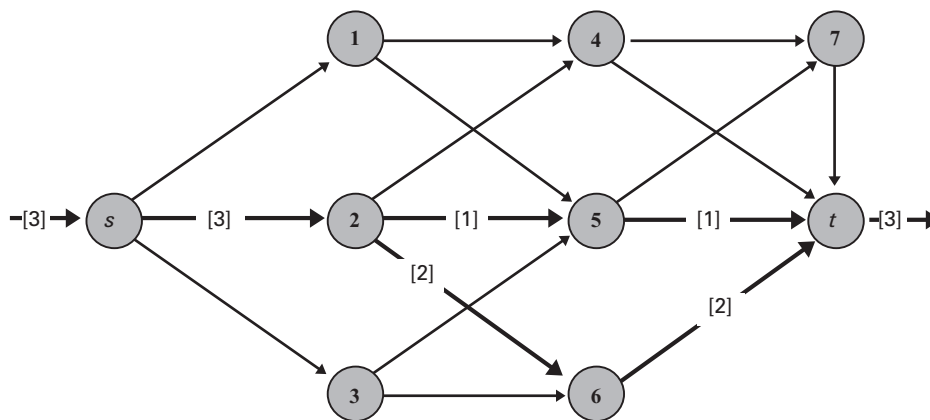
FIGURA 7.42 Rede exemplo.

Passo 1: Obter o potencial dos vértices (menor valor entre o somatório das capacidades que entram ou que saem).

K	s	1	2	3	4	5	6	7	t
v_k	11	4	3	4	8	6	5	8	19

O vértice de referência é 2, com potencial $v_2 = 3$.

Passo 2: Fluir 3 unidades de fluxo a partir do vértice 2 esgotando os arcos que geraram o valor de potencial gargalo (arcos de entrada). Observamos que, como estamos tentando forçar um fluxo a partir do vértice de menor potencial do caminho, esse fluxo sempre será passível de concretização.



(*) A capacidade do arco (2, 6) é esgotada antes de circular fluxo em (2, 5).

FIGURA 7.43 Fluxo forçado pelo vértice 2.

Passo 3: Eliminações e regra de parada.

Os arcos saturados são eliminados e o vértice s ainda possui sucessores.

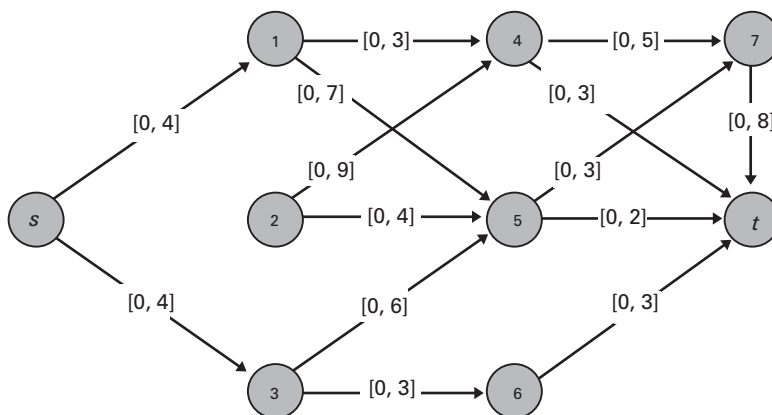


FIGURA 7.44 Primeira atualização dos arcos e capacidades.

Passo 1: Obter o potencial dos vértices (menor valor entre o somatório das capacidades que entram ou que saem).

	s	1	2	3	4	5	6	7	t
v_k	8	4	0	4	8	5	3	8	16

O vértice de referência é 6, com potencial $v_6 = 3$.

Passo 2: Fluir mais 3 unidades de fluxo a partir do vértice 6 esgotando os arcos que geraram o valor de potencial gargalo (arcos de entrada).

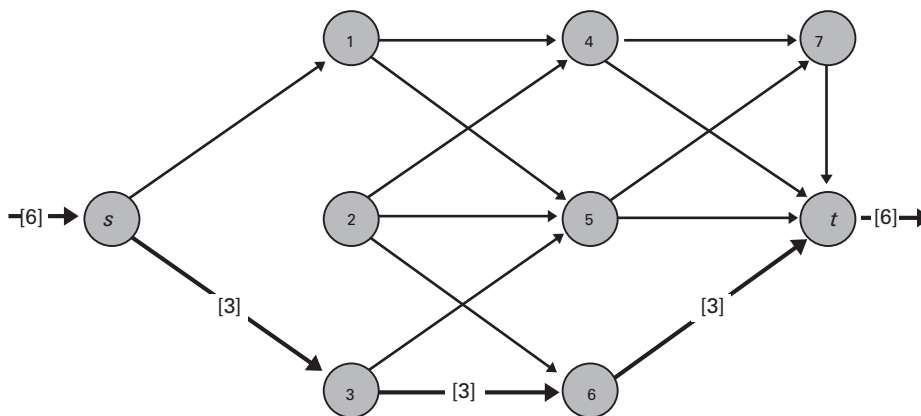


FIGURA 7.45 Fluxo forçado pelo vértice 6.

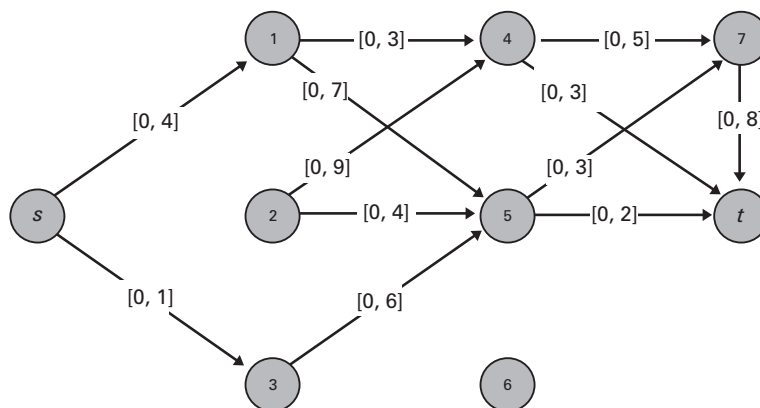
Passo 3: Eliminações e regra de parada.

Os arcos saturados são eliminados, e o vértice s ainda possui sucessores.

Passo 1: Obter o potencial dos vértices (menor valor entre o somatório das capacidades que entram ou que saem).

	s	1	2	3	4	5	6	7	t
v_k	8	4	0	1	8	5	0	8	13

O vértice de referência é 3, com potencial $v_3 = 1$.



(*) A capacidade do arco (3, 6) é esgotada

FIGURA 7.46 Segunda atualização dos arcos e capacidades.

Passo 2: Fluir mais 1 unidade de fluxo a partir do vértice 3 esgotando os arcos que geraram o valor de potencial gargalo (arcos de entrada).

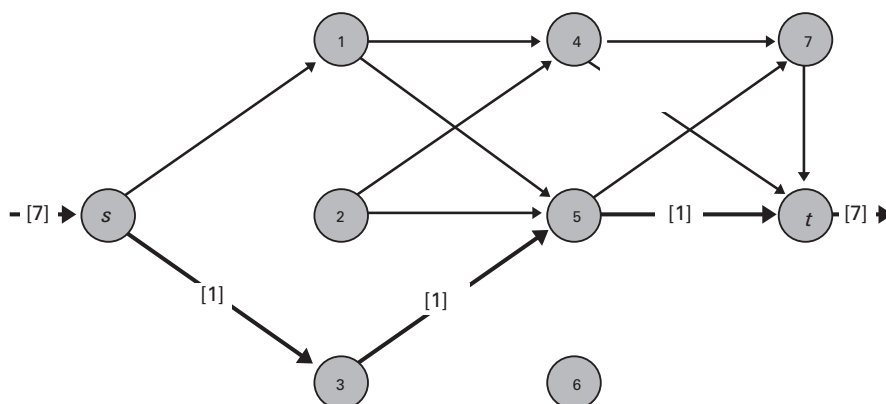
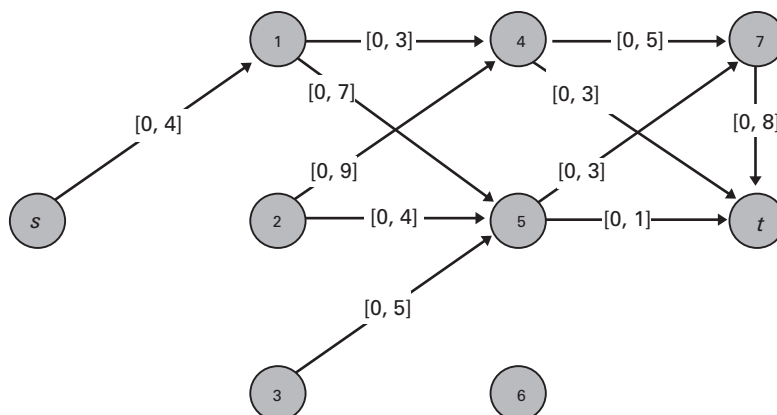


FIGURA 7.47 Fluxo forçado pelo vértice 3.

Passo 3: Eliminações e regra de parada.

Os arcos saturados são eliminados, e o vértice s ainda possui sucessores.



(*) A capacidade do arco (s, 3) é esgotada

FIGURA 7.48 Terceira atualização dos arcos e capacidades.

Passo 1: Obter o potencial dos vértices (menor valor entre o somatório das capacidades que entram ou que saem).

K	s	1	2	3	4	5	6	7	t
v_k	4	4	0	0	8	4	0	8	12

O vértice de referência é o número 1, com potencial $v_1 = 4$. Observar que poderíamos ter escolhido o vértice 5 também. Nesse caso, não podemos esquecer que o fluxo de 5 seria forçado através de vértices com potencial efetivo, ou seja, o vértice 2 não seria computado para o caminho de fluxo apesar de ser antecessor desse vértice.

Passo 2: Fluir mais 4 unidades de fluxo a partir do vértice 1, esgotando os arcos que geraram o valor de potencial de gargalo (arcos de entrada).

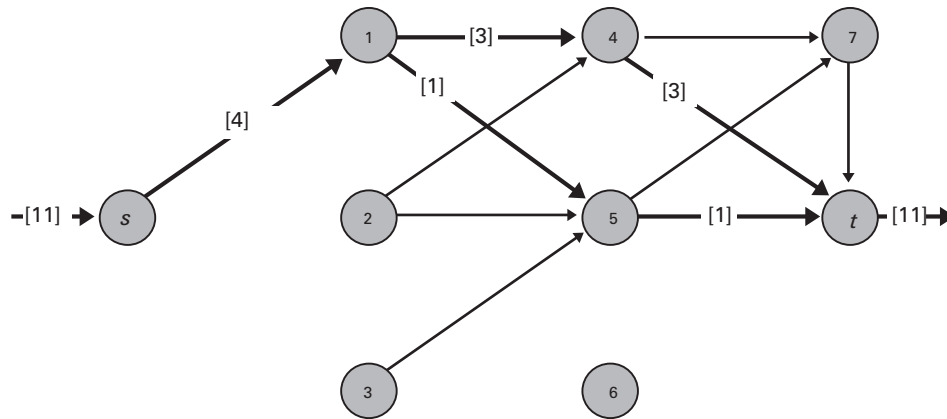
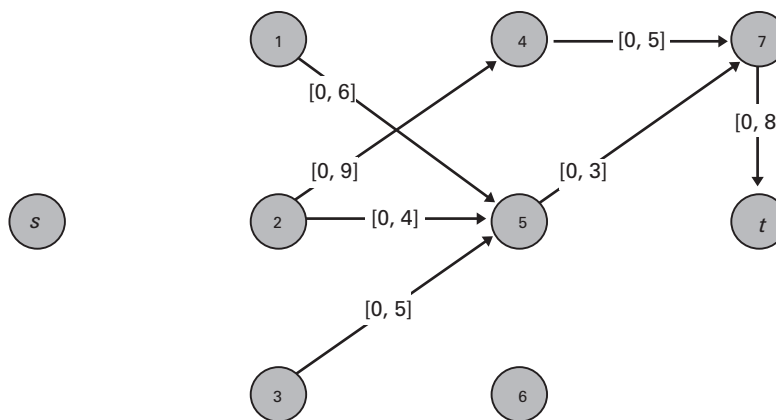


FIGURA 7.49 Fluxo forçado pelo vértice 1.

Passo 3: Eliminações e regra de parada

Os arcos saturados são eliminados, e o vértice s possui potencial zero e será eliminado. Fim.



(*) As capacidades dos arcos $(s, 1)$, $(1, 4)$, $(4, t)$ e $(5, t)$ são esgotadas

FIGURA 7.50 Término com o isolamento de s .

A análise de complexidade do algoritmo MPM pode ser feita dentro de cada etapa de cálculo. Como a cada iteração pelo menos um nó é eliminado (o de referência), a complexidade de determina-

ção desse nó é $O(n^2)$. A atualização da capacidade dos arcos pode ser feita em uma lista circular, mantendo, para cada vértice, os arcos incidentes para o exterior. Assim, a complexidade dessa fase é $O(n^2)$. O problema possui m arcos e cada um deles ficará saturado apenas uma vez, sendo eliminado em seguida. A cada iteração pelo menos um arco é saturado. Assim o total das eliminações de arcos é $O(m)$. Portanto, cada iteração exige no máximo $O(n^2)$ operações. Como existe, no pior caso, a possibilidade de examinarmos $n-2$ vértices, o MPM, nessa proposta de implementação, é $O(n^3)$. O gráfico da Figura 7.51 resume a relação evolutiva de uma série de algoritmos desenvolvidos para o problema de fluxo máximo.

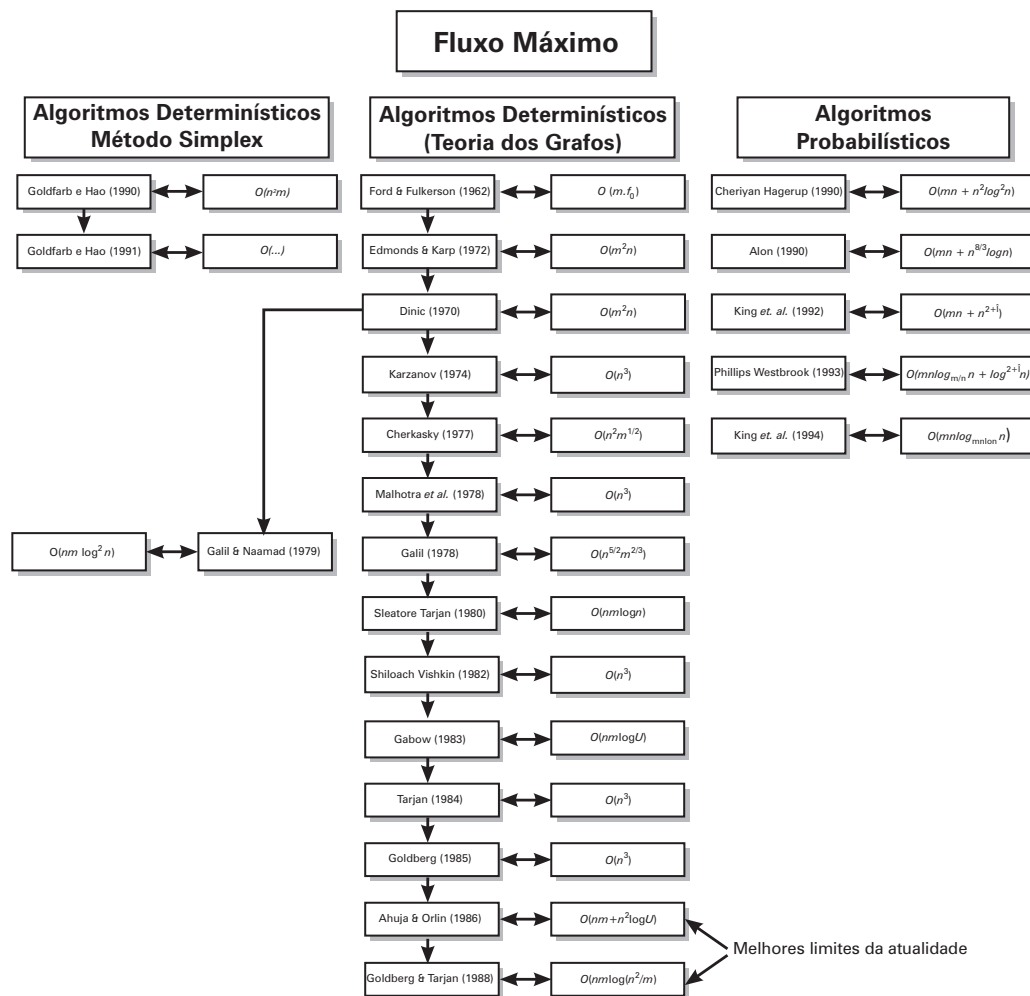


FIGURA 7.51 Evolução dos algoritmos para fluxo máximo.

7.4.3 – O Problema de Fluxo a Custo Mínimo (PFCM)

Esse caso corresponde à necessidade de circular um fluxo em uma rede s - t pagando o mínimo possível pelo tráfego. Normalmente, o fluxo a transitar é conhecido e existem restrições para o máximo e o mínimo fluxo admissível nos arcos, bem como um valor associado ao trânsito em cada arco por unidade de fluxo (c_{ij}) [limite inferior, custo do fluxo, limite superior], como mostra a Figura 7.52.

A Figura 7.53 mostra um resumo evolutivo de alguns algoritmos recentes, baseados no simplex, para o fluxo a custo mínimo em redes:

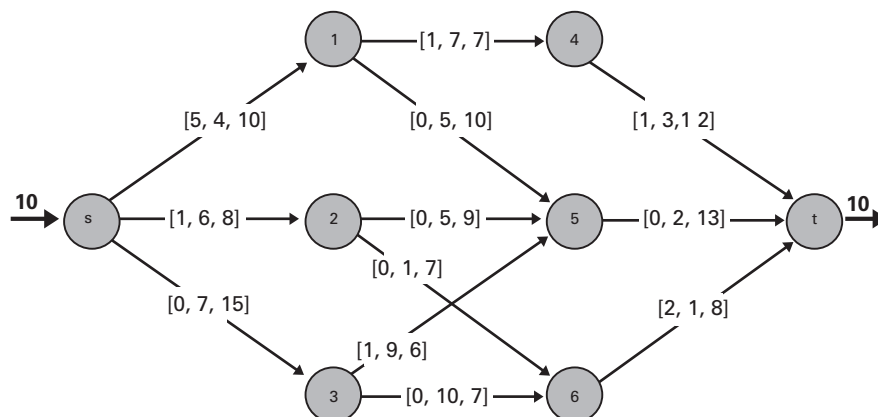


FIGURA 7.52 Exemplo de fluxo a custo mínimo.

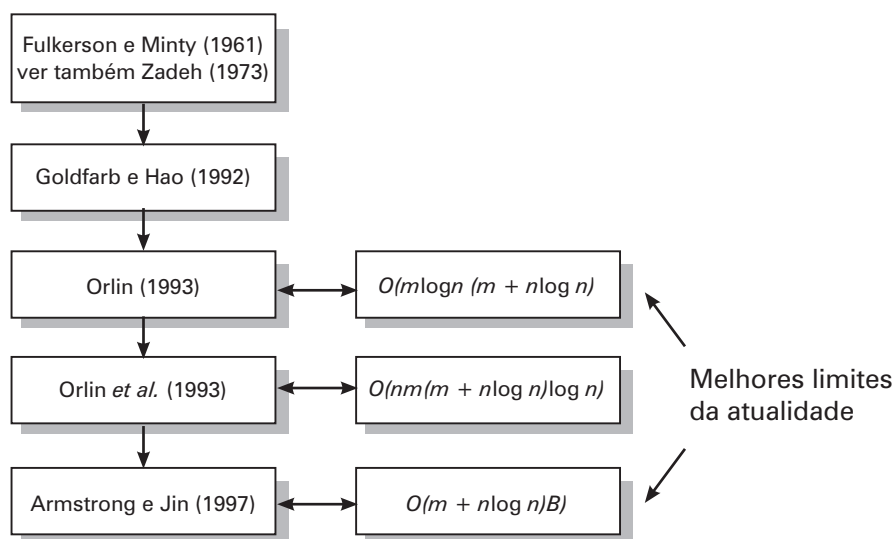


FIGURA 7.53 Evolução do desempenho dos algoritmos

✓ O Algoritmo Simplex para o (PFCM)

Nos itens anteriores mostramos que o algoritmo simplex poderia ser utilizado eficientemente para solucionar alguns casos específicos de fluxos em redes. As aplicações concentraram-se no problema de transporte, designação e no 1º matching. Com um pequeno esforço a mais poderemos generalizar o uso desse algoritmo em qualquer problema de fluxo em rede (formulado pelo modelo estrito). Seja a rede da Figura 7.54.

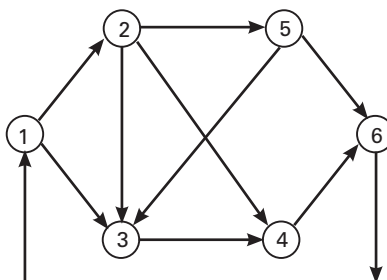


FIGURA 7.54 Rede exemplo para o (PFCM).

Que possui a seguinte matriz de incidência nó \times arco mostrada na Figura 7.55.

Nós	Arcos									
	x_{12}	x_{13}	x_{24}	x_{25}	x_{23}	x_{34}	x_{46}	x_{53}	x_{56}	x_{61}
1	+1	+1								-1
2	-1		+1	+1	+1					
3		-1			-1	+1		-1		
4			-1			-1	+1			
5				-1				+1	+1	
6							-1		-1	+1

FIGURA 7.55 Matriz de incidência da Figura 7.54.

Como já sabemos, essa matriz $A = [a_{ij}]$ é formada por entradas unitárias nas i -ésimas e j -ésimas posições de uma coluna (associadas a um arco $i-j$ na Figura 7.54). Devido a essa propriedade, podemos expressar A como uma soma vetorial de vetores unitários e_s , da seguinte forma:

$$a_{ij} = e_i - e_j$$

Considerando T qualquer árvore do grafo de substrato da rede exemplo, então T , por definição, é um subgrafo conexo da rede exemplo com m nós e $m-1$ arestas e que não contém ciclos. Levando em conta, ainda, que qualquer submatriz de A , por exemplo A_T , uma matriz $m \times (m-1)$ de uma árvore no exemplo da Figura 7.54, então se $m \geq 2$, essa árvore possui pelo menos um nó terminal (um nó com apenas arcos incidentes). Nesse caso, a k -ésima linha de A_T , conterá apenas uma entrada unitária, conforme a Figura 7.56 e a matriz A_T na Figura 7.57 esclarecem.

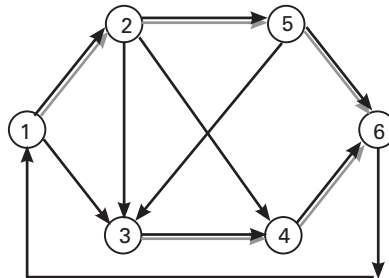


FIGURA 7.56 Uma árvore na rede exemplo.

Nós	Arcos					
	x_{12}	x_{25}	x_{34}	x_{46}	x_{56}	
1	+1					← (oferta)
2	-1	+1				
3			+1			← (oferta)
4			-1	+1		
5		-1			+1	
6				-1	-1	← (demanda)

FIGURA 7.57 Uma submatriz A_T de A .

É fácil constatar que a matriz A_T da Figura 7.57 pode ser triangularizada através de permutações entre suas linhas e colunas. Obviamente, em relação à solução do sistema de equações decorrentes da matriz de restrições uma das linhas da matriz é redundante. Com a remoção de uma linha de A_T o conjunto de variáveis (arcos) que a compõe formará uma base. Se desejarmos trabalhar com uma matriz $m \times m$ a alternativa será introduzir uma variável artificial para acrescentarmos em uma unidade o *rank* de A . A matriz A_T triangularizada e, acrescida da variável artificial, está exibida na Figura 7.58.

	x_{12}	x_{25}	x_{34}	x_{46}	x_{56}	x_a	
1	+1						← (oferta)
2	-1	+1					
3			+1				← (oferta)
4			-1	+1			
5		-1			+1		
6				-1	-1	+1	← (demanda)

FIGURA 7.58 Uma matriz básica ($m \times m$) triangularizada.

A matriz da Figura 7.58 corresponde à seguinte árvore no grafo de substrato da rede do exemplo:

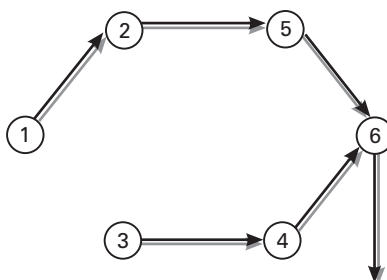


FIGURA 7.59 A árvore associada a uma base em A .

Essa associação entre as árvores no grafo substrato da rede e as soluções básicas do problema pode ser formalizada pelo seguinte teorema:

Teorema 7.3:

Um conjunto de m colunas linearmente independentes da matriz de incidência nó \times arco de uma rede de fluxo corresponde a uma árvore no grafo $G = (V, E)$, substrato na rede.

Em virtude dos exemplos anteriormente desenvolvidos de aplicação do simplex para o cálculo de fluxos em rede em situações específicas, cremos que já estamos em condições de adaptar os três passos básicos do algoritmo para o cálculo de qualquer problema de fluxo formulado no padrão estrito:

- ✓ Obtenção de uma solução inicial.
- ✓ Procedimento de melhoria da solução corrente.
- ✓ Regra de parada e de verificação da viabilidade.

Exemplo Numérico

Determinar o fluxo de menor custo da rede da Figura 7.60.

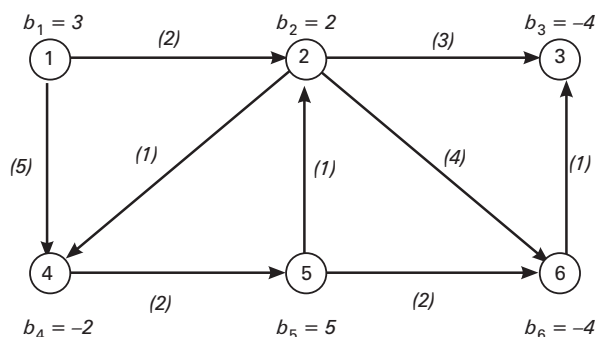


FIGURA 7.60 Exemplo numérico.

O fluxo na rede da Figura 7.60 pode ser formulado em um sistema de equações do tipo $Ax = b$, onde A é a matriz de restrições de fluxo, e $b = [b_i]$ representa o termo independente para as restrições de equilíbrio associadas aos nós da rede, como se segue:

<i>Nós</i>	<i>Arcos</i>										<i>Demanda Oferta</i>
	x_{14}	x_{12}	x_{23}	x_{24}	x_{26}	x_{45}	x_{52}	x_{56}	x_{61}		b_i
1	+1	+1							-1	=	3
2			-1	+1	+1				-1		2
3											-4
4	-1				-1		+1				-2
5						-1	+1	+1			5
6					-1			-1	+1		-4

✓ Obtenção de uma Solução Inicial para o Sistema

Como sabemos, toda árvore na rede R é uma solução básica do sistema de equações de equilíbrio de fluxo. Toda árvore que corresponde a um fluxo viável é uma base viável. Seja a árvore da Figura 7.61.

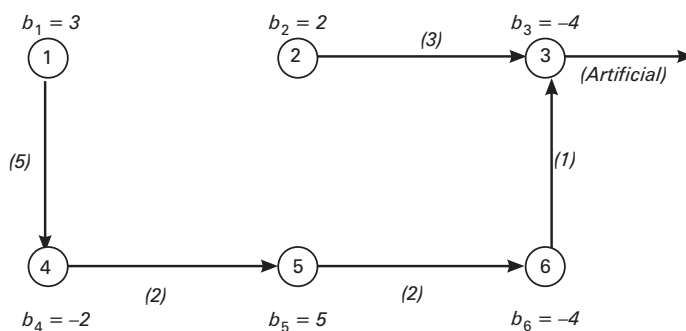


FIGURA 7.61 Uma solução básica inicial.

A árvore da Figura 7.61 corresponde à matriz triangularizada de incidência nó vezes arco da Figura 7.62.

Nós	Arcos						d_i
	x_{14}	x_{45}	x_{56}	x_{63}	x_{23}	x_a	
1	+1						3
4	-1	+1					-2
5		-1	+1				5
6			-1	+1			-4
2					+1		2
3				-1	-1	+1	-4

FIGURA 7.62 Matriz triangularizada.

cuja solução é:

$x_{14} = 3$; $x_{45} = 1$; $x_{56} = 6$; $x_{63} = 2$; $x_{23} = 2$ e $x_a = 0$, que é uma circulação viável, como observamos na Figura 7.63, a um custo de 37 unidades.

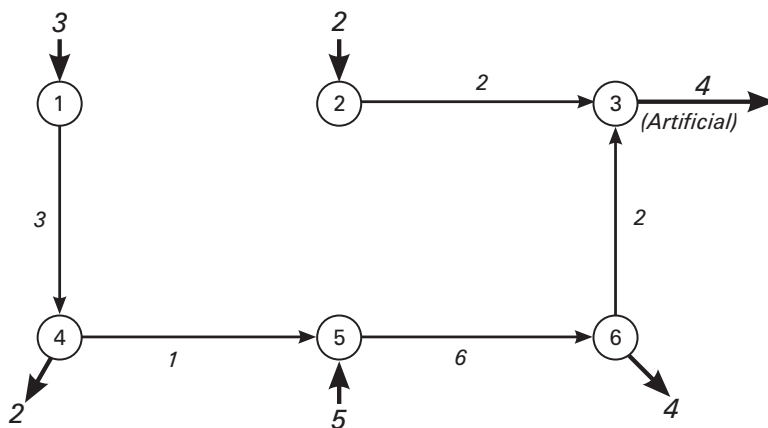


FIGURA 7.63 A circulação viável associada à base.

Contudo, nem toda base (árvore em R) corresponde a uma circulação viável. Pensando em obter facilmente uma base viável inicial, seria natural imaginar o uso de m variáveis artificiais que respondessem pelo fluxo real, o que automaticamente viabilizaria o fluxo nulo para os demais arcos da rede. Cada variável artificial estaria associada a uma linha, assumindo o valor $\pm e_i$, conforme o sinal da componente d_i . Com o mesmo raciocínio utilizado no método simplex, chegaremos à conclusão de que uma boa maneira de eliminar essas variáveis seria dividir o esquema de solução em duas fases. Na primeira, o método trataria de remover as variáveis artificiais da base alcançando uma base viável para o problema. Na segunda, a otimização seria processada como classicamente preconizada no algoritmo primal simplex. O acréscimo das variáveis artificiais está associado à inclusão de um nó artificial, ou seja, a mais uma linha na matriz de restrições, como mostra o quadro da Figura 7.64:

	x_{12}	...	x_{14}	x_{45}	x_{56}	x_{63}	x_{23}	x_1	x_2	x_3	x_4	x_5	x_6	x_7	d_i
1	+1	...	+1					+1							3
4		...	-1	+1					+1						-2
5		...		-1	+1					-1					5
6		...			-1	+1					-1				-4
2	-1	...					+1					+1			2
3		...				-1	-1						-1		-4
7	0	0	0	0	0	0	0	-1	-1	+1	+1	-1	+1	1	0

FIGURA 7.64 Quadro da primeira fase para o exemplo numérico.

A base artificial corresponde aos arcos tracejados na Figura 7.65. Notar que para a primeira fase do método os arcos artificiais receberam um custo unitário igual a 1, enquanto os arcos reais foram quantificados em 0.

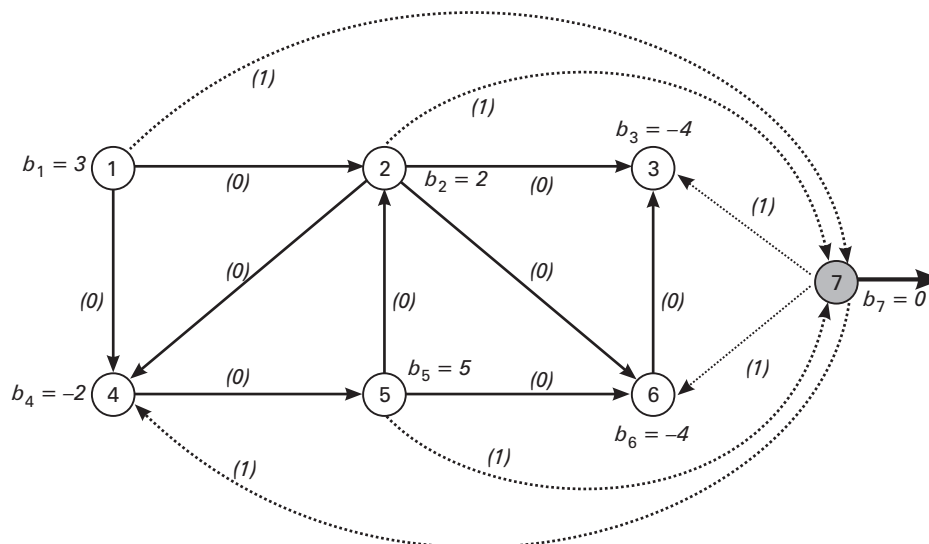


FIGURA 7.65 Primeira fase do problema.

✓ Procedimento de Melhoria da Solução Corrente.

Prosseguiremos em nosso exemplo considerando a primeira fase já concluída. O passo relativo ao cálculo dos custos reduzidos – computação dos $z_{ij} - c_{ij}$ – associados às variáveis não básicas pode ser realizado, como já vimos anteriormente, de duas maneiras:

Cálculo dos $z_{ij} - c_{ij}$ baseado nas variáveis primais:

Nesse caso, o valor dos $z_{ij} - c_{ij}$ poderá ser obtido diretamente do acréscimo do arco não básico. Quando a árvore formada pelos arcos da base receber o acréscimo do novo arco deixará de ser um grafo sem ci-

clos. No ciclo formado e considerando-se como negativo o sentido do arco acrescentado à árvore, somaremos todos os custos de arcos de sentido contrário ao novo arco e subtrairemos os de mesmo sentido. Para o caso do exemplo numérico vamos calcular os custos reduzidos dos arcos 2-6 e 1-2, arcos não básicos.

$z_{26} - c_{26} = -4 - 1 + 3 = -2$ como podemos ver na Figura 7.66:

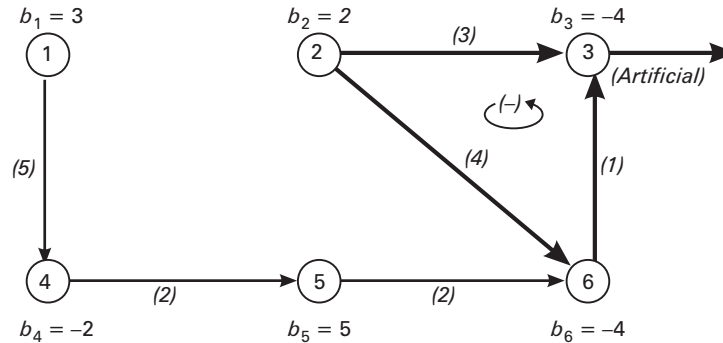


FIGURA 7.66 Cálculo de $z_{26} - c_{26}$

$z_{12} - c_{12} = -2 - 3 + 1 + 2 + 2 + 5 = +5$, conforme Figura 7.67.

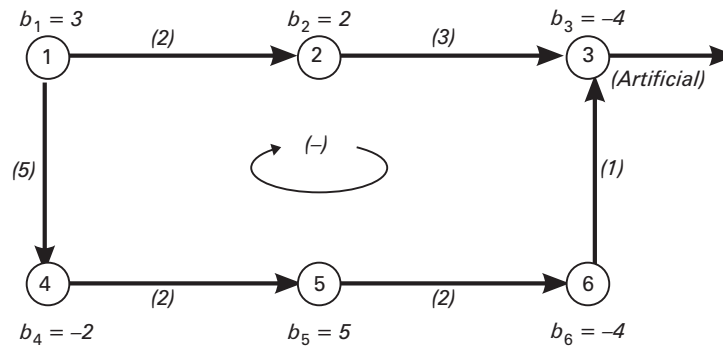


FIGURA 7.67 Cálculo de $z_{12} - c_{12}$

Cálculo dos $z_{ij} - c_{ij}$ baseado nas variáveis duais:

Sabemos que:

$$\begin{aligned} z_{ij} - c_{ij} &= w a_{ij} - c_{ij} \\ &= w (e_i - e_j) - c_{ij} \\ &= w_i - w_j - c_{ij} \end{aligned}$$

Por outro lado, podemos calcular a variável dual w por sua definição, ou seja:

$$wB = cB$$

Para o exemplo numérico podemos calcular $z_{26} - c_{26}$ e $z_{12} - c_{12}$ através das seguintes equações:

$$\begin{aligned} z_{26} - c_{26} &= w_2 - w_6 - c_{26} \\ z_{12} - c_{12} &= w_1 - w_2 - c_{12} \end{aligned}$$

e o vetor w do exemplo pode ser calculado através de sua equação primal:

$$[w_1 \ w_4 \ w_5 \ w_6 \ w_2 \ w_3] \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix} = [5 \ 2 \ 2 \ 1 \ 3 \ 0]$$

que produz:

$$\begin{aligned} w_1 - w_4 &= 5 \\ w_4 - w_5 &= 2 \\ w_5 - w_6 &= 2 \\ w_6 - w_3 &= 1 \\ w_2 - w_3 &= 3 \\ w_3 &= 0 \end{aligned}$$

E, finalmente: $w_1 = 10$, $w_2 = 3$, $w_3 = 0$, $w_4 = 5$, $w_5 = 3$ e $w_6 = 1$. O que conduz a:

$$\begin{aligned} z_{26} - c_{26} &= w_2 - w_6 - c_{26} = 3 - 1 - 4 = -2 \\ z_{12} - c_{12} &= w_1 - w_2 - c_{12} = 10 - 3 - 2 = 5 \end{aligned}$$

Como existe um arco (i, j) tal que $z_{ij} - c_{ij} \geq 0$, a base corrente não é ótima. Como é o maior dos custos reduzidos (solicita-se ao leitor verificar os demais) a variável x_{12} deverá entrar na base. Com a entrada de x_{12} , um ciclo é formado na árvore induzida em G . Para que uma nova base seja formada é necessário que um dos arcos do ciclo seja removido e uma nova árvore seja constituída. O critério de retirada do arco estará associado ao crescimento no fluxo do arco que entra. Chamando por Δ o valor do fluxo no arco que entra na base, à medida que Δ crescer, o fluxo nos arcos contrários ao sentido vai diminuir no mesmo valor. A Figura 7.68 mostra o efeito do crescimento de Δ . O primeiro arco que anula o fluxo é o arco 4-5 e $\Delta = 1$. A variável x_{45} deve deixar a base.

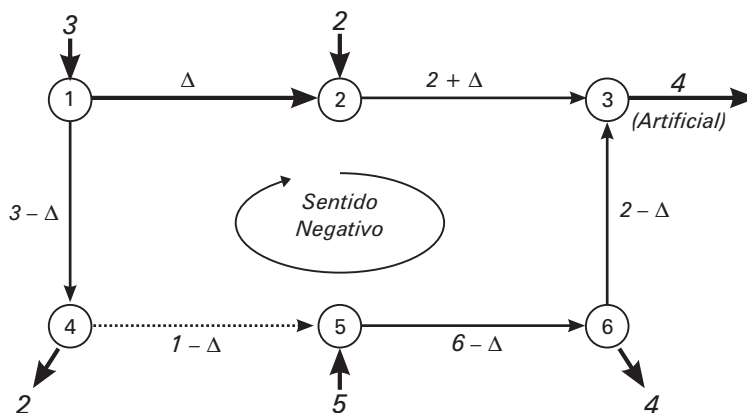


FIGURA 7.68 Identificação da variável que sai da base.

A partir da entrada na base de uma nova variável o fluxo é atualizado conforme mostra a Figura 7.68, chegando-se à solução: $x_{14} = 2$, $x_{12} = 1$, $x_{56} = 5$, $x_{63} = 1$, $x_{23} = 3$ e $x_a = 0$, que é uma circulação viável, como observamos na Figura 7.69, a um custo de 32 unidades.

O algoritmo recalcula os valores dos $z_{ij} - c_{ij}$. Nesse caso, $z_{24} - c_{24} = -2 - 1 + 5 = 2$. A variável x_{24} é candidata para entrar na base. O ciclo que se forma está expresso na Figura 7.70. Nesse caso, $\Delta = 2$ e a variável

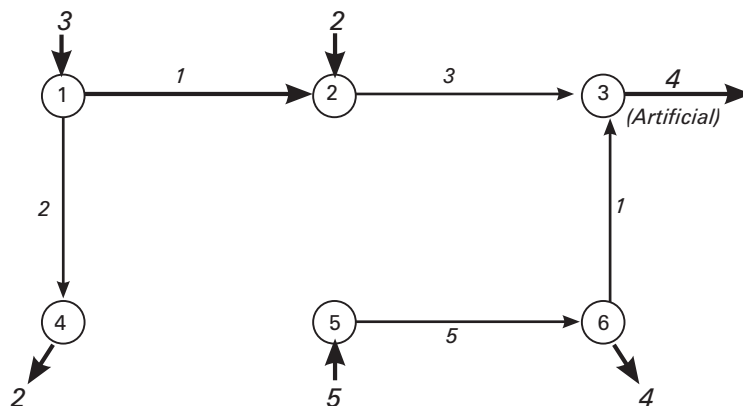
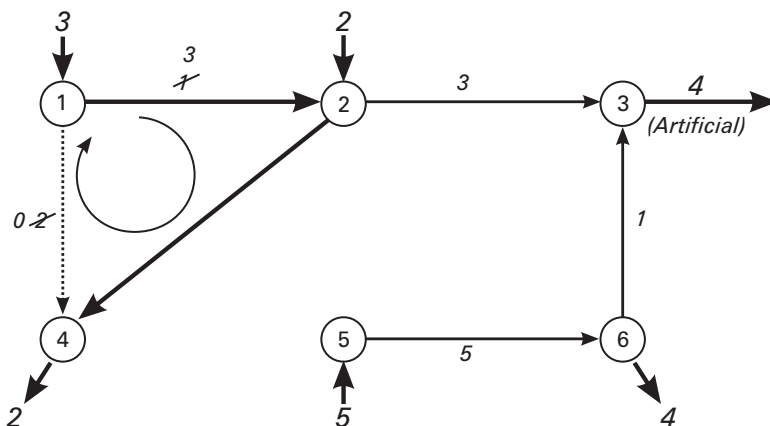


FIGURA 7.69 Identificação da variável que sai da base.

x_{14} sai da base. O fluxo no arco 1-4 é anulado (a variável passa a não básica). A solução para o fluxo passa a ser: $x_{24} = 2$, $x_{12} = 3$, $x_{56} = 5$, $x_{63} = 1$, $x_{23} = 3$ e $x_a = 0$, a um custo de 28 unidades. Nesse momento, todos os $z_{ij} - c_{ij} \leq 0$ e o fluxo é ótimo.


FIGURA 7.70 Ciclo formado pela entrada de x_{24} na base.

7.5 – PROBLEMAS PROPOSTOS

1 – Consolidação dos Conceitos Básicos



Considere a rede definida pelo quadro que se segue:

# Nó	Oferta	Demanda	# Arco	De	Para	Custo Unidade Fluxo	Capacidade do Arco
1	12	1	1	1	1	0	4
2	10	0	2	1	3	3	3
3	5	0	3	2	5	1	8
4	1	11	4	2	4	6	8
5	0	3	5	3	4	5	10
			6	3	5	4	12
			7	5	4	1	15

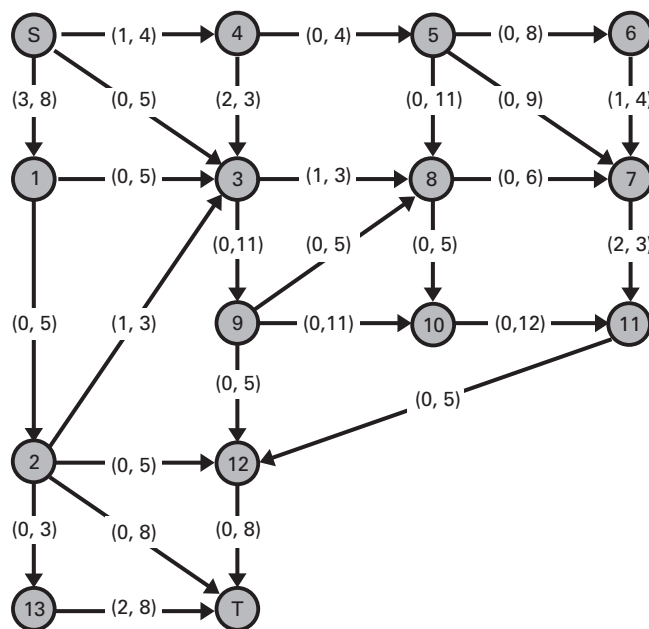
1. Construir esquematicamente a rede definida no quadro, indicando uma solução viável para um fluxo e seu custo.
2. Determinar a matriz de incidência (nó \times aresta) associada à rede.
3. Criar um ponto de demanda fictício, com os respectivos arcos associados, de tal forma que a oferta total seja igual à demanda total, construindo o esquema gráfico da rede assim formada.
4. Reformular o problema de fluxo para que somente exista um único ponto de oferta e um único ponto de demanda.

2 – Fluxo Máximo



Considerando a rede a seguir, calcule o maior fluxo possível entre os vértices s e t através do:

1. Algoritmo de Ford-Fulkerson.
2. Algoritmo MPM.



3 – O Problema da Torcida do Coringas Futebol Clube



Uma grande quantidade de torcedores fanáticos do Coringas, um time de futebol, vive em São Paulo e deseja assistir a uma partida-chave de seu time que será realizada em Natal, Rio Grande do Norte. Os torcedores, todos trabalhadores ocupados, desejam sair no dia do jogo, chegando a Natal, no máximo, às 19:00 h (a partida será às 21:30 h). Como a disponibilidade financeira dessas pessoas é limitada, elas estão optando pelo voo mais barato, mesmo que isso implique mais demora, conexões, deslocamentos aéreos em circuito etc. A companhia escolhida foi a Manysteps Airlines. O quadro de voos da empresa está mostrado a seguir.

Resolveram se reunir para comprar suas passagens em sociedade 150 torcedores, e conseguiram da Manysteps 50% de desconto nos trechos (também denominadas pernas de voo) que passam em Brasília (chegando ou partindo da cidade). Sabendo-se que os torcedores vão ratear a despesa das 150 passagens de forma uniforme, independentemente do esquema de voo que caiba a cada um, formular o problema de minimizar a despesa total de deslocamento do grupo de torcedores. Um esquema de voo entre São Paulo e Natal é válido se, além de chegar antes das 19:00 horas:

1. Não possuir mais que cinco pernas (ou trechos entre cidades).
2. Os tempos de conexão (troca de voos) programados não podem ser menores do que uma hora. Exemplificando: não podemos compor o voo 501, que chega a Recife às 17:00 h, com o voo 595 que sai para Natal às 17:30 h, porque o tempo de conexão seria de apenas 30 minutos.

Os voos possuem capacidade de 200 lugares, e as passagens reservadas não podem mais ser vendidas ao grupo de torcedores.

# do Voo	Origem	Destino	Saída	Chegada	Custo Normal U\$	Reservas ✓
501	São Paulo	Goiânia	08:00	10:00	120,00	120
	Goiânia	Salvador	11:00	13:40	120,00	70
	Salvador	Recife	15:50	17:00	100,00	80
	Recife	Natal	18:00	18:45	80,00	100
513	São Paulo	Brasília	09:00	10:50	50,00	180
	Brasília	Fortaleza	11:45	15:00	180,00	20
	Fortaleza	Recife	15:50	16:20	100,00	25
581	Curitiba	Manaus	06:50	10:20	320,00	180
	Manaus	Belém	10:50	12:00	100,00	20
	Belém	Fortaleza	12:50	14:20	100,00	35
587	Rio de Janeiro	São Paulo	06:00	07:50	80,00	160
	São Paulo	Maceió	08:30	11:00	180,00	100
	Maceió	Recife	12:00	13:00	80,00	120
	Recife	Natal	15:00	15:45	80,00	10
590	Porto Alegre	Curitiba	06:00	07:30	100,00	100
	Curitiba	Brasília	10:00	11:45	150,00	140
	Brasília	Recife	14:00	16:30	150,00	120
592	Brasília	Maceió	12:00	14:30	100,00	180
593	Rio de Janeiro	Brasília	09:00	09:45	100,00	110
594	Rio de Janeiro	Vitória	05:00	06:00	100,00	130
	Vitória	Salvador	08:30	09:45	80,00	20
	Salvador	Maceió	10:45	11:45	80,00	100
	Maceió	Natal	16:45	18:00	100,00	50
595	Recife	Natal	17:30	18:15	50,00	120

4 – O Problema do Transporte da Produção de Soja



A produção de soja do Rio Grande do Sul se realiza principalmente nas regiões de Passo Fundo e Cruz Alta, alcançando valores da ordem de 24 e 36 centenas de milhares de toneladas ao ano. Podemos considerar que praticamente toda a produção é exportada através do porto de Rio Grande, sob a forma dos derivados farelo e óleo. As principais unidades de processamento estão instaladas na área de Porto Alegre ou próximas ao porto de Rio Grande, com capacidades anuais de 50 e 16 centenas de milhares de toneladas.

As cidades de Passo Fundo e Cruz Alta distanciam-se de Porto Alegre em, respectivamente, 240 e 360 quilômetros. Essas três cidades estão, por sua vez, a distâncias de Rio Grande de 600, 500 e 300 quilômetros, respectivamente. Todo o transporte entre Porto Alegre e Rio Grande é realizado pela modalidade hidroviária, estando limitado a 50 centenas de milhares de toneladas por ano; as demais ligações são feitas por transporte rodoviário, com um custo, por quilômetro, duas vezes maior que o hidroviário. Presumindo que o custo de transporte rodoviário seja diretamente proporcional à distância (uma unidade monetária por quilômetro) e que o preço do produto embarcado no porto seja de 300, responda às seguintes questões:

1. Formule o problema de programação linear que calcula os fluxos de exportação a custo mínimo.
2. Mostre que esse problema pode ser classificado como um problema de fluxo em redes capacitadas.
3. Determine a solução ótima do problema (por qualquer método).
4. Quais serão os preços recebidos pelos produtores de soja nas diversas regiões produtivas?

5 – O Problema da Amizade



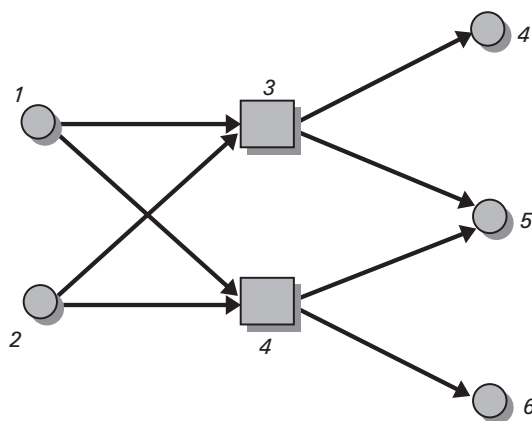
Um grupo de discussão via Internet está interessado em formar pares para debater assuntos de interesse mútuo. Para cada indivíduo i é conhecido o conjunto das n_i pessoas que lhe são compatíveis, $n_i = \{1, \dots, m\}$.

1. Formule o problema de maximizar a formação de duplas compatíveis por meio de um modelo de fluxo em redes.
2. O que seria alterado no modelo anterior se, além da compatibilidade, fosse exigido que as duplas fossem formadas por um homem e uma mulher?

6 – O Problema de Expansão de uma Rede de Comunicação



A rede representada na figura que se segue define o tráfego de mensagens entre as estações 1 e 2 (lado esquerdo da rede), e as estações 5, 6 e 7. O tráfego é realizado por meio do chaveamento localizado nos vértices 3 e 4. A atual rede possui uma certa capacidade t_{ij} de tráfego entre os vértices i e j .



Rede de Comunicação

Um plano de expansão prevê a possibilidade de que a capacidade dos arcos de comunicação seja ampliada e que os nós de chaveamento possam também processar um maior número de mensagens. São conhecidos os custos unitários c_{ij} para a expansão dos arcos que ligam os vértices e os nós de chaveamento. São ainda conhecidos os custos para o aumento unitário do fluxo em um certo vértice de chaveamento v_j , $j = 3, 4$. Sabendo-se que a demanda prevista entre os vértices i e j será f_{ij} , formular o problema de encontrar a expansão que minimize o custo de operação da rede.

8

PROBLEMA DO CAIXEIRO VIAJANTE

8.1 – INTRODUÇÃO

8.1.1 – As Origens do Problema

O problema do caixeiro viajante (PCV) é um dos mais tradicionais e conhecidos problemas de programação matemática (Melamed [1990] e Campello e Maculan [1994]). Os problemas de roteamento lidam em sua maior parte com passeios ou *tours* sobre pontos de demanda ou oferta. Esses pontos podem ser representados por cidades, postos de trabalho ou atendimento, depósitos etc. Dentre os tipos de passeios um dos mais importantes é o denominado hamiltoniano. Seu nome é devido a Willian Rowan Hamilton que, em 1857, propôs um jogo que denominou *Around the World*. O jogo era feito sobre um dodecaedro em que cada vértice estava associado a uma cidade importante na época. O desafio consistia em encontrar uma rota através dos vértices do dodecaedro que iniciasse e terminasse em uma mesma cidade sem nunca repetir uma visita. O grafo do problema é mostrado na Figura 8.1.

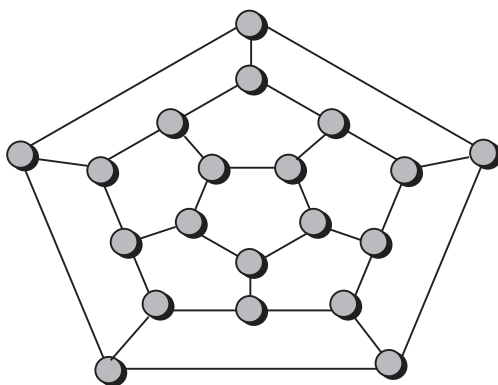


FIGURA 8.1 Jogo de Hamilton.

Uma solução do jogo de Hamilton, em sua homenagem, passou a se denominar um ciclo hamiltoniano. Hamilton não foi o primeiro a propor esse problema (Biggs *et al.* [1986]), mas seu jogo o divulgou. Uma das soluções do jogo está apresentada na Figura 8.2.

Modernamente, a primeira menção conhecida do problema é devida a Hassler Whitney em 1934 em um trabalho na Princeton University. Independente desse trabalho de Hamilton, o problema do caixeiro viajante (PCV) é um problema de otimização associado ao da determinação dos caminhos hamil-

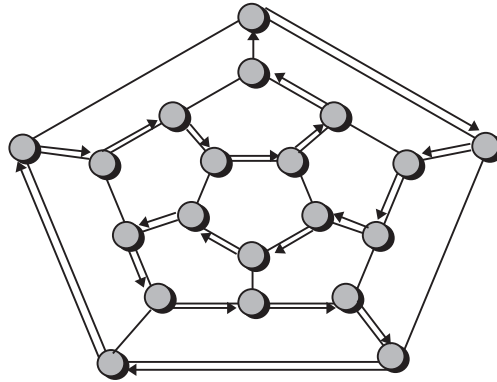


FIGURA 8.2 Uma solução do jogo de Hamilton.

tonianos em um grafo qualquer. O objetivo do PCV é encontrar, em um grafo $G = (N, A)$, o caminho hamiltoniano de menor custo.

8.1.2 – Importância Atual do PCV

A importância do PCV é devida a pelo menos três de suas características:

- ✓ Grande aplicação prática (Bellmore e Nemhauser [1968], Burkard [1979], Reinelt [1994]).
- ✓ Uma enorme relação com outros modelos (Laporte *et al.* [1996]).
- ✓ Grande dificuldade de solução exata (Papadimitriou e Steiglitz [1982], Gould [1991] e Zhang [1997]).

Dessa forma, a importância do modelo é indiscutível, tanto sob o aspecto prático, como o teórico. Considerado intratável por Garey e Johnson (1979) é classificado por Karp (1975) como NP-árduo.

8.2 – FORMULAÇÕES E APLICAÇÕES

8.2.1 – Formulações

Existem várias formulações para esse problema. Devido à sua importância apresentaremos as mais difundidas. Essas formulações podem ser consideradas como canônicas, tanto por sua larga difusão na literatura especializada como por desenvolverem modos peculiares de caracterização do problema. Para uma abordagem mais específica Lawler *et al.* (1985) apresentam um bom estudo introdutório.

• Formulação de Dantzig-Fulkerson-Johnson (DFJ)

Dantzig, Fulkerson e Johnson (1954) (ver Christofides *et al.* [1979]) formularam o PCV como um problema de programação 0-1 sobre um grafo $G = (N, A)$, como se segue:

$$\begin{aligned}
 (\text{PCV1}) \text{ Minimizar } z &= \sum_{j=1}^n \sum_{i=1}^n c_{ij} x_{ij} \\
 \text{sujeito a:} \\
 \sum_{i=1}^n x_{ij} &= 1 \quad \forall j \in N
 \end{aligned} \tag{8.1}$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i \in N \quad (8.2)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset N \quad (8.3)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in N \quad (8.4)$$

Onde a variável binária x_{ij} assume valor igual a 1, se o arco $(i, j) \in A$ for escolhido para integrar a solução, e 0 em caso contrário, e S é um subgrafo de G , em que $|S|$ representa o número de vértices desse subgrafo. Nessa formulação assumimos implicitamente que x_{ii} não existe e que teremos $n(n-1)$ variáveis inteiras 0-1 e $O(2^n)$ restrições. As restrições (8.1) e (8.2) são semelhantes às do problema de designação, de modo que configurações como as mostradas na Figura 8.3 seriam válidas. O conjunto de restrições (8.3) determina a eliminação desses circuitos pré-hamiltonianos.

As equações em $|S|$ tornam os circuitos pré-hamiltonianos ilegais. A Figura 8.3 mostra que a restrição para $|S| = 5$ elimina circuitos pré-hamiltonianos com cinco vértices da seguinte forma:

$$\begin{aligned} x_{15} + x_{54} + x_{49} + x_{96} + x_{61} &\leq 4 \\ x_{23} + x_{310} + x_{108} + x_{72} + x_{87} &\leq 4 \end{aligned}$$

Para cada circuito pré-hamiltoniano possível é necessária uma restrição do tipo (8.3), justificando-se assim o número de $O(2^n)$ restrições.

Essa formulação destaca um importante aspecto do PCV que é sua natureza combinatória. Pela formulação fica claro que solucionar um PCV é determinar uma certa permutação legal de custo mínimo. A formulação de Dantzig também auxilia no entendimento da ligação do PCV aos problemas de sequenciamento de operações, tão comuns em manufatura. Supondo que exista um certo tempo de preparação de cada máquina para receber uma tarefa, e que as tarefas possam ser distribuídas de várias formas no conjunto das máquinas existentes, o problema de encontrar uma sequência de operações que minimiza o tempo de preparação das máquinas e, conseqüentemente, os custos do trabalho, pode ser modelado como um PCV.

• Formulação de Miller-Tucker-Zemlin

Miller, Tucker e Zemlin (1960) propõem uma formulação para o PCV denominada Folha de Cravo, conforme descrita a seguir:

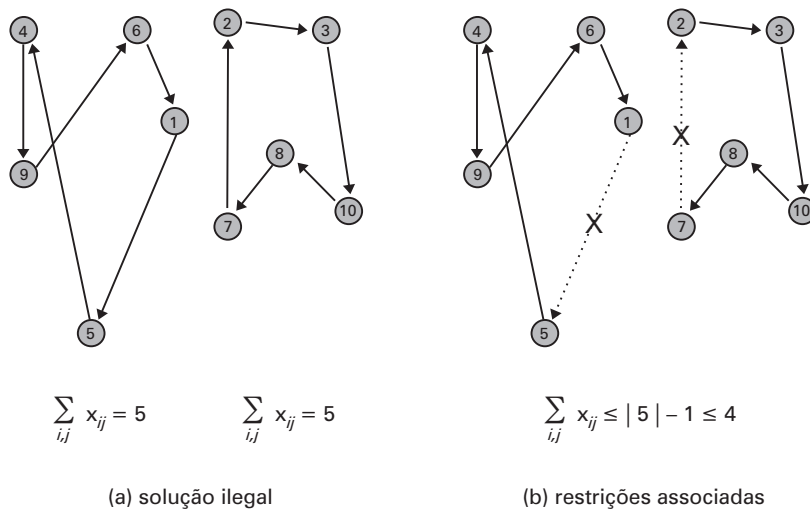


FIGURA 8.3 Restrições de cardinalidade para o PCV.

Denotaremos a cidade 1 como a cidade início-fim ou cidade origem. O caixeiro viajante deve visitar as outras $n - 1$ cidades exatamente uma vez. Durante seu trajeto deve retornar a cidade origem exatamente t vezes, incluindo o retorno final, e não deve visitar mais de p cidades diferentes em um *tour* ou ciclo. A formulação requer que $\left\lceil \frac{n-1}{p} \right\rceil \leq t \leq n-1$, onde $\lceil a \rceil$ denota o menor inteiro maior ou igual ao valor de a , ou teto da divisão, para garantir a existência de *tours* viáveis. Dessa forma o PCV pode ser formulado como se segue.

$$(PCV2) \text{ Minimizar } z = \sum_{j=1}^n \sum_{i=1}^n c_{ij} x_{ij}$$

sujeito a:

$$\sum_{i=2}^n x_{i1} = t \quad (8.5)$$

$$\sum_{i=1}^n x_{ji} = 1 \quad j = 2, \dots, n \quad (8.6)$$

$$\sum_{j=1}^n x_{ji} = 1 \quad i = 2, \dots, n \quad (8.7)$$

$$u_i - u_j + p x_{ij} \leq p - 1 \quad 2 \leq i \neq j \leq n \quad (8.8)$$

$$u_i \geq 0 \quad 2 \leq i \leq n \quad (8.9)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in N \quad (8.10)$$

O primeiro conjunto de restrições de (PCV2) garante que a cidade origem é visitada exatamente t vezes. O arco (i, j) em toda a solução viável 0-1 dos três primeiros conjuntos de restrições obrigam o conjunto de restrições em p a transformar-se em:

$$u_i - u_j \leq -1$$

A formulação (PCV2) envolve $O(n^2)$ restrições em $n^2 - 1$ variáveis. Quando $t = 1$ e $p \geq n - 1$ ela modela o PCV.

• Formulação de Fox-Gavish-Graves

Fox, Gavish e Graves (1980) formularam uma generalização do PCV denominada *time dependent*. Nessa formulação o custo de percorrer o caminho entre as cidades i - j depende da posição t do arco (i, j) em um *tour* (ou passeio) de uma dada cidade origem indexada por 1. Essa formulação foi originalmente proposta por Fox, para o problema do *1-machine n-job scheduling*. As variáveis de decisão desse modelo z_{ijt} são triplamente indexadas. $z_{ijt} = 1$ se o arco de i para j é designado na t -ésima posição do *tour*, $z_{ijt} = 0$ para o caso contrário. Dessa forma temos:

$$(PCV3) \text{ Minimizar } z = \sum_{i=1}^n \sum_{j=1}^n \sum_{t=1}^n c_{ijt} z_{ijt}$$

sujeito a:

$$\sum_{i=1}^n \sum_{j=1}^n \sum_{t=1}^n z_{ijt} = n \quad (8.11)$$

$$\sum_{j=1}^n \sum_{t=2}^n tz_{ijt} - \sum_{j=1}^n \sum_{t=1}^n tz_{ijt} = 1 \quad i = 2, \dots, n \quad (8.12)$$

$$z_{ijt} \in \{0, 1\} \quad \forall i, j, t \quad (8.13)$$

As restrições (8.11) ou restrições de agregação do problema de alocação 3-dimensional garantem que cada cidade é visitada exatamente uma vez e que para cada posição do *tour* exista um arco designado. O conjunto de restrições (8.12) garante a eliminação dos *subtours*.

• Formulação de Claus

Padberg e Sung (1988) apresentam uma importante formulação do PCV devida a Claus, usando conceitos de fluxos em rede. Denotando por s a cidade origem ou fonte e transformando todo ciclo hamiltoniano em um caminho hamiltoniano, por considerar também a cidade origem como um destino t , podemos interpretar o PCV como um problema de determinar um caminho hamiltoniano de s para t no digrafo $G = (V, E)$, onde:

$$V = V^1 \cup \{s, t\}, \quad V^1 = \{1, \dots, n\} \text{ e}$$

$$V = V^* \cup \{t\} \text{ e } E = \{(i, j), \forall i \neq j \in V^1\} \cup \{(s, i), (i, t), \forall i \in V^1\}.$$

Podemos interpretar a variável binária x associada a aresta (i, j) como:

$x_{ij} = 1$ se (i, j) pertence a um caminho hamiltoniano, 0 em caso contrário.

O fluxo na rede envolve $n + 1$ comodidades. Claus define uma variável auxiliar y_{ijk} como o fluxo da k -ésima comodidade no arco (i, j) . Daí temos:

$$(\text{PCV } 4) \text{ Minimizar } z = \sum_{(i,j) \in E} c_{ij} x_{ij}$$

sujeito a:

$$\sum_{j \in V} x_{ij} = \sum_{j \in V} x_{ji} \quad \forall i \in V^1$$

$$\sum_{j \in V} y_{ijk} - \sum_{j \in V} y_{jik} = 0 \quad \forall i \in V^1 \quad \forall k \in V^*$$

$$\sum_i y_{ikk} = 1, \sum_i y_{sik} = 1, \sum_i y_{kik} = 0 \quad \forall k \in V^*$$

$$y_{ijk} \leq x_{ij} \quad \forall i, j, k$$

$$x_{ij} \geq 0 \quad \forall i, j$$

$$y_{ijk} \geq 0 \quad \forall i, j, k$$

$$x_{ij}, y_{ijk} \text{ inteiros } \forall i, j, k$$

A formulação C envolve um conjunto de restrições da ordem de n^3 e $(n^3 + n^2 + 3n)$ variáveis.

8.2.2 – Aplicações

O PCV, em suas diversas versões, está presente em inúmeros problemas práticos. Sem nos estendermos demasiadamente podemos destacar:

- Determinação de acesso a arquivos computacionais (Henry-Labordere [1969]).
- Programação de assistência social (Saskena [1970]).
- Manipulação de itens em estoque (Ratliff e Rosenthal [1981]).
- Na maioria dos problemas de roteamento de veículos (Bodin [1983] e de navios Fagerholt e Christiansen [2000]).
- Programação de transporte entre células de manufatura (Finke e Kusiak [1985]).
- Programação de operações de máquinas em manufatura (Kusiak e Finke [1987]).
- Otimização do movimento de ferramentas de corte (Chauny *et al.* [1987]).
- Otimização de palhetas em turbinas a gás (pante *et al.* [1987]).
- Análise de estruturas cristalinas (Bland e Shallcross [1987]).
- Fabricação de Chips VLSI (Korte [1988]).
- Otimização de perfurações de furos em placas de circuitos impressos (Reinelt [1989]).
- Na solução de problemas de sequenciamento de tarefas (Whitley [1991]).
- Trabalhos administrativos (Laporte *et al.* [1996]).
- Roteamento de entrega postal (Laporte *et al.* [1996]).
- Na solução de problemas de programação e distribuição de tarefas em plantas (Salomon *et al.* [1997]).
- Cortes em chapas de aço e vidro (Hoeft e Palekar [1997]).
- Construção de árvore evolucionária e determinação de sequenciamento genético (Korostensky e Gonnet [2000]).
- Produção de Aço (Tang *et al.* [2000]).
- Testes citológicos (Laporte e Palekar [2002]).
- Planejamento da produção (Ben-Arieh *et al.* [2003]).
- Planejamento de sistemas noturnos de distribuição de valores (Cordone e Calvo [2003]).

8.3 – PROBLEMAS CORRELATOS

O PCV Simétrico (PCVS)

Esse caso particular do PCV é de extrema importância, pois representa muitas das situações reais de roteamento. Quando a matriz de custos do PCV é simétrica, o problema é denominado *simétrico*. Se $c_{ik} \leq c_{ij} + c_{jk}$, $\forall i, j, k \in N$ o PCV satisfaz à chamada *desigualdade triangular*. Um PCV é denominado euclidiano (PCVE) quando sua matriz é simétrica e satisfaz à desigualdade triangular, característica da métrica euclidiana.

Um caso especial do PCV simétrico é aquele em que existem conjuntos de nós que possuem restrições que os obrigam a estar em uma determinada sequência de atendimento. Esses conjuntos de nós são denominados de *cluster* ou grupamentos. O problema é denominado na literatura como *Symmetric clustered traveling salesman problem* (SCTSP). O problema assim definido possui várias aplicações na indústria e no roteamento de veículos (ver Jongens [1985]).

O PCV Generalizado (PCVG)

Caso, como o próprio nome indica, é uma generalização do PCV. Semelhante ao (SCTSP) o problema está sujeito ao agrupamento de nós, só que, nesse caso, para forçar uma generalização no ciclo hamiltoniano. Cada grupamento formado deve contribuir com certo número de nós para o ciclo hamiltoniano do

caixeiro. Obviamente, quando todos os nós dos grupamentos são visitados o problema recai no PCV clássico. Uma versão desse problema, denominada *equality*, exige que em cada *cluster* apenas um vértice seja visitado. Essa versão modela problemas de ligação em circuitos integrados em que os subcircuitos possuem várias posições possíveis de conexão com o conjunto da placa. A Figura 8.4 esclarece a topologia do modelo mostrando uma das muitas alternativas de rotear quatro placas:

Uma outra aplicação do PCVG ocorre na área de entrega de correspondência administrativa e bancá-

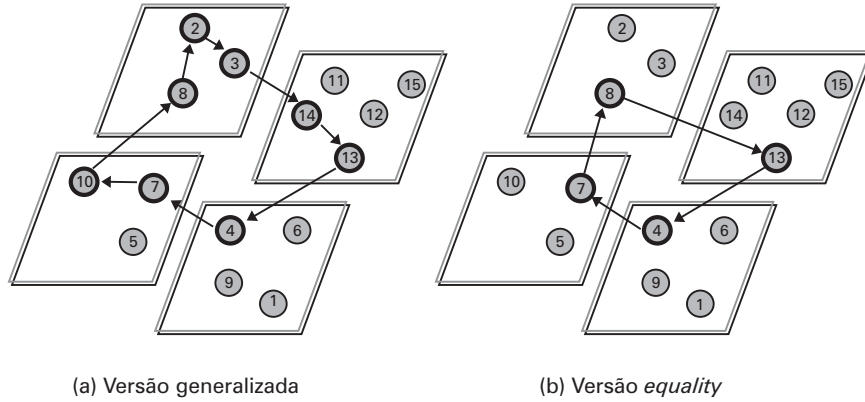


FIGURA 8.4 O PCV Generalizado (PCVG).

ria. Nesse tipo de problema sempre existe uma rede de postos de serviços (agências bancárias, filiais, representações etc.) onde um mensageiro pode executar indistintamente uma determinada atividade. Suponhamos que o problema seja executar r serviços em n possíveis postos, $n > r$. Admitamos que cada tipo de serviço possa ser realizado indistintamente em um ponto dos S_r conjuntos homogêneos que dividem n , ou seja, o mensageiro tem de passar por pelo menos um ponto pertencente a cada conjunto S_r para que possa executar as r tarefas. A solução via PCVG desse problema é o ciclo hamiltoniano de menor comprimento que visita um nó em cada um dos S_r conjuntos de pontos em que n é dividido. Laporte *et al.* (1996) apresenta outras aplicações em seu trabalho.

Suponhamos que o conjunto de nós de $G = (N, A)$ é dividido em grupamentos disjuntos S_1, S_2, \dots, S_r , ou seja, $N = S_1 \cup S_2 \cup \dots \cup S_r$. Seja $R = \{1, 2, \dots, r\}$ o conjunto de índices dos grupamentos. Supondo ainda que não existem arcos ligando esses nós, Noon e Bean (1991) apresentam o seguinte modelo de programação inteira para o PCVG:

$$(PCVG) \text{ Minimizar } z = \sum_{(i,j) \in A} c_{ij} x_{ij}$$

sujeito a:

$$\sum_{i \in S_l} \sum_{j \notin S_l} x_{ij} = 1 \quad (8.14)$$

$$\sum_{i \notin S_l} \sum_{j \in S_l} x_{ij} = 1 \quad (8.15)$$

$$\sum_{i \in N} x_{ij} - \sum_{k \in N} x_{jk} = 0 \quad j \in N \quad (8.16)$$

$$\sum_{l \in R} \sum_{i \in S_l} \sum_{j \notin R} \sum_{j \in S_j} x_{ij} \geq 1 \quad \forall R, 2 \leq |R| \leq r-2 \quad (8.17)$$

$$x_{ij} \in \{0, 1\} \quad (8.18)$$

O PCV com Backhauls (PCVB)

Este problema pode ser considerado um caso especial do (PCVG) em que os nós de G são particionados em dois grupamentos denominados normalmente de L (nós *linehauls*) e B (nós *backhauls*) (Kalan-tari *et al.* [1985]). A matriz de custo do problema atende às condições da norma euclidiana. Uma versão do problema determina que os nós de L sejam visitados inicialmente e, posteriormente, os nós de B . A estratégia dessa versão é dar preferência ao descarregamento dos veículos para, posteriormente, realizar o carregamento em direção ao depósito (nó inicial). A Figura 8.5 exemplifica o problema.

De fato, esse problema pode ser reduzido ao PCV pela consideração de penalidades convenientemente grandes entre as arestas que ligam os nós L e B .

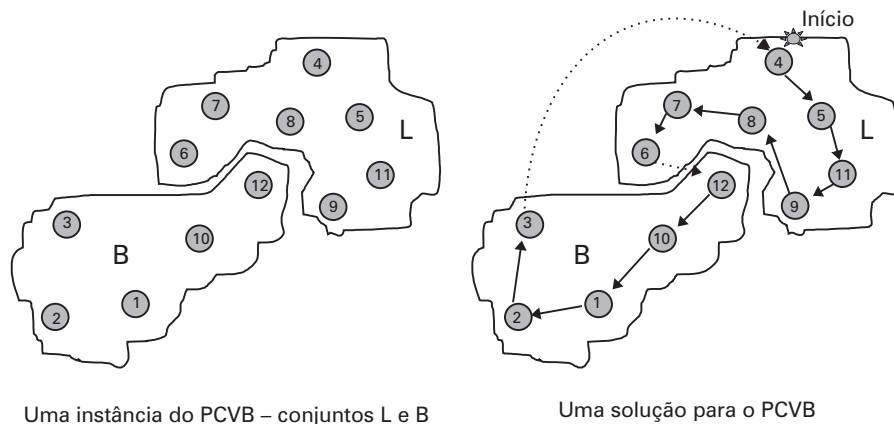


FIGURA 8.5 Exemplo de uma solução para o PCVB.

O PCV com Janelas de Tempo (PCVJT)

Se associarmos uma variável t_{ij} a cada arco do grafo $G = (N, A)$, representando a duração do percurso entre os nós i e j e assumirmos que a matriz $T = [t_{ij}]$ satisfaz à desigualdade triangular, podemos imaginar que a chegada a cada vértice de G possa estar associada a um intervalo de valores para o acúmulo do tempo gasto no deslocamento pela rota. Dessa forma, o valor acumulado do tempo de trajeto deve estar entre um certo intervalo de tempo $[a_i, b_i]$, denominado de janela de tempo, para cada vértice $i \in N$. Ainda associado a esse tipo de problema com restrição nos nós, encontramos outras variantes como o PCVJT que incluem restrições de precedência (PCVJT-RP). As restrições de precedência obrigam que a chegada a certos vértices seja antecedida por um determinado conjunto de outras visitas. Considerando que o vértice x_i seja um vértice de *backhaul*, ou seja, $i \in B$, (possui um carregamento para o depósito), tipicamente os vértices x_j , $j \in L$, representam um conjunto de precedência. Podemos considerar que tanto o PCVB como o PCVJT são casos particulares do PCVJT-RP.

Como, normalmente, os tempos e os custos em um percurso hamiltoniano são positivos e associados aos arcos de G , em algumas situações o PCV poderá ser formulado em função da minimização dos tempos de atendimento. Nesse caso, os t_{ij} poderão ter também impacto no custo do trajeto. O problema, então, poderá ser o de minimizar o tempo total de atendimento, respeitando-se limites de janelas de tempo e gastos com o deslocamento. Tal problema é constituído pelo PCVJT acrescido de restrições de custo.

O PCV Múltiplo (PCVM)

Este problema consiste em obter r tours, todos iniciando e terminado em um certo vértice (x_0 , por exemplo) em G , normalmente associados a r caixeiros, cuja soma total é mínima.

O PCV com Gargalo ou MinMax (PCV MinMax)

Este problema consiste em obter um ciclo hamiltoniano tal que sua aresta de maior comprimento seja mínima. Burkard (1991) formaliza o problema. É interessante notar que a solução do PCV é um limite

inferior para esse problema de otimização. Apesar de se tratar de um problema NP-árduo como os anteriores (Parker e Rardin [1984]) em diversas situações ele pode admitir solução polinomial. Burkard (1991) provou que para o caso em que a matriz de custo desse problema $C = [c_{ij}]$ pode ser decomposta em um produto de dois vetores (a_i) e (b_j) tal que $a_1 \leq a_2 \leq \dots \leq a_n$ e $b_1 \geq b_2 \geq \dots \geq b_n$, então o PCV MinMax pode ser solucionado em $O(n^2)$, enquanto o PCV associado permanece NP-árduo.

O PCV com Bônus (PCV-B)

Este problema foi proposto por Fischetti e Toth (1988) e também estudado por Balas (1989). Associando-se um bônus b_i a cada vértice x_i de um grafo $G = (N, A)$, o PCV-B consiste em determinar um *tour* hamiltoniano de comprimento mínimo de um vértice x_1 a um subconjunto $S \subseteq (N \setminus \{x_1\})$ tal que o retorno total obtido em bonificação nos vértices seja pelo menos igual a um certo valor p .

Consideremos que no problema do grafo da Figura 8.6 (a), em que o vértice x_1 é o vértice 1, seja exigido um bônus mínimo de 25 unidades e o subconjunto S de vértices de N seja de cardinalidade igual a 6. O ciclo de (c) seria uma solução melhor que (b). Notamos que o ciclo hamiltoniano de (b) é menor que o de (c) mas viola a condição de bônus mínimo. A solução 1-4-5-2-3-7-6-1 possui ciclo de comprimento 19 e bônus 29, sendo a solução ótima. Observamos que a solução 1-6-7-3-2-5-4-1, que possui comprimento 19 e bônus 30, é igualmente ótima, uma vez que a condição de bônus é apenas restritiva.

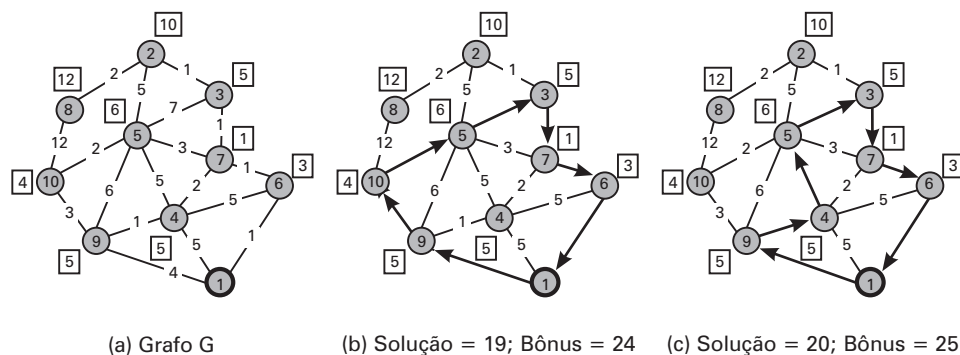


FIGURA 8.6 O PCV com bônus.

O PCV Seletivo (PCV-S)

Este problema foi estudado por Laporte e Martello (1990). Como vimos no item 3.7, o PCV-B utiliza o bônus como um elemento restritivo. Se o bônus ultrapassar o limite mínimo, as soluções com o mesmo comprimento mas com diferentes bônus não são distintas. O PCV-S aborda a situação em que, para um certo subconjunto $S \subseteq (N \setminus x_1)$, deseja-se agora maximizar o bônus, sujeito à restrição de que o comprimento do ciclo não seja superior a certo valor r .

Consideremos que no problema do grafo da Figura 8.7 (a), em que o vértice inicial x_1 corresponde ao nó 1, e que seja exigido que o comprimento do ciclo hamiltoniano não seja maior que 25 unidades, considerando-se $|S| = 6$. No exemplo da Figura 8.7 os ciclos (b) e (c) são legais, mas o ciclo representado na Figura 8.7 (c) seria a solução procurada. Notamos que o ciclo hamiltoniano de (b) é menor que o de (c), mas o critério de otimização agora diz respeito ao bônus.

O PCV Estocástico

O PCV clássico considera que o ciclo hamiltoniano desejado, ou o *tour* do caixeiro, se dará em um grafo em que os elementos constitutivos são determinísticos. No modelo estocástico, as janelas de tempo, os vértices, os tempos e os custos das arestas são elementos aos quais poderemos associar distribuições de probabilidades que lhes definirão a existência ou o valor. Os casos a seguir foram atualmente relatados na literatura:

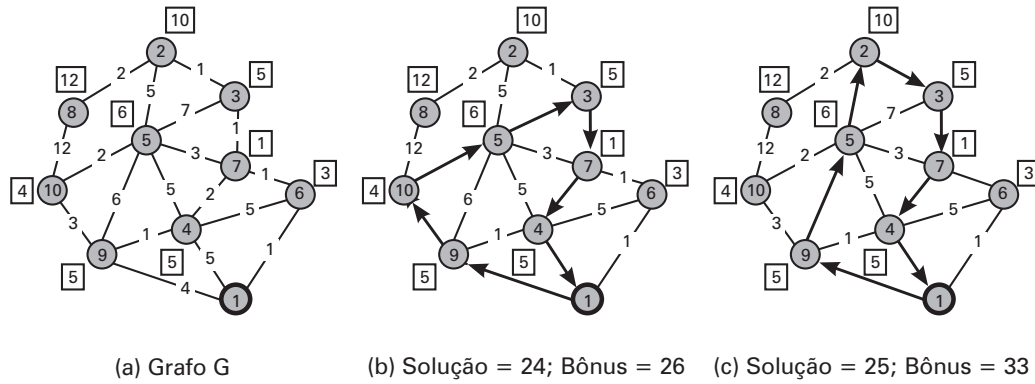


FIGURA 8.7 O PCV seletivo.

✓ PCV com Clientes Estocásticos (PCV-CE)

Nesta versão, o grafo $G = (N, A)$ tem seus vértices ocupados por clientes x_i , $i \in N$ segundo uma probabilidade p_i . O modelo considera, portanto, o caso em que existe a possibilidade de um vértice de G não demandar uma visita. A abordagem para exibir um *tour* legal dentro desse modelo inclui, normalmente, duas etapas:

- Etapa 1: exibir um *tour* sobre todos os vértices de $G = (N, A)$.
- Etapa 2: formar um *tour* legal eliminando os vértices não demandantes.

Esse problema foi introduzido por Jaillet (1985) que demonstrou que a solução determinística pode ser arbitrariamente ruim para o PCV-CE.

✓ O PCV com Tempo de Viagem Estocástico (PCV-TE)

No chamado PCV-TE os coeficientes de custo c_{ij} representam o tempo de viagem entre os vértices i - j e são variáveis aleatórias. Os autores que abordaram esse problema até o momento o fizeram tentando determinar uma solução *a priori* tal que a probabilidade de completar o *tour*, dentro de um dado prazo, fosse maximizada. Até o momento da edição deste trabalho, a literatura não reporta, pelos principais meios de comunicação da área, um modelo matemático para o problema. Kao (1978) é um dos primeiros trabalhos que propõem soluções para esse problema.

✓ PCV Múltiplo com Tempo de Viagem Estocástico (PCV-MTE)

Trata-se da versão múltipla do PCV-TE. Nesse caso, todas as rotas iniciam e terminam no mesmo nó (que obviamente representa uma sede para o caixeiro). O número de caixeiros (ou veículos) é uma variável de decisão que possui custos fixos associados. Os trabalhos que abordaram o tema impuseram um limite de tempo para a duração dos ciclos de cada caixeiro (um comprimento máximo para o ciclo), ver Gendreau *et al.* (1996).

O PCV Min Max-Min Sum (PCV-M³S)

Este caso corresponde ao objetivo Min Max-Min Sum para a função de otimização do PCV. Os problemas chamados de gargalo ou *bottleneck* são tradicionais no contexto da otimização porque estão associados a várias situações práticas que envolvem o deslocamento de facilidades. O Min Max-Min Sum é uma composição do problema clássico com o de gargalo, em que a função objetivo a ser minimizada é a soma algébrica das funções objetivo dos problemas mencionados. São dados p elementos de um conjunto $E = \{e_1, e_2, \dots, e_p\}$ e $F \subset P(E)$, onde $P(E)$ é o conjunto potência de E , uma família finita de subconjuntos de E perfeitamente definidos. Instâncias de F poderão ser, por exemplo, as subárvores de um grafo, os caminhos unindo dois vértices de um grafo ou conjuntos de arestas não adjacentes de um *matching* máximo. A cada elemento e_i , $i = 1, 2, \dots, p$ são associados dois números reais, um representado por c_i ou custo, e outro por p_i ou peso.

Sendo $S \in F$, onde F é o conjunto das soluções viáveis de um certo problema, define-se a função:

$$Z(S) = \max_{e_1 \in S} \{p_i\} + \sum_{e_i \in S} c_i$$

O problema de otimização Min Max-Min Sum pode ser formulado da seguinte forma (M³S1):

$$\min_{S \in F} Z(S)$$

Um dos primeiros trabalhos abordando um problema de otimização combinatória bicritério no formato M³S foi reportado por Hansen (1980) para o problema do caminho mais curto. O M³S foi reportado pela primeira vez como um problema distinto no trabalho de Minoux (Minoux [1986]), aparecendo como um subproblema do problema de decomposição da matriz relacionada a alocação ótima de tráfego de comunicação em satélites. Posteriormente, o problema foi associado, dentre outros, aos problemas de coloração de arestas em grafos valorados e particionamento de vértices e arestas (Minoux [1987]). Martello *et al.* (1984) apresenta um problema similar ao M³S no contexto da alocação balanceada. Berman, Einav e Handler (1990) abordam um problema bicritério com duas medidas de desempenho que poderiam ser modeladas como em M³S, mas que, no estudo, são tratadas separadamente, um problema como restrição do outro. Casos bicritérios semelhantes são solucionados através das superfícies de Pareto nos trabalhos de Braham (1966) e Frisch (1966). O M³S1 pode ser formulado de forma equivalente como expresso abaixo a qual denominaremos (M³S2):

$$\min_{1 \leq k \leq p} \left\{ P_k + \min_{S \in F_i^k} \left\{ \sum_{e \in S} C_i \right\} \right\}$$

De modo que solucionar M³S pode ser equivalente a solucionar a uma sequência de p subproblemas Min_Sum. O algoritmo de Minoux 1989 fundamenta-se exatamente na formulação M³S2.

No contexto das aplicações práticas o PCV-M³S pode modelar situações de roteamento entre células de produção em que o caixeiro representa um aparelho de transporte e a movimentação de produtos entre as células ocasionam impacto no tamanho das filas dentro de cada célula.

O PCV Periódico (PCV-P)

O Problema do Caixeiro Viajante Periódico (PCV-P) consiste em programar uma visita diária do caixeiro em um horizonte de planejamento de m dias de modo a visitar pelo menos uma vez uma das n cidades de um determinado conjunto $I = \{1, \dots, i, \dots, n\}$, considerando-se que o caixeiro retorna à sua cidade diariamente. O objetivo do problema é minimizar a distância total percorrida nos m dias do horizonte de planejamento. As cidades podem ser visitadas mais de uma vez dentro do período de m dias, contudo somente uma vez a cada dia. Não é permitido um trajeto diário vazio – sem visitas. O PCV-P foi primeiramente descrito no trabalho de Palleta (1992).

O PCV com Coleta (PCV-C)

Considerando que o veículo do caixeiro viajante possua uma determinada capacidade e que o caixeiro deva entregar e recolher produtos de seus clientes. Considerando ainda que os produtos coletados possam ser entregues a outros clientes na rota, o PCV com Coleta (PCV-C) consiste em programar um trajeto que, atendendo a todos os clientes de demanda e oferta dentro da capacidade do veículo minimize a distância percorrida. Os primeiros trabalhos no tema são devidos a Anily e Mosheiov (1994), e Mosheiov (1994) que descreve uma aplicação ao transporte escolar. A atual variante possui muitos pontos em comum com o problema de roteamento com entrega e coleta (*Pick-Ups and Deliveries*).

O PCV com Recompletação (PCV-R)

Considerando que pesos inteiros positivos w_i são atribuídos aos nós de um multigrafo direcionado $G = (N, A)$ e que o peso do caminho p é a soma dos pesos dos nós ao longo de p . Um subconjunto dos arcos é chamado de arcos de *recompletação* (*reenchimento*) e notado por $B \subseteq A$. Os demais arcos, denotados por $C = A \setminus B$, são chamados de *ordinários* (*comuns*). Um caminho é dito inviável se seu peso exceder a um determinado limite de peso constante e se ele não contém arcos de recompletação; caso contrário, ele é viável. Um caminho é minimamente não viável se todos os seus subcaminhos são viáveis.

O problema do caixeiro viajante com arcos de recompletação denotado por (PCV-R) é encontrar um tour de custo mínimo, isto é, um ciclo hamiltoniano direcionado, que não contém subcaminhos não viáveis. Nesse contexto faz sentido permitir a possibilidade da existência de dois arcos paralelos entre nós, um sendo um arco de recompletação e o outro um arco ordinário, de modo que o arco ordinário tenha custo mais baixo.

O PCV em Matriz Circulante (PCV-MC)

Trata-se do problema do Caixeiro Viajante em um grafo cuja matriz de distâncias é circulante (Kalman e White [2001]). A complexidade do PCV-MC é um problema em aberto. O primeiro trabalho abordando o tema se deve a Fink (1987). Yang *et al.* (1995) também aborda a solução desse problema.

O PCV Seletivo Euclidiano (PCV-E)

Considerando-se n cidades localizadas no plano euclidiano, o PCV-E consiste em encontrar o menor caminho fechado que visite todas as cidades exatamente uma vez.

O PCV de Máximo Espalhamento (PCV-ME)

Trata-se de determinar em um grafo ponderado e completo um caminho hamiltoniano em que o comprimento da menor aresta é maximizado. Problema descrito em Arkin *et al.* (1997).

O PCV m-Peripatético (PCV-mP)

Trata-se de determinar em um grafo ponderado m ciclos hamiltonianos disjuntos em aresta de forma a minimizar o custo acumulado de todos os ciclos. O problema foi descrito pela primeira vez em Krarup (1975). De Kort (1991) exhibe um limite inferior para o problema, e relata um algoritmo branch-and-bound para a solução do (PCV-2p) em De Kort (1993). Duchenne *et al.* (2003) propõe um algoritmo branch-and-cut para a solução do problema.

O PCV com Alvos Móveis

Dado um conjunto $S = \{s_1, s_2, \dots, s_n\}$ de s_i alvos móveis com velocidade constante v_i e posição inicial p_i , o problema consiste em determinar uma trajetória que inicie e termine em uma posição com velocidade $V > \max_i v_i$, $i = 1, 2, \dots, n$, e intercepte todos os alvos. O problema é descrito pela primeira vez em Helvig *et al.* (2003).

O PCV com Mínimo Desvio

Trata-se de um problema definido sobre o plano euclidiano. O objetivo é encontrar uma trajetória composta por retas que passem por um conjunto de pontos distribuídos em \mathbb{R}^2 de forma a minimizar o número de retas necessárias. O problema é descrito em Stein e Wagner (2000).

O PCV On-line

Trata-se da versão on-line do PCV, ou seja, quando os dados de entrada do problema são alterados ao longo do processo de solução.

O PCV Remoto

Dado um grafo ponderado e completo $G = (N, A)$ e um inteiro k , o problema consiste em determinar um subconjunto de nós de cardinalidade k tal que o custo do menor PCV no grafo formado pelos subconjunto de nós selecionados seja maximizado. Problema abordado por Halldórsson *et al.* (1995).

O PCV de Mínima/Máxima Latência

Dado um grafo não direcionado $G = (N, A)$ e cujo peso da arestas i é w_i . Para um dado caminho de comprimento P em G a latência L_j do j -ésimo vértice do caminho é dada pela soma de todas as arestas do caminho até esse vértice:

$$L_j = \sum_{i=1}^j w_i$$

O PCV de Máxima Latência é o circuito hamiltoniano em G cuja latência acumulada em todos os vértices é maximizada. O problema é abordado por Chalasani e Motwani (1999). Por sua vez Sarubbi (2004) propõe uma formulação de fluxos multiprodutos para o problema de mínima latência.

O PCV Com Custos Dependentes do Tempo

A presente variante é um caso particular de PCV dinâmico em que os custos das arestas variam ao longo do tempo. Um dos primeiros trabalhos no tema deve-se a Picard e Queyranne (1978). Brondén *et al.* (2004) estudam um caso em que o número de arcos com custo dependente é limitado e propõem vários algoritmos heurísticos de solução.

O PCV Com Requisitos de Separação

O problema consiste em determinar um circuito hamiltoniano de custo mínimo cumprindo uma ou mais das seguintes restrições:

- Restrições de inclusão, de modo que um certo nó i seja visitado (no mínimo/no máximo/exatamente) m nós (antes ou depois) de um certo nó j .
- Restrições de exclusão, de modo que um certo nó i não seja visitado (no mínimo/no máximo/exatamente) m nós (antes ou depois) de um certo nó j .

A presente variante é descrita por Balas *et al.* (1995), e uma reformulação do problema é proposta por Wang e Regan (2002).

O PCV Bipartido

O problema consiste em determinar um circuito hamiltoniano de custo mínimo sobre um grafo ponderado que possui n nós vermelhos e n nós azuis, de modo que o circuito inicie em um nó azul e os nós vizinhos no circuito possuem sempre cores diferentes. O problema encontra-se definido em Chalasani *et al.* (1996) juntamente com algoritmos heurísticos para o problema.

8.4 – MÉTODOS DE SOLUÇÃO E ALGORITMOS

8.4.1 – Uma Visão Geral

A literatura que aborda a solução do PCV nas suas diversas versões é uma das mais ricas dentro da pesquisa operacional. As Tabelas de 8.1 a 8.5, sem nenhuma pretensão de esgotar o assunto, organizam os principais trabalhos em uma visão panorâmica. Desde o trabalho de Bellmore e Nemhauser (1968), várias compilações sobre o PCV foram publicadas na literatura. Recentemente, destaca-se o trabalho de Miller e Pekny (1991).

TABELA 8.1 TRABALHOS PARA O PCV CLÁSSICO – SOLUÇÃO EXATA

<i>Problema</i>	<i>Ano</i>	<i>Pesquisador</i>	<i>Trabalho</i>
Problema do Caixeiro Viajante (PCV)	Abordagens Exatas		
	1954	Dantzig <i>et al.</i>	Trabalho referência para o PCV
	1972	Christofides e Eilon	Métodos exatos
	1973	Laporte e Norbert	Métodos exatos
	1980	Carpaneto e Toth	Crítérios para algoritmos B&B
	1980	Crowder e Padberg	B&B
	1981	Balas e Christofides	Restrições lagrangeanas para o PCV
	1985	Fleischmann	Algoritmo com uso de planos de cortes
	1991	Miller e Pekny	Métodos exatos
	1992	Gendreau <i>et al.</i>	Procedimentos de pós-otimização
	1995	Jünger <i>et al.</i>	Relaxações e B&Cut
	1998	Applegate <i>et al.</i>	B&Cut
	2000	Jünger <i>et al.</i>	Cortes
	2001	Applegate <i>et al.</i>	Cortes
	2002	Focacci <i>et al.</i>	Relaxação em restrições globais
	2004	Marcotte <i>et al.</i>	Programação multinível

TABELA 8.2 ALGORITMOS HEURÍSTICOS PARA O PCV CLÁSSICO

<i>Problema</i>	<i>Ano</i>	<i>Pesquisador</i>	<i>Trabalho</i>
Problema do Caixeiro Viajante (PCV)	1973	Lin e Kernighan	Métodos heurísticos
	1980	Golden <i>et al.</i>	Várias Heurísticas
	1980	Kanellakis e Papadimitriou	Busca local
	1983	Adrabinski	Experimentos comp. para heurísticas
	1985	Goldberg e Lingle	Algoritmos genéticos
	1985	Grefenstette <i>et al.</i>	Algoritmos genéticos
	1985	Cerny	Métodos termodinâmicos (Sim. anel.)
	1987	Laarhoven e Aarts	Métodos termodinâmicos (Sim. anel.)
	1988	Burr	Métodos elásticos
	1988	Angeniol <i>et al.</i>	Mapa de auto-organização
	1990	Johnson	Otimização local
	1990	Fiechter	Busca tabu paralela
	1991	Ulder	Algoritmos genéticos
	1991	Whitley	Algoritmos genéticos
	1991	Glover	Busca tabu
	1991	Xu e Tsai	Redes neurais
	1992	Gendreau <i>et al.</i>	Procedimentos de pós-otimização
	1993	Reeves	Métodos aproximativos
	1993	Potvin	Redes Neurais
	1994	Chandra <i>et al.</i>	Procedimento k-ótimo
	1994	Valenzuela e Jones	Divisão e Conquista e Alg Genéticos

TABELA 8.2 CONTINUAÇÃO

<i>Problema</i>	<i>Ano</i>	<i>Pesquisador</i>	<i>Trabalho</i>
Problema do Caixeiro Viajante (PCV)	1994	Fiechter	Busca tabu paralela
	1995	Jünger <i>et al.</i>	Relaxações e B&Cut
	1995	Potts e Van de Velde	Heurística híbrida de k-substituição
	1995	Vakhutinsky e Golden	Abordagem elástica
	1996	Chatterjee <i>et al.</i>	Algoritmos genéticos
	1996	Potvin	Algoritmos genéticos
	1996	Cesari	Estratégia de divisão e conquista p/ heurist
	1997	Somhom <i>et al.</i>	Redes neurais/modelo auto-organizado
	1997	Dorigo e Gambardella	Colônias de Formigas
	1998	Tsubakitani e Evans	Busca Tabu
	1999	Voudouris e Tsang	Busca Local
	1999	Gutin e Yeo	Vizinhanças de busca local
	1999	Shin <i>et al.</i>	Programação Molecular
	1999	Coy <i>et al.</i>	Heurísticas smoothing
	1999	Okano	Heurísticas Construtivas
	1999	Aras <i>et al.</i>	Rede de Kohonen com info estatística
	1999	del Castillo	Aproximação contínua
	2000	Charon e Hudry	Noise Method
	2000	Helsgaun	Implementação eficiente Lin-Kernighan
	2001	Munehisa <i>et al.</i>	Atualização cooperativa / rede Hopfield
	2001	Zeng e Martinez	Reescalamento / rede Hopfield
	2002	Ugajin	Método de difusão inversa
	2003	Eppstein	PCV em grafos cúbicos
	2003	Li. e Gong	Colônia de Formigas
	2003	Applegate <i>et al.</i>	Heurística Chained Lin-Kernighan

TABELA 8.3 TRABALHOS DE REVISÃO DA LITERATURA

<i>Problema</i>	<i>Ano</i>	<i>Pesquisador</i>	<i>Trabalho</i>
Problema do Caixeiro Viajante (PCV)	Trabalhos de Revisão Bibliográfica e Análise		
	1977	Rosenkrantz <i>et al.</i>	Revisão das abordagens heurísticas
	1985	Lawler <i>et al.</i>	Revisão do estado da arte – texto didático
	1990	Melamed <i>et al.</i>	Revisão do estado da arte
	1992	Laporte	Revisão do estado da arte
	1994	Bianco <i>et al.</i>	Revisão do estado da arte
	1994	Reinelt	Estudo computacional
	1995	Juenger <i>et al.</i>	Revisão do estado da arte – texto didático
	1998	Burkard <i>et al.</i>	Casos especiais de solução eficiente
	1999	Larrañaga <i>et al.</i>	Algoritmos Genéticos para o PCV
	2001	Balas e Simonetti	Estudo algorítmico
	2002	Johnson e McGeoch	Análise experimental para o PCVS
	2002	Punnen e Kabadi	Análise de dominância entre heurísticas
	2002	Johnson <i>et al.</i>	Análise experimental para o PCVAS

TABELA 8.4 TRABALHOS PARA VÁRIAS VERSÕES DO PCV – PARTE I

<i>Problema</i>	<i>Ano</i>	<i>Pesquisador</i>	<i>Trabalho</i>
PCV Generalizado (PCVG)	1969	Henry-Labordere	Programação dinâmica
	1985	Jonges e Volgenant	Relaxação lagrangeana
	1987	Laporte <i>et al.</i>	Algoritmo B&B para o caso simétrico
	1991	Noon e Bean	Formulação inteira do problema
	1994	Fischetti <i>et al.</i>	Algoritmo B&Cut / caso simétrico
	1996	Laporte <i>et al.</i>	Aplicações práticas do problema
	1996	Gendreau <i>et al.</i>	Algoritmos heurísticos
	1996	Laporte <i>et al.</i>	Aplicações do problema
	1997	Fischetti <i>et al.</i>	Algoritmo B&Cut
	1997	Chentsov e Korotayeva	Programação dinâmica
	1998	Renaud e Boctor	Heurística composta
	1999	Laporte e Semet	Procedimento de transformação
	1999	Balas	Problemas de solução eficiente
PCV Simétrico (PCVS)	1980	Padberg e Hong	Estudo computacional do problema
	1980	Jonker <i>et al.</i>	Transformações do problema
	1982	Jonker e Volgenant	B&B
	1987	Stewart	Heurísticas para o PCVS
	1989	Carpaneto <i>et al.</i>	Limites para o problema
	1990	Padberg e Rinaldi	Identificação de facetas
	1991	Padberg e Rinaldi	Algoritmos B&Cut
	1991	Grötschel e Holland	Solução de problemas de grande porte
	1994	Knox	Busca tabu
	1996	Zachariasen e Dam	Busca tabu
	1996	Codenotti <i>et al.</i>	Método da perturbação
	1996	Renaud <i>et al.</i>	Heurística composta
	2002	Johnson e McGeoch,	Análise experimental
	2002	Lorena e Narciso	Informação lógica e relaxação lagrangeana
PCV Assimétrico (PCVAS)	1991	Miller e Pekny	Algoritmo exato
	1996	Freisleben e Merz	Algoritmo genético com busca local – memético
	1996	Gambardella e Dorigo	Colônia de formigas
	1998	Brest e Zerovnik	Algoritmo heurístico
	1999	Amin	“Simulated Jumping”
	2000	Zhang	Busca Local e B&B
	2000	Ascheuer	B&Cut
	2001	Cirasella <i>et al.</i>	Algoritmos e instâncias teste
	2001	Glover <i>et al.</i>	Heurísticas Construtivas
	2002	B1aser	Heurísticas para o máximo PCVAS
	2003	Lewenstein e Sviridenko	Heurísticas para o máximo PCVAS
	2003	Choi <i>et al.</i>	Algoritmos Genéticos

TABELA 8.4 CONTINUAÇÃO

<i>Problema</i>	<i>Ano</i>	<i>Pesquisador</i>	<i>Trabalho</i>
PCV com Backhauls (PCVB)	1985	Kalantari <i>et al.</i>	Algoritmos exatos para casos especiais
	1989	Goetschalckx e JB	Vários algoritmos
	1994	Anily e Mosheiov	Modelo de programação
	1996	Gendreau <i>et al.</i>	Algoritmos heurísticos
	1997	Gendreau <i>et al.</i>	Algoritmo heurísticos híbridos
	1997	Hansen e Mladenovic	Heurística de busca em vizinh. var.
	2003	Gharizi e Osman	Redes Neurais
O PCV com J. Tempo (PCVJT)	1980	Psaraftis	Programação dinâmica
	1983	Baker	B&B
	1983	Psaraftis	Algoritmo enumerativo
	1986	Desrosiers <i>et al.</i>	Programação dinâmica
	1992	Malandraki e Daskin	Estudos e algoritmos heurísticos
	1993	Langevin <i>et al.</i>	Programação dinâmica
	1994	Bianco <i>et al.</i>	Programação dinâmica / limites
	1995	Dumas <i>et al.</i>	Testes de dominância e programação dinâmica
	1995	Russell	Heurísticas híbridas
	1996	Malandraki, e Dial	Heurísticas de prog dinâmica restrita
	1997	Mingozzin <i>et al.</i>	Programação dinâmica / limites
	1998	Pesant <i>et al.</i>	Programação Lógica
	1998	Gendreau <i>et al.</i>	Heurísticas de inserção
	1999	Ascheuer <i>et al.</i>	B&Cut
	2000	Calvo	Heurística baseada no assignment
	2000	Fagerholt e Christiansen	Aplicações
	2002	Focacci <i>et al.</i>	Relaxação em restrições globais
O PCV Múltiplo (PCVM)	1977	Russel	
	1988	Desrosiers <i>et al.</i>	Relaxação lagrangeana
	1991	Cvetkovic	Programação dinâmica
	1995	França <i>et al.</i>	PCVM com objetivo minmax
	2000	Tang	Aplicações do modelo
	2002	Sofge <i>et al.</i>	Computação evolucionária
	2003	Pérez e Gil	Colônia de formigas

TABELA 8.5 TRABALHOS PARA VÁRIAS VERSÕES DO PCV – PARTE II

<i>Problema</i>	<i>Ano</i>	<i>Pesquisador</i>	<i>Trabalho</i>
O PCV Periódico (PCV-P)	1992	Palleta	Modelo e algoritmos heurísticos
	1995	Chao <i>et al.</i>	Algoritmo heurístico
	2002	Palleta	Algoritmo heurístico
	2004	Bertazzi <i>et al.</i>	Algoritmo heurístico

TABELA 8.5 CONTINUAÇÃO

<i>Problema</i>	<i>Ano</i>	<i>Pesquisador</i>	<i>Trabalho</i>
O PCV Geométrico (PCV-G)	1992	Bentley	Algoritmos heurísticos
	1996	Zachariasen e Dam	Busca Tabu
	1998	Barvinok	PCV geométrico máximo
	1999	Mitchell	Heurística de subdivisões poligonais
	1999	Fekete	PCV máximo geométrico
	2002	Jonsson	PCV em linhas no plano
	2003	Barvinok <i>et al.</i>	PCV geométrico máximo
O PCV Euclidiano (PCV-E)	1996	Arora	Esquema polinomial de aproximação
	1997	Cerf <i>et al.</i>	Aproximação por ligações aleatórias
	1998	Al-Mulhem e Al-Maghrabi	Método elástico
	1998	Arora	Esquema aproximação em tempo linear
	2000	Aras <i>et al.</i>	Mapa KNIES de auto-organização
	2000	Bonachea <i>et al.</i>	Abordagem Multistart
	2004	Dumitrescu e Mitchell	Aproximação em vizinhanças no plano
O PCV com Recompletação (PCV-R)	2000	Boland <i>et al.</i>	Descrição do modelo
	2001	Mak	Descrição do modelo
O PCV com Coleta (PCV-C) pick-delivery)	1994	Mosheiov	Modelo para o PCV-C
	1994	Anily e Mosheiov	
	1998	Mosheiov	Heurísticas de particionamento
	1999	Gendreau <i>et al.</i>	Heurísticas para o problema
	1999	Anily e Bramel	Heurísticas para o problema
	2002	Renaud <i>et al.</i>	Heurísticas baseadas em perturbação
	2003	Baldacci <i>et al.</i>	B&Cut
	2003	Salazar-González e Hernández-Pérez	Caso de entrega e coleta de um único item
	2003	Bruun <i>et al.</i>	Formulação por Fluxo em Redes
	2004	Salazar-González e Hernández-Pérez	B&Cut
O PCV Min-Max (PCV MinMax)	1995	Frania <i>et al.</i>	PCV Múltiplo com objetivo MinMax
	1997	Averbakh e Berman	Algoritmos heurísticos
	1997	Golden <i>et al.</i>	Heurística com memória adaptativa
	1999	Somhom <i>et al.</i>	Redes neurais
PCV com Bônus (PCV-B)	1988	Fischetti e Toth	Algoritmo de solução
	1989	Balas	Modelo e inequações para o problema
	1996	Blum <i>et al.</i>	Heurística $O(\log^2 k)$. k = nós coletados
	1998	Dell'Amico <i>et al.</i>	Heurística lagrangeana
O PCV Seletivo (PCV-S)	1990	Laporte e Martello	Modelo PCV Seletivo

TABELA 8.5 CONTINUAÇÃO

<i>Problema</i>	<i>Ano</i>	<i>Pesquisador</i>	<i>Trabalho</i>
PCV com Clientes Estocásticos (PCV-CE)	1985	Jaillet	Soluções derivadas do PCV
	1985	Jézéquel	Heurísticas de economia
	1987	Rossi e Gavioli	Soluções heurísticas
	1988	Jaillet	Modelos
	1993	Bertsimas	Soluções heurísticas
	1994	Laporte <i>et al.</i>	Método L-Shaped
	2002	Bianchi <i>et al.</i>	Colônia de Formigas
	2003	Bowler <i>et al.</i>	Características do problema
O PCV com Tempo de Viagem Estocástico (PCV-TE)	1978	Kao	Heurísticas
	1981	Sniedovich	Heurísticas via programação dinâmica
	1989	Carraway <i>et al.</i>	Programação dinâmica generalizada
	1999	Percus e Martin	Heurística "Cavity Prediction"
PCV Múltiplo com Tempo de Viagem Estocástico (PCV-MTE)	1992	Laporte <i>et al.</i>	Método L-Shaped
	1993	Lambert <i>et al.</i>	Penalidades e algoritmos de economia
PCV com custos Estocásticos	2004	Choi <i>et al.</i>	Uma arquitetura para heurísticas

8.4.2 – Limites para o PCV

Ser capaz de obter bons limites para o PCV é um dos caminhos para o sucesso em vários algoritmos de B&B, B&Cut, enumeração implícita etc. São conhecidas diversas formas de obtenção de limites, dentre elas destacamos:

✓ Limites Obtidos da Árvore Geradora Mínima (T_{GM})

Chamando por Φ um circuito hamiltoniano no grafo $G = (N, A)$, um grafo não direcionado, $x_i \in N$ um vértice de Φ , e A_i o conjunto das arestas incidentes sobre o vértice x_i . Se removermos x_v , um dos vértices do ciclo, então teremos um caminho p em G . Se T_{GM} for a árvore geradora mínima do subgrafo $G = (N \setminus \{x_v\}, A \setminus A_v)$ então, claramente, T_{GM} é um limite inferior para p . Os arcos removidos de Φ , são tão grandes quanto os dois menores arcos incidentes em v , (i_1, v) e (i_2, v) por exemplo. Chamando por $G_v = (N_v, A_v)$ o grafo composto pelos arcos de T_{GM} e pelos arcos (i_1, v) e (i_2, v) , e pelo nó v . É claro que o custo total dos arcos em G_v é um limite inferior para Φ . O problema da determinação do grafo G_v que minimize esse limite pode ser expresso pelo programa linear Lim_Arv. Uma outra forma de entender essa relaxação é perceber que se desconsiderarmos as restrições unitárias em estrita igualdade (restrições de alocação linear) a estrutura resultante da solução do problema formulado será uma 1-árvore. A relaxação assim constituída obviamente poderá resultar em um *tour* ilegal; contudo, seu valor será um limite inferior para qualquer *tour* hamiltoniano (legal) em G .

$$\begin{aligned}
 (\text{Lim_Arv}) \text{ Minimizar } z &= \sum_{j=1}^m c_j x_j \\
 \text{sujeito a:} \\
 \sum_{j=1}^m x_j &= n
 \end{aligned} \tag{8.19}$$

$$\sum_{j \in K_t} x_j \geq 1 \quad \forall K_t \equiv (S_t, \bar{S}_t), S_t \subset N \quad (8.20)$$

$$\sum_{j \in A_v} x_j = 2 \quad (8.21)$$

$$x_j \in \{0, 1\} \quad j = 1, \dots, m \quad (8.22)$$

onde $\bar{S}_t = N \setminus S_t$, e $K_t \equiv (S_t, \bar{S}_t)$, $S_t \subset N$ representa o conjunto de arcos (i, j) em que o vértice $i \in S_t$ e $j \in \bar{S}_t$. K_t é também denominado *conjunto de arcos de corte* de G . Em relação a um *tour* hamiltoniano, pelo menos um dos arcos deve pertencer a qualquer conjunto de corte K_t de G . x_j é uma variável binária que vale 1 se o arco j está na solução e 0 em caso contrário. Um limite melhor pode ser obtido se considerarmos todos os nós do grafo no conjunto A_v , da última restrição, ou seja, $v = 1, \dots, n$. Entretanto, fazer isso é aproximar a formulação da formulação original do PCV. Um outro caminho é introduzir essa restrição na função objetivo através do uso de multiplicadores de lagrange λ . Nesse caso, a função objetivo de Lim_Arv se transformará em:

$$\min_x \left(\sum_{j=1}^m c_{ij} x_{ij} \right) + \sum_{i \in N} \lambda_i \left(\sum_{j \in A_i} x_j - 2 \right)$$

Dessa forma, o problema Lim_Arv pode ser modelado em função de λ como:

$$\min_x \left[\sum_{j=1}^m (c_j + \lambda_{i_j} + \lambda_{k_j}) - 2 \sum_{i \in N} \lambda_i \right]$$

Sujeito às demais restrições de Lim_Arv, onde i_j e k_j são os dois nós (i e k) que definem o arco j (representado no índice). Na verdade, a função objetivo em λ representa uma transformação de custos na função original de Lim_Arv tal que:

$$c_j = c_j + \lambda_{i_j} + \lambda_{k_j}$$

Os melhores valores para os λ (ou penalidades) são obtidos da solução do problema em função de λ . Métodos para solucionar o problema em λ foram sugeridos por Held e Karp (1970) Hansen e Krarup (1974). Valenzuela e Jones (1997) resumem e analisam vários métodos derivados do limite Held-Karp.

✓ Limites Obtidos do Problema de *Matching* (Emparelhamento) (PE)

Para o caso do PCV simétrico é possível obter-se um limite inferior normalmente de melhor qualidade do que o anterior através da relaxação do PCV como um PE. A relaxação do PCV pode-se configurar com o problema de programação linear que se segue:

$$(\text{Lim_Emp}) \text{ Minimizar } z = \sum_{j=1}^m c_j x_j$$

sujeito a:

$$\sum_{j \in A_i} x_j = 2$$

$$x_j \in \{0, 1\} \quad j = 1, \dots, m$$

Essa formulação pode ser obtida da relaxação associada à árvore geradora mínima pela desconsideração das restrições de continuidade da estrutura (a árvore é um subgrafo conexo). Podemos, como no caso anterior, incluir a restrição na função objetivo através de multiplicadores de lagrange. O problema é, essencialmente, o mesmo anterior e suas técnicas de solução análogas.

✓ Limites Obtidos do Problema de Alocação Linear (PAL)

Como fica claro na formulação PCV1, os dois primeiros conjuntos de restrições representam um problema de alocação linear. Balas e Christofides (1976) sugerem a introdução das demais restrições do problema, na medida de sua violação em uma relaxação linear, na função objetivo, através de multiplicadores de lagrange. Os detalhes da proposta podem ser vistos em Christofides (1979).

8.4.3 – Algoritmos Aproximativos de Solução

Tendo em vista a extrema importância da obtenção de bons limites para a possibilidade de uma solução exata do PCV em problemas de porte real (muitas vezes com dezenas ou centenas de milhares de clientes) e do elevado grau de sofisticação dos algoritmos exatos da atualidade (que fogem ao escopo do presente trabalho), concentraremos nossa atenção em algumas abordagens aproximativas.

✓ Solução do PCV através de Algoritmos Genéticos (AG)

Os algoritmos genéticos constituem métodos de busca baseados em mecanismos de seleção e evolução natural. Os primeiros trabalhos nessa linha são originários de John Holland (Holland [1962] e [1970]), e objetivavam replicar os processos utilizados pelos sistemas autoadaptativos em um contexto computacional. Obviamente, nem todos os mecanismos de evolução e adaptação dos sistemas biológicos estão sequer modelados, contudo, vários aspectos já são bem conhecidos. Um dos pontos bem aceitos é de que o processo de adaptação depende de uma forte interação entre uma certa configuração genética portada pelo indivíduo e os aspectos ambientais. Lamarck em 1809 formulou a primeira hipótese sistematizada sobre evolução, afirmando que os seres vivos evoluíam sem saltos ou cataclismos e de forma lenta e segura, introduzindo o conceito de caracteres adquiridos. Charles Darwin desenvolveu uma teoria completa explicando como o meio ambiente atuaria sobre os indivíduos, forçando uma seleção dos mais aptos. Johann Gregor Mendel em 1866 apresenta uma nova teoria para explicar como ocorreriam os fenômenos hereditários, o que viria a se transformar no que hoje conhecemos como genética.

Já em meados do século passado, os cientistas conheciam, no núcleo das células que podiam ser observadas pelos primeiros microscópios, filamentos enrolados denominados cromossomos. O nome foi originário da facilidade com que eram marcados por corantes. Thomas Hunt Morgan descobriu que esses filamentos eram constituídos por subgrupos que foram denominados genes, e que estes controlavam, dentre outras coisas, a cor dos olhos de certas moscas. Ele demonstrou que esses grupos eram divididos e recompostos formando um fenômeno chamado de cruzamento ou reprodução sexuada.

Partindo do modelo de cromossomos para representar o processo evolutivo, Holland (Holland [1973] e [1975]) utilizou uma lista de símbolos binários (0, 1) para representar as cadeias do ácido nucleico. Na verdade, os objetivos do trabalho de Holland eram fundamentar uma teoria geral de sistemas de adaptação robusta, contudo, foi capaz de encontrar um caminho de grande e imediata aplicação prática na determinação de máximos e mínimos de funções matemáticas. Isso facilitou sobremaneira a aceitação dos AG no meio acadêmico. Diversos autores contribuíram para a consolidação do trabalho de Holland, como DeJong (1975), Grefenstette (1986), Goldberg (1989) e outros. Em resumo, podemos entender que os AG são uma parte da computação evolucionária que encontrou a possibilidade de aplicação geral e que possui as seguintes características gerais:

- Operam em um conjunto de pontos (denominado população) e não a partir de pontos isolados.
- Operam em um espaço de soluções codificadas e não diretamente no espaço de busca.
- Necessitam como informação somente o valor de uma função objetivo (denominada função de adaptabilidade ou *fitness*).
- Usam transições probabilísticas e não regras determinísticas.

Em resumo, podemos entender que os AG são uma parte da computação evolucionária que encontrou a possibilidade de aplicação geral. Os termos que se seguem são intensamente usados por esses algoritmos:

- **População:** conjunto de indivíduos (isto é, conjunto de soluções do problema).
- **Cromossomo:** representa um indivíduo na população (isto é, uma configuração ou solução). Definido normalmente como um vetor de componentes.
- **Fitness:** medida de aptidão de um indivíduo (isto é, normalmente associada ao valor da função objetivo para uma dada solução).
- **Gene:** representa uma componente do cromossomo (isto é, uma variável do problema).
- **Allele** ou **Alelo:** descreve os possíveis estados de um atributo do indivíduo (isto é, os possíveis valores de uma variável do problema).
- **Locus:** representa a posição do atributo no cromossomo (isto é, a posição da componente no vetor de componentes).
- **Operadores genéticos:** são as regras que permitem a manipulação dos cromossomos que, basicamente, são de:
 - *crossover* (cruzamento), ou o operador que permite a obtenção de indivíduos filhos a partir da combinação (cruzamento) dos cromossomos dos pais.
 - *mutação* ou o operador que permite a produção de um novo indivíduo por alterações diretas no cromossomo pai.
- **Fenótipo:** denota o cromossomo decodificado.
- **Genótipo:** representa a estrutura do cromossomo codificado.
- **Schema:** é um modelo de representação para uma família de cromossomos, normalmente é representado por símbolos * dentro do cromossomo.

A Figura 8.8 exemplifica alguns dos termos.

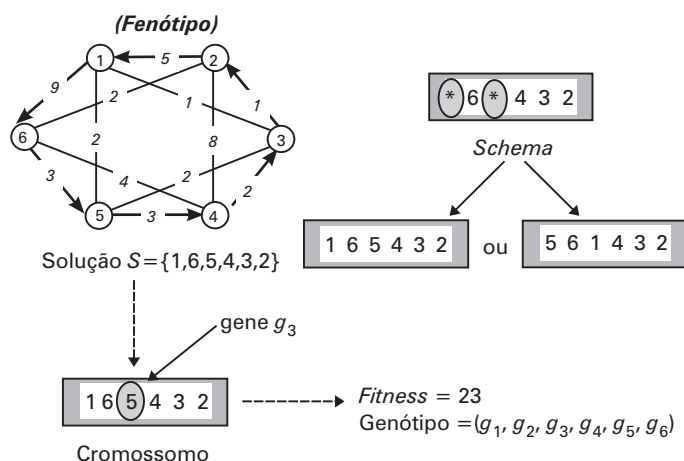


FIGURA 8.8 Um exemplo de formação do cromossomo.

De fato, os AG são algoritmos de busca probabilística inteligente, com as seguintes características básicas:

- *Os indivíduos são representados por cromossomos e competem por recursos e possibilidade de reprodução.* Os cromossomos, normalmente, estão associados a soluções dos problemas modelados. Uma população inicial é formada, em princípio, através de algum mecanismo de avaliação de performance. Quando esse mecanismo não é ainda conhecido, a população pode ser formada de modo

aleatório. Os indivíduos, em geral, são codificados como uma sequência finita em um alfabeto binário $\{0, 1\}$. Cada componente da sequência é dito um gene. É natural supor que um gene esteja associado a uma variável do problema. A partir da solução inicial, o mecanismo de reprodução será aplicado.

- Os indivíduos que tiverem mais sucesso nas competições terão maior probabilidade de reprodução do que aqueles de menor performance. O mecanismo de reprodução é constituído por uma função de avaliação que classifica os indivíduos por sua performance e por regras que permitirão que esses melhores indivíduos se perpetuem e reproduzam. Observamos a importância da função *fitness* ou de avaliação de performance. O chamado método da *roulette wheel* escolhe o indivíduo para ser um genitor com uma probabilidade igual à sua adequação normalizada, ou seja, o resultado de sua adequação absoluta dividida pela adequação média da população avaliada. O método *tournament* sorteia aleatoriamente k membros da população e, nesse grupo, escolhe o indivíduo de melhor desempenho. É importante perceber a natureza probabilística do processo, objetivando manter a diversidade da população.
- Os genes dos indivíduos avaliados como bons se propagam através das populações, de modo que possam ser aperfeiçoados e gerem proles cada vez mais adequadas. Os indivíduos selecionados serão transformados por operadores genéticos em novos indivíduos. Existem operadores associados à reprodução com a utilização de um ou de mais de um indivíduo. O operador primário da reprodução utiliza dois indivíduos, sendo denominado *cruzamento* ou *crossover*. Existem vários modos de realizar um cruzamento entre os indivíduos representados por cromossomos. Basicamente, o objetivo do cruzamento é formar novos indivíduos a partir da troca de genes entre os indivíduos pais. As regras que definem como essa troca de genes ocorrerá são especificadas para cada caso.

A Figura 8.9 mostra a possibilidade de um cruzamento em que a posição de permutação é escolhida de forma aleatória.

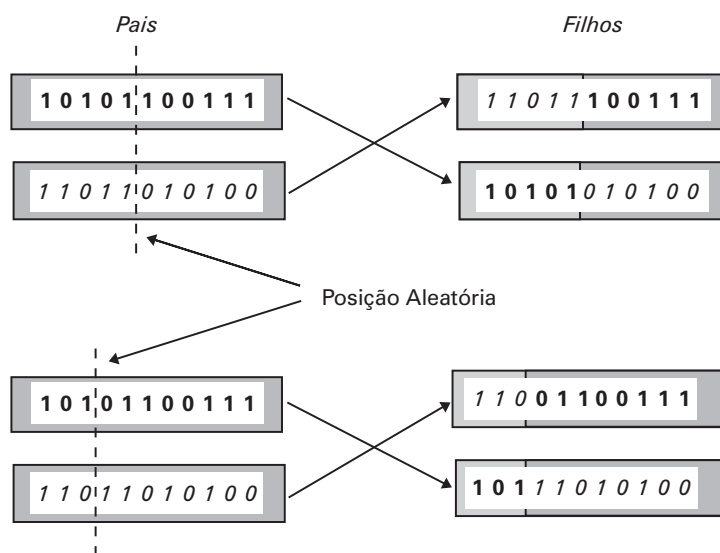


FIGURA 8.9 Cruzamento simples de posição aleatória.

Completada a reprodução por cruzamento, a população pode ser submetida (não obrigatoriamente) a um processo de mutação. Essa alternativa de geração de novos indivíduos pode ser feita diretamente de um outro indivíduo. Tal modo de reprodução é denominado *mutação*. A Figura 8.10 mostra dois tipos possíveis de mutação.

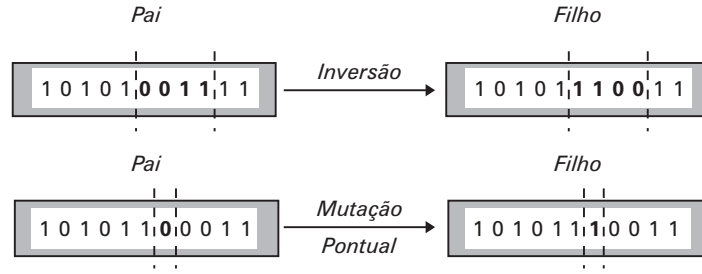


FIGURA 8.10 Mecanismos de muta  o.

Podemos definir formalmente um AG da seguinte forma:

$$AG = (N, P, F, \Theta, \Omega, \Psi,)$$

Onde P   uma popula  o de N indiv duos, $P = \{S_1, S_2, \dots, S_N\}$. Cada indiv duo S_i , $i = 1, \dots, N$   uma cadeia ou um conjunto de valores inteiros de comprimento n representando uma solu  o do problema. F representa a fun  o *fitness* que retorna um valor positivo e real na avalia  o de cada indiv duo.

$$F: S_i \rightarrow \mathbb{R}^+, i = 1, \dots, N$$

Θ   o operador de sele  o de pais que escolhe r indiv duos de P

$$\Theta: P \rightarrow \{p_1, p_2, \dots, p_r\}$$

Ω   um conjunto de operadores gen ticos que inclui o crossover (denominado Ω_C), o operador de muta  o (denominado Ω_M) e qualquer outro operador espec fico que produza s filhos a partir de r pais.

$$\Omega = \{\Omega_C, \Omega_M, \dots\}: \{p_1, p_2, \dots, p_r\} \rightarrow \{f_1, f_2, \dots, f_s\}$$

Ψ   o operador de remo  o que retira s indiv duos selecionados na popula  o P , permitindo, por exemplo, que s filhos (novos indiv duos) sejam adicionados   nova popula  o P_{t+1} :

$$P_{t+1} = P_t - \Psi(P_t) + \{f_1, f_2, \dots, f_s\}.$$

τ   o crit rio de parada

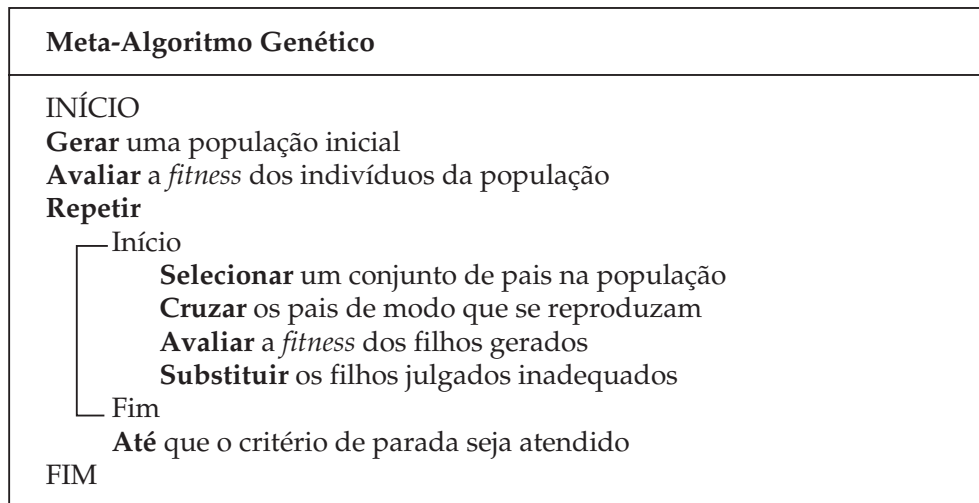
$$\tau: P_t \rightarrow \{\text{verdadeiro}, \text{falso}\}$$

A pr pria concep  o do modelo gen tico o fez flex vel e abstrato. O significado das cadeias de cromossomos   completamente livre. O que est  em foco no modelo   uma concep  o generalista de um processo adaptativo, por isso cabe a denomina  o de *meta-heur stica*. A cada situa  o cabe ao modelador concretizar o processo adaptativo e de sele  o natural preconizado na estrat gia. A representa  o de uma solu  o dentro de uma estrutura cromoss mica pode n o ser uma tarefa trivial. A pr pria seq  ncia bin ria proposta por Holland pode ser inadequada para muitos dos problemas de programa  o linear inteira, impondo-se a considera  o de n meros inteiros, por exemplo. Abordaremos esse tema no momento oportuno. Um algoritmo gen tico exige:

- Uma representa  o gen tica das solu  es vi veis do problema combinat rio.
- A determina  o de uma popula  o inicial de cromossomos.
- A defini  o de uma fun  o de avalia  o dos cromossomos.
- A defini  o dos operadores que v o permitir a produ  o de novos indiv duos e seus par metros, tais como: distribui  es de probabilidade da posi  o de quebra do cromossomo, o tipo de muta  o etc.

- A definição de vários parâmetros como: regras de parada, tamanho da população, esquema de garantia da diversidade etc.

Podemos, então, descrever genericamente um algoritmo genético da seguinte forma:



Bons resultados no uso de algoritmos genéticos para a solução do PCV são relatados por diversos autores (Wetzel [1983], Grefenstette *et al.* [1985], Goldberg e Lingle [1985], Sirag e Weisser [1987]). Utilizaremos o grafo da Figura 8.11 para exemplificar a terminologia e procedimentos dessa técnica

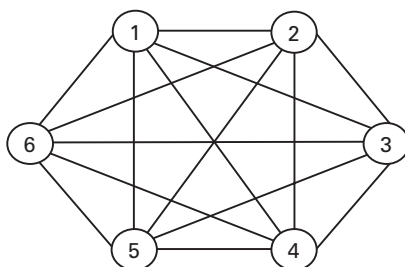


FIGURA 8.11 Grafo de exemplo do PCV.

Para sermos capazes de aplicar a solução via AG ao PCV é indispensável resolvermos as seguintes questões:

- *Obter uma representação genética das soluções viáveis do PCV e associar a essa representação um processo de reprodução.*

Existem várias formas possíveis de representação genética das soluções do PCV. As mais conhecidas são:

1 – Sequência de Inteiros

Nessa forma de representação, o cromossomo é formado pela sequência dos nós no ciclo, ou seja, por uma solução do PCV. No exemplo da Figura 8.12 um cromossomo poderia ser $p_1 = (1, 4, 2, 3, 6, 5)$, que corresponde à solução:

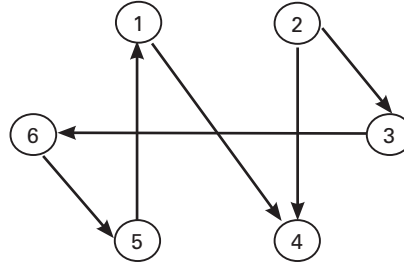


FIGURA 8.12 Uma solução PCV.

Essa representação também é denominada *por caminhos*. É a representação mais natural para um cromossomo, uma vez que o ciclo está diretamente exposto. Grefenstette *et al.* (1985) reporta bons resultados na utilização dessa representação; contudo, uma das dificuldades dela é que o *crossover* pode produzir ciclos inviáveis como por exemplo:

$$\begin{matrix} p_1 = (1, 2, 3, | 4, 5, 6) \\ p_2 = (4, 3, 2, | 5, 1, 6) \end{matrix} \Rightarrow f_1 = (1, 2, 3, | 5, 1, 6) \text{ e } f_2 = (4, 3, 2, | 4, 5, 6)$$

Tendo em vista esse problema, foram desenvolvidos operadores especiais que evitam a produção de filhos inviáveis (geneticamente abortivos), dado que os pais sejam representações de soluções viáveis. Entre eles destacamos:

- **Operador e Partially-Mapped Crossover (PMX)**

O operador PMX propõe o seguinte procedimento: considere um par de cromossomos e sobre eles realize um corte aleatório. Caso sejam gerados dois filhos, f_1 e f_2 , os genes localizados entre os dois cortes de p_1 e p_2 são herdados integralmente, preservando a ordem e a posição de cada um. Por exemplo:

$$\begin{matrix} p_1 = (1, 2, 3, | 4, | 5, 6, 7) \\ p_2 = (4, 3, 2, | 5, | 1, 7, 6) \end{matrix} \Rightarrow f_1 = (1, 2, 3, | 5, | 5, 6, 7) \text{ e } f_2 = (4, 3, 2, | 4, | 1, 7, 6)$$

São ciclos inviáveis. O operador, então, tentará preencher cada gene f_1 em a partir de seu pai p_1 , e cada gene de f_2 , a partir de seu pai p_2 , desde que não formem um ciclo inviável para o PCV. Então:

$$\begin{matrix} p_1 = (1, 2, 3, \text{---}, 5, 6, 7) \text{ e } p_2 = (4, 3, 2, \text{---}, 1, 7, 6) \\ f_1 = (\otimes, \otimes, \otimes, | 5, | \otimes, \otimes, \otimes) \text{ e } f_2 = (\otimes, \otimes, \otimes, | 4, | \otimes, \otimes, \otimes) \end{matrix}$$

Que nos leva a:

$$\begin{matrix} p_1 = (1, 2, 3, \text{---}, 5, 6, 7) \text{ e } p_2 = (4, 3, 2, \text{---}, 1, 7, 6) \\ \downarrow \downarrow \downarrow \quad \times \downarrow \downarrow \quad \times \downarrow \downarrow \quad \downarrow \downarrow \downarrow \\ f_1 = (1, 2, 3, | 5, | \otimes, \otimes, 6, 7) \text{ e } f_2 = (\otimes, 3, 2, | 4, | 1, 7, 6) \end{matrix}$$

Tomamos o primeiro \otimes de f_1 que deveria ser 5, mas não é possível, pois 5 já pertence ao gene que foi herdado de p_2 . Procuramos agora o valor que ocupava a posição de incompatibilidade em f_1 , na quarta posição. O número é 1. Esse número, obviamente, é compatível com o gene 5 herdado, pois está no pai. Infelizmente, 1 já pertence ao cromossomo f_1 . Procuramos novamente em f_1

qual a posição ocupada pelo número 1, o gene incompatível, e procuramos o que lhe corresponde em p_2 . Nesse caso, a posição é preenchida em p_2 pela cidade 4, que agora é compatível com a sequência de f_1 , e chegamos a: $f_1 = (1, 2, 3, | 5, | 4, 6, 7)$.

$$\begin{array}{c}
 p_2 = (4, 3, 2, \text{---}, 1, 7, 6) \\
 \uparrow \quad \swarrow \quad \uparrow \\
 f_1 = (1, 2, 3, | 5, \otimes, 6, 7)
 \end{array}$$

• Operador Order Crossover (OX)

Esse operador constrói um cromossomo filho herdando uma sequência parcial de cidades de um dos cromossomos pai e mantendo a ordem relativa do outro. Considere o exemplo a seguir:

$$\begin{array}{l}
 p_1 = (1, 2, 3, | 4, 5, 6, | 7, 8, 9) \\
 p_2 = (4, 3, 2, | 1, 8, 5, | 6, 7, 9)
 \end{array}
 \Rightarrow
 \begin{array}{l}
 f_2 = (\otimes, \otimes, \otimes, | 4, 5, 6, | \otimes, \otimes, \otimes) \\
 f_1 = (\otimes, \otimes, \otimes, | 1, 8, 5, | \otimes, \otimes, \otimes)
 \end{array}$$

Os filhos f_1 e f_2 herdam as faixas entre os cortes um e dois. Partindo do segundo corte de um pai (por exemplo p_2), removemos dessa lista as cidades contidas entre os dois cortes do outro pai (no caso p_1), ou seja: de 6, 7, 9, 4, 3, 2, 1, 8, 5, vamos remover 4, 5, 6, obtendo a sequência: 7, 9, 3, 2, 1, 8. Essa sequência deve ser indexada em f_1 a partir do seu segundo corte:

$$\begin{array}{ccc}
 2, 1, 8 & & 7, 9, 3 \\
 \downarrow \downarrow \downarrow & & \downarrow \downarrow \downarrow \\
 f_2 = (\otimes, \otimes, \otimes, | 4, 5, 6, | \otimes, \otimes, \otimes)
 \end{array}$$

de 7, 8, 9, 1, 2, 3, 4, 5, 6, vamos remover 1, 8, 5, obtendo a sequência: 7, 9, 2, 3, 4, 6, que, indexada a partir do segundo corte de f_2 , nos dá:

$$\begin{array}{ccc}
 3, 4, 6 & & 7, 9, 2 \\
 \downarrow \downarrow \downarrow & & \downarrow \downarrow \downarrow \\
 f_1 = (\otimes, \otimes, \otimes, | 1, 8, 5, | \otimes, \otimes, \otimes)
 \end{array}$$

• Operador Cycle Crossover (CX)

Trata-se de um operador que preserva a posição absoluta das cidades nos cromossomos dos pais. Sejam os seguintes pais:

$$\begin{array}{l}
 p_1 = (1, 2, 3, 4, 5, 6, 7, 8, 9) \\
 p_2 = (4, 3, 2, 6, 8, 9, 1, 5, 7)
 \end{array}$$

O primeiro filho, f_1 , é obtido tomando-se inicialmente a primeira cidade de p_1 e colocando-a na primeira posição do cromossomo de f_1 :

$$\begin{array}{c}
 p_1 = (1, 2, 3, 4, 5, 6, 7, 8, 9) \\
 \downarrow \\
 f_1 = (1, \otimes, \otimes, \otimes, \otimes, \otimes, \otimes, \otimes, \otimes)
 \end{array}$$

Após definir a posição do último elemento em f_1 , procuramos em p_2 qual a posição correspondente a esse elemento definido:

$$\begin{aligned}
 f_1 &= (1, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset) \\
 &\downarrow \\
 p_2 &= (4, 3, 2, 6, 8, 9, 1, 5, 7)
 \end{aligned}$$

Dessa forma o elemento da primeira posição em p_2 será herdado pelo filho f_1 , contudo preservando a posição em que esse elemento ocupa no cromossomo de seu pai p_1 , ou seja, a quarta posição:

$$\begin{aligned}
 p_1 &= (1, 2, 3, 4, 5, 6, 7, 8, 9) \\
 &\downarrow \quad \downarrow \\
 f_1 &= (1, \emptyset, \emptyset, 4, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)
 \end{aligned}$$

Continuando, vamos encontrar o elemento que ocupa a quarta posição no pai p_2 , que é 6:

$$\begin{aligned}
 f_1 &= (1, \emptyset, \emptyset, 4, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset) \\
 &\downarrow \quad \downarrow \\
 p_2 &= (4, 3, 2, 6, 8, 9, 1, 5, 7)
 \end{aligned}$$

De modo semelhante, 6 será herdado por f_1 na posição que ocupa em p_1 , ou seja, na sexta posição. A Figura 8.13 resume o processo:

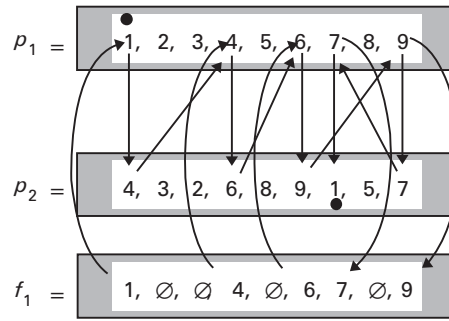


FIGURA 8.13 Primeira fase do CX.

Quando ocorrer que o elemento visitado em p_2 ocupe a primeira posição, e isso acontecerá na sétima posição, diremos que foi completado um ciclo. O ciclo do exemplo foi completado com a inclusão do elemento 7. Quando um ciclo é completado as outras posições são preenchidas do cromossomo de p_2 .

$$\begin{aligned}
 p_2 &= (4, 3, 2, 6, 8, 9, 1, 5, 7) \\
 &\downarrow \downarrow \quad \downarrow \quad \downarrow \\
 f_1 &= (1, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)
 \end{aligned}$$

Resultando em:

$$f_1 = (1, 3, 2, 4, 8, 6, 7, 5, 9)$$

• Operador Edge Recombination Crossover (ERX)

Este operador procura construir filhos maximizando a possibilidade de presença de arestas que sejam comuns a ambos os pais. Essa intenção é justificada pela representatividade genética. Em um processo de reprodução racional, seria razoável esperar que os descendentes herdem de seus pais tantos gens quantos possíveis. Interpretando essa diretiva no contexto do PCV, diríamos que, em um processo de

reprodução que objetivasse preservar a representatividade genética, deveríamos reduzir ao máximo a introdução de arestas aleatórias nos filhos. Para tal foi desenvolvido o operador ERX. Seja o grafo da Figura 8.14:

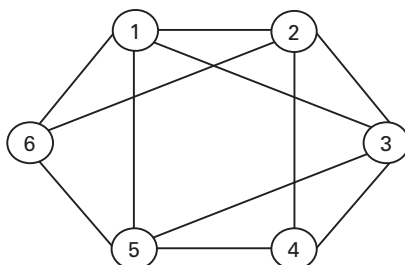


FIGURA 8.14 Grafo para o PCV.

cujas listas de arestas do problema são:

Cidade	Arestas
1	(1, 2), (1, 3), (1, 5), (1, 6)
2	(2, 1), (2, 3), (2, 4), (2, 6)
3	(3, 1), (3, 2), (3, 4), (3, 5)
4	(4, 2), (4, 3), (4, 5)
5	(5, 1), (5, 3), (5, 4), (5, 6)
6	(6, 1), (6, 2), (6, 5)

Sejam os dois cromossomos:

$$p_1 = (1, 3, 2, 4, 5, 6)$$

$$p_2 = (2, 6, 5, 4, 3, 1)$$

que nos levam à seguinte lista de arestas:

Cidade	Arestas
1	(1, 2), (1, 3), (1, 6)
2	(2, 1), (2, 3), (2, 4), (2, 6)
3	(3, 1), (3, 2), (3, 4)
4	(4, 2), (4, 3), (4, 5)
5	(5, 4), (5, 6)
6	(1, 6), (6, 2), (6, 5), (6, 4)

Para utilizar o ERX na versão para a produção de apenas um filho, selecionaremos a cidade inicial de um dos dois cromossomos pais (cidade 1 ou 2) que possuir o menor número de arestas incidentes na lista de arestas dos cromossomos, ou seja, 1 é escolhida. A cidade 1 está ligada diretamente às cidades 2, 3 e 6. Trabalharemos em relação à tabela de arestas dos cromossomos. Escolheremos, dentre as cidades ligadas a 1, a que possuir o menor número de arestas incidentes na tabela de trabalho. Escolheremos a cidade 3. Essa sequência de escolha vai formando o indivíduo da seguinte maneira:

$$f_1 = (1, 3, \emptyset, \emptyset, \emptyset, \emptyset)$$

Aplicando o raciocínio de forma sucessiva incluiremos as cidades 5, 6 e 2, respectivamente, formando-se o seguinte cromossomo filho:

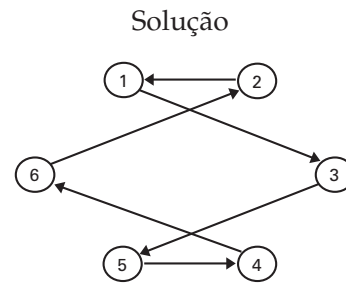
$$f_1 = (1, 3, 4, 5, 6, 2)$$

As experiências computacionais reportam uma certa dificuldade de esse operador reduzir a introdução de arestas aleatórias nos descendentes, especialmente nas últimas arestas do circuito, quando já existe pouca flexibilidade para inclusões.

2 – Lista de Adjacências

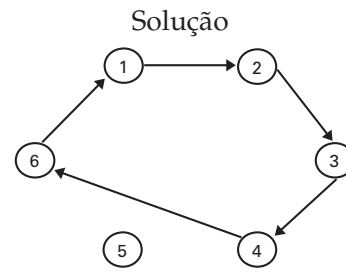
Podemos também representar um cromossomo da seguinte forma: a cidade j do ciclo é listada na posição i se e somente se, o ciclo vai diretamente da cidade i para a cidade j . Assim, um cromossomo $p = (3, 1, 5, 6, 4, 2)$ representa os seguintes arcos em um ciclo associado a p :

Posição	Nó em p	Arco
1	3	$1 \rightarrow 3$
2	1	$2 \rightarrow 1$
3	5	$3 \rightarrow 5$
4	6	$4 \rightarrow 6$
5	4	$5 \rightarrow 4$
6	2	$6 \rightarrow 2$



De modo semelhante à representação por sequência de inteiros, essa representação também pode gerar cromossomos inviáveis, como por exemplo $p = (2, 3, 4, 6, 5, 1)$

Posição	Nó em p	Arco
1	2	$1 \rightarrow 2$
2	3	$2 \rightarrow 3$
3	4	$3 \rightarrow 4$
4	6	$4 \rightarrow 6$
5	5	$5 \rightarrow 5$
6	1	$6 \rightarrow 1$



3 – Lista Ordinal

Em uma lista ordinal, o ciclo (cromossomo) é representado por sua posição em uma lista ordenada de cidades do PCV. A lista l varia de 1 a n cidades. O cromossomo é um vetor de posicionamento da cidade na lista. Seja o problema da Figura 180, então $l = (1, 2, 3, 4, 5, 6)$. Um cromossomo $p = (1, 2, 1, 3, 2, 1)$ representa o seguinte ciclo (que representaremos por $t = (t_1, t_2, \dots, t_n)$):

- $p = (1, 2, 1, 3, 2, 1) p^1 = 1 \Rightarrow$ primeira posição na lista $c = (1, 2, 3, 4, 5, 6)$, ou seja, $t_1 = 1$ e $t = \{1\}$; $t = \{1\}$;
- $p = (1, 2, 1, 3, 2, 1) p^2 = 2 \Rightarrow$ segunda posição na lista atualizada $c = (\otimes, 2, 3, 4, 5, 6)$, onde os valores anulados não são computados, ou seja, $c = (2, 3, 4, 5, 6)$, e $t_2 = 3$, o que leva a $t = \{1, 3\}$
- $p = (1, 2, 1, 3, 2, 1) p^3 = 1 \Rightarrow$ primeira posição na lista atualizada $c = (\otimes, 2, \otimes, 4, 5, 6)$, ou seja, $c = (2, 4, 5, 6)$ e $t_3 = 2$, o que leva a $t = \{1, 3, 2\}$
- $p = (1, 2, 1, 3, 2, 1) p^4 = 3 \Rightarrow$ terceira posição na lista atualizada $c = (\otimes, \otimes, \otimes, 4, 5, 6)$, ou seja, $c = (4, 5, 6)$ e $t_4 = 6$ o que leva a $t = \{1, 3, 2, 6\}$

5. $p = (1, 2, 1, 3, 2, 1)$ $p^5 = 2 \Rightarrow$ segunda posição na lista atualizada $c = (\otimes, \otimes, \otimes, 4, 5, \otimes)$, ou seja, $c = (4, 5)$ e $t_5 = 5$ e $t = \{1, 3, 2, 6, 5\}$.

6. $p^6 = 1 \Rightarrow$ primeira posição na lista $c = (\otimes, \otimes, \otimes, 4, \otimes, \otimes)$, ou seja, $t = \{1, 3, 2, 6, 5, 4\}$

A representação ordinal é perfeitamente consistente aos operadores clássicos de cruzamento. Acompanhemos o seguinte cruzamento utilizando a representação ordinal:

$$c = (1, 2, 3, 4, 5, 6)$$

$$\begin{matrix} p_1 = (2, 1, 3, \mid 1, 2, 1) \\ p_2 = (1, 2, 2, \mid 1, 1, 1) \end{matrix} \Rightarrow p_3 = (2, 1, 3, \mid 1, 1, 1) \text{ e } p_4 = (1, 2, 2 \mid 1, 2, 1)$$

p_1 responde ao ciclo $\{2, 1, 5, 3, 6, 4\}$, ao ciclo $\{1, 3, 4, 2, 5, 6\}$, o filho ao ciclo $\{2, 1, 5, 3, 4, 6\}$ e p_4 ao ciclo $\{1, 3, 4, 2, 6, 5\}$.

• Determinar como obteremos a população inicial de cromossomo

Os cromossomos são soluções do PCV. Para o caso do grafo completo, qualquer permutação dos nós é uma solução viável. Existem várias heurísticas para a obtenção de soluções aproximativas. As heurísticas para o problema de roteamento normalmente podem ser aplicadas para a obtenção de soluções viáveis para o PCV. Como o desenvolvimento dos AG está fortemente associado à herança genética, a população inicial certamente vai influenciar o desempenho dos algoritmos que serão elaborados. Esse é um tema de grande importância.

• Definir a função de avaliação dos cromossomos e os critérios de renovação da população

Os custos totais são um indicador natural para a avaliação da qualidade de um cromossomo (uma solução). É possível obter também limites para o valor ótimo do PCV através da solução de outras estruturas no grafo de substrato como o *matching* mínimo, a árvore geradora mínima etc.

• Solução do PCV Através de Algoritmos Heurísticos Construtivos

Grande parte dos algoritmos heurísticos construtivos para o PCV utiliza a estratégia denominada *gulosa* ou *míope* de solução. Essa estratégia, em última análise, desenvolve, a cada passo das heurísticas construtivas, a ideia de obter o maior proveito ou ganho possível. Evidentemente, a política global de tomada de decisão pode não comportar sempre ações que levem aos maiores ganhos imediatos. Algumas estruturas matemáticas possuem a propriedade de aceitar uma forma de tomada de decisão míope. Tais estruturas denominam-se *matroide* (Edmonds [1971] e anexo). Examinaremos agora algumas heurísticas construtivas míopes para o PCV.

• Heurística de Bellmore e Nemhauser

A heurística consiste em, partindo de um vértice qualquer do grafo de substrato do PCV, estruturar o caminho hamiltoniano dirigindo-se, a cada passo, ao vértice mais próximo do vértice anterior (Bellmore e Nemhauser [1968]). A complexidade dessa heurística é $O(n^2)$. Uma variação repete o algoritmo para todos os nós ou possíveis cidades para minimizar o efeito da influência da escolha da cidade inicial. Nesse caso, a heurística passa a ter complexidade $O(n^3)$. A estratégia gulosa é caracterizada pelo fato de, a cada passo, o algoritmo tomar a decisão de menor custo (nó mais próximo ao nó da extremidade). Uma variante dessa heurística permite que a inclusão possa ocorrer em ambos os nós extremos do ciclo parcial que se forma (ver Figura 8.14). De uma forma geral, a heurística pode ser escrita como:

Heurística de Bellmore e Nemhauser
INÍCIO
Escolher um vértice inicial v_1 ;
Encontrar o vértice v_k mais próximo do último vértice (dos vértices extremos) incluído(s) na solução;
Inserir o vértice v_k após o último vértice (o seu vizinho mais próximo);
Caso o ciclo formado seja Hamiltoniano pare .
Caso contrário volte à etapa 2 .
FIM

Aplicamos o algoritmo de Bellmore e Nemhauser ao grafo da Figura 8.15:

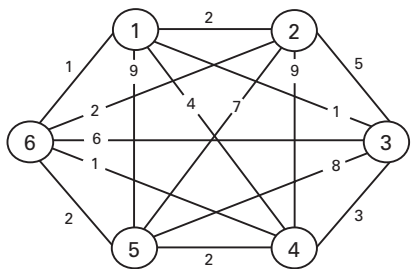


FIGURA 8.15 Grafo exemplo.

Iniciando pelo vértice 6 e aplicando a técnica do *vizinho mais próximo* temos a sequência de inserções da Figura 8.16.

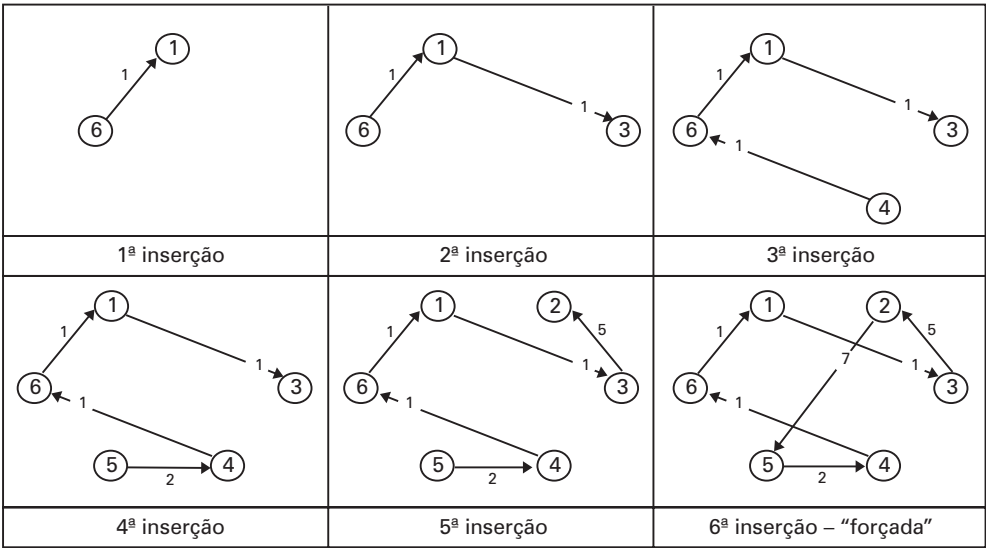


FIGURA 8.16 Exemplo de aplicação da heurística do vizinho mais próximo.

Que nos conduz a um valor final de 17.

• Heurísticas de Inserção

Algumas heurísticas construtivas possuem um processo de decisão mais elaborado do que as simplesmente gulosas. Nesse caso, três níveis de decisão são envolvidos:

- A escolha do vértice a ser inserido na solução.
- A posição de inserção desse novo vértice.
- A decisão de um ciclo inicial.

Essas heurísticas são chamadas de *inserção*. Seu processo de decisão é ainda basicamente míope porém mais elaborado. Normalmente, tais heurísticas partem de um *subtour* inicial (um ciclo normalmente de comprimento 3) e vão selecionando e inserindo vértices ainda não incluídos na solução até completar um ciclo. Alguns dos critérios mais utilizados para a seleção dos vértices a serem acrescentados ao *subtour* são:

- Inserção do vértice mais próximo.
- Inserção do vértice mais distante.
- Inserção do vértice que conduz ao ciclo mais barato (inserção mais barata).
- A inserção aleatória.

No caso das inserções com os vértices mais próximos e mais distantes de qualquer vértice do *subtour*, escolhido o vértice, cabe decidir como ele será inserido no *subtour*. Como o vértice ainda pertence ao *subtour* serão necessárias duas arestas para incluí-lo. Se a inclusão desse vértice ocorrer entre os nós i e $i + 1$, será necessário remover essa aresta para preservar o *subtour*. O critério de decisão sobre quais arestas devemos empregar nesse processo é expresso pelo efeito do balanço entrada \times saída de arestas. Considerando uma inserção entre os vértices i e $i + 1$, o critério para a escolha do ponto de inserção se confunde com o critério de escolha da aresta $(i, i + 1)$ que minimize o seguinte balanço:

$$\text{Minimizar } \{c_{ik} + c_{ki+1} - c_{i,i+1}\}$$

No caso da inserção mais barata, o próprio vértice passa a ser escolhido pelo critério de mínimo balanço, considerando-se todas as inserções possíveis para todos os vértices ainda não pertencentes ao *subtour*.

De uma forma geral, as heurísticas de inserção podem ser descritas pelos seguintes passos:

Heurística de Inserção
<p>INÍCIO</p> <p>Iniciar por um ciclo de vértices $\{v_1 v_2 v_3\}$;</p> <p>Encontrar o vértice v_k não pertencente ao ciclo, mais próximo/mais distante de qualquer dos vértices pertencentes ao ciclo;</p> <p>Encontrar a aresta $(i, i + 1)$ tal que Minimize $\{c_{ik} + c_{ki+1} - c_{i,i+1}\}$</p> <p>Inserir o vértice v_k entre os vértices i e $i + 1$.</p> <p>Caso o ciclo formado seja hamiltoniano então pare.</p> <p>Caso contrário volte à etapa 2.</p> <p>FIM</p>

A inserção mais barata executa os passos 2 e 3 em um só procedimento da seguinte forma:

Encontrar v_k não pertencente ao ciclo tal que sua inserção entre i e $i+1$ minimize $\{c_{ik} + c_{ki+1} - c_{ii+1}\}$

Vamos aplicar as heurísticas de inserção do vértice mais próximo (conforme Figura 8.17) e inserção mais barata (conforme Figura 8.18) ao grafo da Figura 8.15. Apesar de referirem-se ao mesmo grafo, os exemplos serão iniciados com ciclos diferentes.

O resultado da aplicação da heurística de inserção do vértice mais próximo ao grafo da Figura 8.15 é um circuito com 15 unidades de comprimento.

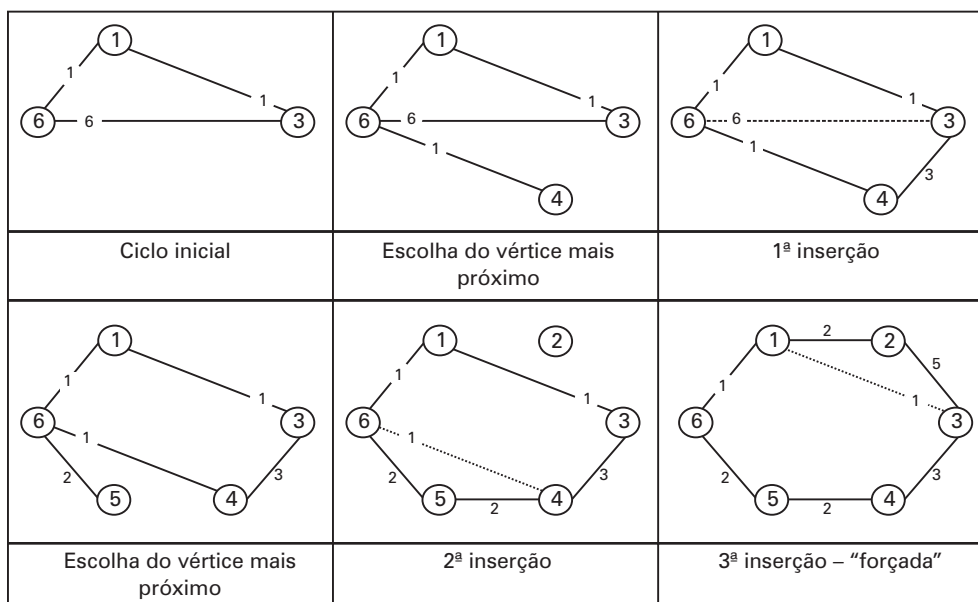


FIGURA 8.17 Inserção do vértice mais próximo.

O resultado da aplicação da heurística de inserção do vértice mais barato ao grafo da Figura 8.15 é um circuito com 15 unidades de comprimento; contudo, destacamos o fato de que, nessa última heurística, a última inserção não foi completamente forçada. O efeito final da política míope de tomada de

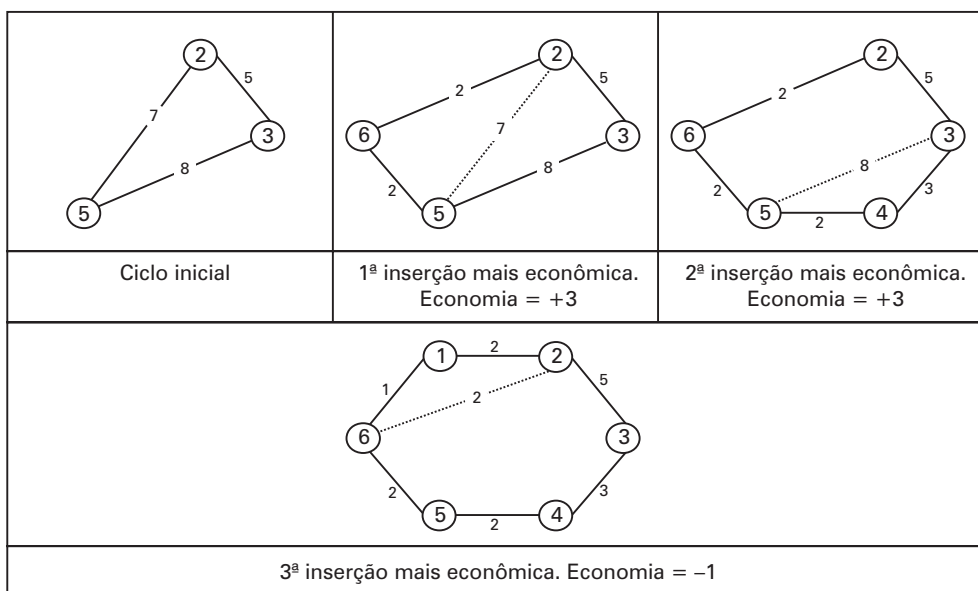


FIGURA 8.18 Inserção mais barata.

decisão das heurísticas de inserção do vértice mais próximo ou mais distante, é diminuída na inserção mais barata.

Frendewey (ver Arthur [1959]) relata bons resultados computacionais no uso dessas heurísticas para PCV de um modo geral. Para o caso do PCV simétrico euclidiano constatou-se superioridade na qualidade de solução para a inserção mais distante.

• Heurísticas de K-Substituições ou k -Opt

As heurísticas de substituição são estratégias de melhoria e partem de um ciclo hamiltoniano inicial. Um dos exemplos clássicos dessas heurísticas encontra-se no trabalho de Lin e Kernighan (Lin e Kernighan [1973]). Existem relatos de que elas têm bom desempenho computacional, especialmente as 2-Opt e 3-Opt (ver Golden *et al.* [1980]) e fazem hoje parte de inúmeros outros algoritmos (Gerdessen [1996]) dentre os quais destacamos:

■ Heurística Dynasearch

Algoritmo proposto por Potts e Van de Velde (1995) combina uma série de movimentos 2-Opt a cada passo do algoritmo para compor uma nova solução. A composição dos movimentos 2-Opt é limitada em $O(n^2)$.

■ Geni and Genius

Gendreau *et al.* (1992) apresentam um procedimento híbrido para construção dos ciclos auxiliado por um processo de otimização local baseado no exame de configurações 2-Opt a 4-Opt. Considerando-se uma certa ordenação de visitas, $c_1-c_2-...-c_n$, o algoritmo pode partir de uma parte do ciclo, $c_1-c_2-c_3$, por exemplo. Cada nova cidade que for sendo acrescentada ao ciclo em construção (c_4 , no exemplo) é testada em seu ponto de inclusão de modo que se examine a possibilidade de uma k -Opt substituição, com $4 \geq k \geq 2$. As possíveis trocas são restritas a uma certa distância (p) da posição original da cidade. Os autores apresentam o resultado de testes computacionais que comparam diversos esquemas para os parâmetros de otimização (k e p).

De uma forma geral as heurísticas k -substituições podem ser descritas pelos seguintes passos:

Heurísticas k -Substituições
<p>INÍCIO</p> <p>Iniciar por um ciclo hamiltoniano $\{v_1, v_2, ..., v_n\}$;</p> <p>Remover k vértices do ciclo corrente, tornando-o incompleto;</p> <p>Construir todas as soluções viáveis que contenham o ciclo anterior;</p> <p>Escolher a melhor solução entre as encontradas;</p> <p>Testar as condições de parada (número de iterações, elementos em subconjunto de controle, limites para o valor da solução etc.), prosseguindo ou não em nova iteração.</p> <p>FIM</p>

A heurística L&K pode ser assim formalizada:

• Keurística Lin-Kernighan (1973)

1. Construir um ciclo hamiltoniano e calcular seu custo c .
2. Atribuir melhor custo $c_{\text{melhor}} \leftarrow c$.
3. Remover uma aresta do ciclo criando um caminho hamiltoniano, H , por um critério a ser definido (por exemplo, a maior aresta do ciclo).
4. Criar duas listas vazias: arestas que saem, L_{Sai} , e arestas que entram, L_{Entra} .
5. Incluir a última aresta removida em L_{Sai} .
6. Selecionar uma aresta, e , entre algum vértice terminal e um vértice v interno de H , tal que $e \notin L_{\text{Sai}}$ e $e \notin L_{\text{Entra}}$.
Caso tal aresta inexista, ir para o passo 12.
Caso contrário incluir a aresta e em H e L_{Entra} , formando-se um ciclo em H .
7. Calcular a função de ganho, g , pelo somatório dos pesos das arestas em L_{Sai} menos o peso das arestas em L_{Entra} .
8. Remover a aresta a adjacente a v tal que $a \notin L_{\text{Entra}}$ e a faz parte do ciclo recém-formado. Se não for possível encontrar aresta a que tende a exigência e ir para o passo 12.
9. Calcular o custo, c_{ind} , do ciclo hamiltoniano induzido por H mais a aresta que une os vértices terminais.
10. Se $c_{\text{ind}} < c_{\text{melhor}}$, então $c_{\text{melhor}} \leftarrow c_{\text{ind}}$ e salvar o ciclo hamiltoniano correspondente a c_{ind} .
11. Se $g > 0$, voltar ao passo 5.
12. Se $c_{\text{melhor}} < c$, voltar ao passo 3, considerando o ciclo hamiltoniano correspondente a c_{melhor} como a nova solução inicial. Caso contrário, FIM.

Passo 1. No grafo da figura 8.11, seja a seguinte solução inicial: (1 4 2 3 6 5) $c = 35$

Passo 2. $c_{\text{melhor}} = 35$

Passo 3. Remover aresta (4, 2) conforme figura 8.19(a)

Passo 4. $L_{\text{Sai}} = \{\}$, $L_{\text{Entra}} = \{\}$

Passo 5. $L_{\text{Sai}} = \{(4, 2)\}$

Passo 6. Incluir aresta (4, 6) conforme figura 8.19(b)
O ciclo (4, 6, 5, 1) é formado. $L_{\text{Entra}} = \{(4, 6)\}$

Passo 7. $g = \text{peso de } (4, 2) - \text{peso de } (4, 6) = 9 - 1 = 8$

Passo 8. Remover a aresta (6, 5) conforme figura 8.19(c) (o caminho é restaurado)

Passo 9. $c_{\text{ind}} = 32$, o custo do ciclo hamiltoniano (5 1 4 6 3 2)

Passo 10. $c_{\text{melhor}} = 32$ e guarda (5 1 4 6 3 2)

Passo 11. ($g > 0$) voltar ao passo 5

Passo 5. $L_{\text{Sai}} = \{(4, 2), (6, 5)\}$

Passo 6. Incluir aresta (4, 5) conforme figura 8.20(a)
 $L_{\text{Entra}} = \{(4, 6), (4, 5)\}$

Passo 7. $g = (9 + 2) - (1 + 2) = 8$

Passo 8. Remover (1, 4) conforme figura 8.20(b)

Passo 9. $C_{\text{ind}} = 25$, o custo do ciclo hamiltoniano (1, 5, 4, 6, 3, 2)

Passo 10. $C_{\text{melhor}} = 25$, guarda (1, 5, 4, 6, 3, 2)

Passo 11. Voltar ao passo 5

Passo 5. $L_{\text{Sai}} = \{(4, 2), (6, 5), (1, 4)\}$

Passo 6. Incluir aresta (1, 6) conforme figura 8.20(c)
 $L_{\text{Entra}} = \{(4, 6), (4, 5), (1, 6)\}$

Passo 7. $9 = (9 + 2 + 4) - (1 + 2 + 1) = 11$

Passo 8. Remover a aresta (4, 6), porém $(4, 6) \in L_{\text{Entra}}$
Portanto, IR para o passo 12

Passo 12. $c_{\text{melhor}} = 25$ e $c = 35$

Portanto, voltar ao passo 3, considerando o ciclo (1, 5, 4, 3, 2) como a nova solução inicial

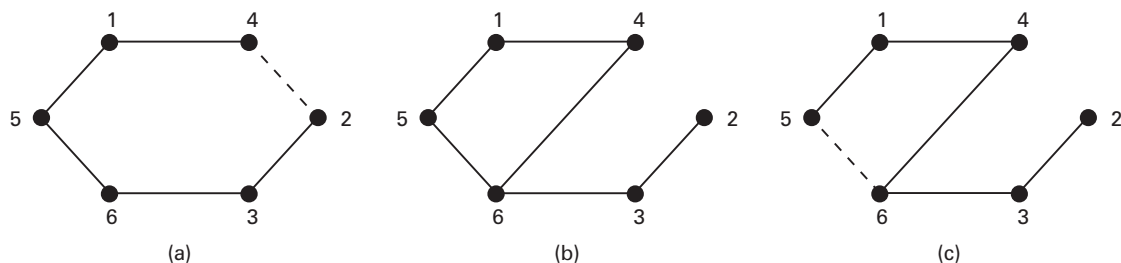


FIGURA 8.19 Primeira parte da execução do algoritmo Lin-Kernighan.

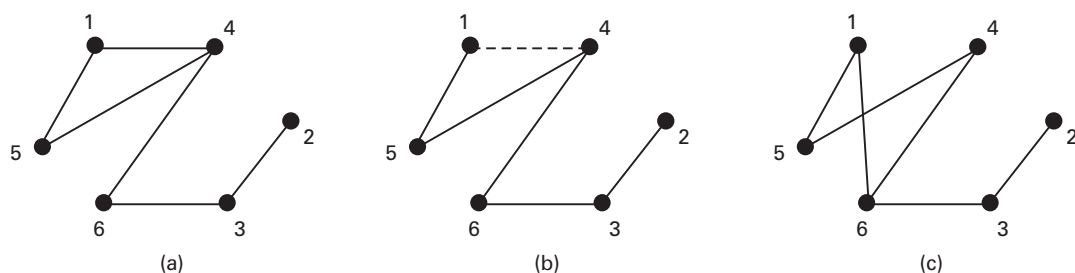


FIGURA 8.20 Segunda parte da execução do algoritmo de Lin-Kernighan.

• Heurística de Saving ou das Economias

Trata-se da aplicação ao PCV de uma estratégia originalmente sugerida para o problema de roteamento (Clark e Wright [1964]). Sua aplicação eficiente sugere a necessidade de um grafo completo, concluindo-se que é uma abordagem bastante razoável para o PCV euclidiano e simétrico. O algoritmo pode ser descrito da seguinte forma:

Heurística das Economias
<p>INÍCIO</p> <p>Iniciar pelo vértice k, selecionado por algum critério ou aleatoriamente;</p> <p>Considerar todos os vértices ligados ao vértice k, ou seja, um circuito não hamiltoniano que passa n vezes pelo nó k.</p> <p>Obter a lista das economias da seguinte forma: $S_{ij} = c_{jk} - c_{ij}$, $i, j = 1, \dots, n$, onde S é a economia feita se o vértice i for ligado diretamente ao vértice j sem passar por k.</p> <p>Ordenar as economias em lista monótona decrescente;</p> <p>Percorrer a lista iniciando pela primeira posição. Tentar a ligação correspondente ao maior S_{ij};</p> <p>Se a inserção da aresta (i, j) e a retirada da aresta (k, i) e (j, k) resultar em uma rota iniciando em k e passando pelos demais vértices, eliminar da lista.</p> <p>Caso contrário, tentar a ligação seguinte na lista.</p> <p>Repetir o procedimento até obter o ciclo hamiltoniano.</p> <p>FIM</p>

A Figura 8.21 exemplifica o uso da heurística das economias no grafo da Figura 8.15.

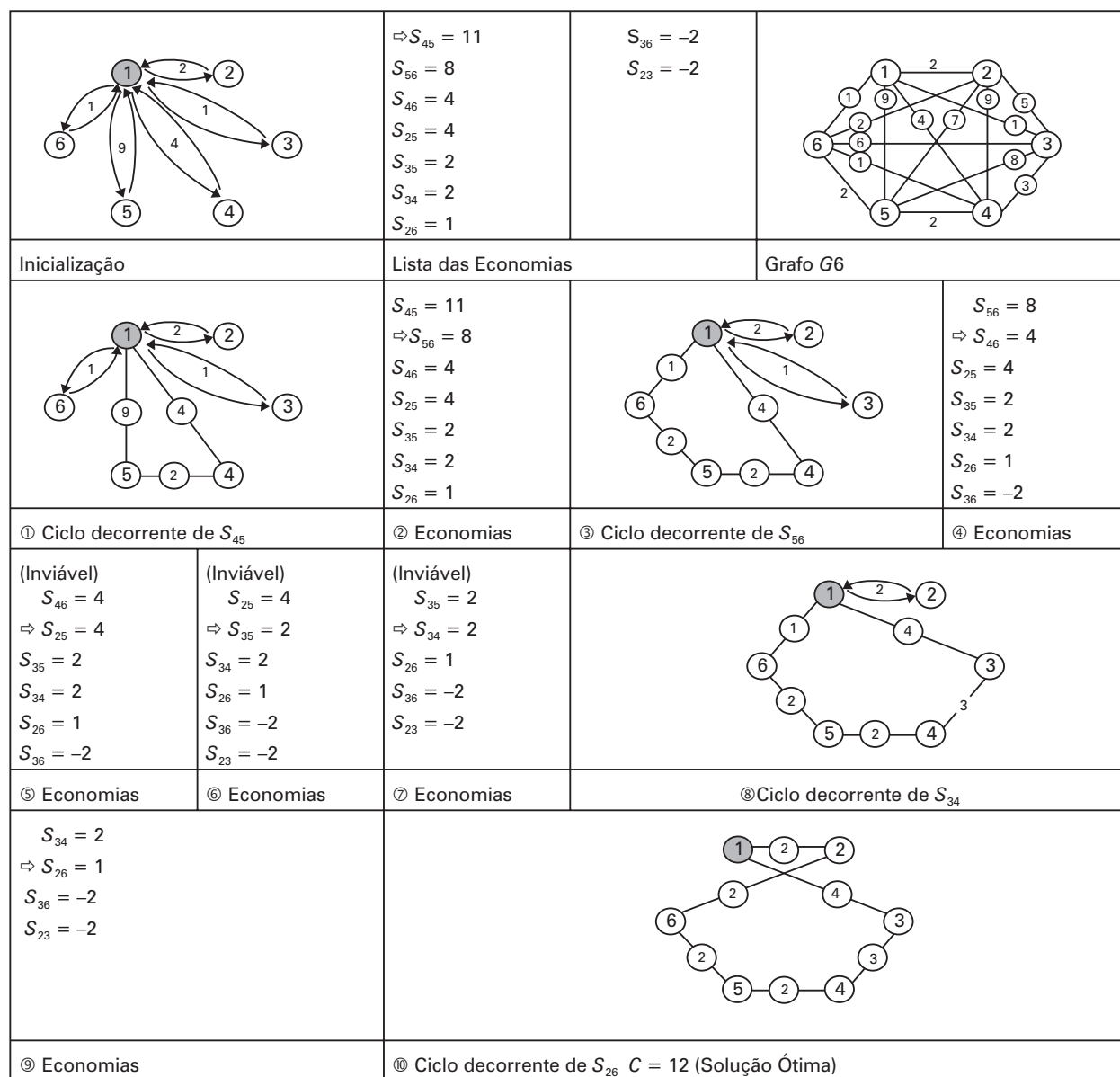


FIGURA 8.21 Etapas da heurística das economias.

8.5 – PROBLEMAS PROPOSTOS

1 – Soluções para o PCV clássico



Seja o PCV simétrico determinado no grafo $G = (N, A)$, em que $N = \{1, 2, \dots, 16\}$ e a matriz de distância entre os nós está transcrita na Tabela 8.6.

TABELA 8.6 DISTÂNCIAS ENTRE OS VÉRTICES DO PROBLEMA 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1		3	5	14	7	11						6			3	
2						3	2		4	9			1			18
3					2	1		3			8	1		2		
4									2	3			8	15		
5							12	1		6	7		7			1
6								5					6			12
7									1		2			11	1	
8										7		13			6	17
9												5		1		
10												1				3
11														5	9	4
12														9		
13																
14															2	3
15																4
16																

✓ Elaborar algoritmos de solução para o problema utilizando as seguintes abordagens:

1. Heurística de Bellmore e Nemhauser.
2. Heurísticas de 2-substituições ou 2-ótima.
3. Heurísticas de economias.

✓ Implementar os algoritmos e comparar os resultados para o problema em pauta.

2 – Soluções para um caso do PCV backhuals (PCVB)



Considerando o mesmo grafo do problema 1 com $B = \{1, 2, 3, 15, 16\}$ e $L = \{4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$, e não levando em conta a exigência da norma euclidiana:

✓ Modificar os algoritmos anteriores de modo que possam solucionar este novo caso.

✓ Comparar o desempenho dos métodos aplicados ao atual problema com o resultado obtido no problema 1, verificando o impacto, para a abordagem computacional, da inclusão da restrição de visita a vértices (melhora, piora, indiferente ao desempenho e qualidade de solução).

3 – Soluções para um caso do PCV com janela de tempo



No grafo do problema 1 os tempos de trajeto entre os vértices estão transcritos na Tabela 8.7:

TABELA 8.7 TEMPOS ENTRE OS VÉRTICES DO PROBLEMA 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1		1	2	3	1	4						6			4	
2						4	4		1	1			1			1
3					1	6		1			6	6		2		
4									1	1			3	2		
5							1	2		1	6		3			4
6								1					3			6
7									1		6			2	1	
8										3		1			8	1
9												1		2		
10												1				2
11														8	8	2
12														8		
13																
14															3	1
15																6
16																

Sabe-se que, a cada conjunto de quatro nós visitados, o tempo de viagem acumulado é anulado. Sabe-se ainda que nenhum ciclo pode acumular mais de 15 unidades de tempo em sua trajetória, ressalvada a regra anterior.

- ✓ Modificar os algoritmos anteriores de modo que possam solucionar este novo caso.
- ✓ Comparar o desempenho dos métodos aplicados ao atual problema com o resultado obtido no problema 1, verificando o impacto, para a abordagem computacional, da inclusão da restrição de tempo (melhora, piora, indiferente ao desempenho e qualidade de solução).

9

PROBLEMAS DE ROTEAMENTO

9.1 – INTRODUÇÃO

9.1.1 – A Importância do Problema e seu Contexto

Um Sistema de Roteamento pode ser considerado como um conjunto organizado de meios que objetiva o atendimento de demandas localizadas nos arcos ou nos vértices de alguma rede de transportes. O sistema de roteamento, como de resto qualquer sistema operacional, pode ser decomposto, sob a ótica da operação, em três partes a saber:

- ✓ Estratégica.
- ✓ Tática.
- ✓ Logística.

O objetivo maior da logística é fazer chegar provisões e/ou serviços a pontos de consumo, a partir de pontos de suprimento (Bodin [1983]). Um sistema logístico completo deve incluir cuidados que se vão desde o processo de obtenção, estoque e distribuição de produtos sobre uma rede de demanda, até os relacionados com os seres humanos, política de investimento e renovação de frota etc. Um sistema dessa natureza é complexo e composto de importantes subsistemas fortemente acoplados. Apesar da verdadeira compreensão do sistema exigir uma visão globalizada, é conveniente analisá-lo através de subsistemas com objetivo de reduzir a complexidade dos modelos de solução. Via de regra os macroaspectos do sistema são decididos pelo empreendedor, independentemente de um modelo formal. Por exemplo, é comum que as decisões relativas à escolha de locais para instalação de depósitos ou fábricas sejam tomadas levando em conta argumentos derivados da estratégia do negócio e de oportunidade (possíveis isenções tarifárias, custo e tamanho dos terrenos, disponibilidade de financiamento, segurança etc.) e antecedam ao modelo de distribuição dos itens fabricados. Dentre outras, as decisões estratégicas envolvem os tópicos da Figura 9.1.

As decisões estratégicas normalmente possuem impacto sobre todo o sistema e um efeito duradouro. É interessante ressaltar que decisões estratégicas errôneas produzirão sérias dificuldades para a futura operação e otimização do sistema. Posto o arcabouço geral, caberá ao nível tático do sistema tomar as decisões que definirão as áreas atendidas, o dimensionamento da frota, o sistema de alocação de tripulação aos veículos e os turnos de trabalho. Circunstancialmente o número de garagens e sua localização, a sistemática de manutenção, bem como as decisões sobre a estocagem podem ainda ficar a cargo do nível tático.

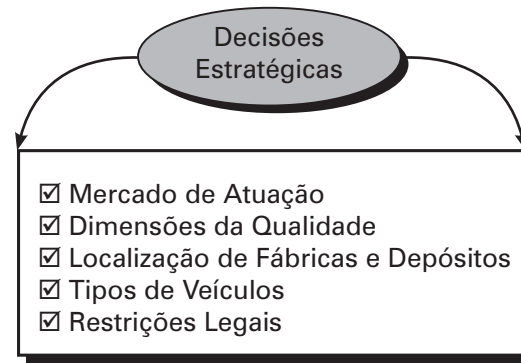


FIGURA 9.1 *Decisões estratégicas para o problema de roteamento.*



FIGURA 9.2 *Decisões táticas para o problema de roteamento.*

Normalmente as decisões que dirão respeito à operação da frota de veículos e o emprego da mão de obra serão deixadas a cargo do nível operacional. A principal preocupação do nível operacional será providenciar um plano econômico e flexível para o atendimento da rede de demanda. É exatamente dentro do nível operacional que os tradicionais Problemas de Roteamento de Veículos (PRV) serão definidos. Contudo, à medida que o estado da arte avança, os problemas do nível tático e estratégico tendem a se incorporar aos PRV tradicionais enriquecendo os modelos e pressionado por algoritmos mais eficientes. Como elementos de entrada mais prováveis dos PRV tradicionais destacam-se:

- ✓ Os clientes servidos pelos depósitos.
- ✓ A demanda dos clientes.
- ✓ O tamanho da frota alocada a cada depósito.
- ✓ O tipo de veículo empregado (capacidade, velocidade, habilitação de transporte etc.).
- ✓ As regras que definirão o carregamento/descarregamento.
- ✓ A possibilidade de uso de equipamento de aluguel.
- ✓ A capacidade das ligações, seu custo operacional, disponibilidade etc.

De posse dessas informações o objetivo do planejamento será estabelecer o roteamento de veículos e o sequenciamento (“scheduling”) de atividades que conduzam à minimização do custo da atividade.

Nesse ponto é pertinente ressaltar que o próprio estabelecimento de uma função objetivo apropriada não é uma tarefa trivial. O objetivo de reduzir custos poderá ser perseguido através da redução de:

- ✓ Prazos de entrega (serviços de emergência, produtos perecíveis etc.).
- ✓ Caminhos a percorrer (combustível, manutenção, tempo de operação etc.).
- ✓ Alocação de mão de obra.
- ✓ Riscos de acidentes ou avarias.
- ✓ Número de veículos.

Ou ainda, redesenhando:

- ✓ Intervalos de trabalho (evitando engarrafamentos, taxa de estacionamento etc.).
- ✓ O carregamento (otimizando a relação carga/rota/meio de transporte).
- ✓ A alocação meio de transporte \times tarefa (otimizando a relação carga/rota/meio de transporte).
- ✓ A política para o atendimento da demanda dos clientes. Observe que a demanda pode ser uma variável de natureza estocástica. O “valor de um cliente” pode ser uma variável composta que inclui desde a perda do cliente e o consequente avanço da concorrência, até a dificuldade de sua futura recuperação.
- ✓ A política de controle de estoques e investimento em instalações e meios diversos (compra ou aluguel de veículos, esquema de manutenção e disponibilidade da frota etc.).

Tais elementos intervenientes sugerem que os Problemas de Roteamento de Veículos mais complexos são tipicamente problemas multiobjetivo. Convém ainda ressaltar a importância, dentro do Problema Geral de Roteamento, do subproblema de distribuição. Essa importância é devida, dentre outros fatores, ao fato de que a distribuição engloba elevados custos. Hollander (1987) mostra que um atraso na entrega de produtos no comércio internacional contribui para o custo final do produto com um acréscimo de 5% por mês de atraso. Essa observação de Hollander é pertinente, pois um ciclo em um roteamento marítimo pode envolver dois ou mais meses. Bodin (1983) mostra que a distribuição física dos produtos contribui com cerca de 16% do custo final do item. Por outro lado, certos produtos carecem de uma distribuição eficiente por motivos não só econômicos como de segurança. Exemplos do caso anteriormente exposto são os medicamentos e os combustíveis.

Os diversos problemas aninhados sob o tema de Roteamento de Veículos têm recebido a atenção de muitos pesquisadores nos últimos 30 anos. Apenas nos últimos 15 anos o problema pôde ser solucionado, por algoritmos exatos, para instâncias da ordem de 30 pontos de demanda (ou nós na rede) conforme Bodin (1983). Com a melhoria do desempenho da mídia computacional e com o surgimento de novos modelos e abordagens, vários relatos de bons resultados constam da literatura, especialmente para casos específicos (Desrosiers [1984]; Laporte [1986] e [1987]; Solomon [1987]; Desrosiers *et al.* [1992]; Bianco *et al.* [1994]; Dumas *et al.* [1995]; Fisher *et al.* [1997]; Fischetti [1997]; Laporte *et al.* [2000]).

9.1.2 – Principais Aplicações Práticas do PRV

Esse problema possui um número extraordinário de aplicações práticas, pois implica tipicamente uma série de situações reais que afetam principalmente a indústria, o comércio, o setor de serviços, a segurança, a saúde pública e o lazer. Dentre outras destacam-se:

- Distribuição de jornais (Golden e Magnanti [1977]; Dillmann *et al.* [1996]).
- Distribuição de manufaturados (Perl e Daskin [1985]).
- Distribuição de produtos diversos (Dantzig e Ramser [1959], Klots *et al.* [1992]).

- Distribuição de bebidas (Golden e Wasil [1987]; Eibl *et al.* [1994]).
- Distribuição de valores (Lambert *et al.* [1993]).
- Distribuição de produtos químicos (Ball *et al.* [1983]).
- Transporte escolar (Newton e Thomas [1974], Li e Fu [2002]).
- Recolhimento de lixo (Beltrami e Bodin [1974], Kulcar [1996], Angelelli e Speranza [2002]).
- Entrega de correspondência (Frederickson [1978]).
- Leitura de medidores elétricos (Stern e Dror [1978]).
- Distribuição de pão (Derigs e Grabenbauer [1993]).
- Recolhimento de borracha (Nambiar *et al.* [1989]).
- Roteamento de helicópteros (Timlin e Pulleyblank [1990]).
- Roteamento em linhas aéreas (Marsten e Shepardson [1981] e Marsten *et al.* [1979], Yan e Tu [1997]).
- Sistemas de transportes coletivos urbanos (táxi , ônibus, trens de metrô) (Ceder e Stern [1981]).
- Serviços de emergência (Daskin [1987]).
- Distribuição de derivados de petróleo (Ronen *et al.* [1987], Brown e Graves [1981]).
- Programação de sondas de produção (Goldbarg, *et al.* [2002]).
- Distribuição de gás (Bell *et al.* [1983]).
- Roteamento entre células de manufatura flexível (Finke e Kusiak [1985]).
- Transporte de pedras (Schneider (1985)).
- Entrega de correspondência bancária (Malmborg e Simons [1989]).
- Distribuição de alimentos (Cassidy e Bennett (1972), Bartholdi *et al.* [1983]).
- Sistemas de proteção contra incêndio (Marianov e ReVelle [1992]).
- Distribuição de material fotográfico (Solot *et al.* [1990]).
- Patrulhamento policial e de segurança (Larson [1972], Calvo e Cordone [2003]).
- Roteamento de navios de longo curso (petroleiros), cabotagem e logística (Brown *et al.* [1987], Ronen [2002]).
- Distribuição de vagões ferroviários (Haghani [1991]).
- Roteamento de auditores bancários (Castellano e Bornstein [1989]).
- Operação de veículos de limpeza de gelo nas ruas e estradas (Eglese [1992]).
- Limpeza de ruas com veículos vassoura (Eglese [1991]).
- Manutenção de elevadores (Blakeley [2003]).
- Distribuição e recolhimento de leite (Basnet *et al.* [1993], Sankaran e Ubgade [1994]).
- Programação de sistemas rollon-rollof (Bodin *et al.* [2000]).
- Montagem de fragmentos de ADN (Pevzner *et al.* [2001]).

- Movimentação de plotter laser (Ghiani e Improta [2001]).
- Gerência de containers (Nishimura *et al.* [2004]).
- Entrega de pizza, fast food, comida, congelados etc. (Golden *et al.* [2001], Tarantilis e Kiranoudis [2002]).
- Manutenção de boias marítimas (Cline *et al.* [1992]).
- Planejamento de transporte de carros por caminhões (Pape [1988]).
- Controle de pragas (Solomon *et al.* [1992]).
- Transporte de açúcar em granel (van Vliet *et al.* [1992]).
- Cultivo de ostras (Wang *et al.* [1996]).
- Roteamento de satélites (Lee *et al.* [2003]).
- Just-in-time (Vaidyanathan *et al.* [1999]).
- Recolhimento de cana-de-açúcar (Cruz [1998]).
- Exploração de poços de petróleo não surgentes (Goldbarg *et al.* [2004]).
- *Rolling batch planning* (Chen *et al.* [1998]).
- Corte de chapas metálicas (Manber e Israni [1984]).
- Recolhimento de sobras de madeira (Alves e Carvalho [2001]).
- Distribuição urbana de concreto.
- Projeto de anéis em redes de telecomunicações.
- Roteamento de robôs em manufatura.
- Transporte de bens pessoais (mudanças).
- Roteamento de pacotes em redes de computadores.
- Roteamento de fluxos de comunicações em redes de telecomunicações.

9.2 – PROBLEMAS DE ROTEAMENTO DE VEÍCULOS

9.2.1 Taxonomias

Os problemas de roteamento de veículos abordam basicamente a determinação de sequências de visitas que objetivem atender a uma determinada função objetivo. As visitas podem tanto estar associadas às ligações (arestas) ou aos pontos de visitas (nós) do grafo que representa as possíveis conexões entre os pontos de visita (ou pontos de ligações entre as arestas).

Os problemas de roteamento de veículos estão entre os mais complexos da área de otimização combinatória. Pelo grande número de variáveis, diversidade de restrições e objetivos apresentados impõem-se o exame de uma cuidadosa taxonomia para seu melhor entendimento. Uma das taxonomias clássicas é a proposta por Bodin e Golden (1981). Segundo os autores o Problema de Roteamento de Veículos pode ser classificado segundo os seguintes critérios:

1. Tempo para servir um determinado nó ou arco
 - Tempo especificado e prefixado.
 - Janela de Tempo (Time Window).

2. Número de domicílios
 - Um domicílio.
 - Mais de um domicílio.
3. Tamanho da frota de veículos
 - Um veículo.
 - Mais de um veículo.
4. Tipo de frota disponível
 - Homogênea.
 - Heterogênea.
5. Natureza da demanda e parâmetros
 - Determinística.
 - Estocástica.
6. Localização da demanda
 - Nos vértices.
 - Nos arcos.
7. Grafo de substrato
 - Direcionado.
 - Não direcionado.
 - Mixto.
8. Restrições na capacidade de veículos
 - Todos sujeitos às mesmas restrições.
 - Restrições diferentes.
9. Tempo de roteamento
 - O mesmo para todos os veículos.
 - Tempos diversos.
 - Sem restrições de tempo.
10. Custos
 - Variáveis (associados à rota escolhida).
 - Fixos.
11. Operação
 - De entrega.
 - De recolhimento.
 - Ambas.
12. Objetivo
 - Minimizar custos fixos.
 - Minimizar custos de operação na rota.
 - Minimizar o número de veículos.
13. Restrições na capacidade dos arcos
 - Imposta a todos os arcos.
 - Impostas a um subconjunto de arcos.
 - Sem restrições.
14. Outras

Uma outra forma de classificar os Problemas de Roteamento é proposta por Maganti (1981) e reforçada por outros autores como Eiselt *et al.* (1995a e 1995b) e Letchford (1996), basicamente separando os clássicos problemas de roteamento em grafos das demais variantes – mais tipicamente voltada para o atendimento de casos práticos e particulares. Nesse sentido os problemas de roteamento em geral poderiam ser classificados em duas grandes classes: Roteamento em Grafos e Roteamento de Veículos propriamente ditos. A classe geral dos problemas de Roteamento em Grafos seria constituída pelas seguintes subclasses:

1. Problema de Roteamento em Nós (associados aos ciclos Hamiltonianos).
2. Problemas de Roteamento em Arcos (associados aos ciclos Eulerianos).

Os Problemas de Roteamento em Nós apresentam-se com variantes do Problema do Caixeiro Viajante englobando principalmente os casos descritos no Capítulo 8 deste livro.

Os Problemas de Roteamento em Arestas (ou arcos) podem ser do tipo capacitado ou não capacitado. Os principais problemas não capacitados, a classe mais comum, são destacados no item 9.2.2.

9.2.2 Problemas de Roteamento em Arcos Não capacitados – Visão Geral

Na literatura são descritos pelo menos os seguintes problemas de roteamento em arestas:

9.2.2.1 O Problema do Carteiro Chinês (PCC)

Um dos mais antigos problemas da teoria dos grafos é o da determinação de um passeio sobre um grafo G que contenha toda aresta de G exatamente uma vez (ver Karp (1975)). Tal circuito é denominado de Euleriano, pelo fato de Euler ter sido o primeiro a reportar um estudo sobre a sua determinação (no ano de 1736). Alguns autores afirmam, inclusive, que a teoria dos grafos nasceu a partir desse problema.

O PCC é um problema de otimização que objetiva cobrir com um passeio (ou *tour*) todos os arcos do grafo, minimizando a distância total percorrida. O passeio do carteiro distingue-se do circuito (ou ciclo) euleriano por nele ser permitida, se necessária, a repetição de arestas. Claramente no caso de o grafo possuir circuitos eulerianos, tais circuitos solucionam o problema. O PCC é um exemplo de um problema de roteamento que admite solução em tempo polinomial (Edmonds e Johnson [1973]). A Figura 9.3 (b) representa a solução para um ciclo Euleriano no grafo representado na Figura 9.3 (a). O caminho fechado 1-2-6-7-2-3-4-5-1-6-8-1 constitui uma solução para o caso.

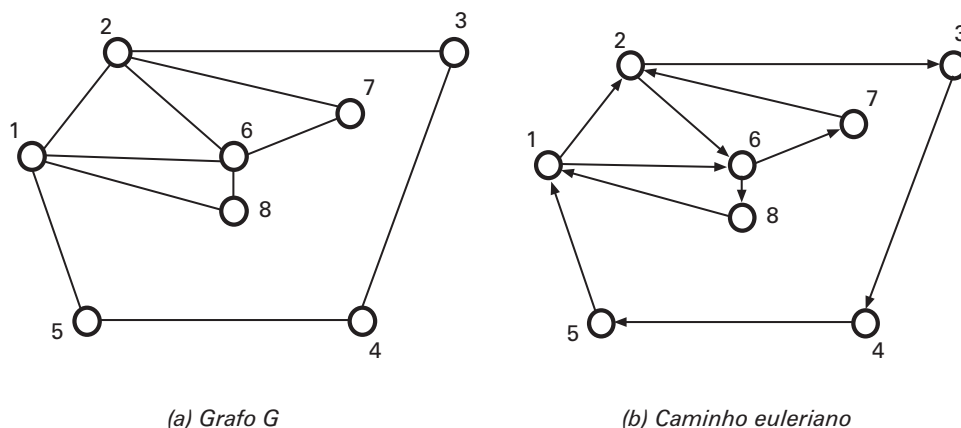


FIGURA 9.3 Caminho Euleriano no Grafo G .

Os seguintes trabalhos apresentam abordagens de solução para o problema:

TABELA 9.1 TRABALHOS PARA A SOLUÇÃO DO PROBLEMA DO CARTEIRO CHINÊS

Ano	Pesquisador	Trabalho
1962	Mei-Ko	Descrição do problema
1965	Edmonds	Algoritmo de solução
1995	Eiselt <i>et al.</i>	Vários modelos e algoritmos de solução

A literatura relata variações do problema de roteamento em arestas que, inclusive, não preservam características polinomiais de solução. As principais variantes do problema são:

• **O Problema do Carteiro Chinês Não direcionado (PCCND)**

A solução do problema do carteiro chinês em grafos não orientados e eulerianos reduz-se à determinação do circuito euleriano. Um grafo conexo G é Euleriano quando possui um número par de arestas incidentes em cada nó. Caso o grafo não seja euleriano, eventualmente será necessário percorrer algumas arestas mais de uma vez para que o passeio completo seja possível.

Kwan Mei-Ko foi o primeiro a relatar o problema assim formulado em uma publicação datada de 1962 (Mei-Ko [1962]) na “Chinese Mathematics”. Em virtude de sua origem, o problema acabou denominado como o Problema por Carteiro Chinês. Considerando um grafo $G = (N, A)$, o problema pode ser formulado como se segue:

$$(PCC1) \text{ Minimizar } z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

sujeito a:

$$\sum_{j=1}^n x_{ji} - \sum_{j=1}^n x_{ij} = 0 \quad i = 1, \dots, n \quad (9.1)$$

$$x_{ij} + x_{ji} \geq 1 \quad \forall (i, j) \in A \quad (9.2)$$

$$x_{ij} \geq 0 \text{ e inteiro} \quad (9.3)$$

Onde:

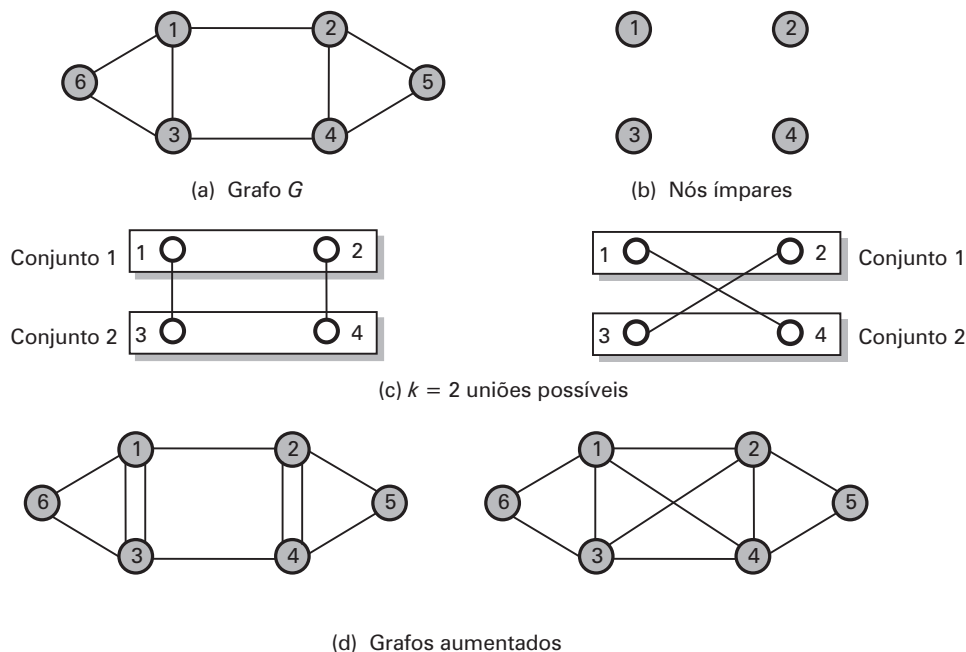
$x_{ij} \equiv$ número de vezes em que a aresta (i, j) é percorrida de i para j ;

$c_{ij} \equiv$ comprimento ou o custo da aresta (i, j) .

No modelo matemático proposto as restrições 9.1 garantem a continuidade do passeio e as restrições 9.2 que nenhuma aresta deixará de ser considerada.

Como em todo grafo conexo existe um número par de nós de grau ímpar, se denominarmos por d_i o grau do nó i e por $|E| = m$ o número de arestas, então $\sum_i d_i = \sum_{i \in \text{ímpar}} d_i + \sum_{i \in \text{par}} d_i = 2m$, pois cada aresta

possui dois nós extremos. Como a primeira parcela da soma é par, a segunda também o será. Fazendo notar por N_i o conjunto de nós de grau ímpar em G e por N_p o conjunto de nós de grau par, e ainda por N o conjunto de todos os nós, desde que o número de nós de grau ímpar é par, $|N_i|$ é par. Utilizando esse fato, podemos particionar N_i em dois conjuntos e formar $k = 1/2 |N_i|$ caminhos entre pares de nós distintos. As arestas (aqui denominadas E^*) contidas nesses caminhos são acrescentadas ao grafo original G como arcos artificiais, obtendo-se um grafo $G_i(E^*)$. O problema então se reduz a determinar os k caminhos que ligam os K pares de nós ímpares (Christofides [1976]). A abordagem de Christofides é a de transformar o PCC em um problema de determinação de um ciclo euleriano em um grafo convenientemente expandido. A Figura 9.4 esclarece o processo de obtenção desse um grafo aumentado $G_i(E^*)$.


 FIGURA 9.4 Processo de obtenção de $G_f(E^*)$.

O grafo G da Figura 9.4 é transformado em um grafo euleriano pelo acréscimo de arestas. As arestas 1-3 e 2-4, bem como as 1-4 e 3-2 representam os caminhos mais curtos entre esses pares de vértices. A solução do problema seria, dentre as duas alternativas possíveis, o percurso total mais barato.

A solução exata desse problema pode ser obtida em $O(n^3)$ como mostra Papadimitriou (1992). Edmonds e Johnson (1973) apresentam interessante algoritmo para a solução do PCC via *matching* (emparelhamento). Podemos resumir o algoritmo da seguinte forma:

Algoritmo Carteiro Chinês

INÍCIO

Ler o grafo $G = (N, A)$;

Se todos os nós em G , o grafo original, possuem grau par então **determinar** um ciclo euleriano em G e **Fim**.

Organizar um grafo K_n da seguinte forma:

Reunir todos os vértices de grau ímpar no grafo K_n e **associar** a cada par de vértices i e j no grafo, uma aresta (i, j) com peso igual ao caminho mais curto que liga i a j no grafo G .

Determinar o 1-matching mínimo em K_n, M^* .

Para cada aresta pertencente a M^* **associar** uma nova aresta em G no caminho mínimo que ela representa, obtendo um grafo G_a .

Determinar a solução do carteiro chinês que é representada por um ciclo euleriano em G_a

FIM

As Figuras 9.5 e 9.6 exemplificam o processo descrito no algoritmo carteiro chinês. A solução para o problema sobre o grafo da Figura é o caminho: $A \rightarrow B \rightarrow F \rightarrow E \rightarrow A \rightarrow D \rightarrow E \rightarrow F \rightarrow C \rightarrow E \rightarrow C \rightarrow B \rightarrow A$ com $C = 19$.

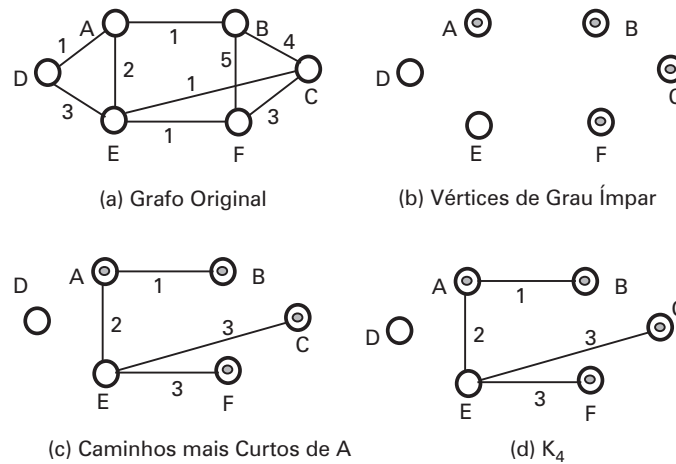


FIGURA 9.5 Obtenção de K_4 .

No caso do grafo orientado uma condição suficiente para a existência do circuito euleriano é que o semigrau interior de cada nó seja igual ao semigrau exterior. Para solucionar o problema vamos determinar dois conjuntos de nós S e D . O primeiro corresponde aos nós determinados de forma que o semigrau interior $d(i)^+$ supere o semigrau exterior $d(i)^-$ (detalhes do conceito de semigrau no anexo), e o segundo, caso contrário. Podemos construir um grafo bipartido com os nós $s_j \in S$ e $d_k \in D$. Cada distância c_{jk} entre dois nós desse grafo representará o menor caminho entre o nó s_j e o nó s_k no grafo original. Tais distâncias podem ser determinadas pela aplicação de um algoritmo de caminho mais curto sobre G . Se existe um nó s_j que não possui caminho que o ligue a todos os nós s_k , então o PCC não possui solução viável. A Figura 9.6 demonstra o fato para um grafo direcionado.

Na verdade a solução do PCC exige a solução de um problema de 1-matching para a formação do grafo $G_i(E^*)$ e, posteriormente, a solução de um problema de circuito euleriano. Como exibido no Capítulo 7, o problema da determinação de 1-matching em um grafo bipartido (conjuntos S e D) pode ser solucionado como um caso do problema de transporte em cerca de $O(n^{2.5})$ operações, onde $n = |N|$. Associando então os s_j a pontos de suprimento, e os d_j a pontos de demanda, constituindo um problema de transporte conforme a seguinte formulação:

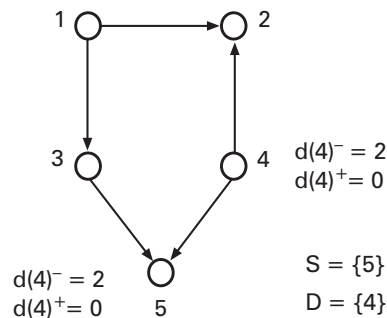


FIGURA 9.6 PCC inviável.

Beltrani e Bodin (1982) apresentam também uma interessante heurística para a solução desse problema.

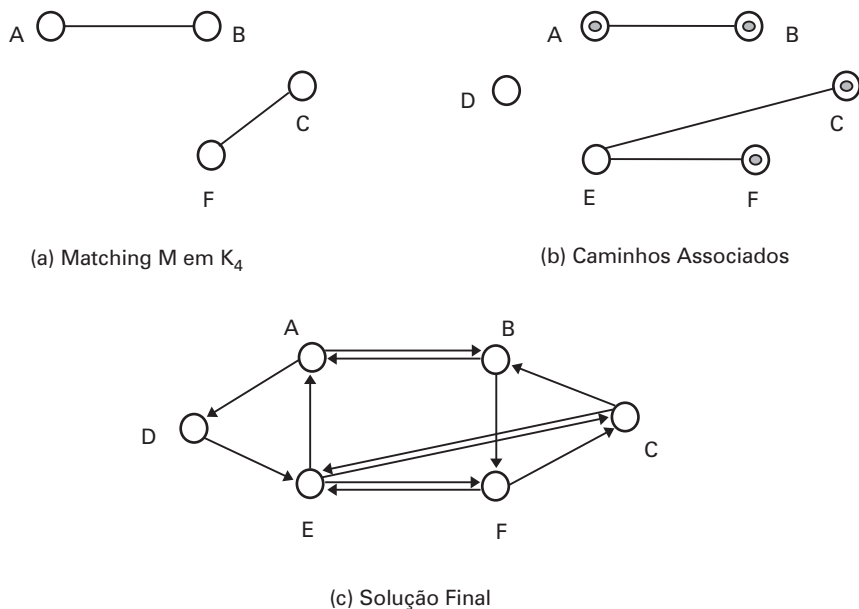


FIGURA 9.7 Solução Final.

• O Problema do Carteiro Chinês Direcionado (PCCD)

Trata-se do caso do PCC com arestas direcionadas. Os trabalhos da Tabela 9.2 apresentam abordagens clássicas para a solução do problema.

TABELA 9.2 TRABALHOS PARA A SOLUÇÃO DO PCCD

Ano	Pesquisador	Trabalho
1973	Edmonds e Johnson	Algoritmo $O(n^3)$
1974	Beltrami e Bodin	Algoritmo $O(m.n^2)$
1988	Lin e Zhao	$O(kn^2)$ fluxo em redes

• O Problema do Carteiro Chinês com Arestas Mistas (PCCND)

Trata-se do caso do PCC com arestas mistas. Esse problema é NP-Árduo. A Tabela 9.3 apresenta alguns dos trabalhos associados ao PCCND.

TABELA 9.3 TRABALHOS PARA A SOLUÇÃO DO PCCND

Ano	Pesquisador	Trabalho
1973	Edmonds e Johnson	Heurística Mix-1
1979	Frederickson	Heurística Mix-2
1979	Kappauf e Koehler	Programação inteira
1979	Minieka	Transformação em problema de fluxo
1984	Christofides <i>et al.</i>	Algoritmo exato
1993	Ralphs	Resultados teóricos
1995	Pearn e Liu	Modificações em Mix-1
1996	Nobert e Picard	Planos de corte (Gomory)

TABELA 9.3 CONTINUAÇÃO

Ano	Pesquisador	Trabalho
1998	Yan e Thompson	B&B
1998	Raghavachari e Veerasamy	Algoritmos heurísticos
1999	Pearn e Chou	Modificações em Mix-1 e Mix-2
2002	Corberán <i>et al.</i>	Algoritmo GRASP
2002	Yaoyuenyong <i>et al.</i>	Algoritmo heurístico

• O Carteiro Chinês Capacitado (PCCC)

Esse problema diz respeito a definir um conjunto de rotas para um conjunto $S = \{1, \dots, s\}$ de carteiros que devem atender à demanda despertada no grafo $G = (V, E)$. Golden e Wong (1981) formulam esse problema de roteamento como um problema de fluxo da seguinte forma:

$$(PCC3) \text{ Minimize } z = \sum_{i \in V} \sum_{j \in V} \sum_{p \in S} c_{ij}^p x_{ij}^p$$

sujeito a:

$$\sum_{j \in V} x_{ji}^p - \sum_{j \in V} x_{ij}^p = 0 \quad \forall i \in V, p \in S \quad (9.4)$$

$$\sum_{p=1}^s (l_{ij}^p + l_{ji}^p) = \left\lceil \frac{q_{ij}}{w} \right\rceil \quad \forall (i, j) \in E \quad (9.5)$$

$$x_{ij}^p \geq l_{ij}^p \quad \forall (i, j) \in E, p \in S \quad (9.6)$$

$$\sum_{i \in V} \sum_{j \in V} l_{ij}^p q_{ij}^p \leq w \quad p \in S \quad (9.7)$$

$$\sum_{j \in V} f_{ij}^p - \sum_{j \in V} f_{ji}^p = \sum_{j \in V} l_{ij}^p \quad \forall i \in E \setminus \{1\}, p \in S \quad (9.8)$$

$$f_{ij}^p \leq (n^2) x_{ij}^p \quad \forall (i, j) \in E, p \in S \quad (9.9)$$

$$f_{ij}^p \geq 0 \quad p \in S \quad (9.10)$$

$$q_{ij} > 0 \quad \forall (i, j) \in E \quad (9.11)$$

$$f_{ij}^p, x_{ij}^p \in \{0, 1\} \quad \forall (i, j) \in E, p \in S \quad (9.12)$$

Onde:

$C = [c_{ij}] \equiv$ matriz do comprimento dos arcos.

$Q = [q_{ij}] \equiv$ matriz das demandas que são despertadas nos arcos e devem ser atendidas pelos carteiros.

$w \equiv$ capacidade dos carteiros, sendo $w \geq \max q_{ij}, \forall (i, j) \in E$.

$l_{ij}^p \equiv$ variável binária que assume valor 1 quando o carteiro p servir a demanda do arco (i, j) , e 0 em caso contrário.

$f_{ij}^p \equiv$ variável de fluxo que assume valor positivo se $x_{ij}^p = 1, f_{ij}^p \in \mathbb{R}^+$.

O depósito central que receberá o índice 1.

As restrições (9.4) garantem a continuidade das rotas dos carteiros. As restrições (9.5) asseguram que o atendimento dos carteiros é considerado em apenas uma das suas passadas pelo arco. As restrições (9.3) obrigam que o carteiro percorra os arcos a que lhe forem designados atender. A restrição (9.4)

garante que os atendimentos dos diversos carteiros não ultrapassem sua capacidade. As restrições (9.5) garantem que o fluxo de atendimento é igual ao computado pela designação aos carteiros.

Golden e Wong (1981) demonstraram que esse problema é NP-Árduo e apresentam um estudo sobre a obtenção de limites inferiores para o problema via 1-*matching*. Nesse trabalho destacamos o seguinte algoritmo para a solução do problema:

Algoritmo de Christofides
<p>INÍCIO</p> <p>Ler o grafo $G = (N, A)$;</p> <p>Fazer todos os arcos serem servidos em um circuito individual.</p> <p>Iniciando com o maior circuito disponível, verifique se um arco de um circuito menor pode ser servido por um circuito maior.</p> <p>Sujeito às restrições do problema procure compor dois circuitos de forma a obter a maior economia possível.</p> <p>Repetir a composição até não existir composição que traga economia à solução;</p> <p>FIM</p>

O algoritmo é descrito de forma vaga, segundo seu autor, exatamente para permitir a absorção, em cada passo, de diversos procedimentos de otimização já existentes. A etapa 3, por exemplo, pode ser implementada com auxílio das heurísticas de Clark e Wright (1964).

• O Problema do k-Carteiro Chinês (k-PCC)

Trata-se do caso do PCC com pelo menos k carteiros, $k > 1$, trabalhando em conjunto. A instância é um grafo $G = (V, E)$, com vértice inicial $s \in V$. A solução é uma coleção de k ciclos contendo o vértice inicial s , de modo que coletivamente cada aresta seja percorrida pelo menos uma vez. O objetivo é minimizar o comprimento dos k ciclos formados. Pearn (1992) e Zhang (1992) apresentam algoritmos de solução para o problema.

• O Problema do k-Carteiro Chinês Min-Max (MM-k-PCC)

Trata-se do caso do k-Carteiro Chinês encontrar k rotas em que a maior aresta é minimizada. Ahr e Reinelt (2002) apresentam heurísticas baseadas no procedimento de Frederickson, Hecht e Kim (1978).

• O Problema do Carteiro ao Vento (PCW) – Windy Postman

Trata-se do caso do PCC em que a matrix de custo do grafo G é assimétrica e não obrigatoriamente atende à desigualdade triangular. O sentido de percurso mais caro é denominado “contra o vento” e o mais barato de “a favor do vento”. A Tabela 9.4 apresenta alguns trabalhos referentes ao problema

TABELA 9.4 TRABALHOS DE SOLUÇÃO PARA PCW

Ano	Pesquisador	Trabalho
1979	Minieka	Primeira descrição do problema
1984	Guan	Heurística que transforma PCW em PCC – $O(n^3)$
1989	Win	Heurística de fluxo mínimo $O(n^3)$
1992	Grötschel e Win	Algoritmo de planos de corte
1994	Pearn e Lin	Heurísticas que alteram as heurísticas de Guan e Win

• O Problema do Carteiro Chinês Hierárquico (PCCH)

Nessa variante o carteiro chinês faz sua rota dentro de grupamentos de arcos obedecendo a relações de precedência entre os grupamentos. O modelo possui aplicação na limpeza de neve e no corte. O problema é solucionado em tempo polinomial. A Tabela 9.5 resume alguns trabalhos relacionados ao tema.

TABELA 9.5 TRABALHOS DE SOLUÇÃO PARA PCW

Ano	Pesquisador	Trabalho
1984	Manber e Israni	Aplicação ao corte de chapas metálicas
1987	Dror <i>et al.</i>	Formalização do problema – algoritmo $O(kn^5)$
1995a	Eiselt <i>et al.</i>	Reformulação do problema
2000	Ghiani e Improta	Algoritmo baseado em <i>matchings</i> e caminhos $O(k^3n^3)$
2002	Korteweg	Algoritmo lexicográfico $O(kn^4)$
2004	Cabral <i>et al.</i>	Formulação via carteiro rural

9.2.2.2 O Problema do Carteiro Rural (PCR)

O carteiro rural é uma variante do carteiro chinês em que o conjunto de arestas que deve ser obrigatoriamente percorrido é um subconjunto das arestas de um grafo G . O problema pretende descobrir o caminho de menor custo em G que percorra o referido subconjunto de arestas. O problema clássico do Carteiro Rural foi formulado em grafos não direcionados. A Tabela 9.6 apresenta trabalhos que abordam o problema.

TABELA 9.6 TRABALHOS DE SOLUÇÃO PARA PCR

Ano	Pesquisador	Trabalho
1974	Orloff	Descrição do problema
1988	Ball e Magazine	Heurísticas de inserção
1995	Eiselt <i>et al.</i>	Algoritmo heurístico
1995	Pearn e Wu	Algoritmos baseados em Chistofides <i>et al.</i> 1986
1996	Lee <i>et al.</i>	Algoritmo genético
1998	Fernandez de Córdoba <i>et al.</i>	Algoritmo heurístico
1998	Cook <i>et al.</i>	Algoritmo genético
1998	Kang e Han	Algoritmo genético
1999	Hertz <i>et al.</i>	Algoritmos heurísticos
2000	Ghiani e Laporte	Algoritmo B&Cut
2003	Fernández <i>et al.</i>	Formulação e limites

O problema é NP-Árduo e admitindo pelo menos as seguintes variantes:

• O Problema do Carteiro Rural Direcionado (PCRD)

Trata-se do caso no qual o carteiro rural é desenvolvido sobre um grafo direcionado. Para o problema destaca-se o trabalho de Christofides *et al.* (1986).

• O Problema do Carteiro Rural com Arestas Mistas (PCRM)

Nessa variante o carteiro rural é desenvolvido sobre um grafo misto. Para o problema destacam-se os trabalhos exibidos na Tabela 9.7.

TABELA 9.7 TRABALHOS DE SOLUÇÃO PARA PCRM

Ano	Pesquisador	Trabalho
1996	Anily <i>et al.</i>	Estudo de caso
2000	Corberán <i>et al.</i>	Algoritmo tabu
2002	Corberán <i>et al.</i>	Carteiro rural com penalidades

• O Problema do Carteiro Rural Periódico (PCRP)

O caso periódico é abordado no trabalho de Ghiani *et al.* (2003) que propõe um algoritmo heurístico de solução.

A literatura cita sem destaques o caso do Carteiro Rural Capacitado (PCRC). O Carteiro Rural Dinâmico (PCRD) é abordado em Moreira *et al.* (2001).

9.2.2.3 O Problema da Empilhadeira – Minimum Stacker Crane (PMSC)

Trata-se de um roteamento em arestas em um grafo $G = (V, E, A)$ em que o objetivo é determinar, a partir de um vértice $s \in V$ um circuito de comprimento mínimo que passe pelo menos uma vez por todos os arcos de E – atendendo a seus sentidos de tráfego. Não existem restrições quanto à repetição de vértices, arcos ou arestas do circuito. A Tabela 9.8 apresenta alguns trabalhos referentes ao problema.

TABELA 9.8 TRABALHOS DE SOLUÇÃO PARA PMSC

Ano	Pesquisador	Trabalho
1978	Frederickson <i>et al.</i>	Descrição do problema e algoritmo 9/5 aproximado
1995	Burkard <i>et al.</i>	Algoritmo de fluxo em redes
1999	Righini e Trubian	Limites para o problema
2002	Hansen e Clausen	Aplicação real
2003	Coja-Oghlan <i>et al.</i>	Extensão para o caso de redes de transporte

9.2.3 Problemas de Roteamento sobre Arcos – Visão Geral

9.2.3.1 Roteamento de Veículos Capacitados (frota homogênea) – PRVC

A literatura em anos passados abordou o problema de roteamento de veículos principalmente em sua versão sem limitação de capacidade e de número de veículos (Golden e Skiscim [1986]; Van Bredam [1996]). Atualmente a maioria dos trabalhos se concentra no caso capacitado. O problema de roteamento de veículos capacitados é relatado com uma frota de m veículos de capacidade W e sediados em ponto garagem. Nesse caso o objetivo é atender a um conjunto de pontos de demanda localizados sobre os vértices do grafo $G = (N, A)$ de modo a minimizar o comprimento total das rotas dos veículos. A Tabela 9.6 mostra alguns trabalhos referentes ao tema.

TABELA 9.9 TRABALHOS DE SOLUÇÃO PARA PRVC

Ano	Pesquisador	Trabalho
1959	Dantzig e Ramser	Trabalho referência para a área
1973	Laporte	Várias técnicas de solução
1980	Psaraftis	Programação dinâmica
1980	Crowder e Padberg	B&B

TABELA 9.9 CONTINUAÇÃO

Ano	Pesquisador	Trabalho
1981a	Christofides <i>et al.</i>	Relaxação lagrangeana e dualidade
1981	Christofides	Teoria dos grafos
1981b	Christofides <i>et al.</i>	Teoria dos grafos
1983	Laporte	Várias técnicas
1984	Desrosier <i>et al.</i>	Geração de colunas
1985	Fleischmann	Planos de corte
1985	Laporte <i>et al.</i>	Eliminação de subtours do PCV
1985	Salvelsbergh	Busca local
1986	Desrosiers <i>et al.</i>	Programação dinâmica
1987	Laporte	Várias técnicas
1989	Agarwal <i>et al.</i>	Geração de colunas e B&B
1990	Haouari	Geração de colunas
1990	Salvelsbergh	Busca local
1991	Dumas <i>et al.</i>	Geração de colunas
1991	Padberg e Rinaldi	Branch-and-Cut
1992	Desrochers <i>et al.</i>	B&B
1994	Fisher	Teoria dos grafos
1995	Miller	Algoritmos exatos
1996	Vigo	Algoritmos heurísticos
1996	Renaud <i>et al.</i>	Heurística de pétala
1997	Kohl e Madsen	Relaxações Lagrangeanas e dualidade
1998	Rodríguez	Heurísticas de busca local
1999	Barbarosoglu e Ozgur	Busca Tabu
1999	Bullnheimer <i>et al.</i>	Algoritmo em colônia de formigas
2000	Hamacher e Hochstättler,	Técnicas de agrupamentos
2000	Backer <i>et al.</i>	<i>Constraint Programming</i> e meta-heurísticas
2000	Campos e Mota	Algoritmos heurísticos
2001	Asano	Roteamento em árvores
2001	Charikar	Vários algoritmos
2001	Bredam	Heurísticas e meta-heurísticas
2002	Toth e Vigo	Métodos exatos e relaxações
2002	Cordeau <i>et al.</i>	Trabalho abrangente
2002	Letchford <i>et al.</i>	Técnica <i>multistart</i>
2003	Ralphs	B&Cut
2003	Baker e Ayechev	Algoritmos genéticos
2003	Jaskiewicz e Kominek	Busca local
2003	Tavares <i>et al.</i>	Influência da representação para algoritmos evolucionários
2003	Toth e Vigo	Algoritmo tabu granular
2003	Achuthan <i>et al.</i>	B&Cut
2004	Prins	Algoritmo Evolucionário
2004	Tarantilis	Algoritmo tabu
2004	Reimann	Colônia de formigas

9.2.3.2 Roteamento de Veículos com Múltiplos Depósitos – PRVMD

Nessa variante os veículos podem partir de vários depósitos. A Tabela 9.10 apresenta alguns trabalhos referentes ao problema.

TABELA 9.10 TRABALHOS DE SOLUÇÃO PARA PRVMD

Ano	Pesquisador	Trabalho
1993	Dell'Amico <i>et al.</i>	Algoritmo heurístico
1993	Chao <i>et al.</i>	Algoritmo híbrido
1997	Filipec <i>et al.</i>	Algoritmo heurístico
1998	Desaulniers <i>et al.</i>	Multidepósito, janela de tempo e custo de espera
1998	Cordeau <i>et al.</i>	Algoritmo tabu
1999	Modares <i>et al.</i>	Algoritmo em rede neural auto-organizada
1999	Salhi e Nagy	Heurística de inserção e grupamento
2000	Irnich	Multidepósito, veículos heterogêneos e entrega/coleta
2002	Wu <i>et al.</i>	Algoritmo heurístico
2002	Giosa, <i>et al.</i>	Algoritmo de localização

9.2.3.3 Roteamento de Veículos com Janelas de Tempo – PRVJT

Nessa variante existe um intervalo de tempo associado aos pontos de demanda ou de coleta “janela de tempo”. A janela de tempo pode se referir ao tempo de chegada, de partida ou de duração do serviço no cliente. A Tabela 9.11 apresenta alguns trabalhos referentes ao problema.

TABELA 9.11 TRABALHOS DE SOLUÇÃO PARA PRVJT

Ano	Pesquisador	Trabalho
1991	Byong-Hun	Janela de tempo e de congestionamento
1992	Savelsbergh	Minimizando a duração da rota
1993	Nagraj	Algoritmo heurístico
1995	Kontoravdis e Bard	Algoritmo GRASP
1995	Russell	Algoritmo híbrido
1996	Chiang e Russell	Algoritmo <i>simulated annealing</i>
1996	Kelly e Xu	Algoritmo tabu
1996	Potvin e Bengio	Algoritmos genéticos
1997	Chiang e Russell	Algoritmo tabu reativo
1997	Bramel e Simchi-Levi	Formulações do problema de cobertura
1997	Kohl e Madsen	Relaxação lagrangeana
1997	Badeau <i>et al.</i>	Tabu paralelo
1997	Marshall <i>et al.</i>	Algoritmos de otimização
1999a	Liu e Shen	Visão geral
1999b	Liu e Shen	Meta-heurística de vizinhança
1999	Schulze e Fahle	Algoritmo paralelo
1999	Gambardella <i>et al.</i>	Algoritmo colônia de formigas
1999	Homberger e Gehring,	Algoritmos evolucionários

TABELA 9.11 CONTINUAÇÃO

Ano	Pesquisador	Trabalho
2001	Cordone e Calvo	Algoritmo heurístico
2001	Bräysy	Busca em vizinhança variável
2001	Gehring e Homberger	Paralelização de meta-heurísticas
2001	Ioannou <i>et al.</i>	Algoritmo guloso
2001a	Tan <i>et al.</i>	Algoritmo genético <i>messy</i>
2001b	Tan <i>et al.</i>	Algoritmo genético híbrido
2001c	Tan <i>et al.</i>	Algoritmos heurísticos
2001	Cordone e Calvo	Algoritmo heurístico
2002	Bard <i>et al.</i>	B&Cut
2002	Cordeau <i>et al.</i>	Vários algoritmos
2002	Rousseau <i>et al.</i>	Operadores de restrição
2002	Jung e Moon	Algoritmo genético híbrido
2002	Czech e Czarnas	Algoritmo simulated annealing paralelo
2003	Braysy	Algoritmo de vizinhança variável reativo
2003	Chaovalitwongse <i>et al.</i>	Algoritmo GRASP
2003	Li e Lim	Algoritmo <i>simulated annealing</i> e reinício
2003	Berger <i>et al.</i>	Algoritmo genético híbrido
2004	Cordeau <i>et al.</i>	Algoritmo tabu
2004	Ioannou e Kritikos	Síntese de várias heurísticas de alocação

9.2.3.4 Roteamento Estocástico de Veículos – PREV

Nessa variante algum dos elementos de entrada do problema como, por exemplo, a demanda, o número de clientes, o custo das ligações etc., é uma variável aleatória. A Tabela 9.12 apresenta alguns trabalhos referentes ao problema.

TABELA 9.12 TRABALHOS DE SOLUÇÃO PARA PREV

Ano	Pesquisador	Trabalho
1992	Cock e Rinnooy Kan	Revisão do problema
1993	Bertsimas e Van Ryzin	Roteamento no plano euclidiano com múltiplos veículos
1993	Belgacem <i>et al.</i>	Roteamento estocástico com <i>split delivery</i>
1995	Bertsimas <i>et al.</i>	Vários algoritmos
1995	Gendreau <i>et al.</i>	Algoritmos para o caso da demanda/clientes estocásticos
1996a	Gendreau <i>et al.</i>	Revisão da literatura
1996b	Gendreau <i>et al.</i>	Algoritmo tabu
1996	Cheung e Power	Modelos e algoritmos
1996	Teodorovic e Pavkovic	Aplicação da teoria dos conjuntos nebulosos
1997	Ong <i>et al.</i>	Demanda estocástica e janela de tempo
2000	Yang <i>et al.</i>	Roteamento estocástico com estoque
2001	Secomandi	Abordagem <i>rollout</i>
2001	Taniguchi <i>et al.</i>	Roteamento com tempo de percurso estocástico

TABELA 9.12 CONTINUAÇÃO

Ano	Pesquisador	Trabalho
2003	Bianchi <i>et al.</i>	Meta-heurísticas
2003	Verweij <i>et al.</i>	Método estatístico de amostragem
2003	Kenyon e Morton	<i>Branch-and-cut</i> e amostragem Monte Carlo

9.2.3.5 Roteamento Dinâmico de Veículos – PRDV

Nessa variante os elementos do problema mudam durante o processo de tomada de decisão. A Tabela 9.13 apresenta alguns trabalhos referentes ao problema.

TABELA 9.13 TRABALHOS DE SOLUÇÃO PARA PRDV

Ano	Pesquisador	Trabalho
1995	Psaraftis	Revisão de métodos de solução
1998	Gendreau e Potvin	Modelos e algoritmo de solução
1999	O'Rourke <i>et al.</i>	Algoritmo tabu
1999	Jih e Hsu	Algoritmo genético
1999	Swihart e Papastavrou	Modelos para o problema
2000	Zhu e Ong	Algoritmo reativo
2002	Larsen <i>et al.</i>	Problema parcialmente dinâmico – modelos e algoritmos
2003	Montemanni <i>et al.</i>	Algoritmo em colônia de formigas
2004	Dua <i>et al.</i>	Sistema híbrido composto de vários algoritmos clássicos

Uma variante desse problema considera a programação de atendimento on-line ou off-line de um conjunto de veículos sobre uma rede de demanda estocástica. Os veículos também são denominados como *servos*.

Psaraftis (1988) considerando que as demandas estão localizadas nos vértices de um grafo e que o veículo (ou servo) pode se deslocar entre quaisquer pares de vértices gastando um tempo que está vinculado aos vértices. O objetivo é minimizar o tempo médio de espera pelo atendimento.

Batta *et al.* (1988) aborda um modelo de serviço igual ao anterior com a obrigação, contanto que o servo retorne à base após cada atendimento.

Bertsimas e Ryzin (1991) estudam o caso em que as demandas ocorrem em qualquer ponto de uma determinada região do plano com uma distribuição de probabilidade uniforme. O servo se desloca entre os pontos de demanda gastando um tempo proporcional à distância entre esses mesmos pontos.

Bertsimas e Ryzin (1993) estudam o caso do atendimento com múltiplos servos.

Tassiulas (1996) aborda um problema semelhante ao de Bertsimas e Ryzin (1991), com o objetivo de estudar o desempenho de várias políticas de roteamento, sugerindo um algoritmo denominado de *congestion focusing algorithm*.

9.2.3.6 Roteamento Periódico de Veículos – PRPV

Na presente variante o planejamento é realizado em um horizonte de tempo que engloba mais de um dia. A Tabela 9.14 apresenta alguns trabalhos referentes ao problema.

TABELA 9.14 TRABALHOS DE SOLUÇÃO PARA PRPV

<i>Ano</i>	<i>Pesquisador</i>	<i>Trabalho</i>
1984	Christofides e Beasley	Apresentação do problema
1984	Tan e Beasley	Algoritmo heurístico
1992	Gaudio e Paletta	Algoritmo heurístico
1995	Chao <i>et al.</i>	Algoritmo heurístico
1997	Cordeau <i>et al.</i>	Algoritmo tabu para o problema multidepósito periódico
2001	Shih e Chang	Aplicação ao recolhimento de lixo contaminado
2002	Angelelli e Speranza	Algoritmo tabu no PRPV com facilidades intermediárias
2002	Baptista <i>et al.</i>	Aplicação ao problema de recolhimento de papel reciclável
2002	Jaillet <i>et al.</i>	Aplicação aos problemas de distribuição com estoque
2003	Blakeley <i>et al.</i>	Aplicação à manutenção de elevadores
2003	Rusdiansyah e Tsao	Heurística híbrida
2004	Alegre e Laguna	Aplicação à coleta de matéria-prima para autopeças.

9.2.3.7 Roteamento com entrega dividida (*Split Delivery*)– *PRED*

No caso desta variante vários veículos fazem o atendimento do cliente dividindo a demanda. A Tabela 9.15 apresenta alguns trabalhos referentes ao problema.

TABELA 9.15 TRABALHOS DE SOLUÇÃO PARA PRED

<i>Ano</i>	<i>Pesquisador</i>	<i>Trabalho</i>
1990	Dror e Trudeau	Modelo e algoritmos heurísticos
1994	Dror <i>et al.</i>	Modelo e algoritmos heurísticos
1995	Frizzell e Giffin	PRED com janela de tempo
2000	Belenguer <i>et al.</i>	Limite inferior para o problema
2002	Feillet	Modelagem do PRED com janela de tempo
2003	Archetti <i>et al.</i>	Algoritmo tabu

9.2.3.8 Roteamento com entrega prioritária (*Backhauls*) – *PREP*

Trata-se a atual variante de um problema de entrega e coleta, em que os veículos devem realizar todas as entregas antes de efetuar qualquer recolhimento. A Tabela 9.16 apresenta alguns trabalhos referentes ao problema.

TABELA 9.16 TRABALHOS DE SOLUÇÃO PARA PREP

<i>Ano</i>	<i>Pesquisador</i>	<i>Trabalho</i>
1988	Casto <i>et al.</i>	Modelos e algoritmos
1989	Goetschalckx e Jacobs-Blecha	Modelos e algoritmos
1992	Min <i>et al.</i>	Variante com múltiplos depósitos
1996	Thangiah <i>et al.</i>	Algoritmos heurísticos
1996	Anily	Algoritmo heurístico
1996	Potvin <i>et al.</i>	Algoritmo genético
1997	Duhamel <i>et al.</i>	Algoritmo tabu

TABELA 9.16 CONTINUAÇÃO

Ano	Pesquisador	Trabalho
1997	Toth e Vigo	Algoritmo exato
1999	Toth e Vigo	Algoritmos heurísticos
1999	Mingozi <i>et al.</i>	Algoritmo exato
2002	Carreto e Baker	Algoritmo GRASP
2002	Osman e Wassen	Algoritmo tabu reativo
2002	Dethloff	Algoritmo heurístico de inserção
2002	Wade e Salhi	Algoritmo de inserção em lista de prioridade variável
2003	Currie e Salhi	Heurística míope construtiva
2004	Wade e Salhi	Algoritmo em colônia de formigas

9.2.3.9 Roteamento com entrega e coleta (*Pick-Ups and Deliveries*) – PREC

Trata-se a atual variante de um problema de entrega e coleta, em que os veículos devem realizar todas as entregas antes de efetuar qualquer recolhimento. A Tabela 9.17 apresenta alguns trabalhos referentes ao problema.

TABELA 9.17 TRABALHOS DE SOLUÇÃO PARA PREC

Ano	Pesquisador	Trabalho
1991	Dumas <i>et al.</i>	PREC com janela de tempo
1993	Bruggen <i>et al.</i>	Busca em profundidade
1998	Mosheiov	Algoritmo em partição de tours
1999	Swihart e Papastavrou	Modelo para o caso estocástico
2000	Nanry e Barnes	Algoritmo tabu reativo/PREC com janela de tempo
2001	Landrieu <i>et al.</i>	Algoritmo tabu para o PREC com janela de tempo
2001	Lau e Liang	Algoritmos e casos testes para o PREC com j/ tempo
2002	Renaud <i>et al.</i>	Heurística com base em procedimento de perturbação
2002	Dethloff	Heurísticas de inserção para vários problemas de roteamento
2003	Xu <i>et al.</i>	Estudo de caso
2003	Malca1 e Semet	Algoritmo tabu
2003	Caricato <i>et. al.</i>	Algoritmo tabu paralelo

9.2.3.10 Roteamento com Frota Heterogenia – PRFH

Trata-se de uma variante onde o objetivo é determinar, simultaneamente, a composição de uma frota de veículos heterogênea sediada em um depósito e suas rotas de atendimento (Taillard, 1999). O problema é também conhecido como:

- O Problema do Tamanho de Frota e Roteamento de Veículos Mistos (*Fleet Size and Composition Vehicle Routing Problem* – Golden, [1984]; Renaud e Boctor, [2002]).
- O Problema da Frota mista de Veículos (*Vehicle Fleet Mix Problem* – Salhi, [1992]; Wassen e Osman, [2002]).

A literatura não relata algoritmo exato para o PRFH. A Tabela 9.18 apresenta alguns trabalhos no problema.

TABELA 9.18 TRABALHOS DE SOLUÇÃO PARA PRFH

Ano	Pesquisador	Trabalho
1984	Golden <i>et al.</i>	Algoritmo de economias adaptado ao PRFH
1984	Gheysens <i>et al.</i>	Formulação inteira mista e comparação de heurísticas
1986	Gheysens <i>et al.</i>	Heurística de duas fases
1988	Ferland e Michelon	Algoritmos exatos e heurísticos
1991	Desrochers e Verhoog	Algoritmo de <i>matching</i> valorado
1992	Salhi <i>et al.</i>	Algoritmo clássico de agrupar e rotear adaptado ao PRFH
1993	Salhi e Rand	Algoritmo de iterativo de perturbação
1996	Osman and Salhi	Algoritmo tabu e λ -interchange
1999	Liu e Shen	Heurísticas de inserção e economia
1999	Han and Cho	Metaheurística baseada no <i>threshold accepting</i>
1999	Taillard	Algoritmo tabu e memória adaptativa
1999	Gendreau <i>et al.</i>	Algoritmo tabu baseado na heurística GENIUS
2002	Wassan e Osman	Algoritmo tabu
2002	Renaud e Boctor	Algoritmo baseado no algoritmo Swapp
2002	Dullaert <i>et al.</i>	Heurísticas para o PRFH com janela de tempo
2004	Tarantilis	Algoritmo com <i>backtracking</i> e <i>threshold accepting</i>
2004	Lima <i>et al.</i>	Algoritmo memético

9.2.3.11 Roteamento Multi-objetivo

A Tabela 9.19 apresenta alguns trabalhos abordando o tema.

TABELA 9.19 TRABALHOS DE SOLUÇÃO PARA PRFH

Ano	Pesquisador	Trabalho
1988	Landeghem	Roteamento com janela de tempo e bicritério
1995	Gudaitis <i>et al.</i>	Algoritmo multicritério usando A*
1997	Golden <i>et al.</i>	Algoritmo de memória adaptativa em problema minmax
1999	Hong e Park	Algoritmo heurístico para PRVJT bi-objetivo
2002	Jozefowiez <i>et al.</i>	Modelos de paralelos híbridos
2002	Applegate, <i>et al.</i>	O problema minmax

9.2.3.12 Outros problemas

• Roteamento com Facilidades Satélites – PRFS

Trata-se de uma variante em que os veículos podem realizar reabastecimento em pontos de depósito (facilidades satélite) distribuídos na rede de demanda. O problema foi recentemente sugerido por Bard *et al.* (1998a). Um algoritmo de solução baseado na técnica de *Branch-and-cut* é sugerido em Bard *et al.* (1998b).

• Roteamento com uso múltiplo dos veículos

O problema é descrito em Taillard *et al.* (1996), em que é sugerido um algoritmo de solução. Hajri-Gabouj e Darmoul (2003) desenvolvem um algoritmo híbrido e evolucionário para a solução de um caso com janelas de tempo.

- **Roteamento com múltiplos veículos de cobertura**

Nesse caso o conjunto de vértice de demanda deve ser coletivamente atendido pela frota de m veículos. O objetivo é minimizar o comprimento total das rotas. Hachicha *et al.* (2000) formulam o problema e sugerem uma heurística que combina o procedimento de Balas e Ho (1980b) com a heurística GENIUS.

- **Roteamento com número limitado de veículos**

Trata-se do problema de roteamento sujeito a um número de veículos. Lau *et al.* (2003) formulam o problema e sugerem um limite superior para o problema de roteamento limitado com janela de tempo. O trabalho sugere também uma heurística tabu.

- **Roteamento e programação multiveículos**

O presente caso refere-se à programação de rota de multiveículos. A frota, diferentemente de PRFH, é fixa. Thompson e Psaraftis (1993) sugerem uma heurística para o problema baseada em uma estratégia denominada *cyclic transfers*. Blanton e Wainwright (1993) desenvolvem um algoritmo genético para o problema multiveículo com restrições de tempo e de capacidade. Liu e Shen (1999) desenvolvem a solução do multiveículos com janelas de tempo.

- **Roteamento com restrição de combustível**

Um problema que pode ser considerado um caso particular do PRFS é tratado em Ichimori *et al.* (1981).

- **Roteamento com restrição de tempo de operação (*time deadline*)**

A variante considera para cada nó de demanda um tempo máximo de atendimento. O problema é um caso especial do roteamento com janela de tempo. Thangiah *et al.* (1993) sugere algoritmos para o problema. Bansal *et al.* (2004) sugere um algoritmo $O(\log n)$ para o caso.

- **Roteamento com Trailers**

Trata-se de um caso interessante de roteamento em que a permanência no vértice da rota é passível de prolongamento significativo. Outro ponto peculiar é que a unidade em rota é composta de um veículo trator e de um trailer. Os veículos tratores podem ser acoplados ou desacoplados dos trailers em conformidade com o planejamento. O estudo da operação de uma frota de veículos com trailer pode incluir operações de transbordo ou troca de trailers. Gerdessen (1996) trata o problema sugerindo várias heurísticas de solução.

- **Gerenciamento de frota**

Trata-se de uma complexa variante em que várias restrições podem ser consideradas, tanto para o dimensionamento da frota como para sua operação. Um exemplo de um trabalho que ataca tal problema é encontrado em Sherali *et al.* (1999).

- **Roteamento com tarefa no destino (*routing and scheduling*)**

A variante se refere, por exemplo, ao caso do uso de frota de veículos de manutenção. No caso os veículos, além de serem designados ao atendimento de um conjunto de pontos de demanda – e consequentemente seguirem uma rota, devem executar tarefas nesses pontos cuja duração é significativa em relação ao tempo de percurso. Modernamente a literatura tem reclassificado certos casos que antes eram denominados genericamente como de *routing and scheduling*. A reclassificação atende às classes anteriormente apresentadas no presente capítulo, especialmente quando o critério mais importante do problema se refere às janelas de tempo, ao comportamento dinâmico do problema ou ao dimensionamento da frota. Laporte e Norbert (1987) e Laporte *et al.* (1988) apresentam uma revisão de métodos exatos para o problema. O problema estocástico é abordado por Jaillet e Odoni (1988). Li *et al.* (1992) trata o caso da minimização de frota. Baker e Schaffer (1986) e Solomon e Desrosiers (1988) reportam resultados para o caso restrito em janelas de tempo. Malmberg (1996) relata um algoritmo genético para o caso da minimização do atraso de atendimento.

• O Problema de Escala de Tripulação (PET – *Crew Scheduling*)

O problema de Escala de Tripulação é constituído de dois casos, a saber:

1. O problema de *pairing*, ou de “planejamento dos trechos embarcado”, determina quais as etapas de viagem que serão cumpridas pelo tripulante. O problema pode incluir (não obrigatoriamente) a função a ser desempenhada por cada tripulante. O problema de designar as funções desempenhadas pelo tripulante embarcado é denominado de *crew complement* ou problema de designar atividades complementares da tripulação.
2. O problema de *rostering*, ou de “planejamento de deveres gerais do tripulante”, define as atividades do tripulante, quer esteja embarcado ou não. No *rostering* as férias, revisão médica, treinamento etc. são obrigações consideradas. Assim o problema de *rostering* inclui a designação de trabalho dos tripulantes a bordo, ou o problema de *crew complement*, no mínimo como dado de entrada. A Figura 9.8 resume as entradas do problema.

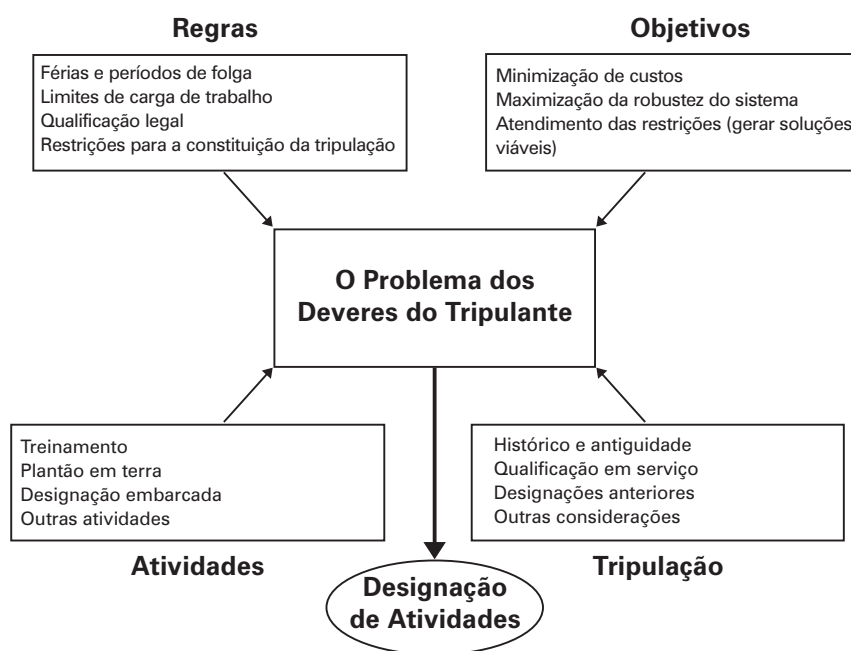


FIGURA 9.8 Entradas do Crew Rostering.

O problema do planejamento dos trechos embarcado é uma sequência de etapas de viagens a serem cumpridas, iniciando em um ponto base e retornando a esse ponto base após certo período de tempo. Planejamento eficiente normalmente cobre as tarefas demandadas pelo serviço de transporte no menor tempo possível (Carrarese e Gallo [1984]; Karraziha *et al.* [2003]). O problema pode envolver diversas restrições, como a de qualificação da tripulação (por exemplo, o caso de pilotos licenciados para determinados aviões ou da exigência do domínio de certa língua pelos tripulantes), considerações de tráfego, atrasos etc. A Tabela 9.20 apresenta alguns trabalhos referentes ao problema.

TABELA 9.20 TRABALHOS DE SOLUÇÃO PARA PET

Ano	Pesquisador	Trabalho
1969	Arabeyre <i>et al.</i>	Revisão da literatura
1979	Marsten <i>et al.</i>	Modelo de programação inteira
1979	Baker <i>et al.</i>	Heurísticas para o problema

TABELA 9.20 CONTINUAÇÃO

Ano	Pesquisador	Trabalho
1981	Marsten e Shepardson	Solução via <i>Set Partitioning Problem</i> – Particionamento
1988	Smith e Wren	Solução via <i>Set Covering Problem</i> – Recobrimento
1988	Lavoie <i>et al.</i>	Geração de colunas
1989	Gershkoff	Aplicação ao problema de <i>pairing</i>
1992	Anbil <i>et al.</i>	Otimização global
1993	Graves <i>et al.</i>	Caso aplicado ao planejamento da United Airlines
1993	Hoffman e Padberg	B&Cut
1995	Barnhart	Fluxo em redes
1996	Mathaisel	Sistema de apoio à decisão
1997	Crainic e Laporte	Modelos de planejamento para transporte de carga
1997	Vance <i>et al.</i>	Formulação e algoritmo de decomposição
1997	Wei <i>et al.</i>	<i>Crew scheduling</i> sob operação irregular
1997	Chu <i>et al.</i>	Problemas de grande porte
1998	Stojkovic <i>et al.</i>	Estudo focalizando aspectos operacionais do problema
1998	Andersson <i>et al.</i>	Aplicação ao problema de <i>pairing</i>
1998	Gamache <i>et al.</i>	Planos de corte e B&B. Aplicado ao caso da Air Canada
1998	Caprara <i>et al.</i>	Modelos e algoritmo de solução
1998	Lavoie <i>et al.</i>	Geração de colunas para o <i>pairing</i>
1999	Lucic e Teodorovic	Simulated annealing para o <i>rostering</i> multiobjetivo
1999	Emden-Weinert	<i>Simulated annealing</i>
1999	Gamache <i>et al.</i>	Algoritmo de geração de colunas para o <i>rostering</i>
1999	Christou <i>et al.</i>	Aplicação ao <i>rostering</i>
2000	Alefragis	Modelo de programação inteira para o <i>crew scheduling</i>
2000	König e Strauss	Serviços integrados de <i>rostering</i>
2001	Freling <i>et al.</i>	Algoritmo Branch-and-Price
2001	Freling <i>et al.</i>	Aplicação na programação de trens na Alemanha
2001	Cordeau <i>et al.</i>	Decomposição de Benders
2001	Butchers <i>et al.</i>	Aplicação ao caso da Air New Zealand
2001	Klabjan <i>et al.</i>	Geração aleatória de rotas para o <i>crew scheduling</i>
2001	Dawid <i>et al.</i>	Modelos p/ <i>rostering</i>
2002	Klabjan <i>et al.</i>	<i>Crew scheduling</i> com janela de tempo
2002	Sellmann <i>et al.</i>	Programação restrita p/ <i>crew scheduling</i>
2002	Fahle <i>et al.</i>	Programação restrita
2003	Freling <i>et al.</i>	Modelos e algoritmos p/ o roteamento e <i>crew scheduling</i>
2004	Kohl e Karisch	Trabalho abrangente
2004	Makri e Klabjan	Esquema de geração de colunas p/ o <i>crew scheduling</i>
2004	Boschetti <i>et al.</i>	Algoritmo exato para uma variante do <i>crew scheduling</i>
2004	Goumopoulos e Housos	Geração de viagens
2004*	Yen e Birge	Programação estocástica para o <i>crew scheduling</i>

* A ser publicado

• Roteamento com coleta de bônus (*Prize Collecting*)

Trata-se de uma variante na qual os clientes que serão servidos devem ser selecionados. Considera-se que cada arco da rota é percorrido segundo um custo c_{ij} , onde i e j são os nós que definem o arco. O atendimento de um cliente i fornece um bônus b_i . O objetivo pode ser encontrar a rota que minimize a diferença entre o somatório dos custos da rota e os bônus obtidos ou a menor rota que garanta um valor mínimo para os bônus. Um caso especial do Problema de Coleta de Bônus é relatado em Goldberg *et al.* (2004), em que o objetivo é maximizar a coleta de bônus, e os veículos estão sujeitos a restrições de tempo total de duração do percurso.

• O Corte de Cana-de-Açúcar

Esse é um caso bastante interessante que aborda o tema de roteamento, pois essa etapa é uma das mais importantes para a execução dessa atividade. Trata-se de um problema de roteamento e *scheduling* com várias tarefas associadas. A Figura 9.9 mostra um esquema básico do problema. A usina recolhe a cana-de-açúcar de várias fazendas de plantio. Cada fazenda é dividida em seções de corte e estas em talhões. No Brasil os métodos de corte e enfeixamento da cana mais utilizados são:

• O método manual

A cana é cortada e enfeixada manualmente, sendo necessários tratores ou carregadores para alimentarem os caminhões de transporte.

• O método mecanizado

Nesse caso as máquinas de colheita cortam, enfeixam e carregam os caminhões.

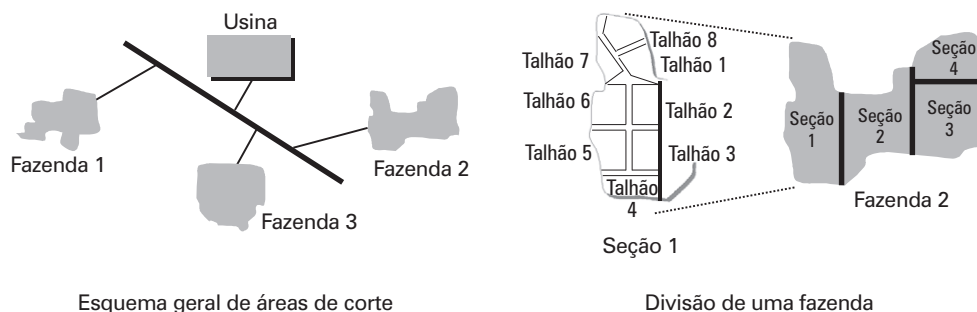


FIGURA 9.9 Esquema de colheita da cana-de-açúcar.

Para que um talhão seja passível de colheita ele deve ser queimado na véspera. Os talhões são colhidos em sequência de vizinhança, uma vez que é necessário ter livre o caminho de acesso aos feixes. Os turnos dos motoristas são, normalmente, de 12 horas. Os caminhões utilizados no transporte possuem carroçaria adaptada ou reboques, uma vez que trata-se de uma carga com grande volume em relação ao seu peso. São comuns caminhões com um ou dois reboques, como mostra a Figura 9.10.

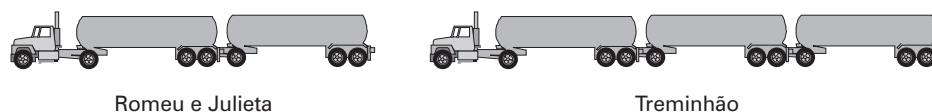


FIGURA 9.10 Tipos de caminhões adaptados ao transporte de cana-de-açúcar.

As usinas trabalham com um depósito-pulmão no seu centro de moagem para absorver flutuações no fluxo de cana originário de problemas operacionais (pneus furados, quebras etc.) e manutenção dos equipamentos e caminhões (quando ocorre manutenção preventiva ou limpeza de carroçaria mais de-

moradas que o previsto) trocas de turnos e a variação do comprimento do percurso ao longo do dia. O depósito-pulmão possui um custo operacional alto, pois, além de representar uma etapa de transbordo desnecessária, ocupa um espaço maior e exige esforços de gerência.

O problema de otimização gerado no corte e transporte da cana-de-açúcar envolve as seguintes variáveis de decisão:

- A definição da frente de serviço.
- Número e tipos de caminhões necessários.
- As rotas associadas a cada caminhão.
- A distribuição dos tratores ou máquinas de corte na frente de serviço.
- O esquema de turnos.
- A gerência do fluxo para manter a usina operando a plena carga.

9.3 – MODELOS E ALGORITMOS DE SOLUÇÃO

9.3.1 Estratégias para Formulação de Problemas de Roteamento

A importância e a influência do modo de formular um problema de otimização, especialmente em áreas complexas como as de roteamento, recobrimento etc., deve, ser bem entendidas. O motivo é evidente: a formulação terá impacto direto no desempenho dos algoritmos de solução. Esse fato não é apenas intuitivo, ele possui justificativas teóricas. Por exemplo, é fácil perceber que existirá uma relação entre o tamanho do espaço das soluções viáveis de um problema, bem como suas características, com a possibilidade de obtermos bons algoritmos de solução para o caso. É ponto pacífico que, quanto maior for o número de variáveis e restrições de um mesmo problema, normalmente maior será o esforço necessário para encontrarmos sua solução. Como vimos no Capítulo 1, o porte das formulações influencia na possibilidade de solução. A característica do espaço de solução são outro ponto fundamental. A característica de integralidade, por exemplo, é um fator conhecidamente complicador.

A interseção do poliedro das restrições de um problema com o reticulado dos inteiros, em algum espaço n -dimensional, estabelece uma correspondência biunívoca com o conjunto de configurações viáveis que respeitam o critério de integralidade. Dessa forma o número dessas configurações pode ser tão grande quanto o é o conjunto dos inteiros.

Denotando por D o conjunto desses pontos discretos e se D é um conjunto finito ou se os dados de formulação envolvem apenas valores racionais, então a envoltória convexa de D (ou $\text{Conv}(D)$) é um poliedro bem definido conforme demonstra Padberg e Sung (1988) (ver também Padberg e Rinaldi (1990)). O teorema de Weyl garante a existência de um conjunto finito de equações lineares ou inequações que descrevem completamente o conjunto $\text{Conv}(D)$. Essa descrição linear completa é também denominada formulação ideal.

Em virtude das características de NP-completude de muitos problemas combinatórios, a formulação ideal pode ser inalcançável em termos práticos. Por outro lado, já verificamos exaustivamente neste trabalho que os problemas de otimização podem admitir diversas formulações diferentes. Como a formulação de um problema pode interferir fortemente em seu processo de solução, o estudo das possíveis alternativas de formulação é um tema relevante.

Dadas duas formulações diferentes A e B de um certo problema, temos imediatamente definidos dois poliedros X_a e X_b . Se $X_a \subset X_b$, então é claro que a formulação A é melhor que a B, uma vez que as fronteiras de seu universo das soluções são mais restritas e estão mais próxima do valor ótimo e limite inferior admitido por X_b (tomado por exemplo o caso de um problema de minimização).

Em última análise, a adequação de uma formulação de um problema Π , inteiro, pode ser medida pela aproximação do poliedro gerado pela formulação e a envoltória inteira do problema. Diferentes

formulações de Π frequentemente conduzem à definição de diferentes conjuntos de vértices no poliedro. Suponhamos que duas formulações A e C modelem o mesmo problema Π mas em espaços n -dimensionais diferentes, e que a dimensão do espaço de C é maior que o de A .

Notando por Z_c e X_a os poliedros associados às formulações A e C , se existir uma transformação afim T que mapeie os pontos inteiros em Z_c sobre os pontos inteiros em X_a , então podemos calcular a imagem $T(Z_c)$ e compará-la a X_a . Se $T(Z_c) \supset X_a$, então a formulação A é evidentemente melhor que C , já que C não providencia nenhuma informação adicional sobre o convexo inteiro do problema. Se $T(Z_c) \supset X_a$ o poliedro $T(Z_c)$ não necessita ser obrigatoriamente uma formulação do problema representado em $T(Z_c)$. Se as variáveis de A estão contidas no conjunto das variáveis de C então teremos em A uma projeção de C que pode ser representada por um caso particular de $T(Z_c)$. Eventualmente alguma projeção de C pode ser, inclusive, diferente de X_a .

Esses conceitos foram aplicados por Miller, Tucker e Zemlin (Padbeg e Sung [1988]), que demonstraram com o auxílio de $n - 1$ variáveis reais, que o Problema Assimétrico do Caixeiro Viajante pode ser reduzido do uso exponencial de restrições para apenas $(n^2 - n + 1)$ restrições lineares, onde n significa o número de cidades envolvidas, o que é melhor observado no item 8.2.1.

Para o desenvolvimento das formulações do problema de roteamento e quando nada for observado em contrário, consideraremos o seguinte:

- ✓ O grafo se substrato não direcionado $G = (N, A)$
- ✓ $S^* = N \setminus S$ para $S \subseteq N$.

9.3.2 Formulação do Problema Geral de Roteamento

Apresentar formulações úteis para o PRV não é uma tarefa trivial.¹ Uma das formulações mais utilizadas como base a diversos métodos de solução é a de Fisher e Jaikumar (1981).

$$(\text{PRV}) \text{ Minimizar } z = \sum_{i,j} \left(c_{ij} \sum_k x_{ijk} \right)$$

sujeito a:

$$\sum_k y_{ik} = 1 \quad i = 2, \dots, n \quad (9.13)$$

$$\sum_k y_{ik} = m \quad i = 1 \quad (9.14)$$

$$\sum_i q_i y_{ik} \leq Q_k \quad k = 1, \dots, m \quad (9.15)$$

$$\sum_j x_{ijk} = \sum_j x_{jik} = y_{ik} \quad i = 1, \dots, n \quad k = 1, \dots, m \quad (9.16)$$

$$\sum_{i,j \in S} x_{ijk} \leq |S| - 1 \quad \forall S \subseteq \{2, \dots, n\}, k = 1, \dots, m \quad (9.17)$$

$$y_{ik} \in \{0, 1\} \quad i = 1, \dots, n \quad k = 1, \dots, m \quad (9.18)$$

$$x_{ijk} \in \{0, 1\} \quad i, j = 1, \dots, n \quad k = 1, \dots, m \quad (9.19)$$

1. Kulkarni e Bhawe (1985) apresentaram uma formulação incorreta no *European Journal of Operational Research* (ver Brodrie e Waters [1988]).

Onde:

$x_{ijk} \equiv$ variável binária que assume valor 1 quando o veículo k visita o cliente j imediatamente após o cliente i , 0 em caso contrário.

$y_{ik} \equiv$ variável binária que assume valor 1 se o cliente i é visitado pelo veículo k , 0 em caso contrário.

q_i é a demanda do cliente i .

Q_k é a capacidade do veículo k .

c_{ij} é o custo de percorrer o trecho que vai do cliente i ao j .

As restrições (9.13) asseguram que um veículo não visite mais de uma vez um cliente. As restrições (9.14) garantem que o depósito recebe uma visita de todos os veículos. As restrições (9.15) obrigam que a capacidade dos veículos não seja ultrapassada. As restrições (9.16) garantem que os veículos não param suas rotas em um cliente. As restrições (9.17) são as tradicionais restrições de eliminação de subtours.

9.3.3 Estratégias para a Solução dos Problemas de Roteamento

Os problemas em que as variáveis assumem valores inteiros ou que possuem funções objetivo com descontinuidades não podem, geralmente, ser solucionados diretamente pelo algoritmo simplex. Esse é o caso de grande parte dos problemas ditos de roteamento. Para tais, a Pesquisa Operacional desenvolve as estratégias mostradas na Figura 9.11:

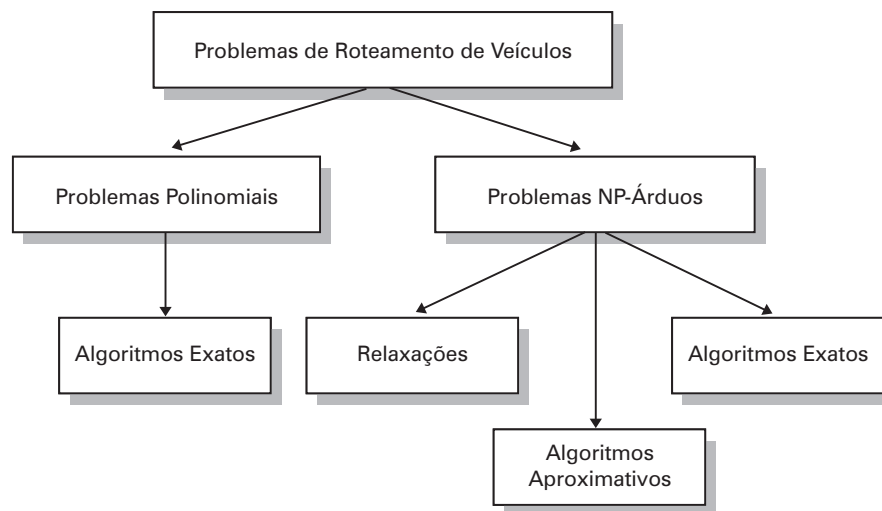


Figura 9.11 Estratégias para solução de PRV.

9.3.4 Abordagens Heurísticas de Solução

Devido à complexidade dos problemas de roteamento o presente texto se concentra na abordagem heurística. As heurísticas para o PRV podem ser classificadas da seguinte forma:

- **Heurísticas Construtivas:**

Para esse tipo de procedimento encontramos variações na forma de:

- Expandir as rotas (critério da economia, expansão radial, grupamentos de proximidade etc.).
- Construir as rotas (sequencial ou paralelo).

Alguns exemplos de heurísticas construtivas são:

Procedimentos de Economia e Inserção

Essas abordagens constroem uma solução através de um conjunto de configurações que a cada passo é atualizado. Os algoritmos realizam a progressão de uma configuração para outra segundo o critério de minimização da função objetivo, também chamado de *saving* ou economia. Arcos de menor custo devem substituir arcos mais caros dentro da rota que vai sendo melhorada em termos de custo. Nos procedimentos de economia e inserção não existe a obrigatoriedade de que a rota seja viável ao longo do processo de melhoria (Paessens, [1988]). Se alguma solução alcançada for viável, então um limite superior para o problema é obtido. Um trabalho clássico dentro dessa linha de atuação é o de Clark e Wright (1964).

• Heurística de Clark e Wright

Considerando que o vetor s_{ij} represente o valor da economia obtida pela reorganização de duas ligações do tipo $(i-1-i)$ e $(j-1-j)$ em uma nova (i, j) como mostra a Figura 9.12.

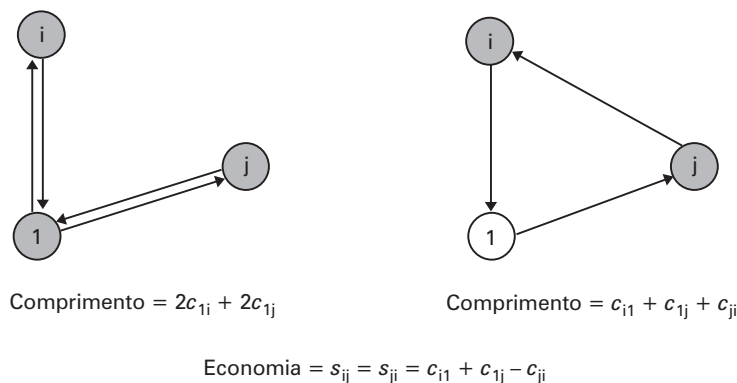


FIGURA 9.12 O procedimento de economia.

Algoritmo Clark e Wright (sequencial)

INÍCIO

Ler $G = (N, A)$, c_{ij} . { *nó 1 é o depósito central do roteamento* }

Inicializar $Rota$: $= (x_1 - x_s - x_1)$,

Calcular a economia $s_{ij} = c_{1j} - c_{ij} + c_{j1}$ para todo o par de clientes x_i e x_j { *nós em G * }

Ordenar as economias em ordem não crescente e colocá-las em uma lista

Enquanto existir ligações na lista **Faça**

Iniciando pela maior economia da lista **Faça**

 Início

Determine a primeira ligação na lista que pode ser utilizada para ser acrescida em um dos dois extremos de $Rota$, aumentando seu comprimento e a retirando da lista;

Se $Rota$ não pode ser expandida da forma anterior **então escolha** a primeira ligação na lista para iniciar uma nova rota, e a retire da lista.

 Fim

FIM

A Figura 9.13 desenvolve um exemplo numérico do procedimento de Clark e Wright aplicado ao grafo da Figura 9.13 (a).

À medida que uma rota vai sendo construída é possível realizar os testes relativos a restrições de capacidade de veículos e, eventualmente, de janelas de tempo. Em relação ao comportamento do algoritmo ele possui a característica de aninhar os clientes nas rotas que vão se formando. Na Figura 9.13 notamos que o nó 3, após a segunda melhoria, passou à condição de *interno* (sem ligação direta com o vértice depósito) não sendo mais candidato a melhoria na rota, uma vez que só são testados para a inserção os nós extremos, ou seja, aqueles que se ligam diretamente ao nó central (vértice 1).

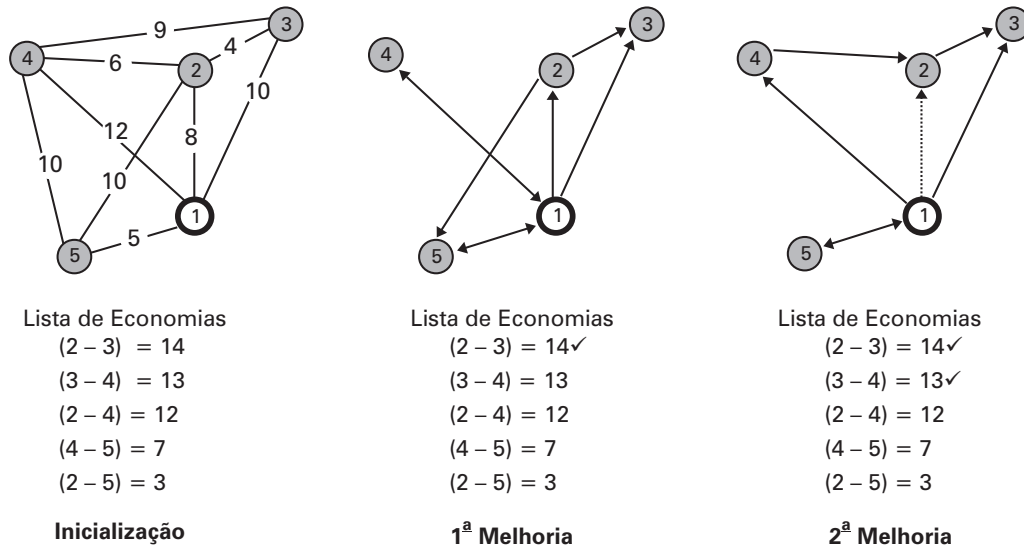


FIGURA 9.13 Exemplo de Aplicação da Heurística de Clark e Wright.

• Heurística de Mole e Jameson

O algoritmo de Mole e Jameson (1976) ataca a fragilidade dos nós internos não serem candidatos a fazer parte do teste de economia, permitindo que a inserção de economia possa ser realizada entre dois nós intermediários. O critério para expandir a rota pela inclusão de novos vértices no algoritmo tem duas condições:

1. $e(i, l, j) = c_{ie} + c_{lj} - \mu c_{ij}$.
2. $\sigma(i, l, j) = \lambda c_{0l} - e(i, l, j)$.

De um modo geral o algoritmo procede da seguinte forma:

Algoritmo Mole e Jameson

INÍCIO

Ler $G = (N, A), c_{ij}$. { *nó 1 é o depósito central do roteamento* }**Inicializar** $F := N \setminus \{x_1\}$ **Enquanto** $F \neq \emptyset$ **Faça**

Início

Rota $= (x_1, x_s, x_l)$ { *vários critérios podem ser utilizados nesta atribuição* } **Para todos** os vértices $x_l \in F$

Início

verifique uma possível inserção na rota em exame que atenda a: $e(i_l, l, j_l) = \min[e(i, l, j)]$ e considere todos os vértices vizinhos x_r e $x_s \in \text{Rota}$ onde x_j^l são os vértices entre os quais x_l será incluído, e que representam a “melhor escolha” (Procedimento MV);

Fim

Inserir o vértice na rota corrente entre os vértices x_i^l e x_j^l e **fazer** $F := F \setminus \{x_l^*\}$

Otimizar a rota **R** utilizando métodos r-óticos²

Fim

FIM

(*) **Procedimento MV** (Melhor Vértice)

INÍCIO

Para todos os nós x_l ainda não incluídos na rota **R** e viáveis, o “melhor vértice” x_{l^*} a ser inserido é o que: $(i_{l^*}, l^*, j_{l^*}) = \max [\sigma(i_l, l, j_l)]$

FIM

✓ **Exemplo numérico do procedimento de Mole e Jameson**

A heurística de Mole e Jameson possui dois parâmetros de controle que podem ser modificados em conformidade com os desejos da estratégia de solução.

- $\lambda = 0$ e $\mu = 1 \Rightarrow$ a inserção será realizada levando em conta a minimização do aumento da rota pela inclusão de um novo cliente.
- $\lambda = 0$ e $\mu = 0 \Rightarrow$ a inserção será realizada levando em conta a minimização da distância entre os dois nós vizinhos.

O caso em que $1 \leq \lambda \leq 2$; $\mu = \lambda - 1$ trata-se de uma generalização do critério de Gaskell (1967).

$\lambda \rightarrow \infty, 0 < \mu < \infty \Rightarrow$ a inserção ocorrerá para o cliente mais distante do depósito.

Se λ cresce, o peso para a distância entre o cliente e o depósito cresce, levando à formação de rotas mais circulares. Quando μ cresce as ligações mais longas são penalizadas.

2. Como o algoritmo de Lin e Kernighan, que será explorado adiante.

Considerando $\lambda = 1$ e $\mu = 1$ e o grafo da Figura 9.14 e considerando a rota parcial da Figura 9.14 (b) em que os nós fora de rota são 4 e 6.

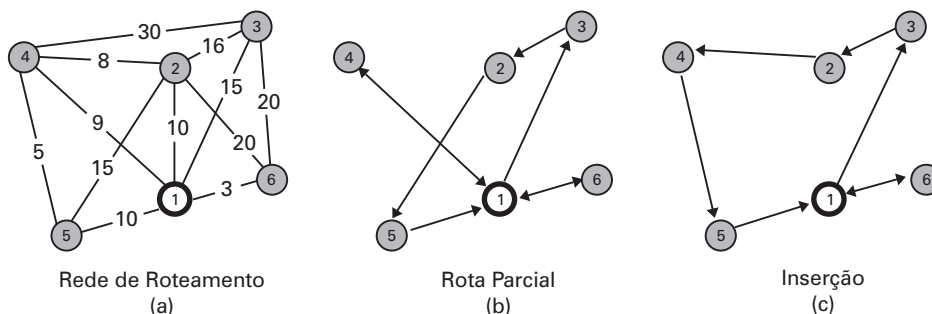


FIGURA 9.14 Exemplo de Aplicação da Heurística de Mole e Jameson.

Para realizar uma inserção o algoritmo calcula inicialmente uma lista de proximidades da seguinte forma:

$I = 4$	$I = 6$
$e(2, 4, 5) = 13$	$e(2, 6, 3) = 40$
$e(5, 4, 1) = 14$	$e(1, 6, 2) = 23$
$e(3, 4, 2) = 38$	

Como exemplo, $e(2,4,5) = 8 + 5 = 13$. Sendo o nó 4 escolhido para a inserção (mais próximo dos vizinhos). Para realizar a inserção propriamente dita o critério da função σ é utilizado conforme a tabela abaixo:

Inserção entre	$I = 4$
2-5	$\sigma(2, 4, 5) = 20$
1-5	$\sigma(5, 4, 1) = 5$
2-3	$\sigma(3, 3, 2) = 14$

Como exemplo, $\sigma(2,4,5) = 18 + 15 - 13 = 20$, o que resulta na inserção que a Figura 9.14 (c) mostra com uma economia de 20 unidades para a rota anterior.

• Heurísticas em Duas Fases:

Estas heurísticas basicamente objetivam reunir os clientes (fase de agrupamento ou *clustering*) designando os veículos aos agrupamentos, e, encontrando a rota dentro do agrupamento (fase de roteamento). A ordem das fases pode ser agrupamento \times roteamento e roteamento \times agrupamento. Dentre essas heurísticas destacamos:

Procedimentos de Primeiro Agrupar e Depois Rotear

• Heurística de Gillet e Miller

Esta estratégia procura obter a solução do problema em duas etapas distintas. A primeira visa a agrupar os pontos de demanda segundo algum critério de proximidade, enquanto na segunda etapa cada grupo (ou *cluster*) é solucionado independentemente. Um exemplo de aplicação dessa abordagem pode

ser encontrado no trabalho de Gillet e Miller (1974). A estratégia parte do princípio de que os trajetos entre nós serão desenvolvidos preferencialmente entre vizinhos. No quadro que se segue desenvolvemos uma versão da heurística que permite evidenciar sua enorme eficiência e simplicidade na fase de grupamento. Em algumas versões dessa heurística a fase de roteamento é sugerida após a organização de todos os grupamentos.

Algoritmo Gillet e Miller
<p>INÍCIO Ler $G = (N, A)$, c_{ij}. { *nó 1 é o depósito central do roteamento*} Obter as coordenadas polares dos clientes em relação ao depósito e ordená-las em ordem de crescimento de seu valor e Fazer:</p> <p style="padding-left: 20px;">Início $F := N \setminus \{x_1\}$ $Ponta_Rota := \{x_1\}$ $Rota_1 := \{x_1\}$ $i := 1$ Fim</p> <p>Enquanto $F \neq \emptyset$ Faça { *Agrupar* }</p> <p style="padding-left: 20px;">Início Enquanto $\exists x_s \in F$ atendendo às condições de viabilidade para $Rota_i$ Início Encontrar o vértice $x_s \in F$ de coordenada polar mais próxima da de $Ponta_Rota$ e Fazer Início $Rota_i := Rota_i \cup \{x_s\}$ $Ponta_Rota := \{x_s\}$ $F := F \setminus \{x_s\}$ $j := j + 1$; Se $Controle(j) = Verdadeiro$ então aplicar <i>Procedimento k-ótimo</i> ($Rota_i$) { *Rotear* } Fim Fim $i := i + 1$ $Ponta_Rota := \{x_1\}$ $Rota_i := \{x_1\}$ Fim</p> <p>FIM</p>
<p>Procedimento K-Ótimo (Melhor Vértice) ver:</p> <ul style="list-style-type: none"> • Lin (1965) e Lin e Kernighan (1973); • Heurística k-ótima no presente Capítulo e no Capítulo 9, heurística de k-substituições.

A Figura 9.15 exemplifica a aplicação do procedimento de **Gillet e Miller** ao grafo da Figura 9.15 (a). A varredura se desenvolve a partir do vértice 5 e no sentido horário.

Nessa classe de heurísticas encontramos também o procedimento de Christofides *et al.* (1979). Nessa abordagem os grupamentos são formados por proximidade e aumentados com critério de inserção e melhoria. Fisher e Jaikumar (1981) fazem o agrupamento pela solução de um problema de designação.

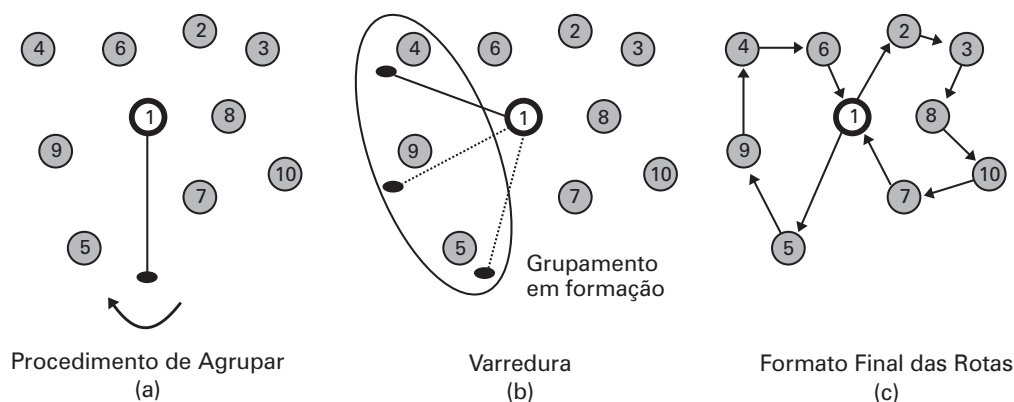


FIGURA 9.15 Exemplo de Aplicação da Heurística de Gillet e Miller.

Procedimentos de Primeiro Rotear e Depois Agrupar

A presente estratégia realiza a operação inversa da número um. A primeira etapa da abordagem busca a identificação de uma rota (normalmente inviável) que englobe todos os pontos de demanda. Em uma segunda fase o circuito é particionado em pequenas rotas viáveis. O trabalho de Newton e Thomas (1974) aplica a presente abordagem.

Procedimentos de Melhoria ou Troca (Improvement or Exchange)

Baseiam-se na aplicação de técnicas conhecidas de divisão (branching) de problemas, combinadas com procedimentos heurísticos para efetuar modificações na solução. A cada passo uma solução viável é alterada de modo a se obter uma nova solução igualmente viável e de menor custo. O procedimento continua até que não seja mais possível efetuar uma redução de custo. Nessa classe citamos o algoritmo de Lin e Kernighan (1973), Christofides e Eilon (1972) e Russell (1977).

• Heurísticas por Aproximação através de Programação Matemática

Alguns algoritmos de solução se baseiam diretamente no uso das técnicas de programação matemática. O trabalho de Fischer and Jaikumar (1981) é um exemplo dessa abordagem.

• Heurísticas por Otimização Iterativa

Esta estratégia possibilita absorver um elevado grau de interação humana no processo de solução. A experiência do especialista pode ser aproveitada ao longo do desenvolvimento do algoritmo. Normalmente o método inclui uma saída de tela em que tanto as rotas como os pontos a atender podem ser visualizados. Jarvis e Ratliff (1981) apresentam um interessante trabalho desenvolvendo uma série de modelos heurísticos nessa linha de raciocínio. Rosenkrantz *et al.* (1977) analisa vários procedimentos heurísticos para a solução do PCV, em um interessante trabalho comparativo.

• Heurísticas de Busca Local

Este tipo de abordagem parte de uma solução inicial e objetiva basicamente intensificar a qualidade da solução obtida. Na verdade a busca local é uma estratégia para reduzir a explosão combinatória de toda a sorte de problemas *NP-Árduos*. Alguns algoritmos construtivos poderiam também ser classificados como de busca local.

Algoritmo Busca Local

INÍCIO

Ler função de custo f e a estrutura de vizinhança N **Selecionar** uma solução s_0 **Enquanto** $f(s) < f(s_0)$ para alguma $s \in N(s_0)$ **Faça**

Início

 Selecione $s \in N(s_0)$ tal que $f(s) < f(s_0)$; $s_0 := s$;

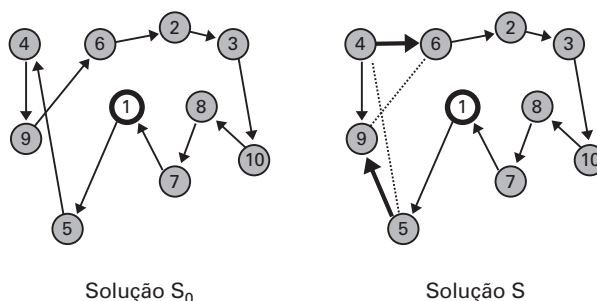
Fim

 s_0 é uma aproximação da solução ótima

FIM

• **Heurística k -Ótima**

A heurística k -Ótima proposta por Lin (1965) é o exemplo de uma heurística de busca local. De fato a heurística propõe uma busca em uma k vizinhança de uma solução de roteamento. A busca se faz pelo exame da possibilidade de troca de k variáveis (arcos) entre s_0 e s . À medida que k cresce o procedimento aproxima-se da enumeração total (caso em que $k = n - 2$). A vizinhança de s_0 cresce exponencialmente na forma de $n! / [(n - k)! k!]$. Para $k = 2$ o tamanho da vizinhança de s_0 é igual a $n(n - 1)/2$. A literatura relata soluções eficientes somente para $k = 2$ e $k = 3$. A Figura 9.16 exemplifica um procedimento de 2-trocas

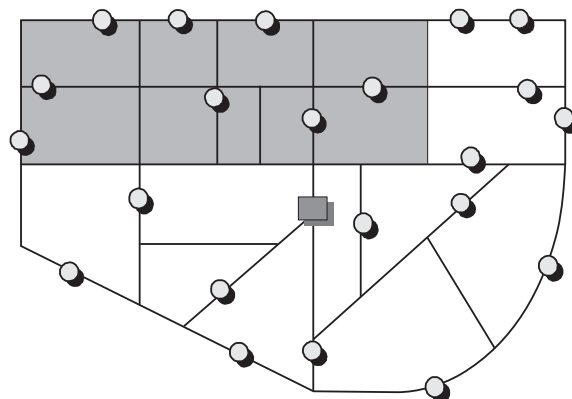
**FIGURA 9.16** Exemplo de aplicação da heurística de Gillet e Miller.

Os arcos 5-4 e 9-6 são substituídos pelos arcos 5-9 e 4-6. O arco 9-4 é percorrido no sentido inverso da solução s . Se o grafo de substrato é não direcionado o procedimento anterior é tipicamente de duas trocas. Se o grafo é direcionado, e os custos de percorrer o arco 9-4 são diferentes do de percorrer 4-9, o procedimento pode ser considerado como de três trocas.

9.4 – PROBLEMAS PROPOSTOS**1 – Formulação**

Suponha que a figura a seguir represente um bairro que é atendido por um depósito de refrigerantes localizado no vértice retangular. A região em negrito vai ser atendida por um certo caminhão de entrega.

- Formule o problema de encontrar o caminho que saia do depósito, atenda aos pontos de demanda e retorne ao depósito minimizando a trajetória percorrida.
- Formule o problema de designar quatro caminhões de entrega para atender a todo o bairro de modo a minimizar o comprimento total das rotas.

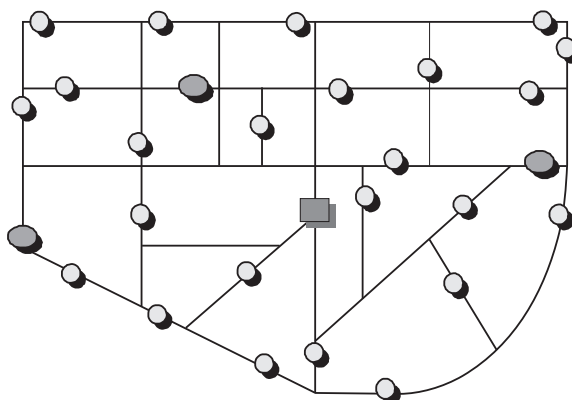


Planta de distribuição

2 – Formulação



Suponha que a figura a seguir represente o mesmo bairro anterior e estamos agora interessados em entregar os resultados de exames laboratoriais e recolher material para novos exames. Os resultados dos exames são entregues nas residências dos usuários, e os materiais para novos exames são apanhados em postos de coleta. Os veículos que fazem a entrega também fazem a arrecadação. Quando um veículo visita um posto de coleta (círculo maior) ele deve voltar imediatamente ao laboratório com o material arrecadado para minimizar o tempo de exposição do material aos riscos do transporte.



Planta de distribuição

Elabore o modelo de programação que minimiza o comprimento total a ser percorrido pelos veículos para o caso de:

- Existirem 3 veículos.
- Existirem 4 veículos.

3 – Modelagem



Analise os seguintes problemas de roteamento e os classifique segundo a metodologia proposta por Bodin e Golden (1981).

- Entrega de pizza, congelados etc.
- Distribuição e recolhimento de leite.

- Distribuição urbana de concreto.
- Projeto de anéis em redes de telecomunicações.
- Distribuição de vagões ferroviários.
- Roteamento de robôs em manufatura.
- Transporte de bens pessoais (mudanças).
- Roteamento de pacotes em redes de computadores.
- Roteamento de fluxos de comunicações em redes de telecomunicações.
- Recolhimento de cana de açúcar.
- Patrulhamento policial.

10

COBERTURA, PARTICIONAMENTO E LOCALIZAÇÃO

10.1 – INTRODUÇÃO

10.1.1 – A Importância dos Modelos e seu Contexto de Aplicação

Os problemas de localização englobam, na atualidade, uma enorme gama de aplicações para a pesquisa operacional. Normalmente, nesses tipos de problema, a escala do comprometimento com variáveis econômicas é enorme. As decisões sobre a localização de facilidades englobam custos fixos elevados e, normalmente, uma análise de retorno a longo prazo. Em uma outra vertente dos modelos associados ao contexto de localização, encontramos uma série de problemas técnicos de menor escala, decorrentes do processo produtivo tais como a determinação de fluxos em plantas, estoques, layout etc. Em qualquer caso, a característica mais marcante do contexto em que os problemas de recobrimento/particionamento e localização são definidos é o grande impacto econômico decorrente do emprego de técnicas de otimização.

10.2 – PROBLEMAS DE RECOBRIMENTO, PARTICIONAMENTO E *PACKING*

10.2.1 – Importância e Formulação

Os problemas de recobrimento (*set covering problem* – PR) e particionamento (*set partitioning problem* – PP) são problemas de programação linear inteira que se enquadram na classe dos mais difíceis problemas de otimização combinatória existentes. O problema de particionamento (PP) é um caso particular do problema de recobrimento (PR) e possui aplicações, dentre outras, nas seguintes áreas:

- Recuperação de informações em bancos de dados (Balas [1980]).
- Alocação de tripulações em linhas aéreas (Vance *et al.* [1997]).
- Distribuição do tráfego de comunicações em satélites (Minoux [1984]), Garcia [1990]).
- Alocação de serviços de emergência (Benvenites [1982]).
- Alocação de serviços diversos (Revelle *et al.* [1970] e [1976]).
- Roteamento de petroleiros (Ronen *et al.* [1987]).

- Seleção de listas (Dwyer *et al.* [1981]).
- Gestão estratégica (Garfinkel e Nemhauser [1970]).
- Lógica simbólica (Fridshal e North [1979]).
- Balanceamento de capacidade em linhas de produção (Salveson [1955]).
- PERT/CPM (Salveson [1955]).
- Roteamento de veículos terrestres (Cullen *et al.* [1981], Bramel e Simchi-Levi [1994]).
- Roteamento de veículos terrestres e navios (Bodin [1990]).
- Distribuição de distritos de venda (Easingwood, [1980]).
- Problema de coloração (Came [1977]).
- Caminhos em grafos (Bellmore *et al.* [1970]).
- Planejamento de tarefas (Bartholdi *et al.* [1980]).
- Projeto de circuitos (Christofides e Brooker [1976], Bartholdi [1981], Eben-Chaine *et al.* [1996]).
- Localização de estações de radar (Goldbarg [1987]).
- Exploração de petróleo em campos submarinos (Goldbarg e Fampa [1995]).
- Análise de amostras de sangue (Nawijn [1988]).
- Alocação arma \times alvo (Campello e Goldbarg [1991]).
- Carregamento de *pallets* (Dowsland [1990]).
- Estocagem (Cattrysse [1990]).
- Planejamento da utilização de ferramentas de usinagem e corte (Finke e Kusiak [1987]).
- Distribuição de serviços médicos (Daskin e Stern [1981]).
- O problema de cortes (Vasko *et al.* [1992]).
- Na tomografia (Bonnet *et al.* [2002]).
- Para o problema de alocação (Savelsbergh [1997]).
- Para designação de bandas de telecomunicações (Parker e Rayan [1994]).
- No planejamento de cadeias de suprimento (Cochrana e Uribeb [2004]).
- Na estocagem de itens sujeitos a deterioração (Hwang, [2004]).

Considerando o conjunto finito $M = \{1, 2, \dots, m\}$, e o conjunto $F = \{M_1, M_2, \dots, M_n\}$ uma família de subconjuntos de M , onde $M_j \subseteq M, j \in J = \{1, 2, \dots, n\}$. Uma *cobertura* é uma coleção de elementos de F , por exemplo $F^* = \{M_{j(1)}, M_{j(2)}, \dots, M_{j(p)}\}$, onde os índices de F^* pertencem ao conjunto $P = \{1, \dots, p\}$ e são denominados por $J^* = \{j(1), j(2), \dots, j(p)\}$, possuindo a seguinte propriedade:

$$\bigcup_{i=1}^p M_{(i)} = M$$

Um particionamento difere da cobertura pelo fato de que além de $\bigcup_{i=1}^p M_{(i)} = M$ ainda é necessário que $M_{j(i)} \cap M_{j(k)} = \emptyset$, para todo i e $j \in P$. Em outras palavras, considerando o caso de um modelo representado por um conjunto de restrições do tipo $Ax = 1$, para obtermos uma solução de particionamento nesse modelo, duas ou mais variáveis pertencentes à solução jamais poderão cobrir uma mesma linha de A . Nesse exemplo, o conjunto M representaria as linhas de A e os conjuntos $M_{j(i)}$ as colunas. Obviamente, uma cobertura ou um particionamento é um subconjunto de colunas de A .

O custo de uma cobertura F^* é $\sum_{j \in J^*} c_j$. O problema de recobrimento (PR) é determina uma cobertura de custo mínimo. Definindo $A = [a_{ij}]$ uma matriz $m \times n$ cujas linhas e colunas estão associadas aos conjuntos M e F , respectivamente, e tal que:

$$M_j = \{i \in M \mid a_{ij} = 1\} \text{ e } \bar{M}_j = \{i \in M \mid a_{ij} = 0\}$$

então formularemos os problemas em pauta da forma que se segue:

✓ *Formulação Clássica do Recobrimento (PR)*

Podemos formular o PR como um problema de programação linear inteira da seguinte forma:

$$\begin{aligned} \text{(PR) Minimiza } z &= \sum_{j=1}^n c_j x_j \\ \text{sujeito a:} \\ \sum_{j=1}^n a_{ij} x_j &\geq 1 \quad i = 1, \dots, m \\ x_j &\in \{0, 1\}, \forall j \end{aligned}$$

onde:

$x_j \equiv$ variável binária que assume o valor 1 quando a coluna j de A está na solução e 0, caso contrário.
 $c_j \equiv$ custo unitário associado à variável x_j .

✓ *Formulação Clássica do Particionamento (PP)*

A diferença entre o problema de particionamento e o problema de recobrimento reside exclusivamente na natureza das restrições dos problemas. No PP o conjunto de restrições de PR fica reduzido a igualdades. Desse modo:

$$\begin{aligned} \text{(PP) Minimizar } z &= \sum_{j=1}^n c_j x_j \\ \text{sujeito a:} \\ \sum_{j=1}^n a_{ij} x_j &= 1 \quad i = 1, \dots, m \\ x_j &\in \{0, 1\}, \forall j \end{aligned}$$

Evidentemente, toda solução para o PP é também uma solução para o PR associado, mas a recíproca não é verdadeira. Somente em ocasiões particulares o PR poderá produzir limites para o PP (ver Goldbarg [1995]).

✓ Formulação Clássica do Packing (PK)

$$\begin{aligned}
 \text{(PK) Maximizar } z &= \sum_{j=1}^n c_j x_j \\
 \text{sujeito a:} \\
 \sum_{j=1}^n a_{ij} x_j &\leq 1 \quad i = 1, \dots, m \\
 x_j &\in \{0, 1\}, \forall j
 \end{aligned}$$

O modelo de *packing* é de grande aplicação prática na solução de problemas de corte geométricos, acondicionamento de embalagens e carregamento de veículos. Sua relação com o PP é bem conhecida e foi motivo de estudos aprofundados em Nemhauser e Sigismondi (1989) e Nemhauser e Trotter (1975).

✓ Transformação do (PP) em (PR)

A transformação do PP em PR é abordado por Garfinkel e Nemhauser (1966), como mostra o modelo (PR₂) que se segue:

$$\begin{aligned}
 \text{(PR}_2\text{) Minimizar } z &= \sum_{j=1}^n c_j x_j \\
 \text{sujeito a:} \\
 \sum_{j=1}^n a_{ij} x_j &\geq 1 \quad i = 1, \dots, m \\
 \sum_{j=1}^n |M_j| x_j &= m \\
 x_j &\in \{0, 1\}, \forall j
 \end{aligned}$$

Como uma solução inteira de PR₂ atende a todas as restrições (\geq) do modelo, a solução S^* , à solução ótima de PR₂, atende a $\bigcup_{i=1}^p M_{(i)} = M$. Por outro lado, a solução inteira S^* também deverá atender a restrição de cardinalidade que exige que a soma das linhas cobertas através de cada variável seja exatamente igual a M , isto é, $M_{j(i)} \cap M_{j(k)} = \emptyset$. Como PR₂ atende às duas condições impostas pelo PP a solução S^* é um particionamento.

A transformação PP em PR pode ser realizada também por uma simples mudança nos pesos do vetor custo do PP (ver Lemke, *et al.* [1971] e Hulme e Baca [1984]).

✓ Transformação do PP em PK

O problema de particionamento também pode ser considerado um caso particular do problema de *packing*. O conjunto de restrições do *packing* condiciona a coleção de F a ser disjunta como no particionamento, ou seja, $M_{j(i)} \cap M_{j(k)} = \emptyset$. Mas a condição de cobertura é relaxada de modo que $\bigcup_{i=1}^p M_{(i)} \subseteq M$. Sob

o ponto de vista do universo das restrições, as exigências do particionamento englobam tanto as referentes a uma cobertura como as de um *packing* sobre o conjunto M . Podemos transformar um PP em um PK utilizando um vetor de custos conveniente. Billionnet (1978) demonstrou que o PP pode ser transformado em um PK da seguinte forma:

$$(PK_1) \text{ Maximizar } z = \sum_{j=1}^n c_j^{\nabla} x_j$$

sujeito a:

$$\sum_{j=1}^n a_{ij} x_j \leq 1 \quad i = 1, \dots, m$$

$$x_j \in \{0, 1\}, \forall j$$

$$\text{onde } c_j^{\nabla} = L \cdot \sum_{k=1}^m a_{kj} - c_j \text{ e } L > \sum_{j=1}^n c_j$$

A presente transformação pode ser útil em várias situações, especialmente devido à possibilidade de utilização da correspondência entre o politopo do *packing* com certos grafos (ver Balas e Zemel [1977]) para a geração de facetas do problema de particionamento (ver também Nemhauser e Sigismondi [1989] e Nemhauser e Trotter [1975]).

10.2.2 – Exemplos de Aplicações e Problemas Correlatos

✓ O Problema de Cobertura em Grafos

Seja um grafo $G = (N, A)$, $n = |N|$ nós e $q = |A|$ arestas. Seja $A_G = [a_{ij}]$ uma matriz de incidência $n \times q$ de G . Seja ainda e_n o vetor dos nós e e_q o vetor das arestas. Considere ainda $B_G = [b_{ij}]$ a matriz transposta de A_G . Exemplificaremos as diversas estruturas associadas ao problema de recobrimento no grafo da Figura 10.1.

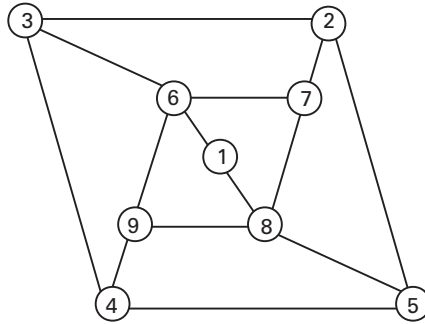


FIGURA 10.1 Grafo exemplo G_c .

- O *node covering* ou cobertura em nós (\hat{N}^*) é o problema de determinar o conjunto \hat{N} de nós, de mínima cardinalidade, $\hat{N} \subseteq N$, tal que toda a aresta em G é incidente a pelo menos um nó de \hat{N} . Formalmente:

$$(NC) \text{ Minimizar } z = \sum_{j=1}^n x_j$$

sujeito a:

$$\sum_{j=1}^m b_{ij} x_j \geq 1 \quad i = 1, \dots, n$$

$$x_j \in \{0, 1\}, \forall j$$

onde

$x_j \equiv$ variável binária que assume o valor 1 quando o nó j está na solução ($j \in \hat{N}$) e 0, caso contrário.

A Figura 10.2 apresenta uma cobertura em nós do grafo G_C , $\hat{N} = \{2, 4, 6, 8\}$. Denominaremos o valor ótimo de NC (cardinalidade da solução) de β_0 ou $\beta(G)$ ou *número de absorção*. No caso, $\beta(G) = 4$.

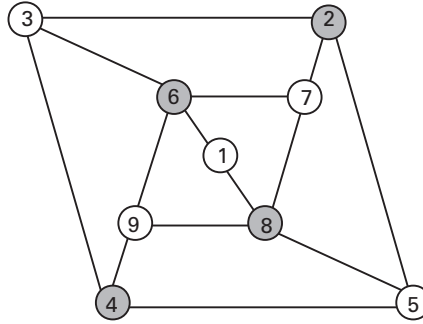


FIGURA 10.2 Uma cobertura em nós para o grafo G_C .

- O *edge covering* ou cobertura de arestas (\hat{A}^*) é o problema de determinar o conjunto \hat{A} de arestas, $\hat{A} \subseteq A$, de mínima cardinalidade, em que todo vértice de G é incidido por pelo menos uma aresta de \hat{A} . Formalmente:

$$(EC) \text{ Minimizar } z = \sum_{j=1}^m x_j$$

sujeito a:

$$\sum_{j=1}^n a_{ij} x_j \geq 1 \quad i = 1, \dots, m$$

$$x_j \in \{0, 1\}, \forall j$$

onde

$x_j \equiv$ variável binária que assume o valor 1 quando a aresta j está na solução ($j \in \hat{A}$) e 0, caso contrário;

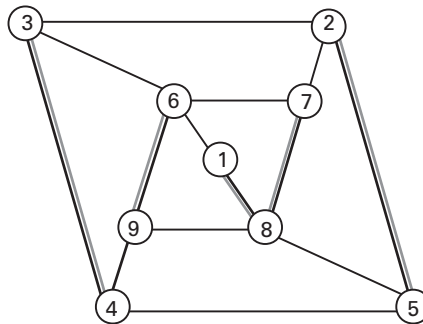
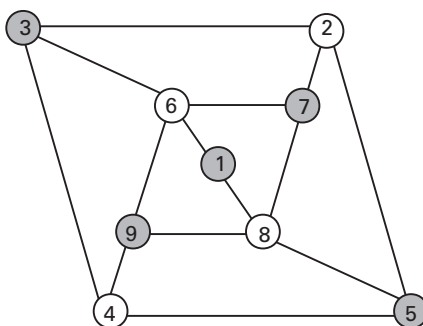


FIGURA 10.3 Uma cobertura em arestas para o grafo G_C .

A cobertura de arestas pode se transformar em um 1-*matching* máximo, bastando para isso remover $l - 1$ das $l > 1$ arestas que incidem sobre um mesmo nó. No caso da solução da Figura 10.3 basta retirar a aresta 1-8 ou a 7-8. Denominaremos o valor ótimo de EC (cardinalidade da solução) de β_1 ou $\beta_1(G)$. Outros dois problemas estão associados ao problema de cobertura em grafos a saber:

- $$\begin{aligned} \text{(NP) Maximizar } z &= \sum_{j=1}^n x_j \\ \text{sujeito a:} \\ \sum_{j=1}^m b_{ij}x_j &\leq 1 \quad i = 1, \dots, n \\ x_j &\in \{0, 1\}, \forall j \end{aligned}$$

A Figura 10.4 exemplifica um *node packing*, $\hat{T}^* = \{1, 3, 5, 7, 9\}$. Denominaremos o valor ótimo de NP (cardinalidade da solução) de $\alpha_0, \alpha(G)$, *número de estabilidade* ou *número de independência* (também denominada por alguns autores de estabilidade interna). Os subconjuntos de \hat{T}^* constituem conjuntos também denominados de *estabilidade*, *estáveis* ou *independentes*. No caso do exemplo da Figura 10.1, temos a solução de NP nos conduzindo à $\alpha(G) = 5$, conforme mostra a Figura 10.4.

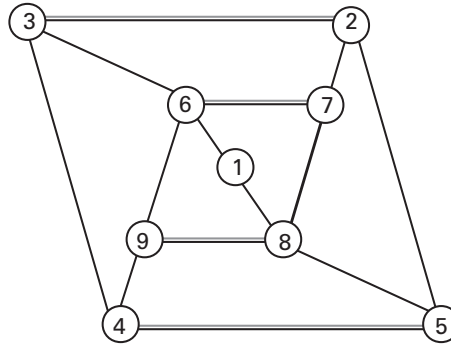


- O *edge matching* (\hat{M}^*) é o problema de determinar o conjunto \hat{M} de arestas, $\hat{M} \subseteq A$, de máxima cardinalidade, em que todo vértice de G é incidido por no máximo uma aresta de \hat{M} . Formalmente:

$$\begin{aligned} \text{(EM) Maximizar } z &= \sum_{j=1}^m x_j \\ \text{sujeito a:} \quad \sum_{j=1}^m a_{ij}x_j &\leq 1 \quad i = 1, \dots, m \\ x_j &\in \{0, 1\}, \forall j \end{aligned}$$

A Figura 10.5 mostra um 1-*matching* ótimo $\hat{M}^* = \{(3, 2); (4, 5); (7, 6); (8, 9)\}$. Denominaremos o valor ótimo do modelo EM (cardinalidade da solução) de α_1 ou $\alpha_1(G)$.

$$\alpha_0 + \beta_0 = n = \alpha_1 + \beta_1$$

FIGURA 10.5 Um Matching para o grafo G_C .

• Núcleo ou Kernel de um Grafo

Quando um conjunto de nós em G for simultaneamente uma cobertura e um conjunto estável diremos que ele é um *núcleo* do grafo G . Um grafo pode não ter um núcleo, como acontece no grafo da Figura 10.1, ou ter mais de um, como acontece no grafo da Figura 10.6. Nesse último caso, identificamos os núcleos 1-3-5 e 2-4-6.

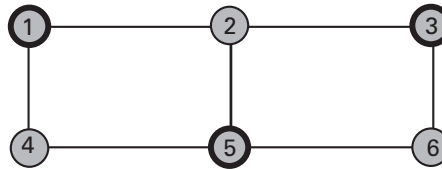


FIGURA 10.6 Um Kernel.

Os conceitos anteriores encontram um grande número de aplicações práticas. No contexto da localização de serviços de emergência, uma cobertura de nós representa uma localização de facilidades sobre os nós de G capaz de atender diretamente ou por vizinhança a todos os vértices do grafo.

✓ O Problema de Cobertura sobre o Contínuo

Em alguns casos o modelo do PR pode modelar importantes e complicadas situações de localização sobre o contínuo. Esse problema possui uma grande complexidade, especialmente face ao número de configurações viáveis (Mehrez e Stulman [1984] e Smallwood [1964]). Suponhamos que exista um conjunto de pontos em \mathbb{R}^2 de coordenadas (x_i, y_i) , $i = 1, \dots, n$, que devam ser atendidos em alguma demanda pela localização de P facilidades (pontos de suprimento). Suponhamos ainda que exista, associado a cada um desses pontos, uma certa distância euclidiana r_{ij} que represente um afastamento físico aceitável entre a localização do ponto de demanda i e o ponto de suprimento j . Define-se a variável x_{ij} binária da seguinte forma:

$x_{ij} = 1$ se $r_{ij} \geq d_{ij}$, onde d_{ij} representa a distância entre a localização do ponto j ao ponto i , e 0 para o caso em que $r_{ij} < d_{ij}$.

Seja ainda $K(\cdot)$ uma função cobertura não decrescente para o problema específico de modo que $K(\cdot)$ não possa decrescer se novos pontos são cobertos. Tem-se, em consequência, que a função objetivo da cobertura máxima para o problema no plano \mathbb{R}^2 pode ser expressa da seguinte forma:

$$(CCI) \text{ Maximizar } z = \sum_{i=1}^n K\left(\sum_{j=1}^p x_{ij}\right)$$

A fim de incorporar as restrições do problema consideraremos $S = \{1, 2, \dots, s\}$ o conjunto dos possíveis pontos candidatos a serem ocupados pela localização de uma posição de suprimento no plano \mathbb{R}^2 e:

$a_{ij} \equiv$ escalar que será igual a 1 se a distância $d_{ij} \leq r_{ij}$ e 0, caso contrário.

$y_j \equiv$ variável binária que será igual a 1 se o ponto j for ocupado por uma localização de uma facilidade e 0, caso contrário.

Nesse caso, podemos reescrever a função objetivo de CC1 e incorporar as restrições de localização no plano, constituindo o modelo CC2 que se segue:

$$(CC2) \text{ Maximizar } z = \sum_{i=1}^n K \left(\sum_{j \in S} a_{ij} y_j \right)$$

sujeito a:

$$\sum_{j \in S} y_j = P$$

$$y_j \in \{0, 1\}, \forall j$$

A exibição da matriz $A = [a_{ij}]$ não é uma tarefa trivial, em virtude do número infinito de possíveis posições para os pontos j , sendo essa dificuldade discutida por White e Case (1980). Podemos, contudo, através de um raciocínio simples, transformar o CC2 em um problema de recobrimento não ponderado, *minimum cardinality set covering problem*, (MCSCP) da seguinte forma:

$M = \{1, \dots, m\}$ representando o conjunto dos pontos a serem atendidos.

$N = \{1, \dots, n\}$ representando o conjunto de posições viáveis de alocação das facilidades.

A matriz $A = [a_{ij}]$, em que $a_{ij} = 1$ se o ponto i puder ser atendido pela facilidade localizada em j e 0, caso contrário.

$x_j \equiv$ variável binária que será 1 se o ponto j for escolhido para receber uma alocação e 0, caso contrário.

A localização de uma facilidade no ponto j permitirá o atendimento de um subconjunto dos pontos de demanda que denominaremos a_j . Esses pontos serão aqueles que possuem uma distância $d_{ij} \leq r_{ij}$. Os pontos j pertencentes ao conjunto $N = \{1, \dots, n\}$ poderão ser determinados de forma precisa se considerarmos que ocorrerão em direção aos limites da distância de operação, bem como em regiões que autoricem pelo menos o atendimento simultâneo a mais de uma facilidade. Se tal raciocínio não fosse válido, a solução de CC2 seria trivial com uma facilidade associada a cada demanda. Nesse caso, o problema CC2 pode ser reescrito com mostra o modelo (PRM – problema de recobrimento mínimo):

$$(PRM) \text{ Minimizar } z = \sum_{j=1}^n x_j$$

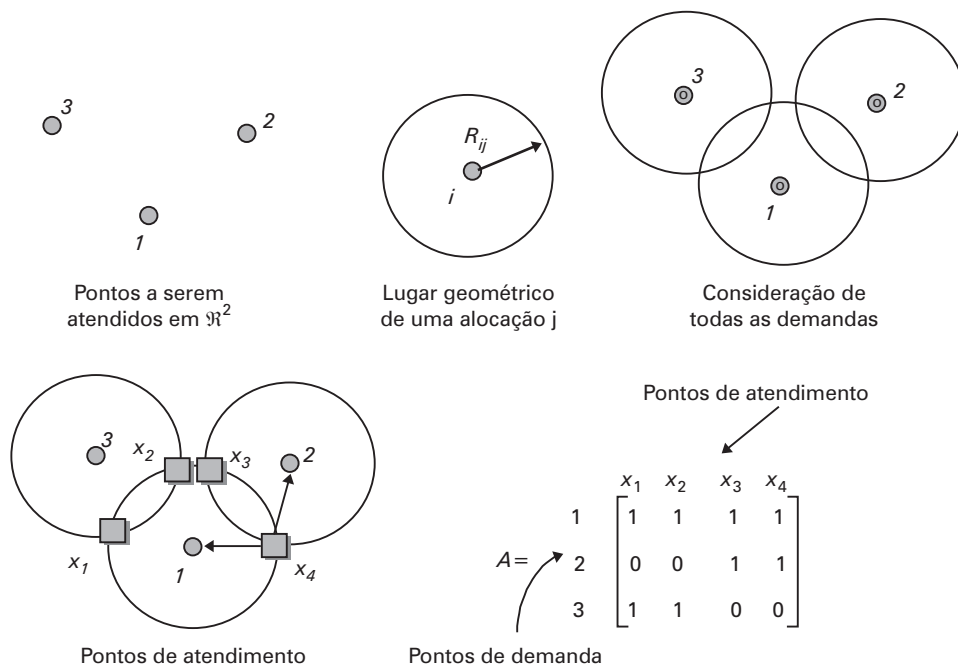
sujeito a:

$$\sum_{j=1}^n a_{ij} x_j \geq 1 \quad i = 1, \dots, m$$

$$x_j \in \{0, 1\}, \forall j$$

Tomando-se o cuidado de obter a matriz A da forma exemplificada na Figura 10.7, com os pontos de localização das facilidades discretizados nas interseções dos círculos de afastamento máximo.

Uma topologia semelhante a anterior utilizando, todavia, um lugar geométrico retangular (em substituição aos círculos) é sugerida por Benveniste (1982). A formulação PRM pode ser aplicada a vários problemas de localização sobre o contínuo. Um exemplo é o problema da localização de postes de iluminação pública, em que os pontos de demanda são os locais em que se deseja providenciar iluminação ou mesmo uma malha desses pontos. O problema de localização de radares de vigilância, a alocação arma \times alvo e a exploração de petróleo em poços submarinos são outros casos que podem ser for-

FIGURA 10.7 Obtenção da matriz A .

mulados dessa maneira (Goldbarg [1987]). Considerando que exista a possibilidade prática de discretizar a demanda, o modelo em pauta pode ser utilizado em outras situações correlatas como a de localização de estações de metrô e a evacuação emergencial em áreas ameaçadas por catástrofes. A Figura 10.8 (a) mostra o processo discretização da demanda de um conjunto de quarteirões para o acesso às estações de metrô. Na Figura 10.8 (b) são associadas as restrições de deslocamento máximo previsto para os usuários até um ponto de acesso à estação (no exemplo nem todos os pontos foram considerados apenas para não sobrecarregar o desenho). A Figura 10.8 (c) mostra uma solução possível para as entradas da estação pesquisada.

• Exploração de Petróleo em Campo Submarino – um Estudo de Caso

Uma das mais interessantes aplicações da formulação PRM pode ser encontrada na otimização da configuração de exploração de um campo submarino de petróleo. Nesse caso, a discretização de alcance é bastante adequada tanto para tratar a necessidade de localização sobre o contínuo, como pelo número de variáveis envolvidas (Vibrantovski *et al.* [1990]). Por outro lado, os enormes custos envolvidos para a construção e operação de sistemas submarinos de exploração de petróleo tornam o processo de otimização tarefa imprescindível.

Definição do Problema: o petróleo é extraído da rocha matriz de um campo submarino através de poços perfurados no solo. Esses poços são dispostos de modo a otimizar o volume de óleo extraído, podendo ser perfurados de forma perpendicular ou oblíqua em relação ao solo. No primeiro caso, o poço é denominado vertical e no segundo, direcional. O local do solo submarino em que a perfuração é iniciada chama-se cabeça do poço, e o local de maior profundidade, no qual a exploração é propriamente realizada recebe o nome de ponto objetivo. O alcance do poço é a distância entre as projeções da cabeça e do ponto objetivo no plano definido pela superfície marítima. Quando o poço é vertical o alcance é nulo, porém para poços direcionais o alcance é uma medida relevante devido à necessidade de atender-se a um padrão de comprimento máximo para o alcance de um poço.

Todo óleo extraído dos poços é enviado para a unidade estacionária de produção ou plataforma (UEP). Face a condicionantes operacionais, o número de ligações diretas entre os poços e a UEP é limitado. O controle do número de ligações diretas a UEP é realizado por *manifolds* que normalmente também são localizados sobre o solo submarino e funcionam como concentradores de óleo.

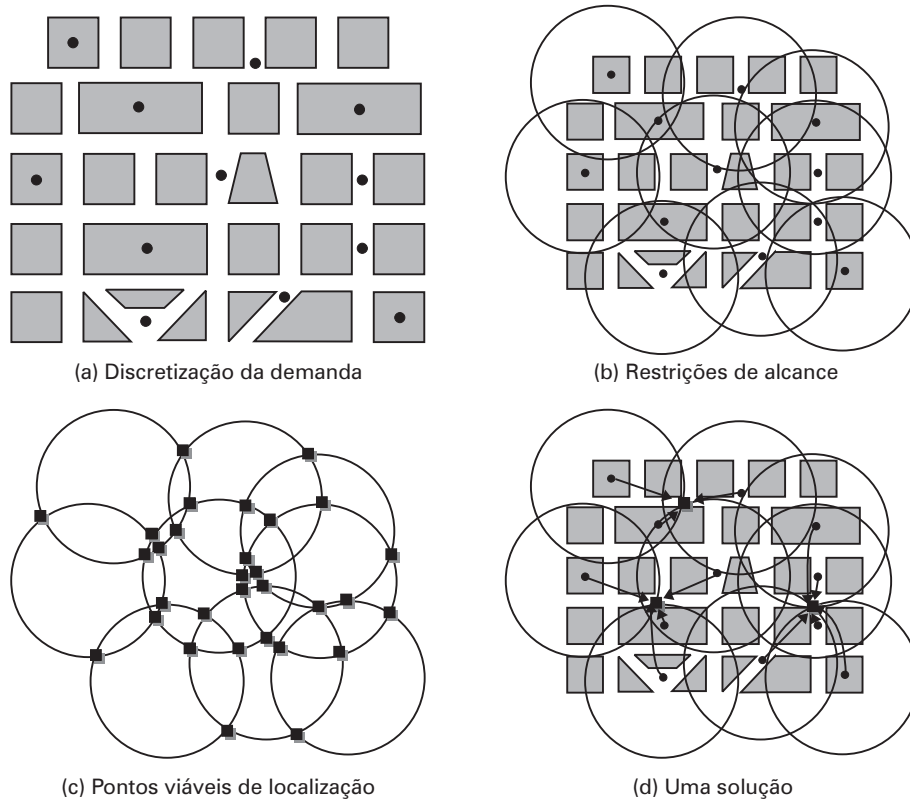


FIGURA 10.8 O caso da localização das entradas de uma estação de metrô.

De forma semelhante a UEP, os *manifolds* também são limitados em sua capacidade de interligação com os poços. Por outro lado, em virtude da necessidade de manutenção da viabilidade econômica, os *manifolds* devem operar com um número mínimo de ligações. A ligação entre poços e *manifolds* é de conveniência, de forma que um poço pode ser ligado diretamente a UEP quando isso for economicamente indicado. As interligações entre poços e *manifolds*, *manifolds* e UEP, e poços e UEP são denominadas *bundles*.

A configuração do campo é influenciada por um grande número de fatores, dentre os quais destacamos, por sua significância:

- Coordenadas dos objetivos dos poços e suas profundidades.
- Perfil de lâmina d'água do campo.
- Número máximo de poços que cada *manifold* submarino suporta.
- Número mínimo de poços por *manifold*, que corresponde ao grau de utilização de cada *manifold* pretendido no projeto.
- Tipos de plataformas disponíveis e suas lâminas d'água máximas de operação.
- Número máximo de interligações provenientes de *manifolds* ou poços (*bundles*), que podem ser interligados à unidade estacionária de produção.
- Alcance máximo permitido para poços direcionais.
- Custos de equipamentos e serviços:
 - Perfuração.
 - *Manifold* submarino.

- *Bundle* de ligação entre *manifold* e plataforma.
- *Bundle* de ligação entre poço e *manifold* ou plataforma.

Solucionar o PCCS significa determinar os seguintes elementos de forma a minimizar os custos de implantação:

- Coordenadas do posicionamento da UEP e plataforma utilizada.
- Coordenadas do posicionamento do(s) *manifold*(s) submarino(s), com os poços a ele(s) associados.
- Coordenadas das cabeças dos poços.

Observe que outros partidos de solução para o PCCS são tecnicamente possíveis, especialmente para a exploração em águas profundas, de modo que seja alterada a necessidade, por exemplo, de que o petróleo seja conduzido até a plataforma flutuante, ou mesmo que a UEP esteja localizada na superfície do mar. Mesmo nesses casos, contudo, a abordagem aqui descrita poderá ser utilizada na definição dos *manifolds*.

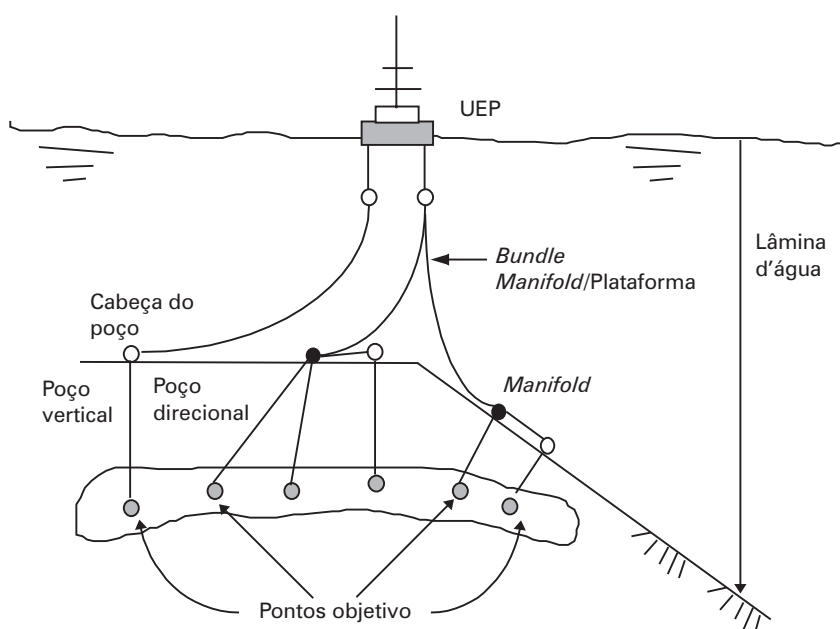


FIGURA 10.9 Vista de perfil de um campo submarino.

Formulação matemática do PCCS – O PCCS1

Alguns dos fatores anteriormente enumerados, apesar de eventualmente significativos, são de difícil quantificação e sujeitos a um comportamento incerto ou variável ao longo do tempo. Por outro lado, pensando-se apenas na alocação de *manifolds*, em um campo composto por m poços e, mesmo de posse de um critério que reduza a localização sobre os infinitos pontos do plano a n localizações viáveis para os *manifolds*, o número de soluções viáveis a examinar seria n^m . Levando-se em conta que o problema completo exige uma consideração combinatória adicional para a localização das cabeças dos poços e UEP, parece claro a impossibilidade do tratamento enumerativo de problemas reais de grande porte (onde, no mínimo, $m > 100$ e $n > 50$). O método heurístico proposto divide o problema em três etapas, a saber:

- A primeira etapa objetiva estabelecer o número de *manifolds* que existirão no campo, suas localizações e o conjunto de poços a eles interligados.

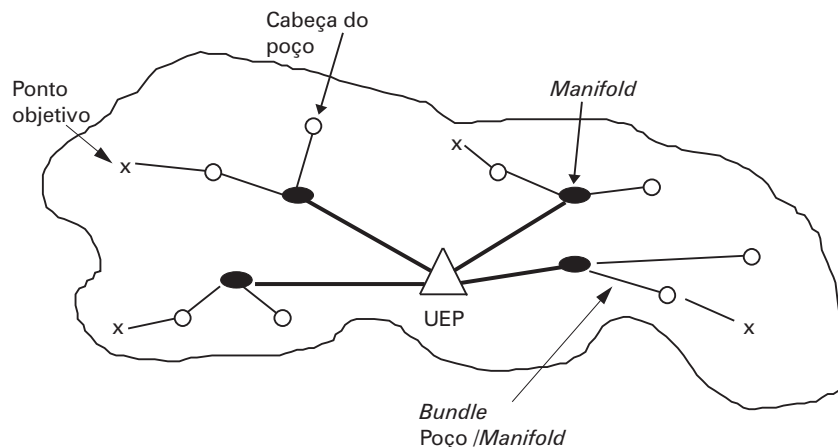


FIGURA 10.10 Planta esquemática de um campo submarino.

- A segunda etapa visa a localizar definitivamente a cabeça dos poços.
- A terceira etapa objetiva localizar a unidade estacionária de produção.

Definindo-se um raio efetivo de operação R para cada poço $r \in M = \{1, \dots, m\}$, como sendo a maior distância admissível entre o poço e o manifold ao qual é ligado, pode-se estabelecer um interessante procedimento de discretização para PCCS1 capaz de reduzir o infinito conjunto das localizações viáveis dos manifolds a um conjunto com no máximo $O(m^2)$ componentes da seguinte forma: levando-se em conta os raios efetivos de operação determinam-se, no plano, pontos de localização para os manifolds $N = \{1, \dots, n\}$ nas interseções dos círculos centrados nos poços, cujos raios equivalem aos seus raios de operação. Esses pontos de interseção permitem a interligação entre, no mínimo, dois poços e um manifold; contudo, podem interligar outros poços cujos círculos de operação os contenham. A utilização do conceito de raios efetivos de operação ou raios de sensibilidade introduz uma nova abordagem para a formulação de problemas de localização sobre o plano. No caso do PCCS a solução obtida no plano poderá ser estendida para o perfil do fundo do mar, considerando-se sua projeção topográfica. É importante observar que a localização dos manifolds, bem como sua designação às cabeças dos poços, é realizada na primeira fase sem que a posição definitiva das cabeças dos poços tenha ainda sido perfeitamente determinada. Considera-se, em princípio, todos os poços verticais, de modo que sejam coincidentes, em planta, os pontos objetivos e as cabeças dos poços. Assim sendo, uma configuração de exploração de um campo submarino de petróleo é reconhecida como admissível quando todos os poços $r \in M$, estiverem cobertos por manifolds alocados sobre os pontos viáveis $i \in N$, obtidos pelas interseções dos círculos. Considerando-se as variáveis 0-1, os custos e os coeficientes são definidos a seguir:

- Variável de decisão x_j associada à localização de um manifold sobre um ponto viável. x_j será igual a 1 se um manifold é alocado em um ponto j , obtido pela interseção de dois círculos e 0, caso contrário.
- Matriz de cobertura $A = [a_{rj}]$. a_{rj} será igual a 1 se o poço r pode ser ligado ao manifold localizado no ponto j , e 0, caso contrário.
- c_j o custo de alocar um manifold em j acrescido do custo de interligá-lo a todos os poços que podem ser por ele atendidos.

A matriz de cobertura $A = [a_{rj}]$ de dimensão $m \times n$ deve ser obtida para cada problema, em função da localização das cabeças dos poços. Essa matriz representa a capacidade de cada ponto viável cobrir mais de um poço simultaneamente. Cada coluna da matriz representa um possível ponto viável sobre o qual será alocado um manifold. Os custos c_j de cada alocação j podem agora ser calculados com muito

boa precisão, levando-se em conta todos os fatores julgados importantes para a posição específica. Dessa forma, tem-se o seguinte modelo:

$$\begin{aligned}
 \text{(PCCS1) Minimizar } z &= \sum_{j=1}^n c_j x_j \\
 \text{sujeito a:} \\
 \sum_{j=1}^n a_{rj} x_j &\geq 1 \quad r = 1, 2, \dots, m \\
 x_j &\in \{0, 1\}, j = 1, 2, \dots, n
 \end{aligned}$$

A restrição de cobertura garante que todos os poços sejam ligados a pelo menos um *manifold*. Como no caso real cada poço deve ser ligado a apenas um *manifold*, a solução de PCCS1 deverá ser revista, eliminando-se sobrecoberturas (Leal [1993]). A solução revista será uma solução de particionamento de custo inferior ao obtido em PCCS1.

A Obtenção da Matriz A

Vamos exemplificar através deste problema a obtenção da matriz *A* para os demais casos de alocação sobre o contínuo. A matriz de cobertura de PCCS1 é obtida a partir da geração de pontos viáveis de localização para os *manifolds* com o auxílio da utilização dos círculos de operação. Cabe, contudo, considerar as demais restrições do problema sob pena da solução de PCCS1 não ser viável para o PCCS. Felizmente, a consideração dessas restrições pode ser realizada em tempo polinomial sobre *N*, na ocasião da organização da matriz *A*, de acordo com as seguintes regras:

- As restrições relativas ao número máximo e mínimo de poços servidos por um *manifold* são garantidas através de:
 1. Considerar todas as colunas nas quais o número de poços cobertos é maior que o limite inferior e menor que o limite superior para a capacidade do *manifold*. Considerar todas as colunas nas quais o número de poços cobertos viola o limite superior em *S* unidades, através da eliminação dos *S* poços mais distantes do ponto de alocação.
 2. Considerar todas as colunas nas quais o número de poços cobertos viola o limite inferior em *R* unidades, através da inclusão dos *R* poços mais próximos do ponto de alocação. As restrições relativas à lâmina d'água são atendidas pela eliminação das colunas associadas a essas violações.
- Restrições de topografia, acesso ou outras, podem ser consideradas semelhantemente à lâmina d'água. Uma vez alterada a configuração de uma coluna, o custo a ela associado deverá ser recalculado diretamente dos dados disponíveis. Caso ocorra a existência de poços não cobertos por *manifold*, ou seja, linhas da matriz *A* não cobertas por alguma coluna, e não for possível designá-los aos *manifolds* mais próximos, essas linhas serão eliminadas da matriz. Esse caso representa a situação da ligação direta entre um poço e a plataforma.
- A designação de um poço não atendido por algum *manifold* pode ser realizada conforme a seguinte regra:

Procuram-se os *k* pontos viáveis para a alocação de *manifolds* mais próximos do poço. Estende-se, então, o raio de operação do poço de maneira a possibilitar a sua ligação a todos aqueles dentre os *k* pontos que não estejam já saturados atendendo a um número de poços equivalente ao limite máximo de ligações possíveis para os *manifolds*. Dessa forma, teremos $a_{rj} = 1$, onde *j* equivale às colunas correspondentes. Todavia, resta ainda a possibilidade de as *k* colunas estarem saturadas. Neste caso, a linha é eliminada da matriz e o poço é ligado diretamente à UEP. Com essa abordagem relaxa-se a segunda hipótese feita no desenvolvimento da formulação do PCCS1, que não admitia *bundles* de ligação do poço à UEP.

• Além das reduções de linhas e colunas necessárias à aplicação das restrições impostas ao problema, outras reduções válidas para o problema de recobrimento podem ser utilizadas com o propósito de simplificar o problema. Mais detalhes podem ser encontrados em Fampa (1992).

Após o pré-processamento e a solução de PCCS1, a localização dos *manifolds* do campo estará definida, bem como o conjunto de poços a eles interligados. O custo de cada coluna é dado por um custo fixo c_m da instalação do *manifold* acrescido de

$$c_{bundle/poço} \times \sum_{r \in P_i} d_{ri}$$

onde $c_{bundle/poço}$ é o custo do *bundle* de ligação do poço ao *manifold*, sendo d_{ri} é a distância euclidiana entre o ponto objetivo i e a cabeça do poço r , e P_j é o conjunto de poços atendidos pela alocação de um *manifold* no ponto viável j .

O PCCS2

No desenvolvimento da segunda etapa de resolução, objetiva-se estabelecer o melhor posicionamento para as cabeças dos poços. Para tal, a hipótese que estabelecia todos os poços como poços verticais é relaxada, admitindo-se então a existência de poços direcionais, tais que suas cabeças localizam-se em cima da linha que une o objetivo do poço e o *manifold* ao qual se interliga, respeitando, contudo, o limite imposto pelo alcance máximo permitido. Evidencia-se que tal admissão é bastante razoável, já que a localização da cabeça do poço fora da linha citada implica custo de implantação obviamente maior pelo aumento do comprimento do *bundle* de ligação e maior complexidade na geometria de perfuração. Assim, o objetivo da segunda etapa é determinar, dentre os pontos viáveis de alocação das cabeças dos poços, aqueles que conduzem a um menor custo, representado pelo custo de perfuração do terreno acrescido do custo do *bundle* de ligação do poço ao *manifold*.

Essa etapa é solucionada através de um problema de programação não linear unidimensional que consiste em determinar $x_{cr}, y_{cr}, z_{cr}, r = 1, 2, \dots, m$, coordenadas definitivas das cabeças dos poços (ver detalhes em Fampa [1992]), de forma a minimizar a soma do custo de perfuração e do *bundle* de ligação entre o poço e o *manifold*.

Para uma localização conhecida do *manifold*, a localização ótima da cabeça do poço independe da localização dos demais poços. A região viável para a alocação da cabeça do poço compreende a linha que parte da projeção do seu objetivo no solo em direção ao *manifold* que o atende, sempre acompanhando a topologia do terreno. O término da linha que representa os pontos candidatos ao posicionamento da cabeça será então estabelecido pelo alcance máximo para poços direcionais ou pela distância entre a cabeça do poço e o *manifold*, caso seja menor que o alcance máximo, conforme mostra a Figura 10.11.

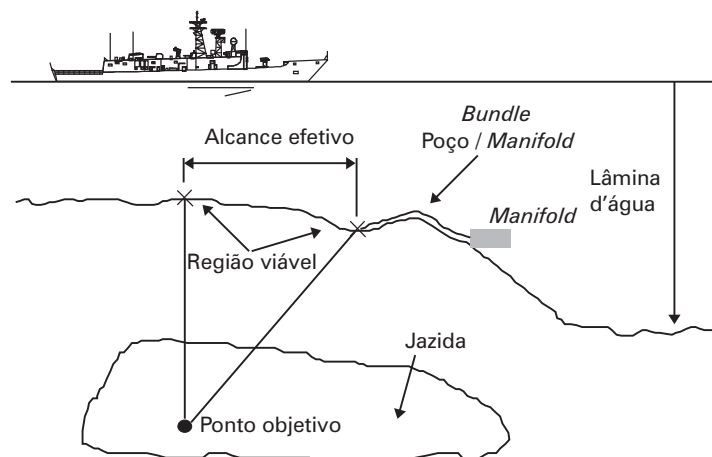


FIGURA 10.11 Região viável para alocação das cabeças dos poços.

✓ O Problema de Maximal Covering Location Planning (MCLP)

Este modelo é uma variante do problema de localização e recobrimento sugerido por Church e ReVelle (1974) e atende a uma classe de problemas bastante próxima a PRM. O modelo foi desenvolvido em evolução ao modelo de Toregas e ReVelle (1972) que, sendo um modelo possuidor de restrições de recobrimento clássicas, objetivava minimizar o número de facilidades necessárias à satisfação da demanda, respeitadas, como no modelo PRM, indiretamente, as restrições de distância.

Church e ReVelle (1974) formularam o MCLP como:

$$\begin{aligned}
 \text{(MCLP1) Maximizar } z &= \sum_{i=1}^m d_i y_i \\
 \text{sujeito a:} \\
 \sum_{j=1}^n a_{ij} x_j - y_i &\geq 0, & i = 1, \dots, m \\
 \sum_{j=1}^n x_j &= p \\
 x_j, y_i &\in \{0, 1\}, i = 1, \dots, m & j = 1, \dots, n
 \end{aligned}$$

onde:

$y_i \equiv$ variável binária que assume o valor 1 se o nó i é coberto e 0, caso contrário.

$x_j \equiv$ variável binária que assume valor 1 se uma facilidade é localizada no vértice j e 0, caso contrário.

$a_{ij} = 1$ se a demanda i vai ser coberta pela facilidade alocada em j e 0, caso contrário;

$d_i \equiv$ demanda no vértice i .

$p \equiv$ número de facilidades a localizar.

Church e Toregas formularam o MCLP considerando implicitamente configurações de atendimento da seguinte forma:

$$\begin{aligned}
 \text{(MCLP2) Minimizar } z &= \sum_{j=1}^n x_j \\
 \text{sujeito a:} \\
 \sum_{j \in N_i} x_j &\geq 1 & i = 1, \dots, n \\
 x_j &\in \{0, 1\}, \forall j
 \end{aligned}$$

onde:

$x_j \equiv$ variável binária que é igual a 1 quando o nó j é ocupado por uma facilidade e 0, caso contrário.

$N_i = \{j \mid d_{ij} \leq R\}$, sendo R a distância máxima prevista para o alcance do serviço a partir de seu ponto de localização.

O modelo MCLP possui uma ligação estrita com os modelos de p -mediana (Khumawala [1973]).

Objetivando possibilitar a consideração de uma restrição associada ao número máximo de facilidades passíveis de alocação, restrição muitas vezes necessária na prática devido a limitações orçamentárias, e maximizar o atendimento de uma demanda por facilidades dentro de um alcance máximo desse serviço, Chung e ReVelle (1986) reformularam o MCLP como se segue:

$$\begin{aligned}
 \text{(MCLP3) Maximizar } z &= \sum_{i \in I} a_i y_i \\
 \text{sujeito a:} \\
 \sum_{i \in N_i} x_j &\geq y_i \quad i \in I \\
 \sum_{i \in J} x_j &= p \\
 x_j, y_j &\in \{0, 1\}, \forall_j
 \end{aligned}$$

onde:

$y_i \equiv$ variável binária que assume o valor 1 se o nó i é coberto e 0, caso contrário.

$I = \{1, \dots, m\}$ é o conjunto dos nós de demanda.

$J = \{1, \dots, n\}$ é o conjunto dos nós que recebem a alocação de uma facilidade.

x_j e N_i : como anteriormente definidos.

$a_i \equiv$ demanda no vértice i .

$p \equiv$ o número de vértices em que residem as facilidades.

Esse problema possui várias aplicações na área de vendas. Normalmente, sua solução estará associada ao melhor investimento em redes de distribuição e representantes de produtos, postos de vendas, além de oficinas autorizadas e serviços públicos ou de emergência. Chung e ReVelle (1986) sugerem uma aplicação para a área de fiscalização do governo. Almiñana e Pastor (1997) sugerem uma interessante heurística (da classe *surrogate heuristics*) para o problema, e Galvão e ReVelle (1996) uma outra heurística baseada na relaxação lagrangeana do problema. Uma das heurísticas mais difundidas para o problema denomina-se *greedy adding algorithm*. Esse algoritmo inicia sem qualquer facilidade localizada e vai, a cada um dos p estágios, atribuindo uma facilidade ao nó que conduz ao maior crescimento na cobertura de atendimento.

✓ O Problema de Localização de Concentradores (PLC)

O projeto de redes de comunicação de dados eficientes é hoje um dos problemas mais importantes na área computacional. Esse modelo é capaz de representar várias instâncias do problema de arquitetura de redes centralizadas. O modelo considera a existência de três níveis na estrutura hierárquica de uma rede de comunicações:

- Terminais, $T_i, i \in N, N = \{1, \dots, n\}$, normalmente idênticos, dispersos geograficamente e de localização conhecida.
- Concentradores, $C_j, j \in M, M = \{1, \dots, m\}$, unidades de porte e capacidade superior aos terminais, podendo ou não estar localizadas junto aos terminais, e cujo número e localização, normalmente, é desconhecido.
- Unidade central, C_0 , equipamento de grande porte quando comparado com os demais elementos e de localização normalmente conhecida.

O problema colocado consiste em responder a cinco questões:

1. Quantos concentradores serão necessários?
2. Onde eles deverão estar localizados?
3. De que forma os concentradores estarão ligados à central?
4. Quais terminais serão designados a quais concentradores?
5. De que maneira os terminais estarão conectados aos concentradores a eles designados?

A prática adotada nos projetos sugere que os terminais sejam ligados diretamente aos seus concentradores e os concentradores diretamente à central. Existe igualmente uma restrição de capacidade dos concentradores, de modo que não é aconselhável que o concentrador C_i não receba mais que k_i terminais para gerenciar. Considerando como d_j o custo de implantação de uma unidade concentradora no ponto j e c_{ij} o custo de ligar um terminal T_i ao concentrador C_j , o problema consiste em encontrar a rede que minimize os custos de implantação. Podemos formular o PLC como se segue:

$$(PLC) \text{ Minimizar } z = \sum_{i=1}^n \sum_{j=0}^m c_{ij} x_{ij} + \sum_{j=1}^m d_j y_j$$

sujeito a:

$$\sum_{j=0}^m x_{ij} = 1 \quad i = 1, \dots, n$$

$$\sum_{i=1}^n x_{ij} \leq k_j y_j \quad j = 1, \dots, m$$

$$x_{ij}, y_j \in \{0, 1\}, \forall i, j$$

onde:

y_i é uma variável binária que assume o valor 1 se o nó i é escolhido para concentrador e 0, caso contrário.

x_{ij} é uma variável binária que assume o valor 1 se o terminal T_i está ligado ao concentrador C_j e 0, caso contrário.

O PLC, é um problema de particionamento (restrições em igualdade) sujeito a uma restrição de cardinalidade.

Os problemas anteriormente descritos possuem aplicações práticas que vão desde os projetos de redes de comunicações até localização de serviços de emergência e de segurança, bem como o auxílio à logística.

✓ Escala de Tripulações Aéreas (Airline Crew Scheduling Problem – ASP)

O problema ASP consiste em designar n tripulações a um conjunto de m voos, minimizando-se o custo total de operação. O problema pode ser sujeito a restrições geográficas, de tempo, de habilitação das tripulações, de número de pousos e decolagens para uma tripulação etc. Recentemente esse problema vem sendo preferencialmente formulado como um PP (Vance [1997]) da seguinte forma:

$$(ASP) \text{ Minimizar } z = \sum_{p \in P} c_p y_p$$

sujeito a:

$$\sum_{p: i \in P} y_p = 1, \quad i \in F$$

$$y_p \in \{0, 1\}, \quad p \in P$$

onde F é o conjunto dos segmentos de voo e P o conjunto de todas as configurações de atendimento consideradas. y_p é uma variável binária assumindo o valor igual a 1 se a configuração de tripulação $p \in P$ está na solução e 0, caso contrário. A coluna p possui uma entrada igual a 1 na linha i se o voo i está atendido pela configuração p e c_p é o custo da configuração p . Para que possamos utilizar essa formulação cumprir gerar configurações viáveis para o problema. A dificuldade dessa enumeração é debatida por Graves *et al.* 1993. O ASP é um caso particular e muito importante do problema de roteamento de veículos (com programação de atividades) que foi abordado no Capítulo 9. Com uma formulação semelhante é possível tratar a programação da escala de outros tipos de tripulação como no caso dos trens.

✓ O tour de Cobertura ou Covering tour Problem (CTP)

Trata-se de um interessante problema que permite, por sua formulação, uma clara visão da interdependência entre os problemas de programação inteira e da teoria dos grafos. Esse problema pode ser estudado também juntamente com problema do caixeiro viajante (PCV). Seja $G = (V \cup W, E)$ um grafo não direcionado em que $V \cup W$ compõem o conjunto de vértices e $E = \{(v_i, v_j): v_i, v_j \in (V \cup W), i < j\}$ o conjunto de arestas. Podemos considerar que v_1 seria um depósito e V o conjunto de vértices que podem ser visitados a partir de v_1 . Denominaremos por $T, T \subseteq V$ um conjunto de vértices que devem ser visitados ($v_1 \in T$) e W um conjunto que deve ser coberto. É definida uma matriz C sobre E que satisfaz à condição de desigualdade triangular. O CTP consiste em determinar um ciclo hamiltoniano de comprimento mínimo sobre o conjunto V de forma que o tour formado contenha todos os vértices de T e todo vértice de W esteja coberto pelo tour, ou seja, todo vértice de W está a uma distância máxima c de algum vértice do tour. Considere o grafo da Figura 10.12 (a) com todos os arcos de peso 1, independentemente de seu sentido. Considerando a distância c também igual a 1, $v_1 = 3$, $W = \{2, 4, 8, 14\}$, e $T = \{3, 5, 6, 7, 8, 10, 13\}$ uma solução do CTP está apresentada na Figura 10.12 (b).

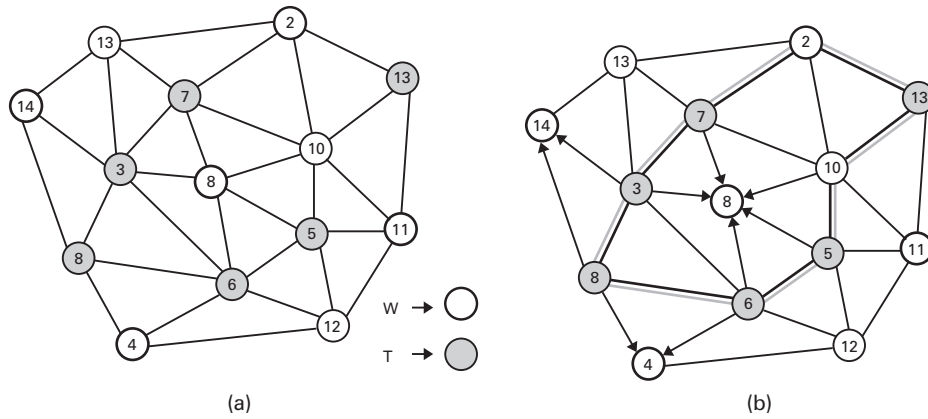


FIGURA 10.12 Solução do CTP.

Podemos formular CTP como um problema de programação inteira (ver Gendreau *et al.* [1997]) da seguinte forma:

$$(CTP) \text{ Minimizar } z = \sum_{i < j} c_{ij} x_{ij}$$

sujeito a:

$$\sum_{v_k \in S_l} y_k \geq 1, \quad v_1 \in V \quad (10.1)$$

$$\sum_{i < k} x_{ik} + \sum_{j > k} x_{kj} = 2y_k \quad v_k \in V \quad (10.2)$$

$$\sum_{\substack{v_i \in S, v_j \in V \setminus S \\ \text{ou } v_j \in S, v_i \in V \setminus S}} x_{ij} \geq 2y_t \quad S \subset V, 2 \leq |S| \leq n-2, T \setminus S \neq \emptyset, v_t \in S \quad (10.3)$$

$$x_{ij} \in \{0, 1\} \quad 1 \leq i < j \leq n \quad (10.4)$$

$$y_k = 1 \quad v_k \in T \quad (10.5)$$

$$y_k \in \{0, 1\}, \quad v_k \in V \setminus T \quad (10.6)$$

onde $y_k = 1$ se e somente se v_k pertence ao tour. Para $v_i, v_j \in V$, e $i < j$, $x_{ij} = 1$ se e somente se a aresta $(v_i, v_j) \in E$ está no tour, e 0, caso contrário. Ainda $\delta_{lk} = 1$ se e somente se $v_k \in V$ e é definido $S_1 = \{v_k \in V \mid \delta_{lk} = 1\}$ para todo $v_l \in W$.

A restrição do tipo (10.1) assegura que cada vértice de W é coberto pelo tour. A restrição (10.2) garante a continuidade do tour (grau dois para cada vértice do tour). A restrição do tipo (10.3) representa as condições de continuidade ou eliminação de subtours. As demais restrições representam as exigências de integralidade. Esse problema foi solucionado por Gendreau *et al.* (1997), por um algoritmo *branch-and-cut* e por um procedimento heurístico. Foram solucionadas de forma ótima instâncias com 600 vértices.

✓ Problemas de Cobertura Multiobjetivo

Dentre os problemas de cobertura multiobjetivo o mais comum é aquele que visa a equilibrar os custos para o estabelecimento da cobertura de facilidades e o atendimento da demanda. Podemos exemplificar uma cobertura multiobjetivo através de um modelo bicritério de recobrimento da seguinte forma:

$$\begin{aligned}
 \text{(PRB) Maximizar } z &= \sum_{i=1}^m d_i y_i \\
 \text{Minimizar } w &= \sum_{j=1}^n c_j x_j \\
 \text{sujeito a:} \\
 \sum_{j=1}^n a_{ij} x_j - y_i &\geq 0, \quad i = 1, \dots, m \\
 x_j, y_i &\in \{0, 1\}, \quad i = 1, \dots, m \quad j = 1, \dots, n
 \end{aligned}$$

onde:

$y_i \equiv$ variável binária que assume o valor 1 se o nó i é coberto e 0, caso contrário.

$x_j \equiv$ variável binária que assume o valor 1 se uma facilidade é localizada no vértice j e 0, caso contrário.

$a_{ij} = 1$ se a demanda i vai ser coberta pela facilidade alocada em j e 0, caso contrário.

$c_j \equiv$ custo de localizar uma facilidade no vértice j .

O modelo bicritério anterior possui uma grande semelhança com a formulação MCPL1. Esse problema e suas soluções são abordados por Church e Davis (1992).

✓ Problemas de Coberturas Capacitadas

Esses problemas correspondem a uma série de situações híbridas em que a tomada de decisão envolve os conceitos de cobertura e localização.

Storbeck (1988) apresenta o seguinte modelo denominado *capacitated maximal covering location model* (CMCLP), ou problema de localização em cobertura capacitada, para casos em que as facilidades são capazes, de alguma forma, de atender aos pedidos que chegam e não violam as restrições de capacidade. Pirkul e Schilling (1989) estudam dois modelos similares ao PRB, desenvolvendo um algoritmo baseado em relaxação lagrangeana e otimização pelo método dos subgradientes.

✓ O Problema da Cobertura Maximal em Caminho mais Curto (PCMCC)

Essa classe de problemas aborda roteamento com a inclusão de aspectos de cobertura. A principal diferença entre o contexto do PCMCC e o do CTP é a consideração multiobjetivo. As aplicações envolvem:

- A definição de traçados de redes de energia elétrica que possam atender a áreas de maior demanda.
- O traçado de rodovias e corredores de escoamento que atendam às diversas demandas de transporte.

- Diversos problemas de roteamento e transbordo bimodal (que envolvem dois tipos de meios de transporte), como distribuição aérea x rodoviária em que um conjunto de aeroportos são selecionados face à possibilidade de uma cobertura rodoviária de conexão e distribuição.
- Traçado de sistemas de transporte urbanos em redes integradas metrô-ônibus.

Nesse modelo as considerações da interação entre o traçado da rota e seu meio ambiente são motivo de exame e considerações (Mesa e Boffey [1996]). Current *et al.* (1985) são os primeiros a relatarem esse problema na literatura e propõem a seguinte formulação:

Considerando um caminho entre os nós O (nó de oferta) e D (nó de demanda) de um grafo $G = (N, A)$ que maximize a cobertura da demanda atendida e minimize os custos de implantação da seguinte forma:

$$\begin{aligned}
 & \text{(PCMCC) Maximizar } z = \sum_k d_k y_k \\
 & \text{Minimizar } w = \sum_{i,j} \delta_{ij} x_{ij} \\
 & \text{sujeito a:} \\
 & \sum_i x_{ij} - \sum_k x_{jk} = \begin{cases} -1 & \text{se } j = O \\ 0 & \text{se } j \notin \{O, D\} \\ +1 & \text{se } j = D \end{cases} \\
 & \sum_k a_{jk} \sum_i (x_{ik} + x_{ki}) - 2y_i \geq 0 \quad j \notin \{O, D\} \\
 & x_{ij}, y_i \in \{0, 1\}, \quad i, j \in A
 \end{aligned}$$

onde:

$y_k \equiv$ variável binária que assume valor 1 se o nó k é coberto e 0, caso contrário (y_O e y_D são, fixados igual a 1).

$x_{ij} \equiv$ variável binária que assume valor 1 se o arco (i, j) está no caminho escolhido e 0, caso contrário;
 $a_{ik} = 1$ se k é coberto de j e 0, caso contrário.

$d_k \equiv$ representa a demanda no vértice k .

$\delta_{ij} \equiv$ representa o comprimento do arco (i, j) .

• Serviços de Atendimento Móvel

Um problema associado ao PCMCC é o que podemos denominar de serviço móvel. Nesse problema, a cobertura de serviços é realizada em um trajeto com várias paradas significativas em tempo. Os tempos das paradas são de ordem de grandeza semelhante ao dos deslocamentos. Nesse caso, o desenvolvimento de caminhos em que o número de nós seja numericamente limitado atende ao contexto do modelo. Como na outra dimensão temporal nós temos as distâncias de deslocamento, a consideração de limites para o comprimento dos caminhos é outra necessidade. A possibilidade da consideração de uma frota de servos permite diversas variantes para o problema (Lindner-Dutton [1991]), inclusive com a possibilidade de visitas sucessivas de um mesmo servo a um mesmo vértice de demanda e, nesse sentido, o modelo se distancia da concepção básica dos modelos de roteamento. Um critério possível para tratar essas situações é a minimização dos retornos a um mesmo ponto de demanda (Malandraki e Daskin [1993]).

• Ligações Físicas

Em várias situações, as ligações de circuitos elétricos e de distribuição de água podem se apresentar em uma configuração que seja análoga aos modelos de serviço de atendimento móvel. Para o caso das ligações de água sob pressão, por exemplo, as voltas e contornos de uma ligação representam pontos de perda de carga (pressão). No caso das ligações em uma rede elétrica o comprimento da linha e suas derivações gera redução de confiabilidade e perda de tensão. Em ambos os casos surge a necessidade de impor limitação ao número total de nós pertencentes a uma mesma derivação ou circuito, bem como buscar minimizar curvas e distâncias percorridas. Modelos para tais problemas podem ser encontrados em Current *et al.* [1986]. A Figura 10.13 mostra o efeito da disposição física no desenho de uma rede de atendimento. A solução 2 é realizada em quatro ramais sem recortes ou derivações nas ligações, enquanto na solução 1 ocorrerem desdobramentos.

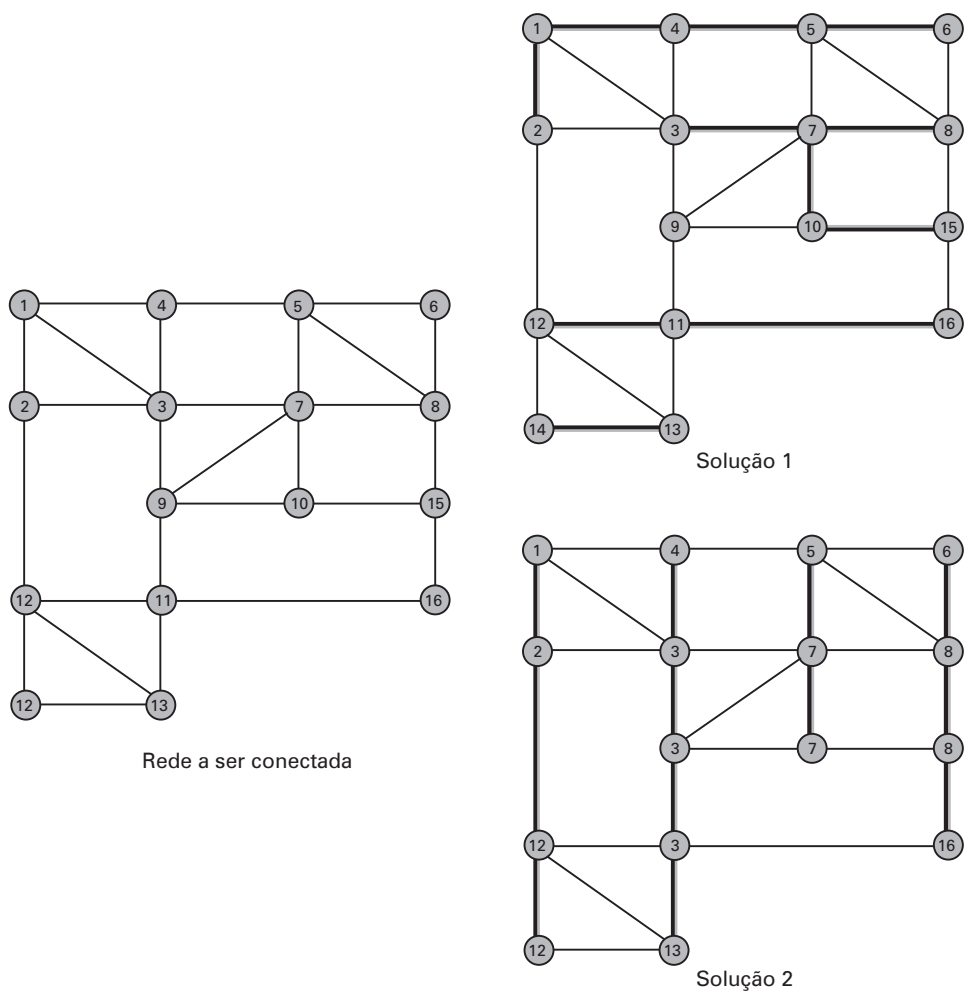


FIGURA 10.13 Soluções para ligação de uma rede.

10.2.3 Métodos de Solução para o PR e PP

Tanto o PP como o PR e o PK são problemas NP-árduos (Karp [1975] e [1976]), o que dificulta sobremaneira a expectativa de um bom desempenho de algoritmos exatos para problemas de grande porte. As técnicas mais utilizadas para a solução abordam duas grandes vertentes: os métodos diretos, desenvolvidos especificamente para a solução desses problemas e os métodos ditos indiretos, ou seja, métodos criados para outros problemas e que são também aplicados com sucesso na solução desses problemas. Destacamos nas Tabelas 10.1, 10.2 e 10.3 alguns dos principais trabalhos da área.

TABELA 10.1 SOLUÇÕES EXATAS

Ano	Autores	Abordando
1971	Lemke	Relaxação linear
1973	Hammer	Álgebra booleana
1974	Marsten	Enumeração implícita
1975	Balas e Padberg	Pivoteamento inteiro e método Simplex
1975	Lin	Geração de colunas
1975	Hammer	Álgebra booleana
1977	Etcheberry	Enumeração implícita
1977	Balas	Planos de corte
1979	Nemhauser e Weber	Relaxação lagrangeana
1979	Balas e Zemel	Teoria dos grafos
1980	Marsten <i>et. al</i>	Agregação de restrições lógicas
1980b	Balas e Ho	Planos de corte p/ recobrimento
1980a	Balas	Planos de corte
1982	Ronen	Geração de colunas
1984	Hulme <i>et al.</i>	Restrições booleanas
1984	Hulme e Baca	Restrições booleanas
1984a	Vasko e Wilson	B&B
1986	Fisher e Kedia	Dualidade
1987	Beasley	Relaxação lagrangeana e dualidade
1987	Campello e Maculan	Relaxação lagrangeana
1988	Ram	Agregação de restrições lógicas
1989	Ali e Thiagarajan	Relaxação lagrangeana e dualidade
1990	Beasley e Jörnsten	Relaxação lagrangeana e dualidade
1990	Fisher e Kedia	Dualidade
1990	Campello e Goldbarg	Enumeração Implícita e restrições lógicas
1992	Desrochers <i>et al.</i>	Geração de colunas
1992	El-Darzi e Mitra	Teoria dos grafos e relaxação / particionamento-recob.
1992	Beasley e Jörnsten	Cortes de Gomory e estratégias de divisão
1994	Harche e Thompson	Subtração de colunas
1995	Saxena e Arora	Técnicas de enumeração p/ problema multiobjetivo
1995	El_Darzi e Mitra	Teoria dos grafos e relaxação / particionamento-recob.
1996	Balas e Carrera	B&B p/ problema de recobrimento
1996	Coudert	B&B / técnica de poda p/ problema de recobrimento

TABELA 10.2 REVISÕES DA LITERATURA

Ano	Autores	Abordando
1966	Garfinkel e Nemhauser	Revisão para o Particionamento
1972	Garfinkel e Nemhauser	Revisão para o Recobrimento
1975	Christofides e Korman	Revisão para o Recobrimento
1976	Balas e Padberg	Revisão para o Particionamento
1979	Balas e Padberg	Revisão para o Particionamento

TABELA 10.3 SOLUÇÕES HEURÍSTICAS

Ano	Autores	Abordando
1979	Chvatal	Míope primal p/ recobrimento
1981	Baker	Míope primal p/ recobrimento
1982	Hochbaum	Teoria dos grafos p/ recobrimento
1984b	Vasko e Wilson	Míope primal p/ recobrimento
1986	Vasko, e Wilson	Abordagem híbrida p/ recobrimento
1988	Goldbarg	Míope primal p/ recobrimento
1988	Barceló e Fernández	Abordagem híbrida (dual x primal) p/ particionamento
1989	Feo e Resende	Heurística GRASP p/ recobrimento
1990	Fisher e Kedia	Abordagem dual p/ particionamento
1990	Beasley	Baseada em relaxações p/ recobrimento
1993	Peleg <i>et al.</i>	Algoritmo de “vizinhaça inteira”
1993	Goldschmidt <i>et al.</i>	Algoritmo míope p/ recobrimento
1993	Sen	Algoritmo <i>simulated annealing</i> p/ recobrimento
1994	Almiñana e Pastor	Abordagem primal
1994	Lorena e Lopes	Heurística <i>surrogate</i> p/ o recobrimento
1994	Huang <i>et al.</i>	Algoritmo genético p/ o recobrimento
1994	Aourid e Kaminska	Redes neurais p/ recobrimento
1995	Caprara, <i>et al.</i>	Baseada em relaxações
1996	Beasley e Chu	Algoritmos genéticos p/ recobrimento
1996	Al-Sultan	Algoritmos genéticos
1996	Levine	Algoritmos genéticos p/ recobrimento e <i>particionamento</i>
1997	Almiñana e Pastor	Baseada em relaxações p/ recobrimento
1997	Grosman e Wool	Algoritmos neurais p/ recobrimento
1997	Haddadi	Heurística baseada na relaxação lagrangeana p/ recobrimento
1997	Lorena e Souza Lopes	Algoritmos genéticos p/ recobrimento
1998	Monfroglio	Heurísticas híbridas p/ recobrimento
1998	Ceria	Heurística com base na relaxação lagrangeana
1999	Caprara <i>et al.</i>	Heurística com base na relaxação lagrangeana
2000	Caprara <i>et al.</i>	Vários algoritmos heurísticos
2000	Hifi	Rede neural p/ recobrimento
2000	Czech	Heurística <i>simulated annealing</i> p/ particionamento
2001	Ohlsson <i>et al.</i>	Redes neurais com <i>feedback</i> p/ recobrimento
2001	Linderoth <i>et al.</i>	Heurística com base na programação linear
2002	Haouari e Chaouachi	Algoritmo GRASP p/ recobrimento
2002	Solar <i>et al.</i>	Algoritmo genético paralelo p/ recobrimento
2002	Aickelin	Algoritmo genético “indireto” p/ recobrimento
2002	Beraldi e Ruszczyński	Versão probabilística do problema de recobrimento
2002	Joseph	Esquema para decomposição p/ particionamento
2004	van Krieken <i>et al.</i>	Heurística baseada em relaxação lagrangeana
2004	Li e Kwan	Meta-heurística p/ recobrimento

✓ Heurísticas Primais

As heurísticas primais são uma generalização dos procedimentos míopes. Basicamente essas heurísticas constroem passo a passo uma cobertura selecionando as variáveis que minimizam determinada função de avaliação.

• Heurística Prime

Sendo $N_i = \{j \in N \mid a_{ij} = 1\}$, $i \in M$, o conjunto das colunas que cobrem a linha i , e $M_j = \{i \in M \mid a_{ij} = 1\}$, $j \in N$, o conjunto de linhas coberto pela variável j . Sendo ainda R o conjunto das linhas não cobertas na matriz A em determinada iteração, então $k_j = |M_j \cap R|$, $j \in M$ será o número de linhas que serão cobertas com a introdução da variável x_j na cobertura. De posse dessas informações, uma heurística primal absolutamente natural para a solução do PR seria aquela que selecionasse uma sequência de inclusões de variáveis maximizando uma função que levasse em conta o número de linhas a serem cobertas pela variável e seu custo. Nesse caso poderíamos escrevê-la da seguinte forma:

Algoritmo Cobertura Prime	
INÍCIO Ler matriz A de cobertura e vetor custo c_j . Inicializar variáveis $R \leftarrow M$, $S \leftarrow \emptyset$, $t \leftarrow 1$. Enquanto $R \neq \emptyset$ fazer Início Escolher $i^* \in R$, tal que $ N_{i^*} = \text{Mínimo } N_i $, $i \in R$ Escolher $j(t)$ tal que $f(c_{j(t)}, k_{j(t)}) = \text{mínimo } \{f(c_j, k_j) \mid j \in N_{i^*} \text{ e } k_j > 0\}$ {*Comando de Avaliação*} Sendo $k_j = M_j \cap R $, $\forall j \in N_{i^*}$ Atualize R e S , ($R \leftarrow R \setminus M$), ($S \leftarrow S \cup \{j(t)\}$) e incremente a iteração ($t \leftarrow t+1$). Fim Ordenar a cobertura S em ordem não crescente de custos ou seja: $S = \{j_1, j_2, \dots, j_t\}$ correspondendo a $\{c_{j(1)} \geq c_{j(2)} \geq \dots \geq c_{j(t)}\}$ Para $j = 1$ até t fazer: Se $S \setminus \{j\}$ é cobertura então $S \leftarrow S \setminus \{j\}$ Caso contrário $z_0 \leftarrow z_0 + c_j$ Escrever $\{S \text{ e } z_0\}$ FIM	

Aplicando a heurística anterior à seguinte instância do PR representada pela matriz de restrições A e o vetor de custos c , e adotando a função de avaliação $f(c_j, k_j) = c_j/k_j$, uma função capaz de quantificar o custo de cobertura por linha coberta, teremos:

$C =$	Coluna	1	2	3	4	5	6	7	8
	Linha								
$A =$		2	3	4	4	6	2	2	3
	1	1	0	0	0	0	0	0	1
	2	0	1	1	1	1	0	1	0
	3	1	1	0	1	1	0	0	0
	4	0	0	1	1	0	0	0	1
	5	0	0	1	1	0	1	0	1
	6	0	0	0	0	1	1	1	0

Inicialização: $R = \{1, 2, 3, 4, 5, 6\}$, $S = \emptyset$ e $t = 1$;

Iteração $t = 1$ do Enquanto:

$\min_{j \in R} \{|N_i|\} = |N_1| = 2$, daí $i^* = 1$ (primeira linha é escolhida para ser coberta).

As duas colunas que cobrem a linha 1 da matriz A são:

J	1	8
k_j	2	3
c_j	2	3
c_j/k_j	1	1

$\text{Min}\{1, 1\} = f(c_1, k_1)$ por uma escolha arbitrária.

$j(1) = 1$

$R = \{2, 4, 5, 6\}$, $S = \{1\}$

Iteração $t = 2$ do Enquanto

$\min_{j \in R} \{|N_i|\} = |N_4| = |N_6| = 3$, daí $i^* = 4$ (quarta linha é escolhida aleatoriamente)

As três colunas que cobrem a linha 4 da matriz A são:

J	3	4	8
k_j	3	3	2
c_j	2	4	3
c_j/k_j	2/3	4/3	3/2

$\text{Min}\{2/3, 4/3, 3/2\} = f(c_3, k_3) = 2/3$

$j(2) = 3$

$R = \{6\}$, $S = \{1, 3\}$

Iteração $t = 3$ do Enquanto

$\min_{j \in R} \{|N_i|\} = |N_6| = 3$ daí $i^* = 3$ (única linha descoberta)

As três colunas que cobrem a linha 4 são:

J	5	6	7
k_j	1	1	1
c_j	6	2	2
c_j/k_j	6	2	2

$\text{Min}\{6, 2, 2\} = f(c_6, k_6) = 2$ por escolha aleatória

$j(3) = 6$

$R = \{\emptyset\}$, $S = \{1, 3, 6\}$

Para $j = 1$ até t fazer

Define que S é uma cobertura Prime, ou seja, não pode ser reduzida de nenhuma de suas colunas sob pena de deixar de ser uma cobertura.

Escrever $\{S = \{1, 3, 6\} \text{ e } z_0 = 9\}$

A complexidade da heurística Cobertura Prime é $O(m^3n)$. A partir do entendimento dessa heurística podemos compreender facilmente outros algoritmos sugeridos na literatura.

• Heurística de Chvatal

Aplicando a heurística Prime sem a verificação de sobrecoberturas (redundância) que o comando “Para $j = 1$ até t fazer” realiza, e substituindo N_i no laço do enquanto $R \neq \emptyset$ por N , então teremos a denominada *Heurística de Chvatal* (Chvatal [1979]). O trabalho de Chvatal foi decorrente do aperfeiçoamento das propostas de Johnson (1974) e Lovász (1975) que podiam ser aplicadas ao recobrimento não ponderado. A heurística de Chvatal possui uma razão de performance Rp calculada de:

$$Rp = \frac{z_0^H(I)}{z_0^*(I)} \leq H(d) < H(m)$$

onde:

$$H(m) = \sum_{j=1}^m \frac{1}{j} \quad \text{e} \quad d = \max_{j \in N} \{ |M_j| \}$$

$z_0^H(I) \equiv \text{valor da solução heurística.}$

$z_0^*(I) \equiv \text{valor ótimo do problema.}$

De fato, Ho (1982) prova que, para toda a classe de heurísticas primais, a razão de performance é a mesma da de Chvatal (Campello e Maculan [1994]).

• Heurística de Balas e Ho

O procedimento proposto por Balas e Ho (1980) sugere uma alteração nas funções de avaliação da heurística Prime da seguinte forma:

1. c_j
2. c_j / k_j
3. $c_j / \log_2(k_j)$
4. $c_j / k_j \cdot \log_2(k_j)$
5. $c_j / k_j \cdot \ln(k_j)$

Se $k_j = 1$ então $\log_2(k_j)$ e $\ln(k_j)$ devem ser substituídos por 1.

• Heurísticas de Vasko e Wilson.

No caso de Vasko e Wilson (1984b) as funções utilizadas para avaliação, além das de Balas e Ho, são:

1. $c_j / (k_j)^2$
2. $\frac{\sqrt{c_j}}{k_j^2}$

Os autores sugerem ainda, sobre o procedimento Prime, uma escolha aleatória, a cada iteração, da função avaliação a ser utilizada. A abordagem de Vasko e Wilson reporta bons resultados (Vasko e Wilson [1984a]). Os autores ainda propõem (Vasko e Wilson [1986]) um uso conjunto das funções de avaliação e de duas versões para as heurísticas primais, uma semelhante a Prime e uma outra sem o comando **Escolher** $i^* \in R$, tal que $|N_{i^*}| = \text{Mínimo } |N_i|, i \in R$.

• Heurísticas Híbridas

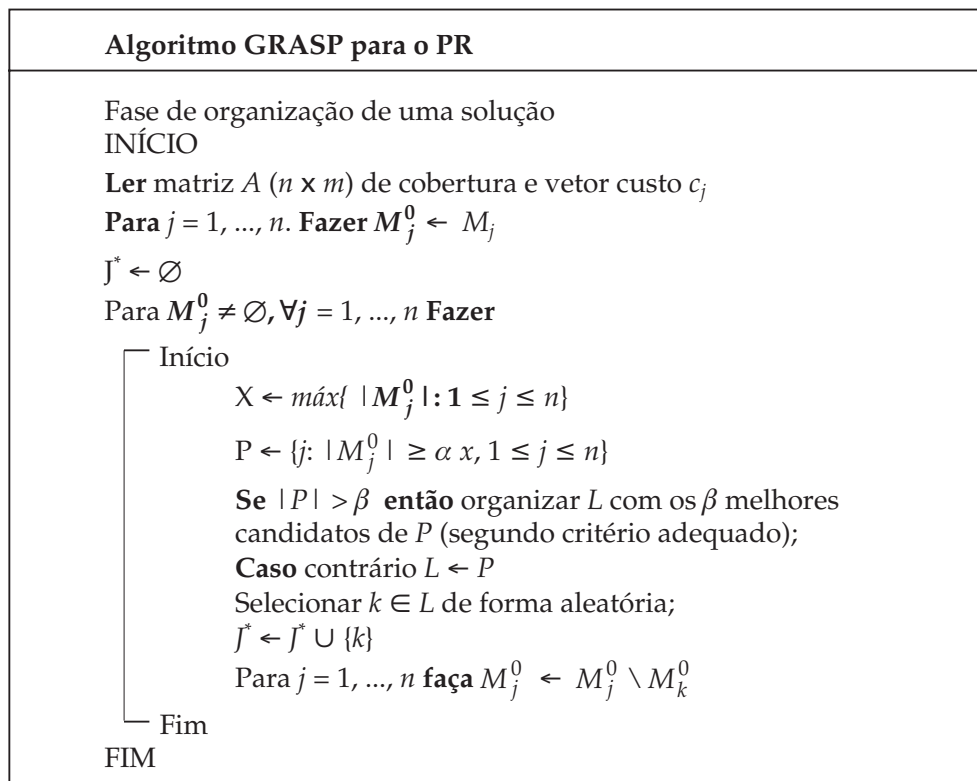
Alguns procedimentos compõem várias abordagens heurísticas e procedimentos exatos desenvolvidos de forma incompleta. Essas heurísticas são chamadas de híbridas. O uso das técnicas de reduções do problema (Nemhauser e Garfinkel [1972] páginas 302-304) para acelerar procedimentos exatos também é um caminho interessante nessa linha de solução (Goldbarg [1988]).

• Heurística de Backer

A heurística de Backer (1981) é interessante pela sua proposta de compor duas soluções de recobrimento para obter uma terceira solução melhor. A estratégia de composição sugerida é igualmente válida para o particionamento. A heurística de Backer pode ser utilizada para auxiliar as operações de *crossover* dos algoritmos genéticos para o PR/PP.

• Uma Aplicação da Meta-Heurística GRASP à Solução do PR

A abordagem denominada GRASP (*greedy randomized adaptive search procedures*) é uma técnica iterativa que visa a diversificar as soluções encontradas através de um critério que equilibra uma estratégia inteligente de escolha e um procedimento aleatório de geração de soluções viáveis. A heurística se desenvolve em duas fases, uma de diversificação e outra de intensificação. A fase de diversificação é peculiar ao método. O quadro que se segue apresenta uma possibilidade de estruturação da fase de diversificação para a solução do PR.



Vamos exemplificar a aplicação da meta-heurística GRASP para a solução do PR através da solução do problema cuja matriz de restrições e vetor de custos estão apresentados na Figura 10.14, definindo os vários parâmetros do algoritmo anterior (L, α, β etc.).

Vetor Custo										
	$c_1 = 1$	$c_2 = 6$	$c_3 = 4$	$c_4 = 7$	$c_5 = 2$	$c_6 = 9$	$c_7 = 3$	$c_8 = 3$	$c_9 = 8$	$c_{10} = 4$
Matriz de Restrições A										
1	1	1	1		1					1
2	1				1		1			
3		1		1			1			
4				1			1			
5				1		1		1		
6			1		1	1			1	
7		1	1			1			1	
8			1	1	1	1				1

FIGURA 10.14 Matriz de restrições do exemplo GRASP.

Como constatamos anteriormente, as heurísticas primais, quando aplicadas à solução do PR, exploram exaustivamente critérios míopes para a formação de uma solução viável. O procedimento GRASP também se utiliza dessa ideia, mas lhes acrescenta uma flexibilidade maior de construção.

Consideraremos que uma escolha míope adequada para o PR será aquela que visa selecionar um conjunto de variáveis M_i que cobre o maior número possível de linhas da matriz A que estejam ainda descobertas, cujo conjunto de índices é representado por M . A função míope adaptativa do GRASP utilizará o critério guloso de decisão que será, contudo, restrito a um certo conjunto P de variáveis. A diferença introduzida pelo GRASP na utilização da estratégia míope fica por conta da flexibilização do critério, o que permitirá sua violação parcial. O critério de escolha sobre esse conjunto restrito (LRC – lista restrita de candidatos ou L) vai se referir a um valor α , percentualmente afastado do critério míope. Essa forma de restringirmos a escolha na lista de candidatos é denominada *restrição em valor*.

Podemos pensar de modo semelhante em relação ao tamanho da própria lista de variáveis candidatas. Nesse caso, poderemos considerar apenas os β melhores elementos dessa lista, segundo o critério míope, com a relaxação definida anteriormente. β é uma restrição de cardinalidade que limita superiormente o tamanho da lista L . Claramente, os dois parâmetros repercutem no tamanho final da lista L , uma vez que atuam implicitamente como restrições lógicas associadas a busca.

Tomemos como exemplo a matriz de restrição da Figura 10.14. Uma escolha míope simples, segundo o critério de máxima cobertura, classificaria as variáveis x_3 , x_4 , x_5 e x_6 como boas candidatas para compor uma solução, uma vez que todas elas maximizam o número de linhas cobertas por uma escolha (todas cobrem 4 linhas da matriz de restrição do problema). Considerando uma relaxação percentual nesse critério míope de cobertura de, por exemplo, 40% (ou seja, $\alpha = 60\%$), estaríamos em condições de aceitar incluir na solução variáveis que cobrissem de 4 (100%) até $4 \times 0,6 = 2,4$ (60%), ou seja, 3 linhas.

Com a adoção do critério relaxado, o conjunto P de variáveis candidatas cresce. No caso do exemplo P cresce de $\{3, 4, 5, 6\}$ para $\{3, 4, 5, 6\} \cup \{2, 7\}$. Se houvesse alguma restrição de cardinalidade como, por exemplo, $\beta = 5$, então a LRC seria obtida de P pela eliminação de uma variável. Um critério possível para essa redução poderia ser o custo da variável. Nesse caso a LRC ou L seria obtida pela retirada do conjunto P da variável mais custosa, ou seja, $L = P \setminus \{6\} = \{3, 4, 5, 7\}$.

Com a lista L estruturada, cabe decidir sobre a inclusão de uma variável na solução. Nesse caso, o GRASP utiliza o método aleatório. Segundo o algoritmo do quadro, se o sorteio recaísse sobre a variável de índice 7, então uma solução parcial seria organizada com as linhas 2, 3 e 4 sendo cobertas. Os conjuntos M_j^0 seriam atualizados face a inclusão de x_7 na solução parcial, removendo-se essas linhas do conjunto de linhas a serem ainda cobertas. A consequência da inclusão da variável de índice 7 na solução parcial pode ser representada pela redução da matriz de restrições, como mostra a Figura 10.15. Nesse ponto, todo o processo é feito novamente.

Vetor Custo										
	$c_1 = 1$	$c_2 = 6$	$c_3 = 4$	$c_4 = 7$	$c_5 = 2$	$c_6 = 9$	$c_7 = 3$	$c_8 = 3$	$c_9 = 8$	$c_{10} = 4$
Matriz de Restrições A										
1	1	1	1		1					1
5				1		1		1		
6			1		1	1			1	
7		1	1			1			1	
8			1	1	1	1				1

FIGURA 10.15 Matriz de restrições atualizada após inclusão de x_7 na solução.

Aplicando novamente o critério guloso para a escolha das colunas candidatas encontramos o máximo $\{|M_j^0|\} = 4$, daí $P = \{3, 4, 5, 9\}$. Como $\beta = 4$ temos $L = P$ e realizamos o sorteio do novo integrante da solução corrente.

Para estudarmos a segunda fase do método vamos supor que a solução final encontrada pelo processo de restrição e sorteio tenha sido $Sol = \{1, 6, 7\}$ com custo total igual a 13. A segunda fase do algoritmo é uma etapa de intensificação elaborada por uma busca local. Consideraremos que iremos examinar a vizinhança de $\{1, 6, 7\}$ utilizando uma estratégia de *economia* para orientar a busca. Nesse caso a busca seria dirigida a identificar uma solução viável e vizinha a $\{1, 6, 7\}$, obtida pela troca de uma ou mais variáveis de Sol e que, pelo critério descendente, levasse a um valor menor que o encontrado para Sol . No caso do exemplo poderíamos trocar a coluna 6 (a mais custosa) pelas colunas 3 e 8. A nova Sol encontrada seria $\{1, 3, 8, 7\}$ a um custo de 11, duas unidades inferior à solução primitiva. Ainda dentro de uma estratégia de busca local a redundância de Sol seria provavelmente eliminada por uma estratégia de intensificação, restando o conjunto $\{3, 7, 8\}$ que representaria um ótimo local. Nesse caso particular a solução ótima para a vizinhança de $\{1, 6, 7\}$ seria também globalmente ótima.

10.3 – O PROBLEMA DA K-DISPERSÃO DISCRETA (PKD)

10.3.1 – Importância e Formulação

A solução de problemas de dispersão está hoje crescendo em importância especialmente pela necessidade de providenciar um destino adequado a uma série de materiais, rejeitos industriais e nucleares e armamentos (Erkut e Neuman [1989]). Dado um conjunto de n facilidades alocadas sobre os nós de um grafo, o problema da k -dispersão discreta (PKDD) consiste em selecionar dentre as n , k facilidades, de modo que a distância mínima entre qualquer par das k facilidades selecionadas seja máxima. A aplicação desse problema diz respeito a:

- Localização de reservatórios de combustível.
- Alocação de agentes competitivos como franquias etc. (Kuby [1987]).
- Distribuição de depósitos de lixo (Church e Garfinkel [1978]).
- Localização de silos de mísseis e instalações nucleares (Erkut e Neuman [1989]).
- Prisões e instalações militares (Berman *et al* [1996]).
- Tratamento de águas residuais (Morales *et al* [1997]).
- Distribuição de frequências em sistemas de comunicação.
- Experimentos estatísticos.
- Exploração de madeira.

O problema da k -dispersão se distingue dos outros problemas de localização em que um conjunto de usuários ou pontos de demanda interagem com um conjunto de possíveis localizações de facilidades, como nos problemas de Anticentro, Antimédiana etc. (Erkut e Neuman [1989], Churk e Garfinkel [1978] e Minieka [1983]). A formulação mais conhecida para esse problema é denominada Max-Min. A formulação que se segue é devida a Erkut (1990).

(PKD1) Maximizar $z = r$

sujeito a:

$$r \leq M(2 - y_i - y_j) + d_{ij}, \quad 1 \leq i \leq j \leq n$$

$$\sum_{i=1}^n y_i = k$$

$$y_i \in \{0, 1\} \quad 1 \leq i \leq n$$

onde:

$y_i \equiv$ variável binária que assume o valor 1 quando a facilidade i é selecionada e 0, caso contrário.

$d_{ij} \equiv$ distância entre as facilidades i e j .

$r \equiv$ distância mínima entre as facilidades selecionadas.

$n \equiv$ número de facilidades potenciais, ou seja, passíveis de escolha.

$k \equiv$ número das facilidades que deverão ser localizadas.

$M \equiv$ número convenientemente grande face a maior d_{ij} .

Essa formulação contém $\frac{1}{2}n(n-1)$ restrições envolvendo R e uma restrição com as variáveis y_i . A relaxação linear de PKD1 pode ser utilizada para definir um limite superior para r , uma vez que as soluções fracionárias dos y_i conduzem, na maioria das vezes, a valores de d_{ij} muito maiores que r . Uma formulação mista inteira é sugerida por Kuby (1987).

$$(PKD2) \text{ Maximizar } z = \sum_i \sum_{j=i+1}^n z_{ij} d_{ij}$$

sujeito a:

$$\sum_{i=1}^n y_i = k$$

$$z_{ij} \leq y_i \quad i < j, j = 1, \dots, n$$

$$z_{ij} \leq y_j \quad i < j, j = 1, \dots, n$$

$$y_i \in \{0, 1\} \quad \forall i$$

$$z_{ij} \in \{0, 1\} \quad \forall i, j$$

onde:

z_{ij} é uma variável binária que assumirá valor igual a 1, se as facilidades são localizadas em i e j , e 0, caso contrário, apesar de sua definição como variável contínua. O controle do valor de z_{ij} é realizado através das restrições em y_i e y_j . Somente se as variáveis y_i e y_j forem ambas iguais a 1 z_{ij} assumirá esse valor. A função objetivo contabiliza a soma das distâncias entre todos os pares de facilidades abertas. PKD2 é NP-completo. Um aspecto bastante interessante desse problema é seu relacionamento com o problema *anti covering* (PAC). O PKD objetiva maximizar a distância de separação mínima r de k facilidades entre si, enquanto o PAC objetiva maximizar o número de facilidades a serem localizadas de forma a manter uma distância mínima r de separação entre si. Moon e Chaudhry (1984) formulam o PAC como se segue:

$$\begin{aligned}
 & \text{(PAC) Maximizar } z = \sum_{j \in S} x_j \\
 & \text{sujeito a:} \\
 & M(1 - x_i) \geq \sum_{j \in Q_i} x_j, \quad \forall i \in S \\
 & x_j \in \{0, 1\}, \quad \forall j \in S
 \end{aligned}$$

onde:

$x_j \equiv$ variável binária que assume valor igual a 1 se a facilidade é localizada em j e 0, caso contrário.

$S = \{1, \dots, n\}$ é o conjunto das facilidades potenciais.

$M \equiv$ um valor numérico suficientemente grande.

$Q_i = \{j \in S \text{ tal que } d_{ij} < r, i \neq j\}, i \in S$, o conjunto de locais que estão a uma distância menor que r unidades desde i , excluindo i .

Nesse modelo, quando $x_i = 1$, todos os x_j devem tomar o valor zero para $j \in Q_i$. Assim a restrição em M representa a separação desejada $d_{ij} > r$.

Uma outra formulação similar para o PKD é proposta por Erkut (1990) e denominada de *r-separação* da seguinte forma:

$$\begin{aligned}
 & \text{(PKD3) Minimizar } z = \sum_{i=1}^n x_{ij} \\
 & \text{sujeito a:} \\
 & x_i + x_j \geq 1, \quad i < j = 2, \dots, n \text{ com } d_{ij} < r \\
 & x_i \in \{0, 1\}, \quad i = 1, \dots, n
 \end{aligned}$$

onde:

$x_j \equiv$ variável binária que assume valor igual a 1 se a facilidade j é selecionada para integrar a solução e 0, caso contrário.

Essa formulação utiliza n variáveis e $O(n^2)$ restrições. O valor ótimo da função objetivo será zero se e somente se existir um conjunto independente de tamanho k . no subgrafo definido pelas arestas com $d_{ij} < r$. Sua similaridade com a formulação do problema *anti-covering* e conjunto independente é forte (Moon e Chaudhry [1984] e Chaudhry *et al.* [1986]).

10.3.2 – Exemplos de Aplicações e Problemas Correlatos

✓ O Problema de Localização de Implantes Radioativos

Trata-se de um dos exemplos interessantes do uso de uma abordagem quantitativa em uma área tipicamente imprecisa. O implante de materiais radioativos sólidos sobre um determinado tecido é uma técnica utilizada em combate a tumores cancerígenos em áreas extremamente sensíveis como o cérebro, por exemplo. O material implantado pode ter a forma de agulhas ou partículas. Uma proximidade excessiva dos agentes ocasionará uma superdosagem, e um afastamento demasiado o efeito contrário. Definido ao alcance de cada elemento radioativo e a área a ser coberta o problema pode ser modelado como um problema de k -dispersão sobre o contínuo. Se os possíveis pontos de localização dos elementos forem definidos previamente o problema pode se reduzir ao PKD.

✓ O Problema de Segurança e Eficácia das Comunicações

Se estamos utilizando um alfabeto $\alpha = \{x_1, x_2, \dots, x_n\}$ para um processo de comunicação e $G = (\alpha, \theta)$, onde θ representa as relações entre os sinais do alfabeto que não podem ser confundidos entre si, a clique (subgrafo completo de G) de tamanho k em G consiste em k sinais para o qual não haverá risco de confusão. Se o problema da confusão de sinais for associado a uma certa variável aleatória de modo que exista para cada aresta (i, j) uma probabilidade p_{ij} de a confusão ocorrer, então podemos definir um PKD da seguinte forma: obter o conjunto de k sinais do alfabeto em que a probabilidade mínima de dois sinais não serem confundidos seja máxima.

✓ O Problema de Marketing de Produtos

Considere um conjunto $S = \{s_1, s_2, \dots, s_n\}$ de produtos e $P = \{p_1, p_2, \dots, p_m\}$ de propriedades desses produtos como manutenibilidade, confiabilidade, aparência etc. A aplicação do PKD é definida como a busca de um lançamento de k produtos tais que, com uma diversidade mínima de propriedades r , cubram a demanda do mercado consumidor. Definindo x_{ij} como uma variável binária que assume o valor igual a 1 se o produto i possui a propriedade j e 0, caso contrário. Obviamente, um produto possui tanto mais similaridades com outro quanto maior for o número de coincidência entre as propriedades dos dois produtos. Então, podemos definir como a distância d_{ik} o número de coincidências entre os vetores x_{ij} e x_{kj} , ou seja, o número de emparelhamentos entre as componentes desses dois vetores. Podemos também representar esse fenômeno através de um grafo $G = (S, A)$, em que o conjunto A de arestas (i, j) de G significa que os produtos s_i e s_j são similares. Nesse modelo obter um conjunto de produtos diversos é equivalente a obter um conjunto independente de tamanho k , o que pode ser solucionado como um caso particular de PKD.

✓ O Problema de Seleção de uma Carteira de Investimento

Considerando um comportamento de mercado que preserve uma tendência de correlação positiva entre as taxas de retorno dos diversos investimentos possíveis nas carteiras. Nessa situação, o problema de escolher uma carteira de investimento seria descrito pela escolha de k propostas dentre um conjunto de n propostas possíveis, de modo a maximizar o retorno mínimo de cada par de propostas.

✓ O Problema da Interferência dos Canais de Comunicações

Esse problema tem uma conexão com o problema da T -coloração (Hale (1980) e anexo: tópicos em teoria dos grafos). Considere um grafo $G = (N, A)$, onde o conjunto $N = \{f_1, f_2, \dots, f_n\}$ representa as frequências disponíveis para alocação em canais de comunicação e uma aresta (i, j) em G representa uma vizinhança de frequência (definida por um certo valor de diferença entre as frequências). Então, a formulação PKD nesse contexto significa selecionar para o uso simultâneo, dentre todas as n frequências disponíveis, um conjunto de k frequências que maximize a separação entre cada par de frequências que será adotado.

✓ O PKD e o Problema k -centro

A dualidade existente entre o problema de k -dispersão e o do k -centro foi estudado por Shier (1977). Uma das abordagens mais comuns para a solução do k -centro tem sido a solução de uma sequência de problemas de recobrimento (Christofides [1975] e Tansel [1982]). Seja um grafo não orientado $G = (N, A)$, com um conjunto de vértices $N = \{v_1, v_2, \dots, v_n\}$ e $i = \{1, 2, \dots, n\}$ o conjunto dos índices dos vértices. Sendo G definido pelo plano euclidiano, seja $d(v_i, v_j)$, a distância entre os vértices v_i e v_j que satisfaz as seguintes propriedades:

$$1 \quad d(v_i, v_j) \geq 0.$$

$$2 \quad d(v_i, v_j) = d(v_j, v_i).$$

$$3 \quad d(v_i, v_j) \leq d(v_i, v_k) + d(v_k, v_j).$$

Dadas as funções $f_i, i \in I$, define-se a função f para $U \subset N$, como $F(U) = \max \{f_i(d(v_i, U)), i \in I\}$ o problema de localização Minmax, onde se requer encontrar um k -centro absoluto $U^* = \{u_1^*, u_2^*, \dots, u_k^*\}$ e um k -raio tal que: $r_k = f(U^*) = \min \{f(U) : |U| = k, U \subset N\}$. O problema pode ser considerado como o de localizar k facilidades ou centros em um conjunto de nós de G de modo a minimizar o risco máximo previsto para uma falha no serviço de atendimento. Para compreender o sentido do risco, suponhamos que o deslocamento dos centros aos possíveis locais de demanda v_i pode ser interpretado como um risco $f_i(d(v_i, U))$ na medida em que o percurso é mais demorado. A função f_i pode ser tomada igual à distância $d(v_i, U)$. Se o conjunto de pontos a atender são todos os nós de G e a facilidade a localizar é apenas uma ($k = 1$), então o problema é denominado de *centro absoluto*. Se o conjunto de facilidades potenciais e de demanda é o mesmo, o problema é denominado de *centro geral* (ver Minieka [1978]). A Figura 10.16 apresenta um grafo em que as distâncias entre os nós valem uma unidade de comprimento. Se localizarmos a facilidade que vai cobrir todos os nós de G no vértice 2 a maior distância entre um vértice e uma facilidade (no caso única) ocorrerá com o vértice 6 e será igual a três unidades. Se a localização for realizada no vértice 1 então a distância cai para duas unidades em relação aos vértices 6 e 4. Se localizarmos a facilidade sobre o meio da aresta 1-3 então a distância cai para 1,5 unidades (nem sempre isso é uma solução legal). O 1-centro geral desse grafo são os vértices 1 e 3 e a distância MinMax é 2.

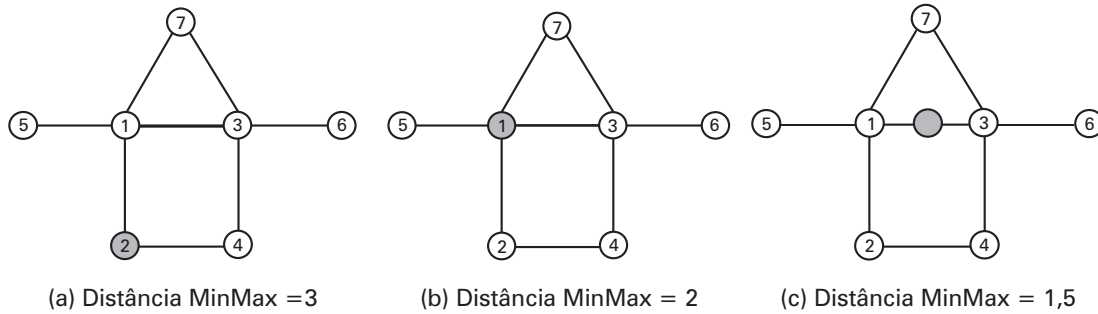


FIGURA 10.16 Solução do k -centro.

O dual do problema do k -centro consiste em escolher $k+1$ pontos de demanda tal que o mínimo dos raios calculados com respeito a todos os pares de demandas escolhidas seja o maior possível. É possível uma relação dual forte entre o problema do k -centro e a k -dispersão em uma árvore e quando não existe restrição de alocação nos vértices para o problema dos k -centros. Nesse caso, a solução ótima MaxMin da $(k+1)$ -dispersão é exatamente o dobro da solução MinMax do k -centros (Shier [1977]). Bar-Ilan (1993) apresenta vários algoritmos heurísticos para a solução desse problema em redes de comunicação.

10.3.3 – Soluções Heurísticas para o PKD

Por se tratar de um problema NP-árduo, as abordagens heurísticas de solução têm sido intensamente utilizadas em problemas de grande porte.

✓ Heurística de Erkut

Erkut (1990) sugere uma heurística que pode ser dividida em fase de solução míope e fase de melhoria. O procedimento, que está formalizado no quadro algoritmo k -dispersão Erkut, possui complexidade $O(n^2)$ na fase de construção e $O(n^2)$ para cada iteração de melhoria.

Algoritmo k -dispersão Erkut

1 – Fase de Construção de uma Solução Míope

INÍCIO

Ler o grafo $G = (N, A)$ e k .

Inicializar variáveis $B := 0$; Gargalo: $= 0$;

Ordenar as arestas d_{ij} de G de forma monótona não decrescente em uma *heap* C

Enquanto $|B| < (n - k)$ fazer

Início

Encontrar a menor distância $d(u, v)$ em C

Se $(u \text{ ou } v) \notin B$ e $|B| < (n - k)$ então **fazer**

Início

 Eliminar aleatoriamente $(u \text{ ou } v) \notin B$

$C \leftarrow C \setminus (u, v)$

$B \leftarrow B \cup \{\text{elemento } u \text{ ou } v \text{ eliminado}\}$

Fim

Se $|B| = (n - k)$ então **fazer** Gargalo $\leftarrow d(u, v)$

Fim

FIM

2 – Fase de Melhoria da Solução Corrente

INÍCIO

Ler $G = (N, A)$, k e Gargalo

Inicializar variáveis $V = A \setminus B$ {organiza o conjunto V onde $i, j \in V \Rightarrow d_{ij} \geq \text{gargalo}, |V| = k^*$ } temp $\leftarrow \emptyset$

Para $j \in B$ fazer

Início

Atribuir $w \leftarrow u$ ou $w \leftarrow v$ se é possível trocar um deles com j nas seguintes condições:

 1. $w \leftarrow v$ se $(d_{uj} > d_{vj})$ e $(d_{uj} > \text{gargalo})$

 2. $w \leftarrow u$ se $(d_{vj} > d_{uj})$ e $(d_{vj} > \text{gargalo})$

 temp $= \{j\} \cup V \setminus \{w\}$

Calcular $N_Gargalo \leftarrow \max, \{d_{ij} \mid i, j \in \text{temp}\}$

Se $N_Gargalo > \text{Gargalo}$ então **fazer**

Início

$V \leftarrow \{j\} \cup V \setminus \{w\}$

 Temp $\leftarrow V$;

Fim

$B \leftarrow B \setminus \{j\}$;

Fim

FIM

Consideremos o problema de localizar três facilidades no grafo da Figura 10.17 (a). Aplicando o algoritmo de Erkut (1990) temos o seguinte resultado por fases:

Fase 1: Solução Míope

$C = \{(3, 6), (1, 6), (2, 3), (1, 4), (3, 4), (4, 6), (4, 5), (5, 6), (2, 5), (1, 3), (2, 6), (1, 2), (2, 4), (3, 5), (1, 5)\}$

$d_{ij} = \{(4), (5), (5), (5), (6), (7), (8), (8), (9), (9), (9), (10), (11), (12), (13)\}$

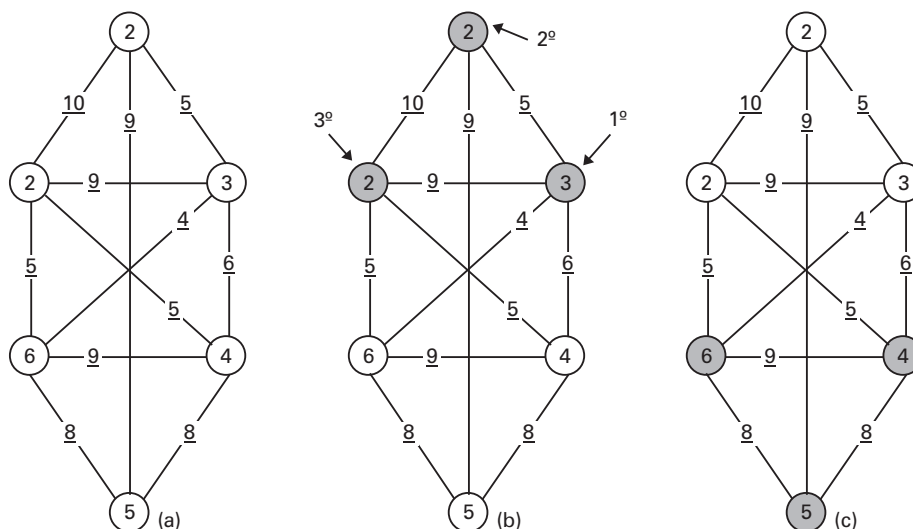


FIGURA 10.17 1ª fase do algoritmo de Erkut para a k -dispersão.

Considerando sempre eliminado o nó possível candidato com o menor índice, a sequência das eliminações e atualizações está apresentada na Tabela 10.4.

TABELA 10.4 PASSO DE ATUALIZAÇÃO DO ALGORITMO DE ERKUT

Distância	Aresta	Nós excluídos	Solução	B	Gargalo
4	(3, 6)	3	(1, 2, 4, 5, 6)	(3)	–
5	(1, 6)	1	(2, 4, 5, 6)	(3, 1)	–
5	(2, 3)	2	(4, 5, 6)	(3, 1, 2)	8

Na terceira inclusão $|B| = 3$ termina a fase de construção de uma solução míope, conforme mostra a Figura 10.17 (b). A solução é composta pelos k vértices remanescentes.

Fase 2: Melhoria da solução

A partir da solução (4, 5, 6), o procedimento busca identifica trocas que melhorem o gargalo existente. A Tabela 10.5 exemplifica a melhoria se o nó escolhido fosse $j = 2$, $j \in B$

TABELA 10.5 MELHORIAS POSSÍVEIS PARA $J = 2$

Nó j	Possíveis pares (u, v)	Distâncias Associadas	Maior Distância	Possível Solução	Gargalo da Solução
2	(4, 5)	$d_{24} = 11$ $d_{25} = 9$	$d_{24} = 11 > 9$ e $d_{25} = 9 > 8$	(2, 5, 6)	$9 > 8$ (*) Melhoria
	(4, 6)	$d_{24} = 11$ $d_{26} = 9$	$d_{24} = 11 > 9$ e $d_{26} = 9 > 8$	(2, 5, 6)	mesmo da anterior
	(5, 6)	$d_{25} = 9$ $d_{26} = 9$	$d_{25} = d_{26} > 8$	–	–

✓ Heurística de Separação por Grupamentos

Uma outra forma de abordar a solução heurística desse problema é através da constituição de *clusters* ou grupamentos de proximidade. A análise de *clusters* é uma abordagem que tenta agrupar os pontos no espaço em conjuntos usualmente disjuntos em que se espera preservar características marcantes em sua população.

Existem várias modalidades de análise de grupamento, sendo a do *enlace simples* uma das mais usadas na computação (Gower e Ross [1969]). A estratégia basicamente une todos os pares de pontos mais próximos para formar grupamentos, que, posteriormente, serão incrementados (unidos ou *joining*) também por proximidade (técnica às vezes denominada de bola de neve; ver McGregor e Shen [1977]). Em cada passo de incremento o número total de grupamentos ($n/2$ grupos) é reduzido em uma unidade pela fusão de dois grupamentos vizinhos em um só.

Uma outra forma de obter os grupamentos de enlace simples é dividir (operação denominada *splitting*) o todo em duas parcelas tais que a distância entre os grupamentos (definido como o mínimo entre os pares de pontos) seja maximizada. Como no caso anterior, as operações de divisão continuam se processando sobre os grupamentos que vão se formando. A propriedade de exigir somente um enlace por combinação de grupamentos justifica o nome de enlace simples.

Os algoritmos de enlace simples para a análise de grupamentos são míopes e possuem normalmente complexidade $O(n^2)$. Para avaliar o desempenho qualitativo das heurísticas de dispersão por agrupamento vamos examinar a instância da Figura 10.18 e seu processo de solução.

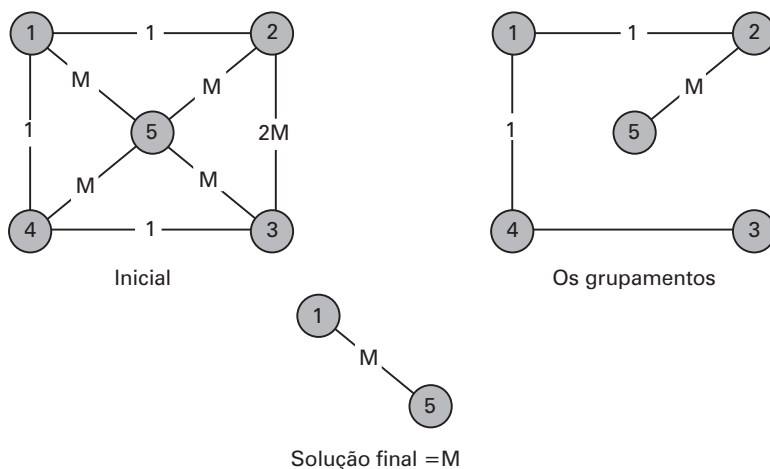


FIGURA 10.18 Qualidade da solução das heurísticas por agrupamento.

A solução ótima do problema da Figura 10.18 é $2M$, mostrando que essa abordagem não possui um limite ε -aproximado absoluto.

Um procedimento também eficiente na geração de k *clusters* ou grupamentos de nós é a eliminação das $k - 1$ arestas mais caras da árvore geradora mínima de G . A Figura 10.19 esclarece esse processo para a formação de quatro grupamentos, baseado no fato de que a retirada de $k - 1$ arestas em uma árvore forma k subgrafos desconexos.

Evidentemente que, para qualquer localização dentro dos grupamentos formados pelo procedimento anterior, as distâncias entre os grupamentos serão um limite superior para o gargalo da k -dispersão. O grafo G pode ser então substituído por um grafo de distâncias em que cada grupamento será representado por um nó, conforme mostra a Figura 10.20.

Examinando o grafo de distâncias da Figura 10.20 verificamos que o gargalo do afastamento é representado pela aresta de menor valor. Cada grupamento possui um conjunto de nós que estabelece sua conexão com os grupamentos vizinhos e a eles o gargalo está relacionado. Em um caso geral, normalmente, os vértices pertencentes aos grupamentos não estão todos na fronteira, ou seja, nem todos relacionam-se com os vértices dos outros grupamentos através das arestas do grafo de distâncias. Esses

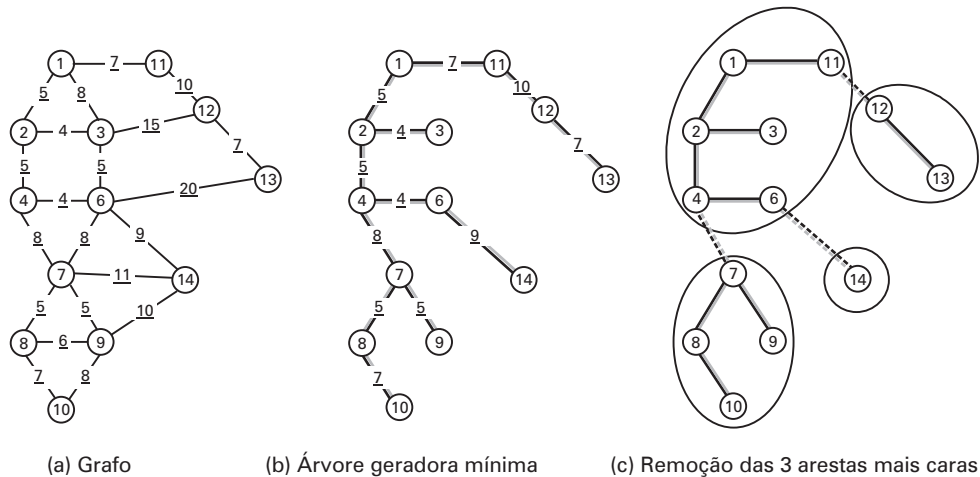


FIGURA 10.19 Procedimento para gerar 4-clusters de proximidade.

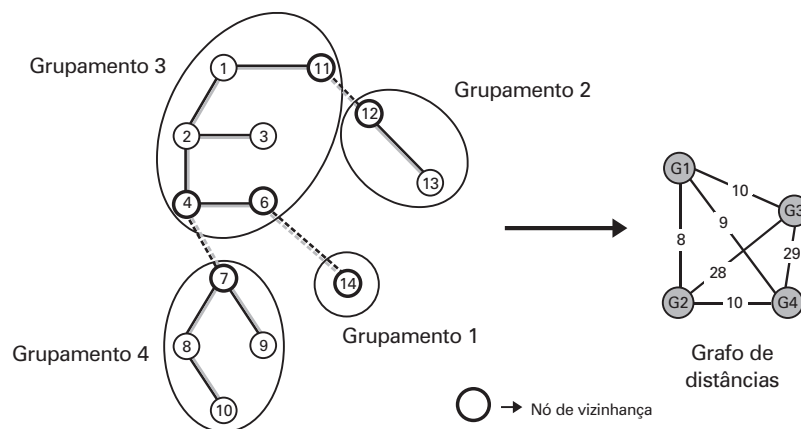


FIGURA 10.20 Grafo de distâncias da Figura 226.

nós internos, ou não relacionados com as distâncias de gargalo, são candidatos a, se utilizados para a solução do problema, melhorar o gargalo associado ao seu grupamento. Se o grafo possuir somente arestas maiores que zero e todos os grupamentos possuírem nós internos, então, com certeza, o gargalo obtido pelo procedimento anteriormente descrito pode ser melhorado. A Figura 10.21 mostra a escolha de nós internos aos grupamentos 1 e 4 e seu efeito sobre o gargalo.

Observamos que o gargalo de afastamento do grafo de distância entre os grupamentos salta para 18 ao escolhermos os nós internos (nós 2 e 10) nos grupamentos 3 e 4 respectivamente. A complexidade de um algoritmo que encontre os grupamentos de nós através da árvore geradora mínima terá complexidade mínima $O(n^2)$. A escolha dos nós internos dentro de cada grupamento exigirá, para um caso geral, uma complexidade $O(n^2)$.

10.4 – PROBLEMAS DE LOCALIZAÇÃO COMPETITIVA – O PROBLEMA DOS K-SERVOS (PKS)

10.4.1 – Importância do Problema e sua Definição

Alguns problemas de localização englobam um aspecto de operação dinâmica que pode transcender ao modelo tradicional de localização, se aproximando de modelos de roteamento. Uma forma de aumentar a capacidade de atendimento de uma facilidade sobre uma rede de demanda é considerar a

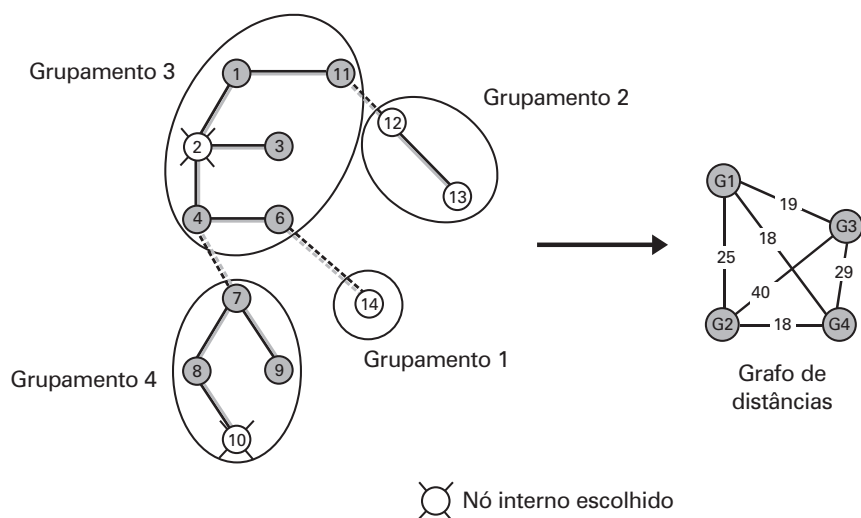


FIGURA 10.21 Localizando alocações em nós internos dos grupamentos.

possibilidade de que ela atenda mais de um ponto. Nesse caso, a facilidade deverá, de alguma forma, realizar uma trajetória na rede para possibilitar o contato oferta \times consumo. As unidades de atendimento das facilidades são denominadas de servos. Um dos exemplos desse tipo de problema é o que se denomina k -servos (PKS). Seja o grafo $G = (N, A)$ e k -servos distribuídos sobre os vértices de G , sujeitos a um vetor de demanda que se caracteriza sobre esses mesmos vértices. O problema dos k -servos é o de planejar o deslocamento desses servos sobre os vértices de G de modo a atender à demanda e demais possíveis restrições, minimizando uma função determinada. De uma forma geral a função de minimização está associada à distância percorrida pelos servos. O k -servos se distingue dos problemas de roteamento pelo fato de o servo permanecer ocupando o último vértice visitado e não se caracterizarem rotas fechadas no grafo. Além disso, o posicionamento inicial dos servos tem grande influência no resultado e desempenho dos algoritmos de solução. A Figura 10.22 esclarece o deslocamento de dois servos em um grafo de sete nós. Os vértices marcados com \checkmark representam os vértices de demanda. As setas com traços reforçados marcam o caminho realizado pelo servo S_i .

A Figura 10.22 exemplifica, dentre as muitas possibilidades de atendimento, duas. Na primeira, os dois servos foram empregados. Na segunda, os dois atendimentos são realizados pelo servo S_1 . Observamos que os servos permanecem estacionados no último nó por eles atendido.

Podemos destacar as seguintes aplicações para o problema:

- Problemas de paginação de memória computacional (Karlin *et al.* [1988] e Raghavan *et al.* [1988]).
- Localização de cabeças de leitura de discos rígidos (Calderbank [1985]).
- Atendimento de segurança e de emergência.

O problema dos k -servos se configura em duas situações específicas:

- ✓ *Off-line* ou quando o vetor de demanda é conhecido *a priori*.
- ✓ *On-line* ou quando a demanda vai ocorrendo ao longo de certo horizonte de tempo, sendo caracterizada em intervalos de tempo. Realmente, as situações de atendimento *on-line* são comuns nesse tipo de problema, especialmente nos modelos industriais (robôs) e de serviços de emergência. Em alguns casos é possível a determinação da distribuição de probabilidade associada a essas demandas. A literatura denomina de *on-line* o caso em que, em princípio, a probabilidade de ocorrência de uma demanda em um nó é igual para todo vértice do grafo, sendo esses eventos independentes.

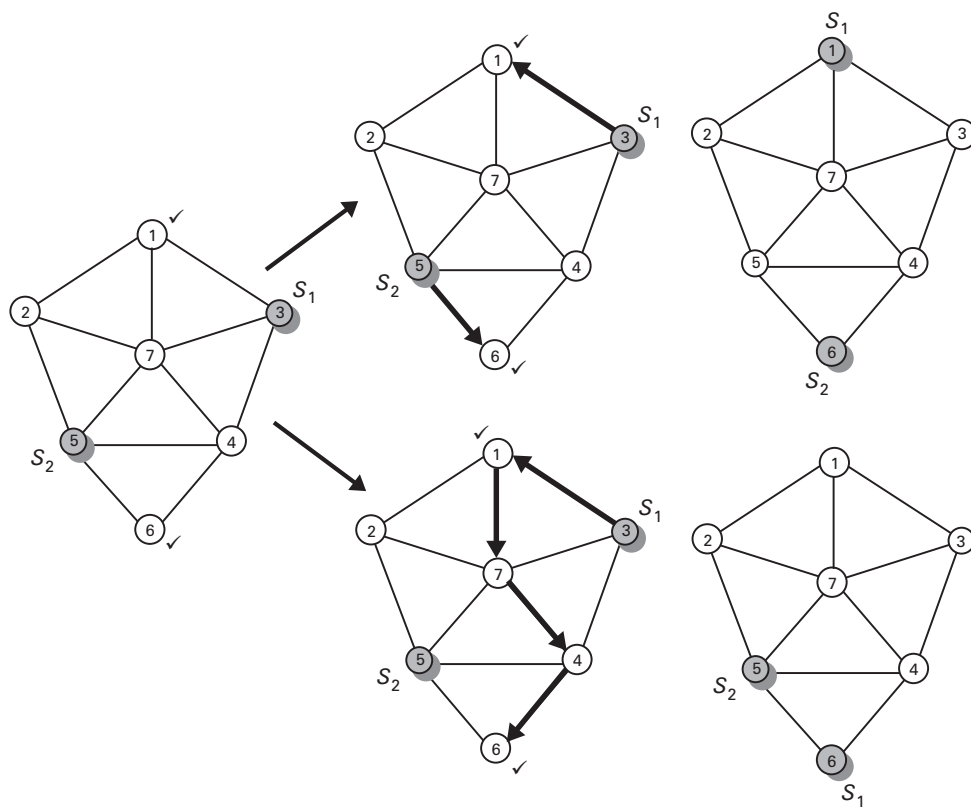


FIGURA 10.22 Soluções para o 2-servos.

10.4.2 – Exemplo de Aplicação e Problemas Correlatos ao k -Servos

✓ O Problema de Distribuição de Transportadores em Manufatura Flexível

Os sistemas de manufatura flexível são complexos tecnológicos que envolvem inúmeros problemas de planejamento e distribuição de recursos compartilhados. O problema de otimizar o transporte entre as células da manufatura envolve, em várias situações, o tratamento de ciclos tipicamente k -servos (ver Koenigsber [1993], Co *et al.* [1989] e Bozer *et al.* [1989]).

✓ O k -servos com Janelas de Tempo

Em situações reais de atendimento de múltipla demanda com múltiplos servos, um dos fatores mais importantes para o desempenho é o tempo de atendimento. Quando esse fator é preponderante, obviamente ele será considerado como o custo do deslocamento. Em outras situações, entretanto, sua consideração será tipicamente restritiva. Dessa forma, em sistemas de atendimento de emergência, como no caso de equipes volantes de manutenção, é necessário considerar a restrição de tempo. Oliveira (1996) sugere um algoritmo *off-line* para a solução desse problema.

✓ Problema de Múltiplos Servos

Batta e Mannur (1990) estudam situações de atendimento de emergência com vários tipos de servos em que um atendimento exige a presença de um servo de cada tipo no ponto de demanda. Esse problema pode ser formulado da seguinte forma:

$$(PMS) \text{ Maximizar } z = \sum_{i=1}^m d_i y_i$$

sujeito a:

$$\sum_{j=1}^n a_{ij}^k x_j^k - y_i \geq 0, \quad i = 1, \dots, m, k = 1, \dots, r$$

$$x_j^k - z_j \leq 0, \quad j = 1, \dots, n, k = 1, \dots, r$$

$$\sum_{j=1}^n x_j^k = p^k, \quad k = 1, \dots, r$$

$$x_j^k, y_i \in \{0, 1\}, \quad j = 1, \dots, n \quad i = 1, \dots, m \quad k = 1, \dots, r$$

onde:

$x_j^k \equiv$ variável binária que assume o valor 1 se um servo do tipo k está alocado à cidade j e 0, caso contrário.

$y_i \equiv$ variável binária que assume o valor 1 se um ponto de demanda i é coberto e 0, caso contrário.

$z_j \equiv$ variável binária que assume o valor 1 se existe uma facilidade alocada no vértice j e 0, caso contrário.

$d_i \equiv$ demanda no ponto i .

$p_j^k \equiv$ número de servos do tipo k permitidos.

$a_{ij} = 1$ se i pode ser coberta pelo servo do tipo k alocado na cidade j e 0, caso contrário.

As restrições que obrigam todos os tipos de servos a atenderem a uma demanda (a primeira restrição de PMS) podem ser substituídas por restrições de compromisso com um montante orçamentário disponível (Boffey e Narula [1997]) para o deslocamento. Marianov e ReVelle (1991) estudaram o PMS em um contexto de segurança contra incêndio para o caso de dois tipos diferentes de servos. No caso em pauta é necessário reunir pelo menos dois servos de um tipo e três de outro para que o ponto de demanda seja considerado atendido por uma equipe.

A Figura 10.23 mostra uma possibilidade de emprego desse modelo para o atendimento múltiplo. A satisfação da demanda é caracterizada pela chegada dos tipos de servos existentes e na quantidade definida. A origem desses servos pode ser variada (várias bases). O vértice 8 representa um ponto de atendimento múltiplo, em que os servos e as demandas (✓) são coloridos em conformidade com a peculiaridade do serviço demandado. Os servos são especializados em atenderem às demandas cuja cor está marcada em seu uniforme.

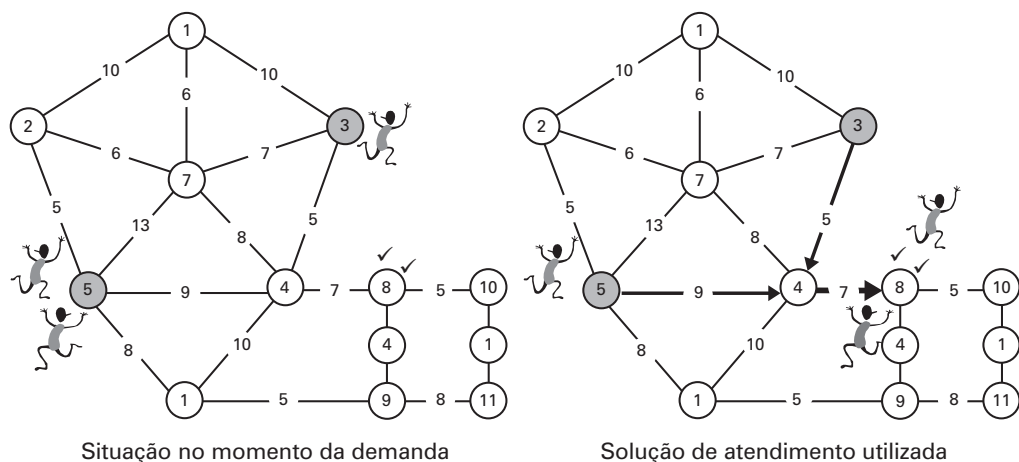


FIGURA 10.23 Atendimento em um modelo de múltiplos servos.

✓ Modelos hierárquicos para atendimento dinâmico

Eventualmente os clientes demandantes podem ser classificados de modo hierárquico de forma que as exigências para o atendimento possam ser relaxadas para certos subconjuntos de usuários. Charnes e Storbeck (1980) abordam situações hierarquizadas dessa natureza através da programação por objetivos. Alguns trabalhos nessa área são reportados por:

- Chuch e Eaton (1987) para localização de clínicas e hospitais.
- Gerrard e Chuch (1994) para sistemas de saúde com a capacidade de considerar várias funções objetivo e sobrecoberturas.
- ReVelle (1996) com um modelo multiobjetivo.

10.4.3 – Algoritmos Heurísticos de Solução

Apresentaremos dois algoritmos de solução para o problema, um para o caso *on-line* e outro para o *off-line*.

✓ Algoritmo de Sleator e Tarjan (para o caso *on-line*)

Sleator e Tarjan (1985), apresentam o algoritmo BAL para a solução do k -servos *on-line*. A estratégia é basicamente escolher os movimentos que permitam que os diversos servos desloquem-se de forma igualitária. Para cada servo k o algoritmo mantém a distância total de movimento equilibrada que será denominada D_k . Se existe uma demanda descoberta em um certo vértice j , o servo k é escolhido para realizar o atendimento de modo a mín $\{D_k + d_{kj}\}$, onde d_{kj} é a distância a ser percorrida pelo servo k até o vértice j , $K = \{1, \dots, k\}$ um conjunto de índices $S = \{s_k^i, i \in N, k \in K\}$ conjunto dos $|K|$ servos localizados nos vértices k , e, finalmente, j , o vértice onde a demanda se configurou. O algoritmo abaixo descreve a estratégia de Sleator para o equilíbrio dos deslocamentos a cada nova demanda surgida.

Algoritmo Sleator e Tarjan	
INÍCIO	
Ler o grafo $G = (N, A)$, D , S e J ;	
Inicializar variáveis $Sol \leftarrow \emptyset$;	
Para todo $s_k^i \in S$ calcule d_{kj}	
Determinar k tal que $\min \{D_k + d_{kj}\}$	
Fazer	
<div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;"> Início $Sol \leftarrow \{k\};$ $D_k \leftarrow D_k + d_{kj};$ $s_k^i \leftarrow s_k^j;$ </div>	
Fim	
FIM	

Examinando o algoritmo de Sleator e Tarjan, fica evidente que a estratégia não é capaz de garantir a minimização do deslocamento dos servos. Trata-se de um algoritmo *on-line*, ou seja, um algoritmo próprio para o tratamento de situações em que a demanda vai sendo informada à medida que os servos vão sendo movimentados. A situação exemplificada na Figura 10.24 mostra como essa heurística pode se afastar da política ótima. Os servos ocupam inicialmente os vértices 3 e 5 do grafo e o primeiro cha-

mado ocorre no vértice 4. Como ambos os servos nunca se deslocaram, o servo escolhido para o atendimento é o mais próximo ao ponto de demanda, ou seja, o servo S_1 que estava estacionado no vértice 5. O segundo chamado ocorre no vértice 2. Nesse caso, como S_2 já se deslocou em nove unidades, o designado é o servo S_1 . O total percorrido pelos servos para fazer ambos os atendimentos é $9 + 13 = 22$ unidades. Se S_2 fosse designado ao primeiro atendimento e S_1 ao segundo, o total de espaço percorrido seria $10 + 5 = 15$ unidades.

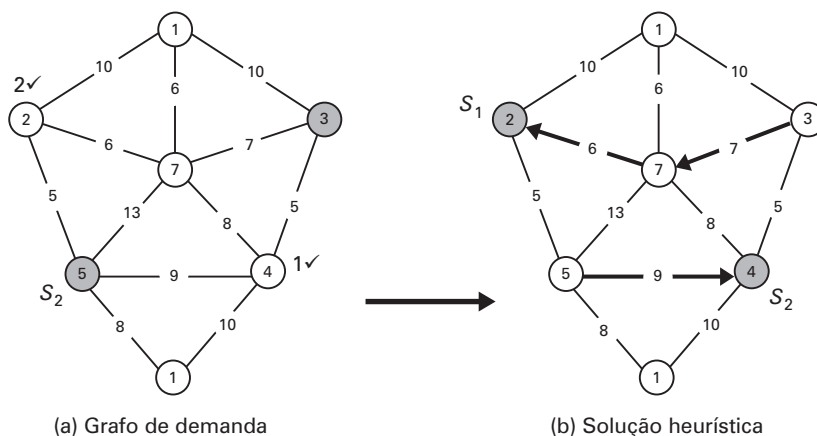


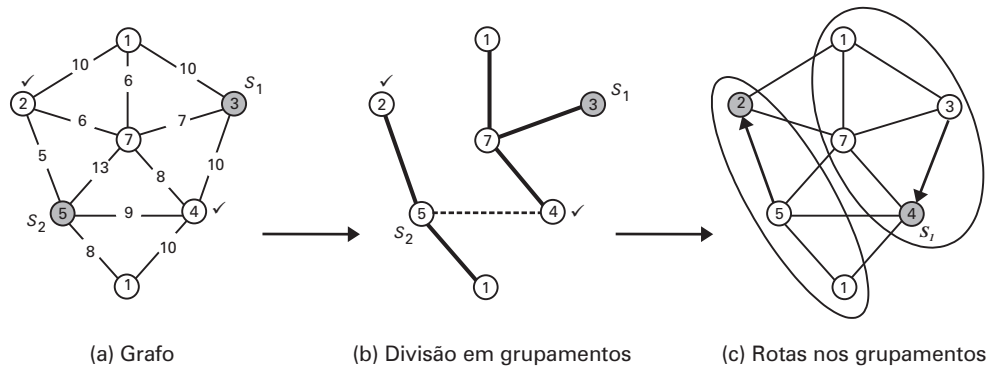
FIGURA 10.24 Solução online para o k -servo.

Se dispomos da informação de demanda logo no início do problema é possível evitar que uma situação como a verificada na aplicação do algoritmo anterior ocorra.

Manasse e Sleator (1990) apresentam outros algoritmos para o problema dos $(n - 1)$ servos e 2-servos. O seguinte algoritmo explora a relação entre o k -servos e o PDK para a solução de um problema *off-line*. Denominando por $J, J \subset N$ o conjunto do índice dos vértices que demandam por serviços. Separando G em k grupamentos de proximidade tendo por representantes os vértices do conjunto P , solução do PDK em $G, P = \{p_1, p_2, \dots, p_k\}$, e denominando ainda por $G_i = \{1, \dots, g_i\}$ os vértices alocados em cada um dos k grupamentos temos:

Algoritmo k _separação
<p>INÍCIO</p> <p>Ler o grafo $G = (N, A), S$ e J;</p> <p>Encontrar k grupamentos de proximidade em G</p> <p>Encontrar k vértices dispersos (solução do PDK)</p> <p>Atribuir cada grupamento de vértices G_i ao nó servo mais próximo ao representante de G_i</p> <p>Encontrar o caminho mais curto que une s_k^i aos nós de demanda pertencentes ao seu grupamento</p> <p>FIM</p>

Observamos que o objetivo da separação proposta para o problema dos servos é diferente do modelo de Erkut (1990) e denominado de *r-separação*. A Figura 10.25 exemplifica a aplicação do algoritmo k -separação ao caso da Figura 10.25 (a). No primeiro passo se formam os grupamentos de proximidade. Na segunda etapa, dentro de cada grupamento, a melhor rota é estabelecida para o servo que lhe foi designado.

FIGURA 10.25 Solução on-line para o k -servo.

O algoritmo da k -separação busca balancear os esforços dos servos ao separar uma área de trabalho para cada um deles. Por outro lado, o critério de minimização do deslocamento é considerado diretamente através do caminho mais curto que une os vértices demandados dentro de cada grupamento.

10.5 – PROBLEMAS PROPOSTOS

1 – O Problema da Exploração Sustentada de Madeira – Modelo por Lotes



Deseja-se explorar economicamente a madeira de uma certa área florestal de modo a reduzir o impacto ambiental da atividade, transformando-a em uma ação autossustentável. Para realizar tal objetivo, um determinado número de árvores devem ser retiradas anualmente dentro de uma configuração esparsa e uniformemente distribuída dentro da área, chamada de configuração anual de exploração.

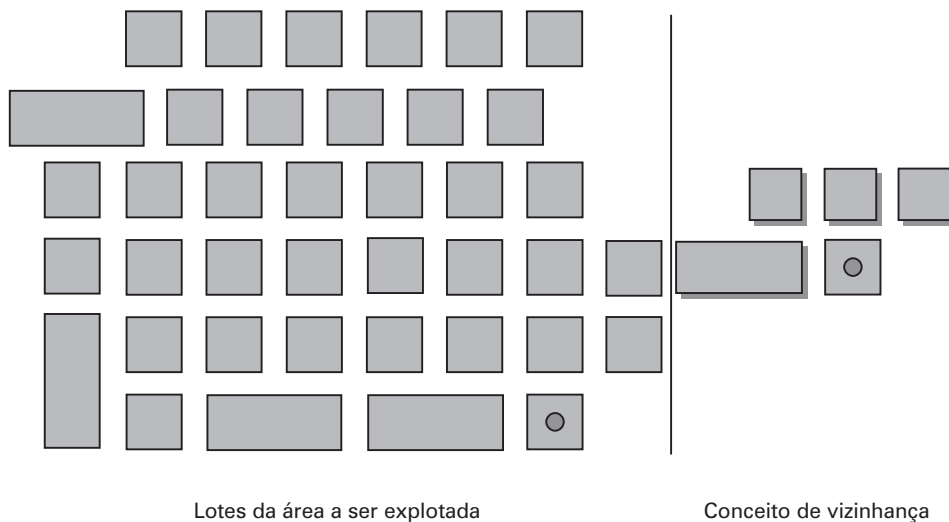


FIGURA 10.26 Loteamento de exploração.

Para operacionalizar o processo, a área de exploração foi loteada em r quadras, como mostra a Figura 10.26. Uma configuração anual de exploração é constituída pela escolha de um certo conjunto de quadras a serem atacadas e o número de árvores que serão retiradas em cada quadra. Dentro das quadras só poderão ser retiradas até n_i árvores $i = 1, \dots, r$. Se uma quadra for incluída em uma certa configuração de exploração, pelo menos duas quadras vizinhas não poderão ser incluídas na mesma configuração. Por motivos econômicos, é necessário que todas as quadras designadas na configuração do ano s possuam pelo menos uma quadra vizinha também na configuração do ano s . O conceito de vizi-

nhança entre quadras é esclarecido na figura em relação à quadra marcada. Observamos que a confrontação entre os vértices das quadras é considerada como uma relação de vizinhança. Deverão ser retiradas anualmente pelo menos k árvores da área. Uma vez tendo sido designada uma certa quadra à configuração do ano s , ela só pode voltar a ser designada novamente para a exploração a partir de $s + 3$. Formule o problema considerando que se deseja maximizar o número total de árvores a serem retiradas em um horizonte de 4 anos.

2 – O Problema da Exploração Sustentada de Madeira – Modelo Detalhado



Neste problema o objetivo é basicamente o mesmo do caso anterior, só que o fator de interferência ecológica é mais grave, por se tratar de um vegetal extremamente valioso, o que obriga a considerar individualmente as árvores. Seja o esquema abaixo, representando a localização de cada espécie vegetal pelo pequeno círculo escuro.

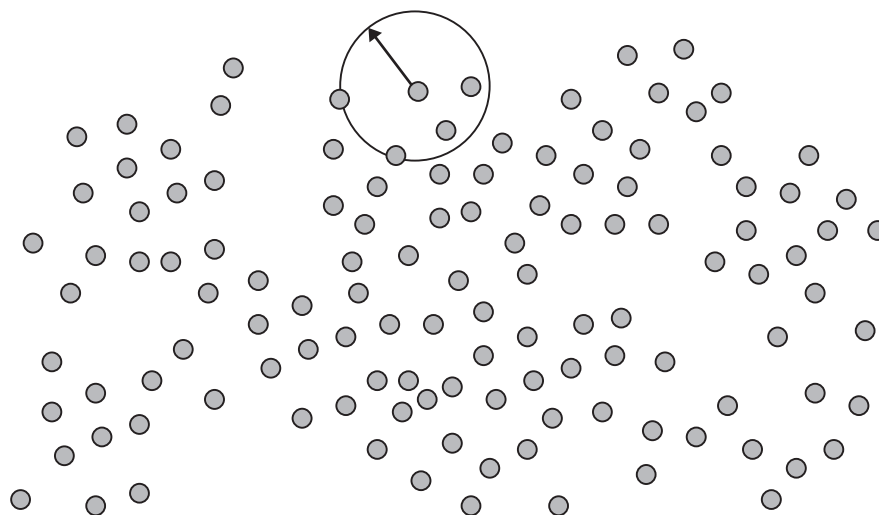


FIGURA 10.27 Distribuição vegetal.

Cada configuração anual de exploração é composta de um certo número de árvores. Se uma árvore é designada a uma configuração anual, nenhuma outra, dentro do círculo de exclusão (constante para todas as árvores) mostrado na figura do problema, pode ser designada na mesma configuração. Se uma árvore é retirada, um novo indivíduo é plantado em seu lugar, necessitando de 15 anos para alcançar a idade de corte. Se uma árvore não possui vizinhos (dentro do raio de exclusão) então não poderá ser cortada. Formular o problema de maximizar a retirada no horizonte de cinco anos.

3 – O Problema da Evacuação de Emergência – Caso da Usina Nuclear



Em uma certa região existe uma usina nuclear para geração de energia elétrica. Face à possibilidade da ocorrência de um vazamento de material radioativo (remota, mas real) é necessário a preparação de um plano de evacuação de emergência para a região circunvizinha. O plano deverá prever a retirada segura de pessoas, animais e o patrimônio essencial (obras de arte, símbolos da cultura e expressão nacional etc.) antes que possam sofrer os efeitos nocivos da exposição radioativa. A região da usina está esquematicamente representada na Figura 10.28:

O modelo proposto para a evacuação idealiza a concentração das pessoas, animais e patrimônio, em centros de triagem e evacuação. Nesses centros, as pessoas e animais poderão ser, inclusive, atendidas em caráter emergencial, caso já tenham sido expostas a algum tipo de efeito, minimizando-se com isso o impacto do acidente. A determinação do número e localização dos centros de triagem e evacuação transcende a dimensão econômica, uma vez que, para que sejam implantados,

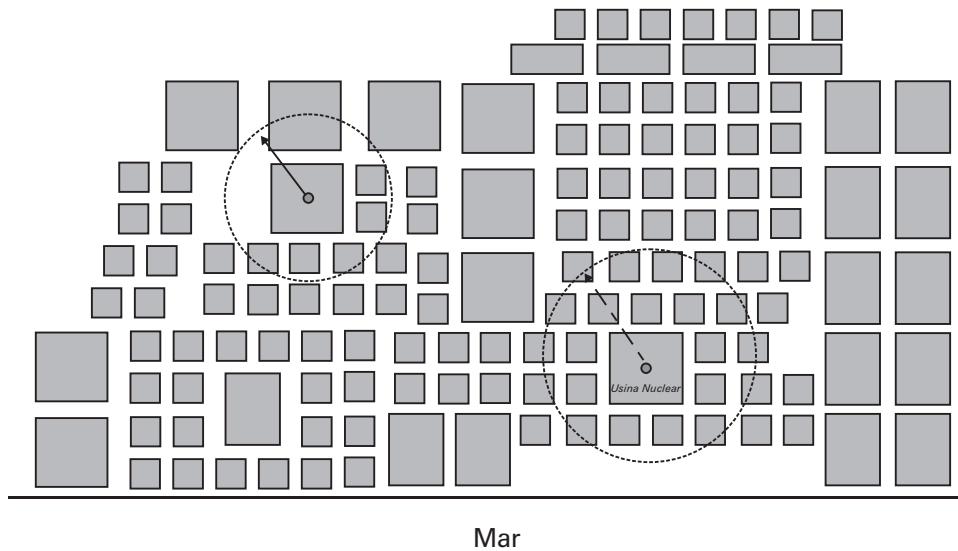


FIGURA 10.28 Distribuição de demanda.

pessoas devem ser introduzidas na área de risco, aumentando o risco humano. Se, por um lado, um número excessivo de centros dificultará a coordenação da evacuação e aumentará o risco de exposição dos seres humanos, por outro, um número pequeno ocasionará, certamente, insuficiência no atendimento.

As unidades de discretização (retângulos da figura) possuem as seguintes demandas:

$p_i \equiv$ número de pessoas que residem na área i .

$v_i \equiv$ número de animais que serão evacuados da área i .

$o_i \equiv$ volume patrimonial demandado pela área i .

Considere que cada unidade de discretização deve ser designada a um centro de triagem. Considere ainda que os centros de triagem serão unidades padronizadas para facilitar o treinamento do pessoal de atendimento, bem como permitir flexibilidade de redistribuição de pessoal (condições típicas das operações de emergência), podendo atender g pessoas, k animais e l unidades de volume patrimonial, dentro do horizonte de tempo julgado seguro para a operação. Levando em conta que nenhum ponto de demanda pode estar afastado mais que r quilômetros de um centro de triagem (como mostra o círculo menor da figura) e que nenhum centro de triagem poderá ser localizado dentro área de risco máximo da usina (como mostra o círculo maior da figura), formule o problema de minimizar o número de centros de triagem do sistema de evacuação.¹

4 – O Problema do Abastecimento de Água em Áreas de Chuva Irregular



Em uma região de regime de chuvas irregular deseja-se construir um sistema de captação e estocagem de águas pluviais e de poços artesianos, de modo que seja possível, através de uma operação integrada, realizar uma conveniente distribuição de água na época da seca. O sistema será composto de barragens subterrâneas e poços artesianos. Os pontos de demanda estão sinalizados na figura que se segue através dos pequenos círculos escuros, demandando cada ponto i uma quantidade f_i de litros de água por segundo. As barragens subterrâneas devem manter uma distância mínima d de outras barragens ou poços artesianos. A capacidade de cada poço é de p litros por segundo, enquanto a das barragens é de q litros por segundo. Os pontos de demanda devem ficar a uma distância máxima de r metros de sua unidade de abastecimento, tendo em vista os problemas operacionais decorrentes do transporte da água.

¹ Considere os pontos de triagem localizados no centro de gravidade dos retângulos de discretização.

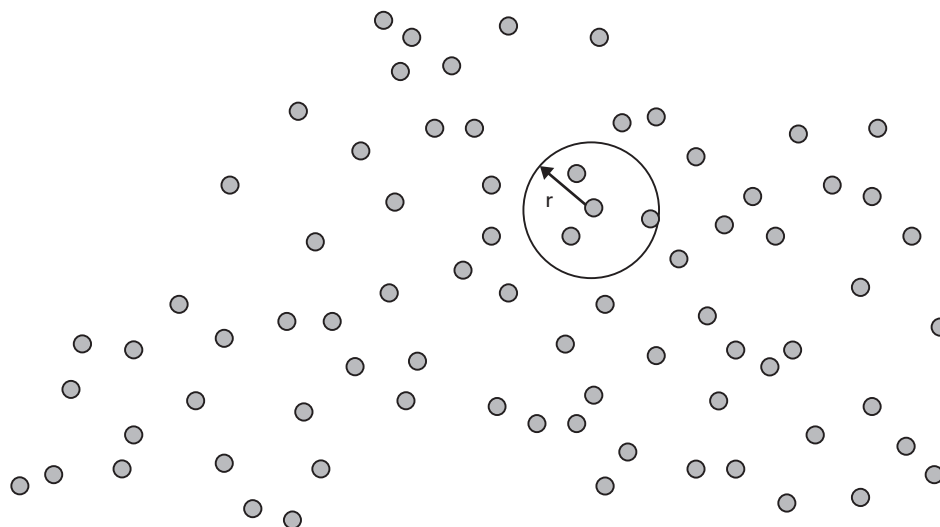


FIGURA 10.29 Localização das demandas por abastecimento de água.

Uma barragem custa s unidades monetárias, enquanto um poço t unidades. Barragens e poços podem ser construídas em qualquer ponto da área em análise.

Formule o problema de modo a minimizar o custo total do sistema e atender a todas as demandas identificadas.

5 – O Problema do Exame de Sangue



Para manter a qualidade do sangue utilizando-se a transmissão de doenças infecciosas, o governo mantém um eficiente sistema de exames. Existem cinco pacotes de exames capazes de verificar a utilizabilidade de cada bolsa de sangue que será distribuída aos hospitais. Os pacotes são compostos de t testes diferentes, cobrindo, cada um, áreas específicas. Consideraremos, neste caso, que os pacotes serão constituídos por quatro tipos de testes que verificarão quatro áreas e que, tanto as áreas como os testes são equivalentes, variando apenas em eficácia e em custo (ou seja, $t = 4$ em todos os pacotes). Cada pacote de exames é mais eficiente em uma certa parcela da população e para determinadas doenças. É possível testar uma bolsa simultaneamente em vários pacotes (ou até mesmo em todos), mas esse procedimento é extremamente custoso e resulta em um acréscimo de segurança pouco expressivo. O governo possui uma certa quantidade Q (em unidades monetárias) destinada ao processo de garantia de qualidade do sangue e pretende esgotá-la completamente com os exames. Sabendo-se que, através das informações que são colhidas na entrevista com o doador, é possível estabelecer-se um fator de adequação f_{ij}^k do teste i do pacote j à bolsa k . Sabe-se também que, por questão de consistência, cada bolsa deve passar pelo menos por todos os testes i de um pacote j , e que cada teste i do pacote j custa c_{ij} unidades monetárias para ser realizado, formular:

1. O problema de maximizar a adequação dos testes de um conjunto de $K = \{1, \dots, k\}$, bolsas de sangue, sujeita à restrição de esgotar os recursos disponíveis para tal fim.
2. Minimizar os gastos com os testes de um conjunto de $K = \{1, \dots, k\}$ bolsas de sangue, exigindo-se um padrão mínimo de adequação R_i , $i = 1, 2, 3, 4$, associado a cada tipo de teste que uma bolsa k será submetida, valor esse obtido pelo somatório dos fatores de adequação entre os testes realizados e a bolsa em exame.

A N E X O A

TÓPICOS EM ÁLGEBRA LINEAR E CONVEXIDADE

A.1 – MATRIZES E OPERAÇÃO COM MATRIZES

A.1.1 – Conceitos Elementares

Matriz é uma entidade matemática representada por um arranjo dimensional de números ou tabela. Como uma entidade matemática, toda matriz atende a certas propriedades. As matrizes mais conhecidas são arranjos bidimensionais em que denominamos as m fileiras horizontais de linhas e as n verticais de colunas. Exemplo:

$$A = \begin{bmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$m = 2; n = 3.$$

$$B = \begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$$

$$m = 2; n = 2.$$

A matriz B é chamada de matriz quadrada. Normalmente representamos uma matriz através de uma letra maiúscula, reservando as letras minúsculas para a representação de seus componentes. Como um arranjo multidimensional, podemos considerar que as matrizes são constituídas por unidades unidimensionais ou vetores. Nesse caso, as linhas ou colunas da tabela podem ser interpretadas como vetores linha ou coluna. Quando uma matriz for reduzida a um vetor linha ou coluna mesmo assim podemos ainda denominá-la de matriz, esse é o caso das matrizes coluna ou linha. Exemplo:

Matriz Coluna

$$C_{3 \times 1} = \begin{bmatrix} 4 \\ 7 \\ 3 \end{bmatrix}$$

Matriz Linha

$$D_{1 \times 3} = [4 \quad 5 \quad 6]$$

Os elementos individuais de uma matriz são representados pela letra minúscula da matriz acompanhada dos índices que correspondem às suas coordenadas no quadro. Ao primeiro índice associamos a coordenada horizontal ou o número de sua linha. Ao segundo índice o número da coluna correspondente. Exemplo:

$$C = [c_{ij}] = \begin{bmatrix} c_{11} = 4 & c_{12} = 3 \\ c_{21} = -1 & c_{22} = 5 \end{bmatrix}$$

Genericamente:

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

A.1.2 – Operações Elementares

✓ Soma e Subtração de Matrizes

A condição para que duas matrizes sejam passíveis de operarem em soma ou subtração é que possuam o mesmo número de linhas e colunas.

O operador neutro na soma de matrizes é a matriz nula (ver matriz C). A soma ou subtração de matrizes é efetuada operando-se elemento a elemento de cada matriz envolvida. Exemplo:

Somar as seguintes matrizes: $D = A + B + C$, onde

$$A = \begin{bmatrix} 4 & -1 \\ 7 & 2 \end{bmatrix}, B = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}, C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Nesse caso obtemos D somando elemento a elemento as matrizes A , B e C (matriz nula).

$$D = \begin{bmatrix} 4 + 1 + 0 = 5 & -1 + 1 + 0 = 0 \\ 7 + 2 + 0 = 9 & 2 - 1 + 0 = 1 \end{bmatrix}$$

✓ Multiplicação por um Número Real

Dada uma matriz A e um número escalar (ou real) λ , o produto $\lambda \times A$ é uma matriz em que todos os elementos de A foram multiplicados pelo valor de λ . Exemplo:

$$\text{Se } A = \begin{bmatrix} 4 & -1 \\ 7 & 2 \end{bmatrix} \text{ e } \lambda = -1 \text{ então } \lambda \times A = \begin{bmatrix} -4 & 1 \\ -7 & -2 \end{bmatrix}$$

✓ Produto de Matrizes

Trata-se de uma das operações mais importantes da álgebra linear. Sejam as matrizes $A = [a_{ij}]$; $B = [b_{ij}]$; $C = [c_{ij}]$; e sabemos que $C = A \times B$. Então poderemos obter os elementos de C , os seus c_{ij} , através da seguinte regra:

O elemento i, j da matriz produto será obtido através da soma dos produtos dos elementos correspondentes da linha i da primeira matriz, pelos elementos da coluna j da segunda matriz. Exemplo:

Sejam:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, B = \begin{bmatrix} 2 & 1 \\ -1 & -1 \end{bmatrix}$$

Então, o elemento c_{12} da matriz C deve ser obtido pela soma do produto entre os elementos da primeira linha de A , (a_{11}, a_{12}) com os elementos da segunda coluna de B , (b_{12}, b_{22}) . Nesse caso:

$$c_{12} = (a_{11} \times b_{12}) + (a_{12} \times b_{22}) = +1 - 2 = -1$$

E assim por diante. No exemplo:

$$C = [c_{ij}] = \begin{bmatrix} c_{11} = 0 & c_{12} = -1 \\ c_{21} = 2 & c_{22} = -1 \end{bmatrix}$$

O operador neutro no produto de matrizes é chamado de matriz identidade. Exemplo:

$$I = [i_{ij}] = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

Pela definição da operação de produto de matrizes ela só poderá ocorrer quando o número de colunas da primeira matriz for igual ao número de linhas da segunda.

A matriz resultante de um produto $A \times B$ possuirá o número de linhas da primeira e o número de colunas da segunda. Exemplo:

$$A_{s \times p} \times B_{p \times v} = C_{s \times v}$$

O produto de matrizes não é uma operação comutativa como o é o produto de número reais. Normalmente $A \times B \neq B \times A$. Se $A \times B = B \times A$, então diremos que A e B são matrizes comutativas. Outro fenômeno interessante de observar é que se $A \times C = A \times B$ não podemos afirmar que $C = B$.

No caso do produto com a matriz identidade $A \times I = A$ se a matriz A for quadrada então $A \times I = I \times A$.

O produto de duas matrizes não nulas pode ser uma matriz nula.

✓ Transposição de Matrizes

Trata-se de uma operação em que trocamos as linhas da matriz operada por suas colunas e vice-versa. O símbolo de transposição é o expoente t .

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ e } A^t = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

$$A_{s \times p} \Rightarrow A^t_{p \times s}$$

Algumas propriedades da transposição:

1. $(kA)^t = k(A)^t$, para todo número escalar k .
2. $(A \times B)^t = (B)^t \times (A)^t$.
3. $((A)^t)^t = A$.
4. $(A + B)^t = (A)^t + (B)^t$.

✓ *Diagonal Principal e Escalonamento*

$$A = [a_{ij}] = \begin{bmatrix} \otimes & Bs & \dots & Bs \\ Bi & \otimes & \dots & Bs \\ \ddots & \ddots & \dots & \ddots \\ Bi & Bi & \dots & \otimes \end{bmatrix}$$

A diagonal formada pelo símbolo \otimes é dita principal. A diagonal principal divide a matriz na banda superior e banda inferior. Quando uma matriz apresenta a banda superior ou inferior composta por zeros e a diagonal principal unitária, dizemos que a matriz está escalonada à direita ou à esquerda.

Exemplo de escalonamento à esquerda:

$$A = [a_{ij}] = \begin{bmatrix} 1 & ??? & \dots & ??? \\ 0 & 1 & \dots & ??? \\ \ddots & \ddots & \dots & \ddots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

A.1.3 – Particionamento de Matrizes

As matrizes podem ser particionadas em submatrizes. Essa providência pode ser muito útil no cálculo de invariantes, demonstração de teoremas e solução de sistemas de equação. Nesse caso, serão utilizadas as notações dos exemplos que se seguem.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

pode ser particionada em quatro matrizes a saber:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

ou ainda,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

pode ser particionada em quatro matrizes colunas a saber:

$$A = [A_{11} \quad A_{12} \quad A_{13} \quad A_{14}]$$

A.1.4 – Equações Matriciais

As equações matriciais são, na verdade, formas compactas de expressar as tradicionais equações da álgebra, por exemplo, dados:



$$Ax = B$$

A.1.5 – Inversão de Matrizes

A inversa de uma matriz A , quadrada, denominada A^{-1} , é a matriz que atende a seguinte condição:

Propriedade básica das matrizes inversas:

$$A \times A^{-1} = I = A^{-1} \times A$$

outras propriedades:

$$\begin{aligned}(A^{-1})^{-1} &= A \\ (I^{-1})^{-1} &= I \\ (k \times A)^{-1} &= k^{-1} \times A^{-1} \\ (A \times B)^{-1} &= B^{-1} \times A^{-1} \\ (A \times B \times C)^{-1} &= C^{-1} \times B^{-1} \times A^{-1}\end{aligned}$$

Nem todas as matrizes quadradas possuem inversas. Uma matriz que atende à propriedade anterior é dita *inversível*. Em caso contrário a matriz é denominada *singular*. A condição necessária e suficiente para que uma matriz quadrada seja inversível é que seu determinante seja diferente de zero. E o que vem a ser um determinante?

Definição 1:

Chama-se determinante (Δ) associado à matriz M , ao polinômio obtido fazendo a soma algébrica de todos os termos constituídos pelos produtos de n elementos do quadro, tomados de tal sorte que os primeiros índices formem a permutação natural e os segundos as permutações possíveis, devendo-se ainda atribuir a cada termo o sinal positivo ou negativo conforme a permutação dos índices seja de primeira ou segunda classe.

O determinante de uma matriz é um valor escalar, ou seja, um número real e pode ser obtido pela expressão que se segue:

$$\det A = \sum_{i=1}^n a_{i1} A_{i1}$$

onde A_{i1} é o cofator de a_{i1} definido pelo produto de $(-1)^{i+1}$ com o determinante de uma submatriz de A , obtida pela retirada da i -ésima linha e da primeira coluna de A .

$$\det \begin{vmatrix} 2 & 3 & 7 \\ 4 & 5 & 11 \\ 6 & 7 & 9 \end{vmatrix} = 2A_{11} + 4A_{21} + 6A_{31} =$$

$$\det \begin{vmatrix} 5 & 11 \\ 7 & 9 \end{vmatrix} - 4 \det \begin{vmatrix} 3 & 7 \\ 7 & 9 \end{vmatrix} + 6 \det \begin{vmatrix} 3 & 7 \\ 5 & 11 \end{vmatrix}$$

O sistema de redução recursiva proposto por essas fórmulas acaba invariavelmente no cálculo de determinantes de matrizes 2×2 cuja expressão é muito simples. Por outro lado, através do cálculo dos determinantes podemos obter o valor das variáveis ou incógnitas dos sistemas lineares através da expressão:

$$x_j = \frac{\det A_j}{\det A}, j = 1, 2, \dots, n$$

São clássicas três grandes linhas de atuação para realizar a determinação da inversa de A , a saber:

- Aplicando estratégias baseadas na definição da inversa de A .
- Utilizando os determinantes.
- Através de operações elementares.

Demonstraremos o método básico da inversão via operações elementares exatamente porque esse método dará suporte à ação do algoritmo simplex.

✓ Inversão de Matrizes pelo Método das Operações Elementares

Seja a equação matricial:

$$Ax = B$$

Utilizando as propriedades, não alteramos $Ax = B$ quando multiplicamos qualquer dos membros pela matriz identidade, assim:

$$Ax = IB$$

é equivalente a $Ax = B$.

O processo de solução de $Ax = B$ diz que:

$$x = A^{-1} B$$

ou, pelo mesmo raciocínio anterior,

$$Ix = A^{-1} B$$

Comparando as equações acima observamos que o processo de inversão de matrizes pode ser pensado como um processo de transformação de uma matriz A qualquer em uma matriz identidade, através de operações que não violem as condições do sistema linear original. Tendo em vista o exposto, podemos imaginar um método de inversão para a matriz A , através de operações elementares em suas linhas. Seja a matriz A a seguir representada e a matriz I ao seu lado. Quando A for transformada em uma matriz identidade a matriz I terá acumulado o valor da matriz A^{-1} .¹

$$A = \begin{bmatrix} 1 & -1 & 1 \\ 0 & -2 & 0 \\ 0 & 1 & 3 \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Vamos utilizar uma estratégia de alteração de A em uma matriz identidade, partindo do canto superior esquerdo e progredindo coluna a coluna.

Operação 1: Tornar o elemento $a_{11} = 1$. Nada para fazer, pois $a_{11} = 1$ em A .

¹ Devemos utilizar uma estratégia de mudança racional e cuidadosa para evitar cálculos desnecessários. Isso permitirá que o escalonamento de A (os números zero na matriz) seja planejado de forma metódica. Observe, no exemplo, que o elemento a_{23} já era zero inicialmente, como se faria necessário, entretanto seu valor será alterado com a primeira operação. Sem uma estratégia racional de cálculo esse fenômeno pode ocorrer em células já calculadas.

Operação 2: Tornar o elemento $a_{21} = 0$. Para tal devemos multiplicar a primeira linha por -1 e somar com a segunda linha resultando:

$$A = \begin{bmatrix} 1 & -1 & 1 \\ 0 & -1 & -1 \\ 0 & 1 & 3 \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Operação 3: Tornar o elemento $a_{12} = 0$. Soma-se a terceira linha à primeira resultando:

$$A = \begin{bmatrix} 1 & 0 & 4 \\ 0 & -1 & -1 \\ 0 & 1 & 3 \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Operação 4: Tornar o elemento $a_{22} = 1$. Multiplica-se a segunda linha por -1 resultando:

$$A = \begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 1 \\ 0 & 1 & 3 \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Operação 5: Tornar o elemento $a_{23} = 0$. Multiplica-se a segunda linha por -1 e soma-se com a terceira resultando:

$$A = \begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 & 1 \\ 1 & -1 & 0 \\ -1 & 1 & 1 \end{bmatrix}$$

Operação 6: Tornar o elemento $a_{33} = 1$. Divide-se a terceira linha por 2 resultando:

$$A = \begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 & 1 \\ 1 & -1 & 0 \\ -1/2 & 1/2 & 1/2 \end{bmatrix}$$

Operação 7: Tornar o elemento $a_{23} = 0$. Multiplica-se a terceira linha por -1 e soma-se à segunda resultando:

$$A = \begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 & 1 \\ 3/2 & -3/2 & -1/2 \\ -1/2 & 1/2 & 1/2 \end{bmatrix}$$

Operação 8: Tornar o elemento $a_{13} = 0$. Multiplica-se a terceira linha por -4 e soma-se à primeira resultando:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad I = \begin{bmatrix} 1 & -2 & -1 \\ 3/2 & -3/2 & -1/2 \\ -1/2 & 1/2 & 1/2 \end{bmatrix}$$

A.1.6 – Propriedade da Inversão da Matriz Aumentada

Seja a matriz A e o vetor B , conforme se segue:

$$A = \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix}, B = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Suponhamos que desejemos realizar a seguinte operação com as matrizes A e B :

$$A^{-1} B$$

Podemos realizar a operação diretamente, obtendo a inversa de A e a multiplicando por B ou através do próprio processo de inversão de A , utilizando uma matriz aumentada como se segue:

$$A^+ = \left[\begin{array}{cc|c} 1 & 1 & 2 \\ -1 & 0 & 1 \end{array} \right] = [A|B]$$

Aplicando o processo de inversão na matriz aumentada através do método das operações elementares temos:

- Agregação da matriz identidade

$$A^+ = \left[\begin{array}{cc|c} 1 & 1 & 2 \\ -1 & 0 & 1 \end{array} \right] \quad I = \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right]$$

- Operação de canonização da primeira coluna, somando a primeira linha à segunda:

$$A^+ = \left[\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 & 3 \end{array} \right] \quad I = \left[\begin{array}{cc} 1 & 0 \\ 1 & 1 \end{array} \right]$$

- Operação de canonização da segunda coluna, multiplicando a segunda linha da aumentada por -1 e somando à segunda linha, concluindo-se o processo de obtenção da inversa de A

$$A^+ = \left[\begin{array}{cc|c} 1 & 0 & -1 \\ 0 & 1 & 3 \end{array} \right] \quad I = \left[\begin{array}{cc} 0 & -1 \\ 1 & 1 \end{array} \right]$$

Observe que:

$$A^{-1}B = \left[\begin{array}{cc} 0 & -1 \\ 1 & 1 \end{array} \right] \times \left[\begin{array}{c} 2 \\ 1 \end{array} \right] = \left[\begin{array}{c} -1 \\ 3 \end{array} \right]$$

foi obtido diretamente no quadro aumentado. Ou seja, quando realizamos a inversão da matriz A , no quadro aumentado, os vetores agregados resultantes representam o produto dos vetores originais pela inversa de A . O quadro final do processo será:

$$[I|A^{-1}B][A^{-1}]$$

A.2 – ESTUDO DOS SISTEMAS DE EQUAÇÕES LINEARES

A.2.1 – Tipos de Sistemas Lineares

Um sistema de equações lineares pode admitir uma solução, infinitas soluções ou ser incompatível. Na determinação das soluções dos sistemas lineares o teorema de Cramer é de extrema valia:

Teorema de Cramer:

Qualquer sistema de n equações lineares com n incógnitas, cujo determinante dos coeficientes das incógnitas é diferente de zero, admite uma única solução.

Um sistema linear da forma:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= 0 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= 0 \\ \vdots &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= 0 \end{aligned}$$

é denominado *homogêneo*. Nesse caso:

1. Se $\Delta \neq 0$ os valores das incógnitas são todos zero e a solução é denominada *trivial*.
2. Se $\Delta = 0$ o sistema admite infinitas soluções distintas e é dito *indeterminado*.

Dado um sistema linear com p equações e n incógnitas, $p < n$ da forma:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1p}x_p + a_{1p+1}x_{p+1} + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2p}x_p + a_{2p+1}x_{p+1} + \dots + a_{2n}x_n &= b_2 \\ \vdots &\vdots \\ a_{p1}x_1 + a_{p2}x_2 + \dots + a_{pp}x_p + a_{pp+1}x_{p+1} + \dots + a_{pn}x_n &= b_n \end{aligned}$$

Consideremos a matriz formada pelos coeficientes das incógnitas:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1p} & a_{1p+1} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2p} & a_{2p+1} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{p1} & a_{p2} & \dots & a_{pp} & a_{pp+1} & \dots & a_{pn} \end{bmatrix}$$

Dela podemos deduzir diversos determinantes de ordem p . Supondo pelo menos um deles diferente de zero e organizando as incógnitas e restrições podemos ter:

$$\Delta = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{p1} & a_{p2} & \dots & a_{pn} \end{vmatrix} \neq 0 \quad (02)$$

O sistema proposto pode ser reescrito da seguinte forma:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1p}x_p &= b_1 - a_{1p+1}x_{p+1} - \dots - a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2p}x_p &= b_2 + a_{2p+1}x_{p+1} - \dots - a_{2n}x_n \\ \dots &\dots \\ a_{p1}x_1 + a_{p2}x_2 + \dots + a_{pp}x_p &= b_n + a_{pp+1}x_{p+1} - \dots - a_{pn}x_n \end{aligned} \quad (03)$$

Como o sistema da direita possui $\Delta \neq 0$, podemos determinar os valores das p incógnitas x_1, x_2, \dots, x_p (incógnitas principais), em função das $n - p$ restantes (incógnitas não principais) de tal sorte que a cada conjunto de valores atribuídos arbitrariamente a estas corresponderá um conjunto determinado de valores para aquelas. Vemos que o sistema proposto, dentro da hipótese feita, é sempre indeterminado.

A.2.2 – Dependência Linear entre Polinômios

Antes de abordarmos o conceito de *dependência linear* vamos definir a propriedade de *combinação linear*, que pode ser expressa pela seguinte equação:

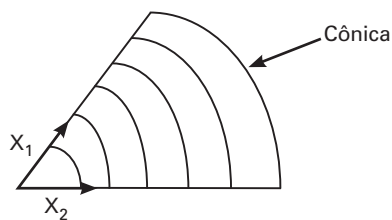
$$x = \lambda_1 x_1 + \lambda_2 x_2 + \dots + \lambda_n x_n$$

onde, lembrando novamente, x é um vetor de dimensão n e λ um valor escalar.

Existem diversos casos particulares de combinação linear.

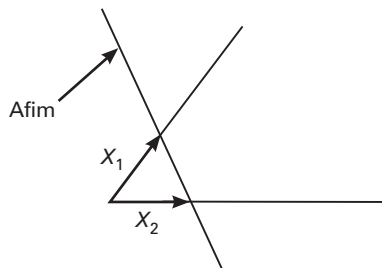
✓ Combinação Cônica

$$x = \lambda_1 x_1 + \lambda_2 x_2 + \dots + \lambda_n x_n \text{ quando } \lambda_i \geq 0$$



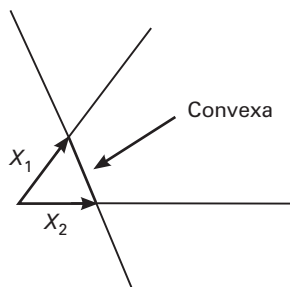
✓ Combinação Afim

$$x = \lambda_1 x_1 + \lambda_2 x_2 + \dots + \lambda_n x_n \text{ quando } \sum_{i=1}^n \lambda_i = 1$$



✓ Combinação Convexa

$$x = \lambda_1 x_1 + \lambda_2 x_2 + \dots + \lambda_n x_n \text{ quando } \sum_{i=1}^n \lambda_i = 1 \text{ e } \lambda_i \geq 0$$



A noção de dependência linear pode ser encarada como uma generalização do conceito de proporcionalidade entre várias grandezas. Consideremos dois conjuntos de números, a saber:

$$x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)} \text{ e } x_1^{(2)}, x_2^{(2)}, \dots, x_n^{(2)}$$

dizem-se proporcionais um ao outro quando é possível determinar duas constantes λ_1, λ_2 não simultaneamente nulas tais que:

$$\lambda_1 x_i^{(1)} + \lambda_2 x_i^{(2)} = 0 \quad (i = 1, 2, \dots, n)$$

Quando $\lambda_1 \neq 0$, deduzimos:

$$x_1^{(1)} = \frac{-\lambda_2}{\lambda_1} x_1^{(2)}, x_2^{(1)} = \frac{-\lambda_2}{\lambda_1} x_2^{(2)}, \dots, x_n^{(1)} = \frac{-\lambda_2}{\lambda_1} x_n^{(2)}$$

Quando $\lambda_2 \neq 0$, deduzimos:

$$x_1^{(2)} = \frac{-\lambda_1}{\lambda_2} x_1^{(1)}, x_2^{(2)} = \frac{-\lambda_1}{\lambda_2} x_2^{(1)}, \dots, x_n^{(2)} = \frac{-\lambda_1}{\lambda_2} x_n^{(1)}$$

A definição anterior tem a vantagem, como caso particular, de que um dos conjuntos é constituído por zeros. Os polinômios f_1, f_2, \dots, f_m , anteriormente definidos, se dizem linearmente dependentes quando podemos determinar m constantes, não todas nulas $\lambda_1, \lambda_2, \dots, \lambda_m$ de tal sorte que:

$$f_1 \lambda_1 + f_2 \lambda_2 + \dots + f_m \lambda_m \equiv 0$$

A.2.3 – Discussão de Sistemas de Equações com Matrizes de Coeficientes Quadradas

Sabemos que, basicamente, um sistema de equações pode admitir as seguintes condições de solução.

- Sistema Compatível:
 - Com uma solução.
 - Com infinitas soluções.
- Sistema Incompatível.

Vamos agora explorar o método das operações elementares para identificar configurações típicas para a discussão dos sistemas lineares em que o número de incógnitas é igual ao número de equações. Para tal vamos introduzir o conceito de matriz aumentada.

Denominaremos a matriz formada pelos coeficientes do sistema de equações lineares e seus termos independentes como *matriz aumentada do sistema*.

Exemplo 1:

Seja o sistema de equações lineares abaixo:

$$\begin{aligned} x_1 + 2x_2 + x_3 &= 1 \\ 2x_1 - x_2 + 2x_3 &= 2 \\ 3x_1 + x_2 + 3x_3 &= 4 \end{aligned}$$

a matriz aumentada do sistema é:

$$S = \left[\begin{array}{ccc|c} 1 & 2 & 1 & 1 \\ 2 & -1 & 2 & 2 \\ 3 & 1 & 3 & 4 \end{array} \right]$$

Por operações elementares vamos transformar a matriz S em uma matriz escalonada e estudar o comportamento do sistema de equações:

Operação 1: Anulando o elemento S_{21} multiplicando a linha 1 por -2 e somando-a à linha 2:

$$S = \left[\begin{array}{ccc|c} 1 & 2 & 1 & 1 \\ 0 & -5 & 0 & 0 \\ 3 & 1 & 3 & 4 \end{array} \right]$$

Operação 2: Anulando o elemento S_{31} multiplicando a linha 1 por -3 e somando-a à linha 3:

$$S = \left[\begin{array}{ccc|c} 1 & 2 & 1 & 1 \\ 0 & -5 & 0 & 0 \\ 0 & -5 & 0 & 1 \end{array} \right]$$

Operação 3: Tornando unitário o elemento S_{22} dividindo a linha 2 por -5 :

$$S = \left[\begin{array}{ccc|c} 1 & 2 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & -5 & 0 & 1 \end{array} \right]$$

Operação 4: Tornando unitário o elemento S_{32} multiplicando a a linha 2 por 5 e somando-a à linha 3:

$$S = \left[\begin{array}{ccc|c} 1 & 2 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right]$$

Como podemos observar na última linha da matriz escalonada S , existe uma inconsistência no sistema de forma que a solução é impossível.

Exemplo 2:

Seja o sistema de equações lineares que se segue:

$$\begin{aligned} x_1 + 3x_2 + x_3 &= 5 \\ 3x_1 - 2x_2 - 8x_3 &= -7 \\ 3x_1 + x_2 + 3x_3 &= 4 \end{aligned}$$

A matriz aumentada do sistema é:

$$S = \left[\begin{array}{ccc|c} 1 & 3 & 1 & 5 \\ 3 & -2 & -8 & -7 \\ 3 & 1 & 3 & 4 \end{array} \right]$$

Por operações elementares vamos transformar a matriz S em uma matriz escalonada e estudar o comportamento do sistema de equações:

Operação 1: Anulando o elemento S_{21} multiplicando a linha 1 por -3 e somando-a à linha 2:

$$S = \left[\begin{array}{ccc|c} 1 & 3 & 1 & 5 \\ 0 & -11 & -11 & -22 \\ 3 & 1 & 3 & 4 \end{array} \right]$$

Operação 2: Anulando o elemento S_{31} multiplicando a linha 1 por -3 e somando-a à linha 3:

$$S = \left[\begin{array}{ccc|c} 1 & 3 & 1 & 5 \\ 0 & -11 & -11 & -22 \\ 0 & -8 & 0 & -11 \end{array} \right]$$

Operação 3: Tornando unitário o elemento S_{22} dividindo a linha 2 por -11 :

$$S = \left[\begin{array}{ccc|c} 1 & 3 & 1 & 5 \\ 0 & 1 & 1 & 2 \\ 0 & -8 & 0 & -11 \end{array} \right]$$

Operação 4: Tornando unitário o elemento S_{32} multiplicando a linha 2 por 8 e somando-a à linha 3:

$$S = \left[\begin{array}{ccc|c} 1 & 3 & 1 & 5 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 8 & 5 \end{array} \right]$$

Operação 4: Tornando unitário o elemento S_{33} dividindo a linha 3 por 8:

$$S = \left[\begin{array}{ccc|c} 1 & 3 & 1 & 5 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 5/8 \end{array} \right]$$

Solução: Sistema determinado, com uma solução.

A.2.4 – Discussão de Sistemas de Equações com Matrizes de Coeficientes Retangular

Apesar de genericamente os sistemas de equações lineares serem do tipo (01) e, no caso de serem solucionados, acabarem reduzidos a conjuntos de sistemas do tipo (03), vamos aplicar a técnica das operações elementares para sua solução e verificar como pode ocorrer na prática esse processo de redução de sistemas do tipo (01) a sistemas do tipo (03). Seja o exemplo seguinte:

$$\begin{aligned} x_1 + x_2 - x_3 + x_4 + 2x_6 &= 12 \\ -x_1 + 2x_2 + 2x_3 + x_5 &= 6 \end{aligned} \tag{04}$$

que nos levará à seguinte matriz aumentada:

$$S = \left[\begin{array}{cccccc|c} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & b \\ 1 & 1 & -1 & 1 & 0 & 2 & 12 \\ -1 & 2 & 2 & 0 & 1 & 0 & 6 \end{array} \right]$$

Observando a matriz aumentada do sistema (04) encontramos uma solução imediata para o sistema, ou seja, uma submatriz do tipo (03) a ser composta pelas variáveis x_3 e x_5 . Então, para efeito de uma solução, o sistema indeterminado (04) pode ser reescrito da seguinte forma:

$$S = \begin{array}{cc|c} x_4 & x_5 & b \\ \hline 1 & 0 & 12 \\ 0 & 1 & 6 \end{array}$$

fixando-se as demais variáveis a zero. Nesse caso a solução seria: $x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 12, x_5 = 6$, ou em:

$$Ax = B$$

$$x = (0, 0, 0, 12, 6)$$

Nesse ponto é interessante esclarecer melhor o processo de seleção das submatrizes (03). Como ficou claro na dedução de (03) não é qualquer submatriz que pode solucionar nosso sistema de equações. Se houvéssemos optado por selecionar a submatriz abaixo, por exemplo:

$$S = \begin{array}{cc|c} x_4 & x_5 & b \\ \hline 1 & 2 & 12 \\ 0 & 1 & 6 \end{array}$$

O determinante da submatriz (03) seria singular e a decomposição não seria válida! Apesar de esse fato ser, matematicamente, já de nosso conhecimento, vamos agora justificá-lo também à luz do cálculo vetorial.

Realmente podemos considerar as colunas da matriz aumentada como vetores e uma solução para o conjunto de equações do sistema como uma combinação linear entre esses vetores. No exemplo a combinação linear que escolhemos inicialmente para solucionar o nosso sistema será:

$$\lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6) = (0, 0, 0, 12, 6, 0)$$

o que leva à seguinte combinação linear:

$$\begin{aligned} X &= \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 + \lambda_4 x_4 + \lambda_5 x_5 + \lambda_6 x_6 = \\ &= 0 \begin{pmatrix} 1 \\ -1 \end{pmatrix} + 0 \begin{pmatrix} 1 \\ 2 \end{pmatrix} + 0 \begin{pmatrix} -1 \\ 2 \end{pmatrix} + 12 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + 6 \begin{pmatrix} 0 \\ 1 \end{pmatrix} + 0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 12 \\ 6 \\ 0 \end{pmatrix} \end{aligned}$$

Os seguintes conceitos são úteis agora:

Definição 2:

Um *espaço vetorial* a n dimensões é o conjunto de todos os vetores de n componentes.

As variáveis e a solução de um sistema de equações lineares podem ser considerados vetores de um espaço de dimensão m , como acabamos de demonstrar no exemplo anterior.

Definição 3:

Um conjunto de vetores é dito *conjunto gerador* de um espaço vetorial, se todos os vetores do espaço puderem ser expressos como uma combinação linear dos vetores deste conjunto gerador.

Definição 4:

Uma *base* de um espaço vetorial é um conjunto gerador do espaço, composto de m vetores *linearmente independentes*. As variáveis associadas a essas colunas denominamos variáveis básicas.

No exemplo que examinamos x_4 e x_5 formam, inclusive, uma base canônica do espaço R^2 ($m = 2$). Dessa forma, existe uma equivalência de conceitos extremamente interessante: obter um sistema consistente do tipo (03) seria uma providência que dependeria apenas de operações algébricas se não fosse o fato de que nem todos os subconjuntos de m variáveis da matriz A possuem em determinante diferente de zero. Os subconjuntos que podem solucionar de forma independente (permitindo fixar os demais em zero) o sistema de equações constituem uma base de dimensão m , ou seja, são variáveis linearmente independentes no conjunto das restrições do sistema de equações, ou ainda: uma base de A é uma matriz de m vetores coluna linearmente independentes de A .

Definição 5:

A cada base corresponde uma e somente uma solução para o sistema. A essa solução denominaremos solução *básica*.

Definição 6:

Quando em uma solução básica todas as componentes do vetor solução são positivas, ela é denominada solução *básica viável*.

É claro que os sistemas indeterminados com um grande número de variáveis irão possuir um grande número de submatrizes que atenderão ao requisito de constituírem uma base. Sabendo que existem n variáveis no sistema e os subconjuntos devem ser formados por apenas m , então no máximo $\binom{n}{m} = \frac{n!}{m!(n-m)!}$ dessas bases poderiam teoricamente ser formadas. É nesse ponto que temos de esclarecer por que estaríamos nos preocupando tanto em trabalhar com esse enorme número de subconjuntos de variáveis da matriz A , em vez de nos aprofundarmos em métodos que solucionassem rapidamente os sistemas indeterminados, mesmo que as variáveis tivessem de compartilhar valores. Enfim, por que não estaríamos interessados em obter soluções do tipo $(1, 0, 0, 6, 7, 3)$, soluções não básicas, mas que, eventualmente, poderiam ser até mais fáceis de calcular?

A.3 – NOÇÕES DE UNIMODULARIDADE

A Unimodularidade é uma propriedade matricial de grande aplicação na programação matemática.

Definições básicas

- ✓ Uma matriz quadrada de elementos inteiros (matriz denominada inteira) $A = [a_{ij}]$ é *unimodular* se $|\det A| = 0$ ou 1.

- ✓ Uma matriz de elementos inteiros A é *totalmente unimodular* (t.u.) se toda submatriz quadrada não singular de A é unimodular.
- ✓ Uma matriz A é *Euleriana* por coluna (ou linha) se a soma dos elementos de cada coluna (ou linha) for par.
- ✓ Uma matriz A é *Euleriana* se for, simultaneamente, euleriana por linha e por coluna.
- ✓ Se A é t.u., toda solução básica $Ax = b$, onde A e b são matrizes de elementos inteiros, é inteira.
- ✓ É condição necessária para que a matriz A seja t.u. que $\forall a_{ij} \in A \Rightarrow a_{ij} = 0, +1, -1$.

Teoremas úteis

Teorema de Hoffman e Kruskal:

Seja (P) : Maximizar $z_0 = cx \mid Ax = b, x \geq 0$, onde A e b são matrizes de elementos inteiros. As seguintes condições são equivalentes:

- O determinante de toda base B , definida de A , é ± 1 .
 - Os vértices do poliedro $\{x \geq 0: Ax = b\}$ são inteiros.
 - A inversa da base B, B^{-1} , definida em A , é uma matriz inteira.
-

Teorema de Camion:

Uma matriz A com componentes $\{0, 1, -1\}$ é t.u. se e somente se a soma dos elementos de cada submatriz Euleriana quadrada de A é múltipla de 4.

Teorema de Truemper:

Uma matriz A com componentes $\{0, 1, -1\}$ é t.u. se e somente toda submatriz quadrada euleriana for singular.

Teorema de Heller & Tompkins:

Em uma matriz A com componentes $\{0, 1, -1\}$ t.u.:

- Não mais do que dois elementos não nulos aparecem em cada coluna.
 - As linhas de A podem ser particionadas em dois subconjuntos Q_1 e Q_2 tais que:
 1. Se uma coluna contém dois elementos não nulos com o mesmo sinal, cada um deles está em um dos subconjuntos.
 2. Se uma coluna contém dois elementos não nulos com sinais diferentes, ambos estão no mesmo subconjunto.
-

Para mais detalhes sobre unimodularidade, ver: Truemper (1976a e 1976b), Chandrasekaran (1969), Hoffman e Kruskal (1957), Heller e Hoffman (1962).

A.4 – ELEMENTOS GEOMÉTRICOS NO ESTUDO DA SOLUÇÃO DE SISTEMAS INDETERMINADOS

O presente tópico objetiva esclarecer a importância das soluções básicas na estratégia de solução de problemas de programação matemática, indicando também os elementos para o entendimento dos critérios de escolha e troca dessas bases em algoritmos iterativos de solução. O estudo dos elementos geométricos associados ao problema permitirá igualmente uma via elegante e de fácil compreensão para o problema da formalização matemática.

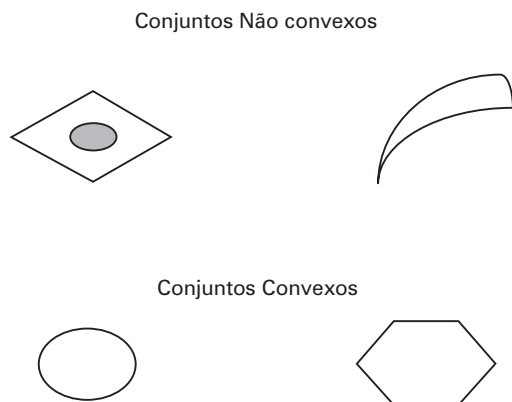
A.4.1 – Algumas Propriedades dos Conjuntos Convexos

Os conjuntos convexos possuem diversas propriedades geométricas. Algumas dessas propriedades e características são importantes para esclarecer o processo de solução dos sistemas indeterminados de equações lineares.

Definição 7:

Um subconjunto $X \in \mathbb{R}^n$ é *convexo* se e somente se, para todos os vetores $x_1, x_2 \in X$ tem-se que $(\lambda x_1 + (1 - \lambda) x_2) \in X$ e $\lambda \in (0, 1)$.

Exemplos de conjuntos convexos e não convexos:



Definição 8:

Denomina-se *ponto extremo* de um conjunto X a todo o ponto que não pode ser expresso por combinação linear convexa de quaisquer outros dois pontos distintos e igualmente pertencentes a X .

A Figura A.1 mostra que os pontos x_1 e x_3 podem ser obtidos por uma combinação linear convexa de outros dois pontos do conjunto C . O mesmo não ocorre com um ponto extremo.

A Figura A.2 mostra que o ponto x_2 só pode ser obtido por uma combinação linear cônica ou afim, e nunca por ambas.

Definição 9:

A reunião dos pontos extremos de um conjunto convexo X é denominada *perfil*.

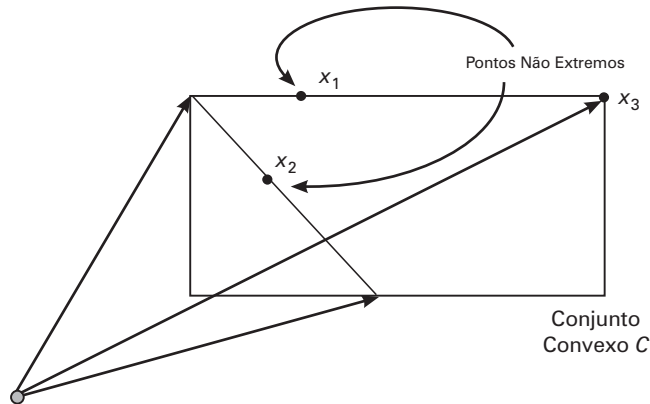


FIGURA A.1 Pontos não extremos.

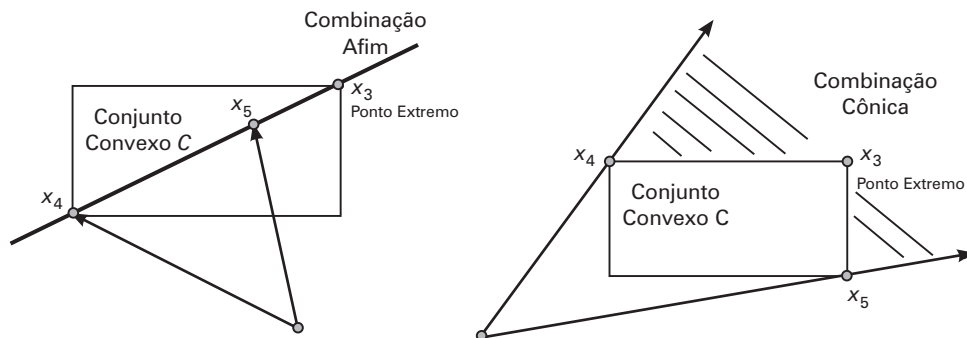


FIGURA A.2 Geração de pontos extremos.

Quando realizamos a associação entre as variáveis dos sistemas lineares e os vetores, estabelecemos uma associação direta entre o conjunto das soluções desses sistemas e as figuras geométricas geradas pelas combinações entre vetores em \mathbb{R}^n . Dessa forma podemos demonstrar que:

Teorema 1:

O conjunto de todas as soluções viáveis do modelo de PL é um *conjunto convexo*.

Alguns dos pontos gerados como soluções viáveis podem ser internos ao conjunto, como os pontos x_1 e x_2 da Figura A.1. Esses pontos incluirão variáveis que são linearmente dependentes entre si, uma vez que podem ser obtidas por combinações lineares convexas de outros pontos desse mesmo conjunto. Outros serão externos. Dentre os pontos externos alguns serão extremos. Os pontos extremos não poderão ser obtidos por combinação linear convexa de outros pontos do sistema, ou seja, serão correspondentes às soluções básicas do sistema. De fato, essa correspondência é biunívoca. A cada ponto extremo corresponde uma base e vice-versa, decorrendo daí o seguinte teorema:

Teorema 2 :

Toda solução viável básica do sistema $Ax = b$ é um *ponto extremo* do conjunto das soluções viáveis.

O que torna as soluções extremas dos conjuntos convexos tão atrativas e centrais para a nossa discussão está associado à sua interpretação geométrica. As inúmeras soluções dos sistemas indeterminados seriam, em termos matemáticos, indistintas se elas não pudessem ser associadas a uma função externa ao sistema. Na verdade, o problema da programação matemática não é solucionar sistemas de equações indeterminados, e sim otimizar uma função sujeita a esses sistemas como restrições. Em cada solução do sistema de restrições a chamada função objetivo possuirá um determinado valor. Nosso desejo é otimizá-la. Considerando uma função objetivo sobre o universo das soluções viáveis do sistema $Ax = b$, os seguintes teoremas se verificam:

Teorema 3:

Se uma função possui um máximo ou um mínimo finito sobre um conjunto convexo, então pelo menos uma solução ótima é um *ponto extremo* desse conjunto convexo.

Teorema 4:

Se a função objetivo assume o máximo ou o mínimo em mais de um ponto extremo, então toma o mesmo valor para qualquer combinação linear convexa desses pontos extremos.

Com base nos resultados teóricos anteriormente relacionados, no caso de a matriz de restrições ser constituída por equações lineares que formem uma figura convexa e compacta, então a solução do modelo de programação linear será encontrada nos pontos extremos dessa figura.

Para entender o comportamento dessa função sobre o universo das soluções viáveis dos sistemas de restrições passaremos a discutir agora a influência da geometria dos conjuntos convexos na otimização de funções lineares.

A.4.2 – Elementos Geométricos

✓ Variedade Linear

Um conjunto V é denominado *variedade linear*, se $\forall x_1, x_2 \in V$ tem-se que $\lambda x_1 + (1 - \lambda)x_2 \in V$ para qualquer $\lambda \in \mathbb{R}$. Como exemplos de variedades lineares podemos citar a reta e o plano. Toda variedade linear é um conjunto convexo. Ressalte-se que a recíproca não é verdadeira.

✓ Geometria em \mathbb{R}^n

• Hiperplano

Generalizando a noção de reta em \mathbb{R}^2 e plano em \mathbb{R}^3 temos o conceito de hiperplano (H) expresso da seguinte maneira:

$$H = \{x \in E^n \mid \alpha x = k\}, \text{ onde } \alpha \neq 0 \text{ e } k \in \mathbb{R}$$

A Figura A.3 representa abstratamente a formação de um hiperplano H através de um ponto $x_0 \in H$ e α .

O que é equivalente a:

$$H = \{x \in E^n \mid \alpha(x - x_0) = 0, x_0 \in H\}$$

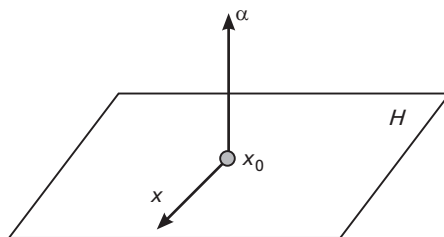


FIGURA A.3 Hiperplano H .

• Semi-Espaço

Um hiperplano divide o espaço vetorial em regiões denominadas semiespaço, da seguinte forma:

$$H^+ = \{x \mid \alpha x \geq k\} \neq \emptyset \text{ e } \alpha \neq 0$$

$$H^- = \{x \mid \alpha x \leq k\} \neq \emptyset \text{ e } \alpha \neq 0$$

ou então:

$$H^+ = \{x \mid \alpha(x - x_0) \geq 0\}$$

A Figura A.4 esclarece o conceito de semiespaço

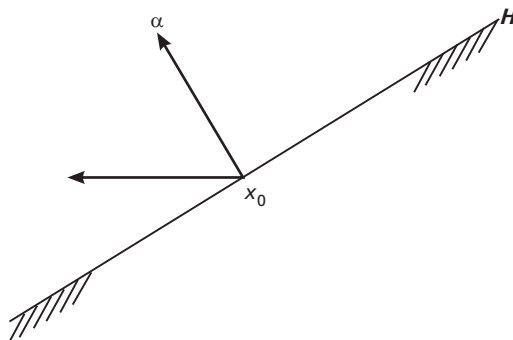


FIGURA A.4 Semiespaço.

• Raios e Direções

Um *raio vetor* é uma coleção de pontos da forma:

$$R = \{x_0 + \lambda d : \lambda \geq 0\} \text{ } d \neq 0$$

onde:

$x_0 \equiv$ vértice.

$d \equiv$ direção do raio vetor.

A Figura A.5 mostra a composição de um raio para a direção d

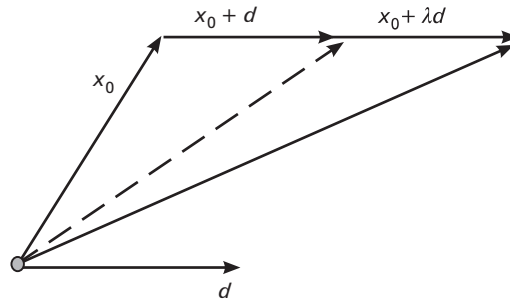


FIGURA A.5 Raios e direção.

• Direção de um Conjunto

Seja X um conjunto convexo, e $X = \{x \in E^n \mid Ax = b, x \geq 0\}$. Diremos que $d \in E^n$ é uma direção de X se:

$$\forall x_0 \in X, \{x_0 + \lambda d : \lambda \geq 0\} \subseteq X$$

ou seja:

$$\left. \begin{array}{l} A(x + \lambda d) = b \\ x + \lambda d \geq 0 \end{array} \right\} \forall x \in X, \lambda \geq 0$$

Conclusão imediata: Se um conjunto é limitado não tem direção. A Figura A.6 mostra uma direção d em um conjunto qualquer.

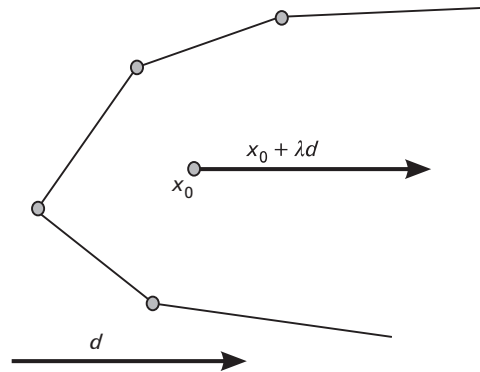


FIGURA A.6 Direção em um conjunto.

• Direções Extremas de um Conjunto Convexo

Uma direção d de X é extrema se não puder ser representada como combinação linear positiva de duas direções d_1, d_2 ($d_1 \neq kd_2$ $k \in R$), do conjunto X .

Em um exemplo numérico sejam os pontos $(0, 1)$, $(0, 2)$ e $(4, 2)$ e as direções d_1 e d_2 conforme mostra a Figura A.7. Então:

$$\begin{array}{ll} x_1 - 2x_2 + \lambda(d_1 - 2d_2) \geq -6 & d_1 - 2d_2 \geq 0 \\ x_1 - x_2 + \lambda(d_1 - d_2) \geq -2 & d_1 - d_2 \geq 0 \end{array}$$

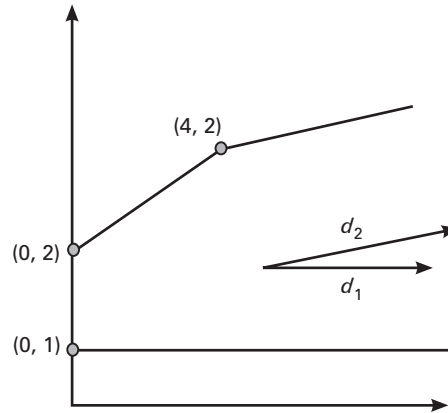


FIGURA A.7 Direções extremas de um conjunto não limitado.

• Conjuntos Poliédricos

Uma interseção de um número finito de semiespaços fechados denomina-se *poliedro*

$$\begin{aligned} H_1 &= \{x \text{ tal que } x_1 \leq 10\} \\ H_2 &= \{x \text{ tal que } x_2 \leq 20\} \\ H_3 &= \{x \text{ tal que } x_3 + x_2 \leq 5\} \\ H &= H_1 \cap H_2 \cap H_3 \end{aligned}$$

De um modo geral:

$$H = \{x \mid Ax \leq b\} \text{ é um poliedro } b \in E^n$$

Representação em função dos pontos extremos e das direções extremas de um conjunto convexo

Seja o poliedro:

$$\begin{aligned} -3x_1 + x_2 &\leq -2 \\ -x_1 + x_2 &\leq 2 \\ -x_1 + 2x_2 &\leq 8 \\ -x_2 &\leq -2 \end{aligned}$$

Para um ponto qualquer, interno ao poliedro, por exemplo, (4, 3), teremos:

$$(4, 3) = \lambda_1 (4/3, 2) + \lambda_2 (2, 4) + \lambda (4, 6) + \mu_1 (1, 0) + \mu_2 (2, 1)$$

Que nos conduz a, por exemplo:

$$\begin{aligned} \lambda_1 &= \lambda_2 = 1/2 \\ \lambda_3 &= 0 \\ \mu_1 &= 7/3 \text{ e } \mu_2 = 0 \end{aligned}$$

• Envoltória convexa

Seja $S = \{x_1, x_2, \dots, x_p, \dots\} \subset E^n$. A envoltória convexa de S , ou $\text{Conv } S$, é o menor conjunto convexo contendo S , ou seja, é o conjunto de todas as combinações convexas dos elementos de S , isto é:

$$\text{Conv } S = \{x \mid x = \sum_j \lambda_j x_j, \text{ onde } \sum_j \lambda_j = 1, \lambda_j \geq 0, x_j \in S\}$$

A Figura A.8 mostra uma envoltória convexa de um conjunto de pontos não convexo.

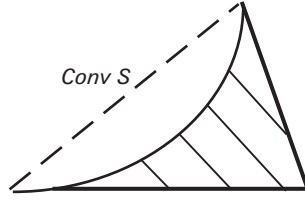


FIGURA A.8 Envoltória convexa.

✓ Interpretação Geométrica no Espaço das Commodidades

Seja o seguinte problema de programação linear:

(P): Mín $\{X_0 = cx \text{ tal que } Ax = b, x \geq 0\}$, onde $A = [a_j]$ é uma matriz $m \times n$ das comodidades a_1, a_2, \dots, a_n . Queremos agora debater a viabilidade de (P), ou seja, saber em que condições o problema não terá solução, ou ainda $x = \{x \text{ tal que } Ax = b, x \geq 0\} = \emptyset$. Consideremos o cone gerado pelos vetores de comodidades:

$$Y = \{y \text{ tal que } y = \sum_{j=1}^n \lambda_j a_j, \lambda_j \geq 0 \quad j = 1, 2, \dots, n\}$$

Seja $K = \{y \text{ tal que } y = \sum_{j=1}^n \lambda_j a_j, \sum_{j=1}^n \lambda_j = 1, \lambda_j \geq 0 \quad j = 1, 2, \dots, n\}$, a envoltória convexa dos pontos defini-

dos pelas comodidades a_1, a_2, \dots, a_n .

Se $x \in X$, isto é, $X \neq \emptyset$, então teremos $Ax = b, x \geq 0$ ou ainda $\sum_{j=1}^n x_j a_j = b, x_j \geq 0 \quad j = 1, 2, n\}$

$$\sum_{j=1}^n \frac{x_j}{\xi} a_j = \frac{b}{\xi}, \text{ com } \xi = \sum_{j=1}^n x_j$$

Definindo $\lambda_j = \frac{x_j}{\xi}$, $0 \leq \lambda_j \leq 1$, com $j = 1, 2 \dots n$, segue-se que $\frac{b}{\xi} \in K$, ou seja, se $X \neq \emptyset$, então $b \in \mathbb{R}^n$ deve pertencer ao cone gerado pelas comodidades, caso contrário $X = \emptyset$ conforme Figuras A.9 e A.10.

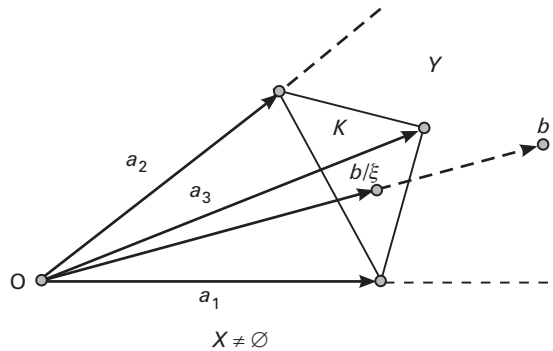


FIGURA A.9 Caso 1 – problema viável.

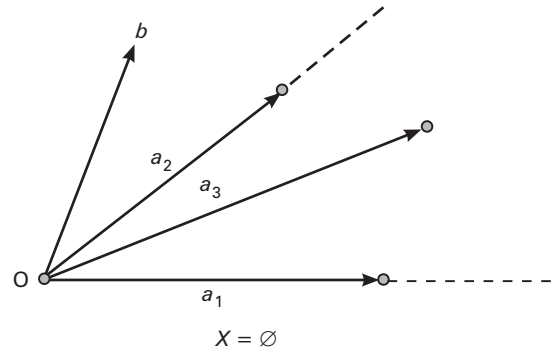


FIGURA A.10 Caso 2 – problema inviável.

✓ Interpretação Geométrica no Espaço das Atividades

Seja o problema abaixo:

$$(P) \text{ Maximizar } x_0 = 2x_1 + 4x_2$$

sujeito a:

$$x_1 + x_2 \leq 9$$

$$x_1 \leq 4$$

$$x_2 \leq 7$$

$$x_1 \geq 0, x_2 \geq 0$$

Pode ser representado graficamente como na Figura A.11:

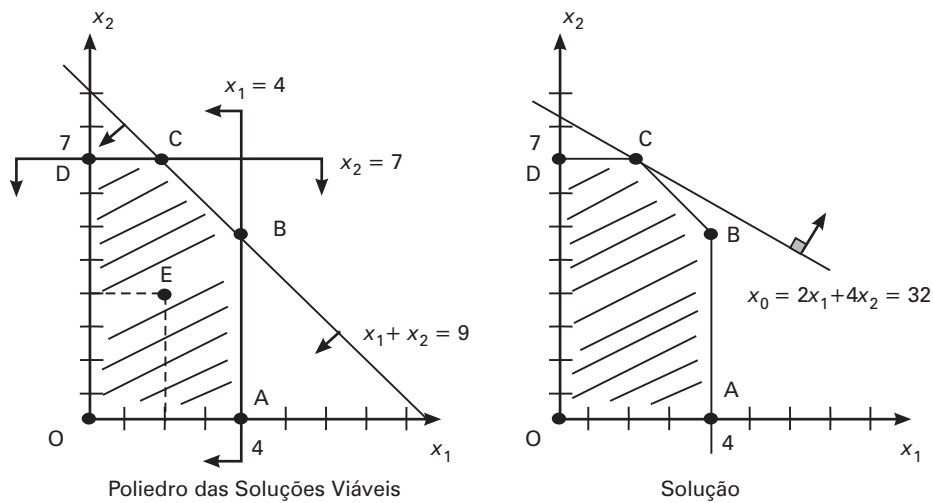


FIGURA A.11 Solução do problema (P).

A N E X O B

TÓPICOS EM TEORIA DOS GRAFOS

B.1 – CONCEITOS FUNDAMENTAIS EM TEORIA DOS GRAFOS

B.1.1 – Definições e Notações Básicas

O conceito de Grafo é extremamente simples e até mesmo intuitivo. Podemos considerar que um grafo nada mais é que uma representação gráfica de interdependência entre elementos que são representados por *nós*. Elementos que atendem à relação imaginada são simbolicamente unidos através de um traço denominado *aresta*. O modelo possui uma interpretação gráfica muito confortável; contudo, tal desenho não tem o poder, em várias situações reais, de formalizar completa e satisfatoriamente a estrutura imaginada. Devido a esse fato é necessário que um grafo seja também definido analiticamente. Alguns autores brasileiros que cobrem o tema são Barbosa (1974, 1975) e Furtado (1973).

Definição de Grafo

Um grafo é uma estrutura de abstração que representa um conjunto de elementos denominados *nós* e suas relações de interdependência ou *arestas*.

Representação Matemática

Denominando por N o conjunto de vértices da estrutura, e por M o conjunto das arestas ou ligações entre os vértices, um grafo pode ser representado por: $G = (N, M)$.

Os termos nós e vértices são empregados usualmente como sinônimos.

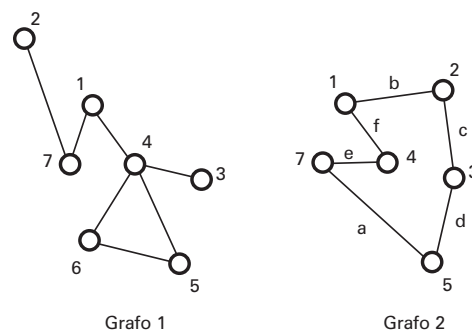


FIGURA B.1 Exemplo de grafos.

O conjunto $N = \{1, 2, \dots, n\}$ será composto dos n nós do grafo, e $M = \{1, 2, \dots, m\}$ conterá as m arestas. É comum a utilização da variável x_i , $i = 1, 2, \dots, n$ para a representação dos nós. No grafo da Figura B.1, $N = \{1, 2, 3, 4, 5, 7\}$ e $M = \{a, b, c, d, e, f\}$.

O conceito de grafo pode ser generalizado para o caso em que a relação entre os nós não é constituída com apenas um par de nós ou vértices. Os hipergrafos são modelos que permitem a representação de arestas que englobam mais de dois nós.

Definição de Hipergrafo

Um hipergrafo H é um par $H = (N, \xi)$ onde n representa o conjunto dos nós de H e ξ é uma família das partes de N .

Na Figura B.2 temos $N = \{1, 2, 3, 4, 5, 6\}$ e $\xi = \{(1, 2, 3); (3, 4); (4, 5, 6)\}$.

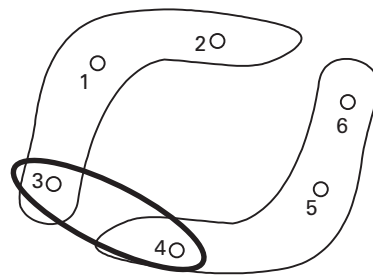


FIGURA B.2 Exemplo de hipergrafo.

Como podemos observar nas figuras dos exemplos, um grafo pode conter informações associadas aos seus vértices e arestas. Definimos essas informações como *rótulos* ou *pesos*. Formalmente temos:

Definição de Grafo Ponderado

Um grafo $G = (N, M)$ é *ponderado* se existem valores numéricos associados a suas arestas ou nós.

Definição de Grafo Rotulado

Um grafo $G = (N, M)$ é *rotulado* se existem atribuições associadas a seus nós (tanto numéricas como alfabéticas).

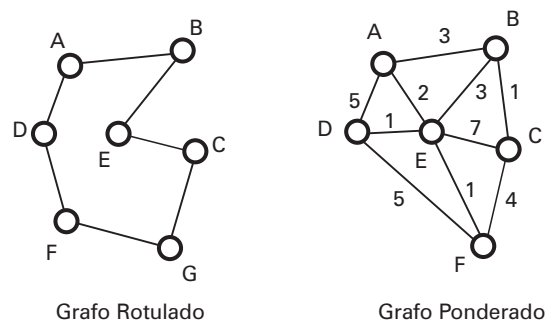


FIGURA B.3 Grafos ponderados e rotulados.

Eventualmente um grafo pode possuir mais de uma aresta distinta ligando dois vértices.

Definição de Multigrafo

Um grafo $G = (N, M)$ é um *multigrafo* se existem mais de uma aresta ligando o mesmo par de vértices.

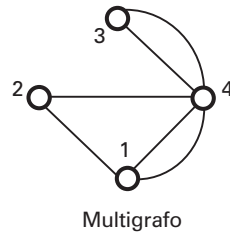


FIGURA B.4 Multigrafo.

Alguns grafos se notabilizam por peculiaridades em suas estruturas. Essas peculiaridades podem ser bastante úteis na representação de situações reais ou na utilização de algoritmos de solução para os problemas em grafos. Dentre eles destacamos:

✓ Grafo Direcionado

Definição de Grafo Direcionado

Um grafo é dito *direcionado* quando o sentido das ligações entre os vértices é importante. Nesse caso normalmente as arestas são chamadas por *arcos*.

Representação Matemática

Denominando por N o conjunto de vértices da estrutura, e por M o conjunto dos pares ordenados do produto cartesiano $n \times n$ das ligações existentes em G , um grafo Orientado é também representado por $G = (N, M)$.

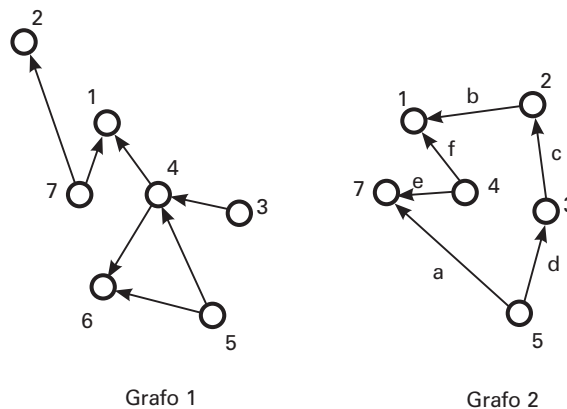


FIGURA B.5 Grafos direcionados.

Os grafos aqui denominados por *direcionados* são igualmente chamados de *orientados*. No presente trabalho utilizaremos a denominação $G = (N, A)$ para representar grafos não direcionados, e $G = (V, E)$ para os direcionados.

✓ Grafo Bipartido

Outra significativa classe é a denominada *grafos bipartidos*. Esse tipo de estrutura pode representar situações como as geradas por alocação de pessoas a tarefas, ferramentas a máquinas etc.

Definição de Grafo Bipartido

Um grafo G é dito *bipartido* quando seu conjunto de Nós, N , pode ser dividido em dois conjuntos N_1 e N_2 tais que $N_1 \cap N_2 = \emptyset$ e $N_1 \cup N_2 = N$ e somente existem arestas em G ligando algum nó de N_1 com algum nó de N_2 e vice-versa.

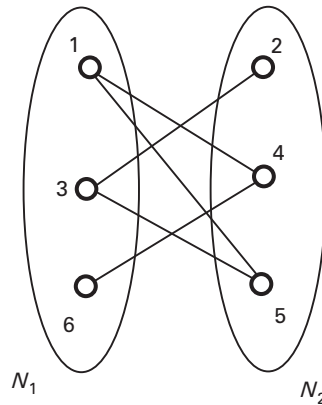


FIGURA B.6 Grafo bipartido.

✓ Grafo Completo

Definição de Grafo Completo

Um grafo G é dito *completo* se existir ao menos uma ligação associada a cada par de vértices. No caso não orientado isso significa exatamente uma ligação.

Os grafos completos não orientados são também denominados *cliques* e denotados como K_n , onde n representa o número de nós do grafo completo.

Por analogia os grafos *completos bipartidos* são denotados por K_{pq} , sendo p e q as cardinalidades das duas partições do grafo.

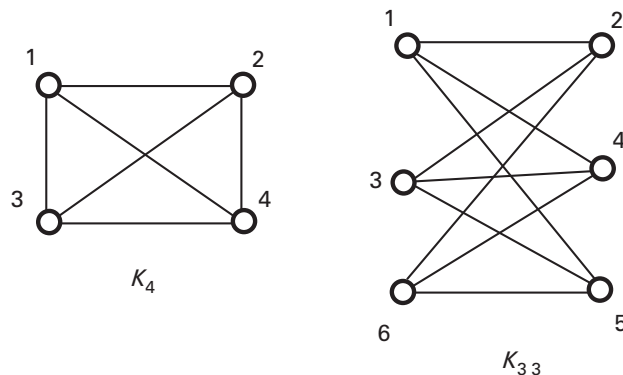


FIGURA B.7 Grafos completos.

✓ Grafo Regular

Definição de Grafo Regular

O grau de um vértice de um grafo é o número de arestas incidentes no vértice.

Um grafo G é dito *regular de grau r* se cada vértice em G possuir o grau r .

(*) O maior grau de G é denominado $\Delta(G)$.

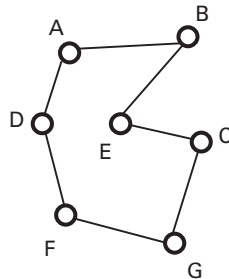


FIGURA B.8 Grafo regular.

Alguns autores fazem distinção entre o conceito de grau para grafos orientados e não orientados. No caso dos grafos orientados, o grau pode ser decomposto em duas parcelas: o grau interno ou o número de arcos chegando ao nó, e o grau externo, ou o número de arcos partindo do nó. Essas parcelas do grau do nó são denominadas semigrau. No caso dos grafos direcionados, a soma do semigrau interior $d(i)^+$ e exterior $d(i)^-$ conduz ao valor final do grau do nó. A expressão para a obtenção do grau em grafos orientados é:

$$d(i) = |d(i)^-| + |d(i)^+|$$

No caso da Figura B.9 podemos calcular o grau do vértice 4 da seguinte forma:

$$\begin{aligned} d(4) &= |d(4)^-| + |d(4)^+| = \\ &= |-2| + |2| = 4 \end{aligned}$$

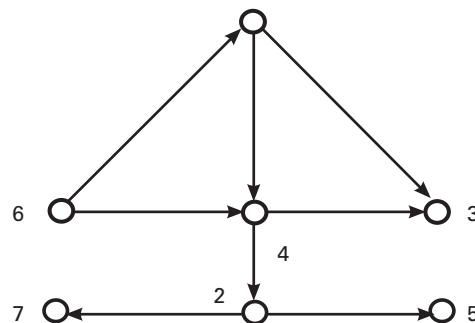
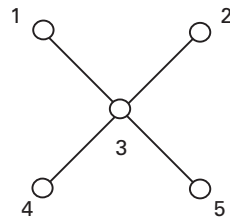


FIGURA B.9 Grau dos vértices em grafos direcionados.

✓ Árvore

Definição de Árvore

Um grafo é denominado por *árvore* se for conexo e não possuir ciclos.



Árvore

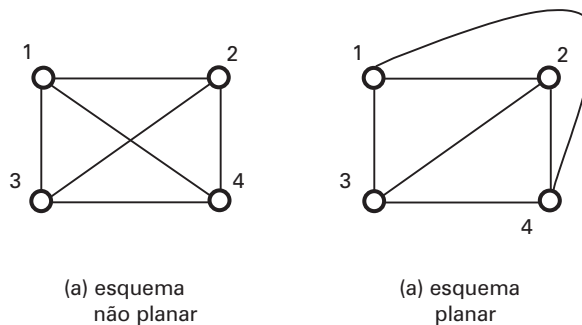
FIGURA B.10 Uma árvore.

✓ Grafo Planar

Um tipo de grafo muito importante por representar os problemas de topologia plana é o chamado *grafo planar*. Redes de estradas, mapas, circuitos eletrônicos, vários tipos de *layout* etc., quando representados via grafos, acabam configurando, na maioria das ocasiões, grafos planares.

Definição de Grafo Planar

Um grafo G é dito *planar* quando seu esquema admitir pelo menos uma representação planar, ou seja, quando traçado em um plano duas arestas quaisquer se toquem, no máximo, em alguma extremidade.



(a) esquema não planar

(a) esquema planar

FIGURA B.11 Grafo planar.

✓ Rede

Definimos uma rede $R = (E, V, F)$ como um grafo direcionado $G = (E, V)$ atravessado por um Fluxo $F = \{f_1, f_2, \dots, f_m\}$ que circula em suas m arestas. Em uma rede normalmente dois nós são destacados: o nó *fonte* e o nó *sumidouro*. Qualquer tipo de rede pode ser reduzida a uma rede com apenas um nó fonte e um nó sumidouro, mesmo que artificialmente configurados. Os arcos da rede podem ser limitados em capacidade em relação ao fluxo. Esses mesmos arcos podem impor custos à circulação do fluxo. De uma forma geral, uma rede poderia ser representada como na Figura B.12. Os nós são representados pelos círculos e os arcos pelas setas. O sentido convencional do fluxo está indicado pelas setas (o grafo de substrato é direcionado).

onde:

$l_{ij} \equiv$ limite inferior (ou mínimo) para o fluxo no arco $i-j$.

$L_{ij} \equiv$ limite superior (ou máximo) para o fluxo no arco $i-j$.

$c_{ij} \equiv$ custo de circulação da unidade de fluxo no arco $i-j$.

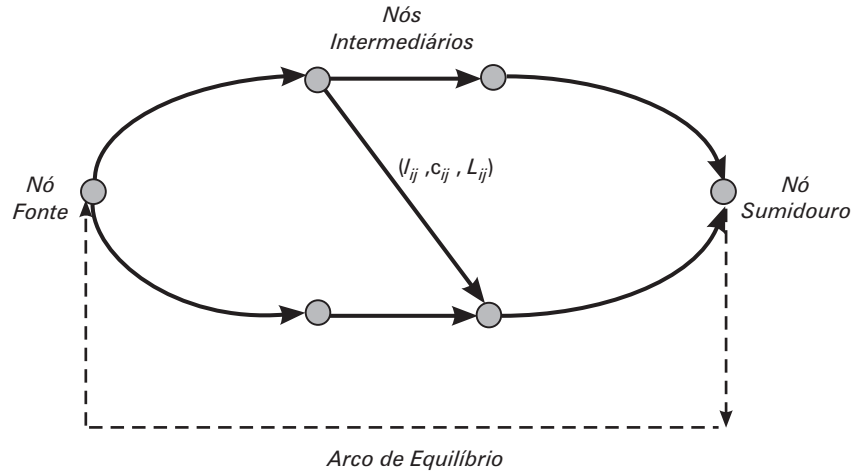


FIGURA B.12 Representação típica de uma rede.

A Figura B.13 representa uma rede valorada:

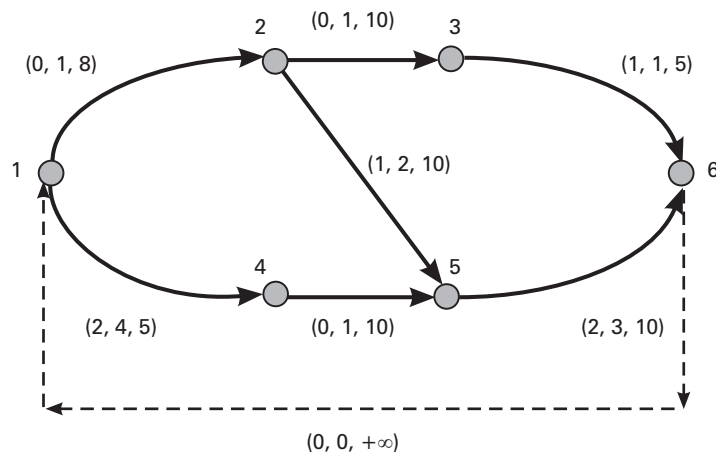


FIGURA B.13 Rede valorada.

B.1.2 – Outros Conceitos Básicos

Um dos pontos mais fundamentalmente associados à noção de grafos é o conceito de ligação entre vértices. De fato, o modelo é, basicamente, uma estrutura adequada a representar topologicamente formas de conexão. É, portanto, indispensável esclarecer como os vértices podem estabelecer vínculos ou ligações através das denominadas arestas ou arcos. Nesse sentido, existem duas formas específicas de entender vizinhança entre nós e arestas, uma para o caso dos grafos direcionados e outra para os não direcionados.

✓ Vizinhança de Vértices

Definição de Vizinhança de Vértices

Dizemos que dois vértices x_i e x_j são vizinhos ou adjacentes quando existe uma aresta que liga x_i a x_j ou vice-versa.

(*) A noção de vizinhança é associada a grafos não orientados.

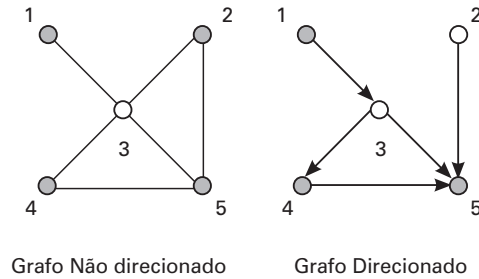


FIGURA B.14 Vizinhos ao vértice 3.

✓ Sucessores e Antecessores de um Vértice

Como o conceito de vizinho para o caso dos grafos não direcionados pode ser insuficiente no estabelecimento da compreensão de determinadas relações, para esses grafos introduziremos o conceito de *sucessor* e *antecessor*.

Definição de Sucessor e Antecessor de um Vértice

Dizemos que um vértice x_j é *sucessor* de x_i se existe pelo menos um arco ligando x_i a x_j .

(*) No caso da ocorrência da relação inversa diremos que o vértice x_j é *antecessor* de x_i .

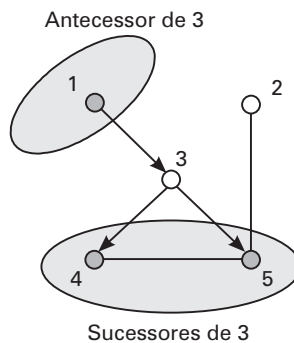


FIGURA B.15 Antecessores e sucessores.

Denominamos por $\Gamma^+(x)$ o conjunto de todos os sucessores de x e por $\Gamma^-(x)$ o dos antecessores. No exemplo da Figura B.15 $\Gamma^+(3) = 4, 5$ e $\Gamma^-(3) = 1$.

✓ Fechos Transitivos

Podemos generalizar o conceito de vizinhança considerando a estrutura de proximidade através de uma ideia recursiva. Os vizinhos dos vizinhos e assim por diante. No modelo de grafos esse tipo de informação poderá ser útil para representar fenômenos de propagação de informação ou de capacidade de comunicação. Um fecho transitivo nada mais é que um desdobramento de níveis de vizinhanças que ressalta o fenômeno da alcançabilidade.

Definição de Fecho Transitivo

Denominaremos por *fecho transitivo direto* do vértice x , $\hat{\Gamma}^+(x)$, ao conjunto de vértices que podem ser alcançados, a partir de x , através de sucessivas relações de vizinhança.

(*) No caso inverso dos vértices que podem atingir x por sucessivas relações de vizinhança, definiremos o *fecho transitivo inverso*.

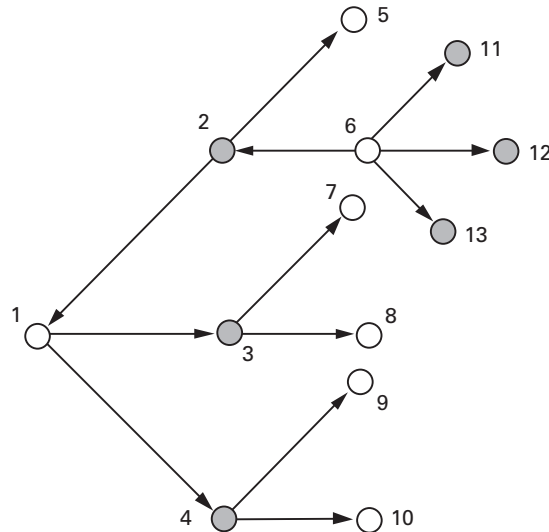


FIGURA B.16 *Fecho transitivo direto de x.*

Na Figura B.16 podemos verificar que o primeiro nível de vizinhança do vértice 1 são os nós 3 e 4. Em um segundo nível encontramos mais os nós 7, 8, 9 e 10. Verificamos também que esses últimos vértices pertencem também ao terceiro nível de vizinhança do nó 2. Assim:

$\hat{\Gamma}^1(1)$ – relação de vizinhança até nível 1 (afastados do vértice 1 em até uma aresta ou arco do) = {3, 4}

$\hat{\Gamma}^2(1)$ – relação de vizinhança até nível 2 (afastados do vértice 1 em até duas arestas ou arcos) = $\hat{\Gamma}^1(1) \cup \{7, 8, 9, 10\} = \{3, 4\} \cup \{7, 8, 9, 10\} = \{3, 4, 7, 8, 9, 10\}$

$\hat{\Gamma}^3(6) = \{2, 11, 12, 13\} \cup \{5, 1\} \cup \{3, 4\} = \{1, 2, 3, 4, 5, 11, 12, 13\}$ etc.

$\hat{\Gamma}^{-3}(3) = \{1\} \cup \{2\} \cup \{6\} = \{1, 2, 6\}$

Alguns autores denominam os *fechos diretos* como *fechos positivos* e os *inversos* como *negativos*.

✓ Cadeia de Arestas

As trajetórias dentro do modelo de grafos são características muito importantes. As cadeia de arestas são obrigatoriamente formadas através de percursos sobre os nós do grafo.

Definição de Cadeia de Arestas

Dizemos que uma *cadeia de arestas* é uma sequência de arestas em que todas são distintas (não repetidas).

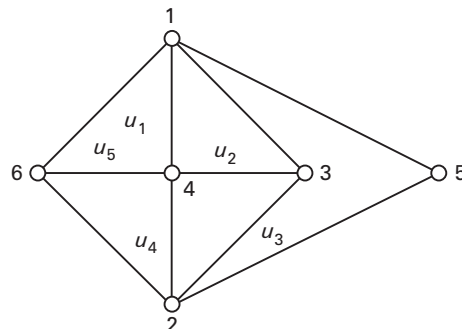


FIGURA B.17 *Cadeia de arestas.*

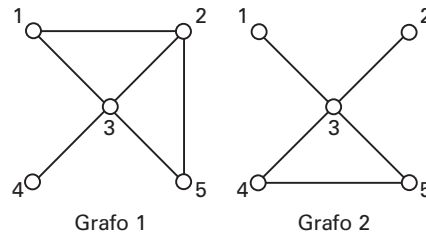


FIGURA B.20 Grafos com ciclos.

✓ Ciclos e Circuitos Especiais

Definição de Ciclo Euleriano

Trata-se de um ciclo que passa por todas as arestas do grafo G .

Definição de Circuito Hamiltoniano

Trata-se de um circuito que passa por todos os nós de G .

(*) Percurso é uma denominação genérica para qualquer trajetória sobre nós e arestas de G .

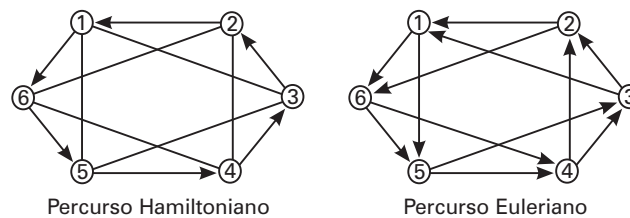


FIGURA B.21 Percursos especiais.

✓ Conexidade

Definição de Grafo Conexo

G é *conexo* se para todo par de vértices existe pelo menos uma cadeia entre eles.

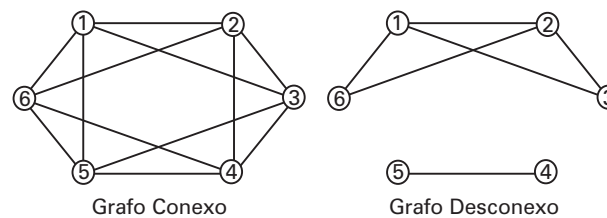


FIGURA B.22 Conexidade.

B.2 – REPRESENTAÇÃO DO MODELO

Além da representação geométrica, um grafo pode ser representado através de, pelo menos, quatro outras formas diferentes:

✓ Representação através da Matriz de Adjacência

Trata-se de uma representação bastante simples. O grafo é expresso em uma matriz $A = [a_{ij}]$ através dos nós e de suas relações de vizinhança. As linhas e as colunas da matriz estão associadas aos nós do grafo. A matriz é, normalmente, booleana, ou seja, seus elementos são 0 e 1. Quando existem arestas paralelas, o valor de a_{ij} pode passar a representar o número de arcos paralelos.

Definição de Matriz de Adjacência

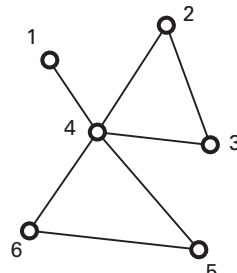
Uma matriz $n \times n$ $A = [a_{ij}]$ é denominada como de *adjacência* do grafo $G = (N, M)$ se:

$$a_{ij} = 1 \Leftrightarrow \exists \text{ Ligação } (i, j)$$

$$a_{ij} = 0 \Leftrightarrow \nexists \text{ Ligação } (i, j)$$

	1	2	3	4	5	6
1	0	0	0	1	0	0
2	0	0	1	1	0	0
3	0	1	0	1	0	0
4	1	1	1	0	1	1
5	0	0	0	1	0	1
6	0	0	0	1	1	0

Matriz de Adjacência



Grafo G

FIGURA B.23 Matriz de adjacências do grafo G.

✓ Representação através da Matriz de Incidência

Nesse caso as colunas da matriz correspondem às arestas do grafo e as linhas aos nós.

Definição de Matriz de Incidência

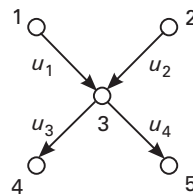
Uma matriz $n \times m$ $A = [a_{ij}]$ é denominada como de *incidência* do grafo $G = (N, M)$ se, para todo arco j que liga o nó k ao nó l temos:

$$a_{ij} = +1 \Leftrightarrow i = k$$

$$a_{ij} = -1 \Leftrightarrow i = l \text{ (para grafo direcionado, senão } a_{ij} = 1)$$

$$a_{ij} = 0 \text{ nos outros casos}$$

	u_1	u_2	u_3	u_4
1	1	0	0	0
2	0	1	0	0
3	-1	-1	1	1
4	0	0	-1	0
5	0	0	0	-1



Grafo Direcionado

FIGURA B.24 Matriz de incidência grafo direcionado.

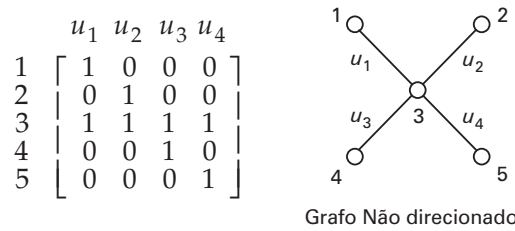


FIGURA B.25 Matriz de incidência grafo não direcionado.

✓ Representação através de Listas Encadeadas

A representação utilizando listas encadeadas é extremamente conveniente para a eficiência computacional de alguns algoritmos, bem como para a economia de memória de armazenamento. A mais tradicional representação por listas encadeadas possui uma configuração baseada nos vértices do grafo e em suas vizinhanças. O vetor Lista_1 possui o comprimento igual ao número de nós do grafo, ou seja, n posições. A estrutura ocupa, para o caso dos grafos não direcionados, mais $2n$ posições na memória, uma vez que as vizinhanças são recíprocas.

✓ Representação Vetorial

Essa representação é útil para o caso de grafos esparsos, ou seja, com poucas arestas. São utilizados dois vetores para resumir as informações de vizinhança. O vetor SX registra o número de vizinhos do nó correspondente à posição x , enquanto o vetor NX os enumera. A Figura B.27 esclarece a estrutura de representação. A representação vetorial simula a representação por listas encadeadas de nós.

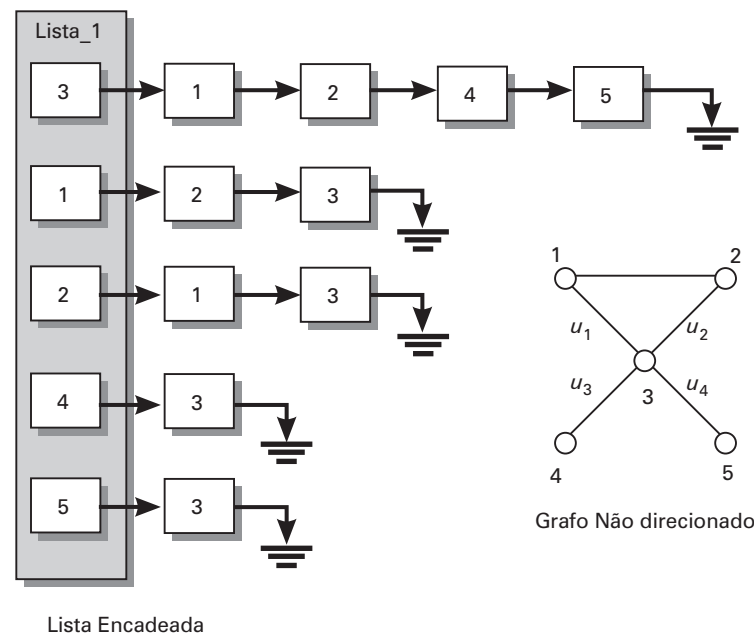


FIGURA B.26 Representação por listas encadeadas de nós.

B.3 – O PROBLEMA DA COLORAÇÃO

O problema da coloração é um dos mais conhecidos na teoria dos grafos. Colorir um grafo $G = (N, A)$ é atribuir cores aos seus vértices de forma que vértices adjacentes recebam cores distintas. Simplesmente colorir um grafo é tarefa trivial, uma vez que podemos imaginar distribuir uma cor para cada vértice. O problema da coloração realmente surge quando desejamos colorir o tal grafo utilizando o menor número possível de cores.

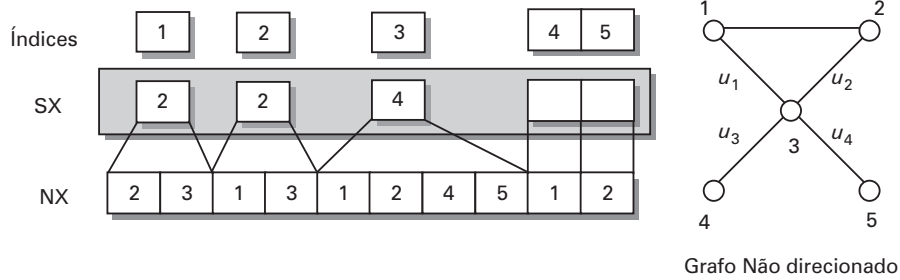
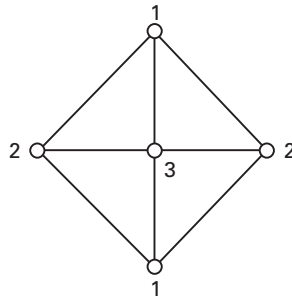


FIGURA B.27 Representação vetorial.

Ao menor número de cores que pode ser utilizado para a coloração de um grafo denominaremos *número cromático* ou $x(G)$. Distinguindo as cores através de números, a Figura B.28 representa uma 3-coloração do grafo G .

FIGURA B.28 3-coloração de G .

Quando o grafo é planar sabe-se que $x(G) \leq 4$ (prova exaustiva em Appel e Haken [1989]). O problema da coloração em grafos planares é denominado de *O Problema das Quatro Cores* e tem uma longa tradição na teoria dos grafos (ver Saaty e Kainen [1977] para uma ampla visão sobre o assunto).

O número cromático está relacionado com outros invariantes de um grafo.

Dado um grafo G e $S \subseteq N$, S é denominado conjunto independente se $\forall i, j \in S, (i, j) \notin A$. Um conjunto independente é máximo se $\forall S' \subset N, |S'| < |S|$. A cardinalidade do S máximo é chamada de *número de independência* de G , ou *número de estabilidade* ou $\alpha(G)$. Como resultado imediato tem-se que vértices coloridos com a mesma cor formam um conjunto independente e que $x(G)$ é o menor número de conjuntos independentes de vértices em que G pode ser particionado. Dessa forma podemos provar que:

$$n - \alpha \leq x \leq n - \alpha + 1$$

Um subconjunto $N' \subseteq N$ de tamanho t é uma t -clique se ele induz um subgrafo completo. Uma clique é dita máxima se não houver outra clique em G com cardinalidade maior que ela. A cardinalidade da clique máxima em G é denotada por $w(G)$. Os grafos em que $x(G) = w(G)$ são chamados de *fracamente perfeitos*. Quando G e todo subgrafo induzido G' de G tiver a propriedade anterior então o grafo será denominado de *perfeito*.

O conceito de coloração pode ser estendido às arestas. Uma k -coloração das arestas de G é uma atribuição de k cores a estas arestas de forma que duas arestas incidentes ao mesmo vértice recebam cores distintas. O número mínimo de cores capaz de colorir as arestas de um grafo G atendendo às exigências anteriores é denominado *índice cromático* ou $x_1(G)$. Alguns resultados são imediatos para o índice cromático:

No caso de grafos bipartidos $x_1(G) \geq \Delta(G)$.

No caso geral $x_1(G) = \Delta(G)$ ou $x_1(G) = \Delta(G) + 1$ (teorema de Vizing).

Podemos colorir um grafo de modo a que vértices e arestas sejam atribuídas cores de forma que dois elementos incidentes ou adjacentes recebam cores diferentes. O número mínimo de cores necessário a essa tarefa é chamado de *número cromático total* ou $\chi_2(G)$.

Colorir os vértices de um grafo com inteiros módulo n de forma que aos vértices adjacentes sejam atribuídas cores com a maior distância possível é a generalização proposta por Vince (1989). Nesse caso, a coloração clássica seria um caso particular desta teoria quando a distância considerada for igual a 1.

Considerando k e d inteiros positivos e $k \geq 2d$. Uma (k, d) -coloração de G é um mapeamento $c: N \rightarrow \mathbb{Z}_k$, de forma que para $\forall(x_i, x_j) \in A$, $|c(x_i) - c(x_j)|_k \geq d$, onde $|x|_k = \text{Mínimo}\{|x|, k - |x|\}$. A Figura B.29 exemplifica uma $(4, 2)$ -coloração, ($d = 2$ e $k = 4$). Nesse caso, o conjunto $\mathbb{Z}_4 = \{0, 1, 2, 3\}$.

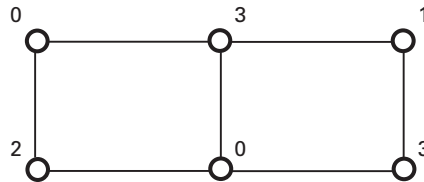


FIGURA B.29 $(4, 2)$ -coloração.

O invariante nesse caso chama-se *número star-cromático* ou $X^*(G)$, sendo o menor k/d para o qual G admite uma (k, d) -coloração.

Uma outra recente extensão dos problemas de coloração em grafos é a T -coloração. Essa forma de coloração foi apresentada por Hale (1980) modelando o problema de alocação de frequências de comunicação. Considerando que exista em uma certa região n transmissores, x_1, x_2, \dots, x_n aos quais se deseja alocar frequências $f(x)$ sobre as quais eles possam operar. Para esse conjunto de transmissores é definido um grafo de interferência $G = (N, A)$, em que $N = \{x_1, x_2, \dots, x_n\}$ e as arestas $(x_i, x_j) \in A$ representam o fato de haver interferência entre os transmissores x_i e x_j . Um conjunto T de inteiros positivos representa as distâncias proibidas entre os pares de transmissores. Assume-se que 0 pertence a T . A restrição de distâncias mínimas que garante a eliminação das interferências entre um par x_i, x_j de transmissores é definida pela seguinte expressão:

$$(x_i, x_j) \in A \Rightarrow |f(x_i) - f(x_j)| \notin T$$

Assim, a T -coloração de um grafo é definida como uma função $f: N \rightarrow \mathbb{N}$ que satisfaz à expressão anterior. A Figura B.30 mostra uma T -coloração para $T = \{0, 1, 2\}$ em um certo grafo G de interferências.

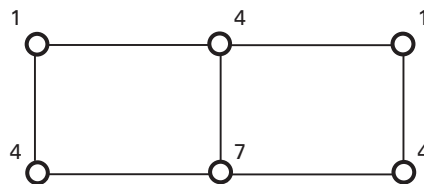


FIGURA B.30 Uma T -coloração válida.

O *número T -cromático* é definido como sendo a menor ordem de uma T -coloração, sendo notado por $X_t(G)$. A ordem da T -coloração vai representar o número de canais abertos ou o espaço de frequências não utilizado. Podemos interpretar que na rede de comunicação do grafo da Figura B.30 são utilizados os canais (UHF) 1, 4 e 7. Nesse problema é ainda importante um segundo invariante, o T -span. De fato, nesse tipo de situação existe o interesse de minimizar o número de canais (frequências) utilizados e o espalhamento dessas frequências na banda de comunicação. O T -span de uma certa alocação é denotado por $sp_T(f)$. O menor espalhamento dentre todos os possíveis em G é denotado por $sp_T(G)$. Se determinarmos que $T = \{0, 1, 4, 5\}$ então a Figura B.31 mostra que o espalhamento ótimo não está associado à

melhor ordem da T-coloração. A ordem ótima da T -coloração utiliza três canais (1, 4 e 7) mas seu espalhamento compromete uma faixa de sete canais. Por outro lado, se forem utilizados os canais 1, 2, 3, 4, 5, os canais 6 e 7 não serão afetados pela faixa de compromisso.

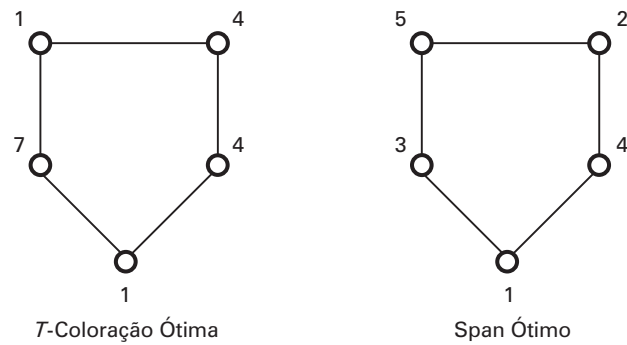


FIGURA B.31 Ordem \times span na T -coloração.

A N E X O C

TÓPICOS EM COMPLEXIDADE DE ALGORITMOS

C.1 – CONCEITOS EM ALGORITMOS

C.1.1 – Conceito de Algoritmo

O dicionário da língua portuguesa editado pelo Ministério da Educação define *algoritmo* (termo matemático) como *um processo formal de cálculo*. A execução de um algoritmo não deve incluir decisões subjetivas ou o uso da intuição e criatividade. Quando utilizamos essa palavra estamos, *strito senso*, implicitamente aceitando pensar em termos computacionais. Nesse caso, seria aceitável incluir dentro da definição de algoritmo os métodos sistemáticos utilizados para a solução de problemas. Os métodos para efetuar as operações de soma, multiplicação, radiciação etc. indiscutivelmente atendem às exigências da definição de algoritmo. O conceito não é tão simples como pode inicialmente parecer. Coermen *et al.* (1991) define algoritmo como *uma sequência bem-definida de procedimentos computacionais (passos) que levam uma entrada a ser transformada em uma saída*. A ideia de associar o conceito de algoritmo ao fenômeno do processamento de uma entrada traduzível em termos computacionais implica a necessidade da definição formal de um alfabeto para a codificação dessa entrada/saída, bem como as estruturas de transformação. Nesse foco podemos também pensar o algoritmo como uma espécie de função processamento sobre o alfabeto de entrada/saída. Nesse caso, diríamos que um algoritmo *é a descrição do cálculo ou avaliação sistemática dessa função*.

A associação do termo à solução de problemas computáveis pode trazer algumas confusões que cabe esclarecer. O processo de solução de um problema qualquer pode incluir etapas como: “adicionar açúcar a gosto”, o que não é uma decisão objetiva. A dificuldade aqui reforça a necessidade de que tenhamos em mente que nem todos os tipos de problemas admitem soluções algorítmicas, apesar de serem, eventualmente, muito importantes: ser ou não ser, eis a questão... Diversos autores denominam os problemas que não podem ser solucionados através de processos formais de cálculo como problemas não decidíveis. Nesse caso, os algoritmos são estratégias para a solução de problemas decidíveis, ou seja, para problemas cuja solução admita um processo formal de ações. O item 5 da presente parte do anexo detalha três tipos possíveis de problemas decidíveis: problemas de decisão, localização e otimização.

C.1.2 – Correção dos Algoritmos

A verificação do funcionamento adequado de um algoritmo é um dos pontos mais importantes dentro desse tema. Muitas vezes um algoritmo é capaz de solucionar somente um certo conjunto de casos de um problema. Esses casos específicos são denominados *instâncias* do problema. Para garantirmos a correção de um algoritmo na solução de um certo problema é necessário que possamos garantir seu acerto em todas as instâncias possíveis do mesmo. Basta a exibição de um caso de fracasso para demonstrarmos sua ineficácia. Por outro lado, para provar seu acerto serão, normalmente, necessárias providências árduas. Uma forma de reduzir a dificuldade da prova da correção de um algoritmo é o estabelecimento de *domínios de definição*, ou seja, regiões ou limites em que podemos garantir que as instâncias são solucionadas corretamente.

C.1.3 – Eficiência dos Algoritmos

Ser capaz de estabelecer um conjunto de instruções que soluciona certo problema decidível não significa que tenhamos encontrado uma estratégia razoável, sob qualquer que seja o nosso ponto de vista. O objetivo maior de um algoritmo é solucionar o problema de uma forma rápida e econômica. O termo *rápido* diz respeito à realidade do fim prático do algoritmo. O termo *econômico* responde pela realidade da limitação dos recursos humanos ou computacionais. *Analisar um algoritmo significa prever os recursos que o algoritmo irá requerer quando em passo de execução*. Como recursos entendemos a *memória*, o *tempo de processamento*, a *natureza das operações* etc.

C.2 – NOTAÇÃO O

C.2.1 – Notação Assintótica

Fazendo \mathbb{N} e \mathbb{R} representar o conjunto dos números naturais e o conjunto dos números reais, respectivamente, o conjunto \mathbb{N}^* representando o conjunto dos reais não negativos e \mathbb{R}^+ o conjunto dos reais estritamente positivo, podemos definir:

$$f: \mathbb{N} \rightarrow \mathbb{R}^* \text{ e } t: \mathbb{N} \rightarrow \mathbb{R}^* \text{ funções quaisquer}$$

então podemos estabelecer a seguinte notação que relaciona as funções $t(n)$ e $f(n)$, $n \in \mathbb{N}$:

$$O(f(n)) = \{t: \mathbb{N} \rightarrow \mathbb{R}^* \mid (\exists c \in \mathbb{R}^+) (\exists n_0 \in \mathbb{N}) (\forall n \geq n_0) [t(n) \leq cf(n)]\}$$

A notação O é chamada de ordem de $f(n)$, e representa o conjunto de todas as funções $t(n)$ limitadas superiormente por um múltiplo positivo de $f(n)$, considerando-se n um valor suficientemente grande para que esse efeito ocorra. Diremos, portanto, que $t(n)$ é da ordem de $f(n)$ se $t(n) \in O(f(n))$. Essa notação é extremamente importante para a análise de algoritmos, especialmente para expressar sua complexidade de execução em função do tempo. Assim, se desejamos avaliar a função tempo para a execução de um algoritmo, em que a variável de entrada é, por exemplo, o comprimento da entrada de dados, $O(f(n))$ significa que o tempo gasto por um algoritmo para alcançar seu resultado é proporcional a $f(n)$. A notação O objetiva proporcionar um embasamento matemático de avaliação de eficiência. Quando usamos a notação O para representar o comportamento de uma função estamos abrindo mão de seu valor exato e nos concentrando na ordem de sua grandeza, ou no seu comportamento assintótico, daí o nome da notação. Essa simplificação é de extraordinário efeito para o cálculo da eficiência do algoritmo, sem deixar, contudo, de ser um indicador bastante aproximado de seu comportamento para valores suficientemente grande de n . Se $f_1(n) = n^4 - 2n^2 - 20n$ e n for suficientemente grande, o termo em n^4 dominará amplamente o resultado final da expressão. Realmente, se pensarmos uma $f_2(n) = n^4 + 2n^2 + 20n$, para $n = 1.000.000$, por exemplo, os valores percentuais de $f_1(n)$ e $f_2(n)$ estarão surpreendentemente próximos. Então, se existir uma função $g(n)$ tal que $f(n) \leq c$

$\times g(n)$ para uma constante $c > 0$, diremos que $O(f(n)) = O(g(n))$. No exemplo da $f(n)$ anterior, para um conveniente valor da constante c o termo em n^4 seria dominante e poderíamos fazer $g(n) = n^4$. Nesse caso $O(n^4 - 2n^2 - 20n) = O(n^4 + 2n^2 + 20n) = O(n^4)$.

C.2.2 – Notação Assintótica com Vários Parâmetros

Como ficou claro no item anterior, a notação assintótica O tem um objetivo bem claro dentro do cálculo da eficiência dos algoritmos. Em muitos casos o tempo de execução de um algoritmo depende simultaneamente de mais de um parâmetro. Esse é o caso, por exemplo, de algoritmos em grafos que envolve a busca em vértices (valor n) e em arestas (valor m). Nesse caso, fazendo:

$$f: \aleph \times \aleph \rightarrow \Re^* \text{ uma função arbitrária}$$

então podemos estabelecer a seguinte notação:

$$O(f(m, n)) = \{t: \aleph \times \aleph \mid (\exists c \in \Re^+) (\exists m_0, n_0 \in \aleph) (\forall n \geq m_0) (\forall m \geq n_0) [t(m, n) \leq cf(m, n)]\}$$

C.2.3 – Notação Assintótica Condicional

Alguns algoritmos são facilmente analisáveis se consideramos instâncias que satisfazem a certas condições. Uma dessas condições mais comuns é a potência de 2. A notação assintótica condicional preenche esse tipo de necessidade para a análise de eficiência de algoritmos. Definindo B como uma constante booleana (falsa, verdadeira), e:

$$f: \aleph \rightarrow \Re^* \text{ e } P: \aleph \rightarrow B \text{ duas funções quaisquer}$$

então podemos estabelecer a seguinte notação que relaciona as funções $P(n)$ e $f(n)$, $n \in \aleph$:

$$O(f(n) \mid P(n)) = \{t: \aleph \rightarrow \Re^* \mid (\exists c \in \Re^+) (\exists n_0 \in \aleph) (\forall n \geq n_0) [P(n) \Rightarrow t(n) \leq cf(n)]\}$$

Lemos a notação acima como a ordem $f(n)$ quando $P(n)$, e corresponde ao conjunto de todas as funções $t(n)$ limitadas superiormente por um múltiplo real positivo de $f(n)$ onde n é suficientemente grande e $P(n)$ ocorre.

C.3 – COMPLEXIDADE DE ALGORITMOS — CONCEITOS BÁSICOS

Estimar a complexidade de um algoritmo é uma tarefa crucial. O caminho de medir empiricamente o desempenho de um algoritmo esbarra em dificuldades que vão desde o equipamento e compilador empregado até as habilidades do programador. Desde o final da década de 1960 um interessante modelo de análise vem sendo adotado. Uma importante providência desse modelo foi propor um modelo geral para a máquina computacional, facilitando sobremaneira a análise da computação das instruções. O modelo RAM (*Random Access Machine*) está expresso na Figura C.1.

O modelo RAM simula o funcionamento de um computador elementar. À memória cabe armazenar os dados do programa. Um programa é constituído por um conjunto de instruções ou comandos que implementa o algoritmo. O modelo considera que cada instrução I possuirá um tempo associado $t(I)$ para ser operacionalizada em RAM. Então, se para executarmos um algoritmo codificado em um programa P , com uma certa entrada fixa, são processadas r_1 instruções do tipo I_1 , r_2 instruções do tipo I_2 , até r_m instruções do tipo I_m . Nesse caso o tempo para executar o programa P será dado por:

$$\sum_{j=1}^m r_j t(I_j) \quad (05)$$

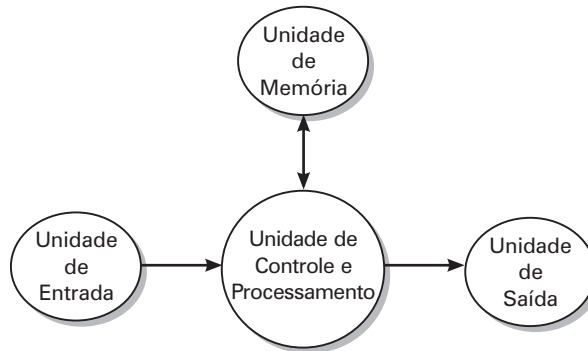


FIGURA C.1 Modelo RAM.

Em última análise, no modelo RAM, o estudo da complexidade de um algoritmo poderia ser resolvido através da avaliação do somatório (05). Para simplificar o problema do cálculo do tempo de duração da computação das instruções do tipo I_j , $j = 1, \dots, m$, considera-se $t(I) = 1$ para qualquer instrução I . Essa simplificação é perfeitamente coerente com o uso da notação $O(\cdot)$ para a avaliação da complexidade em tempo computacional, uma vez que as relações entre a duração dos diversos tipos de instruções é obviamente de natureza constante, o que seria irrelevante no cálculo da ordem de complexidade. Outra vantagem em adotarmos $t(I) = 1$ é que assim o valor do tempo de execução de um programa iguala-se ao número total de instruções computadas.

Denominando por *passo* de um algoritmo α à computação de uma instrução do programa P que o implementa, a *complexidade local* do algoritmo α é definida como o número total de passos necessários à perfeita computação de P , para uma certa *entrada de dados* E de comprimento n . Raciocinando assim, a complexidade de um algoritmo α confunde-se com seu tempo de execução, que denominaremos $T(n)$, e será fundamentalmente dependente da entrada de dados.

Nesse ponto, é interessante firmar o discernimento entre um problema e suas instâncias. O termo *instância* refere-se a uma especificação de valores dados aos parâmetros de entrada num determinado momento, satisfazendo às condições ou restrições próprias do problema. O tamanho de uma instância é o total de códigos necessários à sua identificação, considerando o tipo e a estrutura de dados utilizada. E , no cálculo de complexidade, refere-se ao tamanho da instância, independentemente dos valores associados aos parâmetros que alimentarão o programa P . Nesse sentido, torna-se fundamental bem avaliar o tamanho da entrada.

Como normalmente estamos interessados em avaliar o desempenho de algoritmos sobre *instâncias* de grande tamanho, seria muito útil se pudéssemos definir um limite superior para o número de passos do algoritmo α quando, através de P , atuasse sobre a tal entrada E . Define-se *complexidade local assintótica* de um algoritmo como sendo um limite superior da sua complexidade local, para uma certa entrada E , julgada suficientemente grande. Em tese, estamos sempre interessados no verdadeiro valor da complexidade do algoritmo, mas ele poderá ser extremamente difícil de ser calculado. O uso do conceito de complexidade local assintótica é uma simplificação adequada quando pudermos encontrar um limite superior para a complexidade suficientemente próximo ao valor exato procurado. Nesse ponto buscaremos o auxílio da notação $O(\cdot)$. Usando essa notação estaremos aptos a encontrar com mais facilidade um limite superior para função complexidade em entradas de grande tamanho.

Para que possamos completar uma análise introdutória do tema só falta a consideração do comportamento das instâncias. A complexidade assintótica de um algoritmo não será única pois, mesmo para entradas de igual comprimento, P poderá exigir um número de passos completamente diferente. Dentro de uma grande diversidade de entradas duas são importantes: a que conduz a um desempenho mais otimista e a que leva a um mais pessimista.

Sabendo que $T(n) \leq c \times f(n)$, para algum valor de $c > 0$ e $\forall n$, definimos *complexidade de pior caso* $O(f(n))$ ao valor máximo dentre todas as suas complexidades assintóticas, para entradas de tamanho suficientemente grandes. Definimos *complexidade de melhor caso* $\Omega(f(n))$ ao valor mínimo máximo dentre todas as suas complexidades assintóticas, para entradas de tamanho suficientemente grandes. A complexidade de melhor caso corresponde a um limite superior do número de passos necessários à computação da entrada mais desfavorável. Eventualmente o comportamento assintótico do algoritmo poderá ser limitado simultaneamente por ambas as complexidades anteriormente definidas. Nesse caso particular, introduziremos a noção de *ordem exata* de $f(n)$, $\Theta(f(n))$:

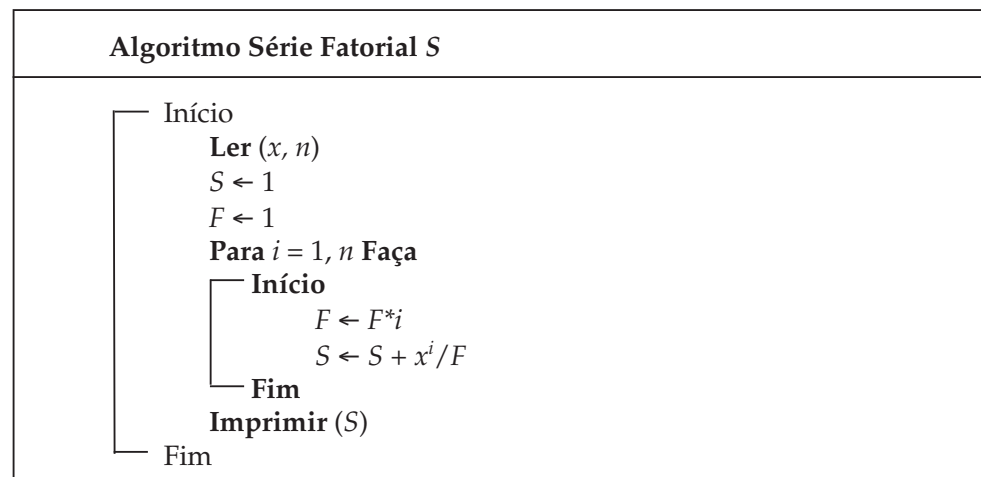
$$\Theta(f(n)) = O(f(n)) \cap \Omega(f(n))$$

Algumas propriedades das funções assintóticas:

1. $O(f(n)) + O(g(n)) = O(\max\{f(n), g(n)\})$
2. $O(cf(n)) = cO(f(n)) = O(f(n))$
3. $O(f(n)) \cdot g(n) = O(f(n)) \cdot O(g(n))$

Vamos exemplificar o cálculo da complexidade através do algoritmo (S) descrito no quadro que se segue. O algoritmo (S) produz uma série matemática de extrema importância, a saber:

$$S(x, n) = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots + \frac{x^n}{n!}$$



Pela propriedade 1 sabemos que a complexidade de S será obtida da complexidade de sua componente mais importante. Podemos dividir as ações de S em três blocos ou subfunções que denominaremos f_1, f_2 e f_3 .

O primeiro bloco diz respeito às ações de leitura e atribuições de inicialização. Poderíamos denominar o primeiro bloco, f_1 , de *passo de inicialização*. A complexidade de f_1 será dada pelas operações nele previstas, a saber:

f_1 = leitura de dois números (2 unidades de tempo) e duas atribuições (2 unidades de tempo)

$f_1 = 4$

O segundo bloco possui um controle de *loop*, uma atribuição com um produto, e uma outra atribuição com uma soma e uma divisão. Notamos que são desenvolvidos $i - 1$ produtos para obter x^i . O *loop* é realizado n vezes ($i = 1, \dots, n$) com 4 operações e $i - 1$ produtos, logo a computação final de f_2 será:

$$f_2 = \sum_{i=1}^n [(i-1) + 4] = \sum_{i=1}^n [i + 3] = \frac{n^2 + 7n}{2}$$

O terceiro bloco é o procedimento de saída e:

$$f_3 = 1$$

Assim a função $T(n) = O(\max \{f_1, f_2 \text{ e } f_3\}) = O(\max \{4, \frac{n^2 + 7n}{2}, 1\}) = O(n^2)$

C.5 – DIFICULDADE DE SOLUÇÃO DOS PROBLEMAS DE PROGRAMAÇÃO LINEAR

Quando formulamos um problema Π estamos implicitamente considerando que temos a esperança de solucioná-lo através de algum algoritmo α . Obviamente solucionar um problema não é apenas definir um algoritmo que o solucione teoricamente, mas obter, na prática, segundo a função objetivo, a posse da configuração desejada. Entendemos por *configuração* um arranjo de variáveis que atende às condições impostas pelas restrições de Π , ou seja, pertence ao espaço das soluções viáveis de Π . Para alguns problemas é fácil elaborarmos algoritmos que nos disponibilizem essa tão desejada configuração através de programas P que rodem rapidamente (sejam eficientes). Para alguns problemas é relativamente fácil exibir algoritmos eficientes de solução. Infelizmente, para outros, isso é uma tarefa surpreendentemente difícil. Entendemos como eficiente um algoritmo cuja $T(n)$ no pior caso é de ordem polinomial.

Quando pudermos encontrar um algoritmo com $T(n)$ polinomial para Π diremos que esse problema é *tratável*. Para provarmos que Π é tratável basta exibir um algoritmo que o solucione em passo polinomial. Para o caso de provarmos que Π é intratável é necessário demonstrarmos que tal algoritmo não existe.

Um problema algorítmico (que admite uma solução via um algoritmo) pode ser caracterizado por um conjunto de todas as possíveis entradas de dados do problema e pela questão a ser solucionada, ou seja, o objetivo do problema. Solucionar um problema é desenvolver um algoritmo que seja capaz de, com base na entrada de dados e no objetivo determinado, encontrar uma configuração de resposta que seja adequada. Supomos que as instâncias de Π serão convenientemente codificadas em E (com os dados estritamente necessários).

Os problemas algorítmicos podem ser classificados em classes específicas, conforme vários critérios. Considerando a natureza da resposta esperada para o problema Π encontramos a seguinte classificação:

Problemas de Decisão

Nesse caso o objetivo proposto em Π é decidir sobre a existência de uma configuração S que atenda às restrições de Π . Uma questão usual nesse tipo de problema é: *existe alguma configuração S que satisfaça às propriedades exigidas em Π* ? A resposta para esse tipo de problema é sim ou não.

Problemas de Localização

Agora o objetivo proposto em Π é localizar ou exibir uma configuração S que atenda à resposta sim do problema anterior ou demonstrar que ela não existe. Uma questão usual nesse tipo de problema é: *exista, se existir, uma configuração S que satisfaça às propriedades exigidas em Π* . A resposta para esse tipo de problema é exibir uma configuração qualquer que atenda às restrições de Π .

Problemas de Otimização

O último tipo de problema algorítmico aprofunda mais a questão levantada em Π . Nesse caso, além de estarmos interessados em exibir a configuração S , estamos interessados em escolhê-la através de um critério de otimização. Uma questão usual nesse tipo de problema é: *exiba, se existir, uma configuração S que satisfaça as propriedades exigidas em P e minimize (ou maximize) uma função $z(x)$ qualquer*. A resposta para esse tipo de problema é uma configuração que atende às restrições de Π e otimiza uma função $z(x)$.

Se o nosso problema Π está associado a percursos hamiltonianos em G :

O Problema de decisão típico seria: existe em G um percurso Hamiltoniano?

O Problema de localização típico seria: determine um percurso Hamiltoniano em G (supomos que a resposta mais provável do problema de decisão é SIM).

O Problema de otimização típico seria: determine o percurso Hamiltoniano em G que minimiza a soma de suas arestas.

Considerando um problema de decisão Π . Se Π for solucionável através da aplicação de algum algoritmo, então existe necessariamente uma justificativa para a solução de Π . Apresentaremos agora uma classificação para os problemas algorítmicos baseada nos estudos de Cook (1971).

Define-se a classe dos problemas de decisão P como sendo aquela que compreende os que admitem um algoritmo polinomial de solução. Como existe um algoritmo polinomial que verifica se um grafo é conexo, por exemplo, então o problema da *conexidade em grafos* é P .

Define-se a classe de problemas NP como aquela que compreende todos os problemas de decisão Π , tais que existe uma justificativa à resposta sim para Π , cujo passo de reconhecimento pode ser realizado por um algoritmo polinomial no tamanho da entrada E de Π .

A inclusão de um problema na classe NP anteriormente formalizada não obriga que exista uma solução polinomial para o problema associado a Π . Exige-se apenas que o tamanho da justificativa do problema de decisão seja polinomial no tamanho da entrada E . Nada é exigido também em relação à justificativa para não. Podemos dar inúmeros problemas pertencentes à classe dos NP . O problema de determinação do ciclo hamiltoniano em um grafo G pertence a essa classe. A justificativa para a resposta sim do problema de decisão consiste no reconhecimento do ciclo hamiltoniano, ou seja, a exibição de uma sequência de vértices (exatamente n vértices) e sua verificação se de fato o ciclo é hamiltoniano. Obviamente todo problema P é automaticamente NP , contudo a recíproca é uma questão em aberto.

Quando existe uma transformação em tempo polinomial de um problema de decisão Π_1 para outro Π_2 (escrevemos $\Pi_1 \propto \Pi_2$), mostra-se que (ver Garey e Johnson) que se $\Pi_1 \in P \Rightarrow \Pi_2 \in P$, ou ainda se $\Pi_1 \notin P \Rightarrow \Pi_2 \notin P$. Podemos dizer que, nesse caso, Π_1 é um caso particular de $\Pi_2 \in P$, ou seja, Π_2 é pelo menos tão difícil quanto Π_1 . A relação \propto é também transitiva, daí se $\Pi_1 \propto \Pi_2$ e $\Pi_2 \propto \Pi_3$ implica que $\Pi_1 \propto \Pi_3$.

Um problema é definido como NP-difícil ou NP-hard se $\Pi_0 \propto \Pi$ para todo $\Pi_0 \in NP$.

Por definição todo problema NP-difícil é tão difícil quanto qualquer problema pertencente a NP . A denominação de problemas de decisão NP-completo foi introduzida por Cook (ver Cook [1983]). Cook provou que era possível dividir NP em classes de problemas equivalentes entre si. Verificou também que a classe P não comportava a possibilidade de que todos os problemas pertencentes a NP fossem nela transformados polinomialmente. Assim a classe NP-completo foi definida como constituída por problemas Π que:

1. $\Pi \in NP$.
2. Para todo problema de decisão $\Pi' \in NP$ então $\Pi' \propto \Pi$.

PUBLICAÇÕES CIENTÍFICAS CITADAS POR SUAS TRADICIONAIS ABREVIATURAS

<i>Amer. Math. Monthly</i>	American Mathematical Monthly
<i>AIIE Transactions</i>	American Institute of Industrial Engineers Transactions
<i>AJMMS</i>	American Journal of Mathematical and Management Science
<i>Ann. Disct. Math.</i>	Annals of Discrete Mathematics
<i>Annals of Maths. and AI</i>	Annals of Mathematical and Artificial Inteligency
<i>Applied Math. Let.</i>	Applied Mathematical Letters
<i>Applied Math. Modelling,</i>	Applied Mathematical Modelling
<i>Bell. Syst. Tech. J,</i>	Bell Systems Technical Journal
<i>Bull. Amer. Math. Soc.</i>	Bulletin of American Mathematical Society
<i>Canad. J. Math.</i>	Canadian Journal of Matematics
<i>Comp. & Maths. with Appls.</i>	Computers & Mathematics with Applications
<i>Computers & Ops. Res</i>	Computers & Operations Research
<i>Commun, Statist. Stochastic Models</i>	Communications Statistic and Stochastic Models
<i>EJIS</i>	European Journal of Information Systems
<i>EJOR</i>	European Journal of Operational Research
<i>Decision Sci.</i>	Decision Science
<i>Discrete Applied Math.</i>	Discrete Applied Mathematics
<i>DMJ</i>	Duke Mathematical Journal
<i>Eng. Opt.</i>	Engineering and Optimization
<i>Geogr. Anal.</i>	Geographic Annals
<i>IEEE-AC</i>	IEEE Transactions on Automatic Control
<i>IEEE. Trans. Eletromag. Compat.</i>	IEEE Transactions on Eletromagnetism and Compatibility
<i>IIE Transactions</i>	Institute of Industrial Engineers Transactions
<i>IEEE Trans. Comm.</i>	IEEE Transactions on Communications
<i>IEEE Trans.Comp.</i>	IEEE Transactions on Computers
<i>IBM J. R&D</i>	IBM Journal of Research and Development
<i>IBM Systems, J.</i>	IBM Systems Journal
<i>Inf. Processing Letters</i>	Information Processing Letters
<i>IJAMT</i>	International Journal of Advanced Manufacturing Technology
<i>Intern. J. Computer Math.</i>	International Journal of Computer Mathematics
<i>Int. J. Math. Educ. Sci. Technol..</i>	International Journal of Mathematic, Education, Science and Tecnology
<i>Int. J. Prod. Res.</i>	International Journal of Production Research
<i>Journal of Alg.</i>	Journal of Algorithms
<i>JORS</i>	Journal of the Operational Research Society
<i>J. Assoc. Comput. Math.</i>	Journal of the Association for Computing Machinary
<i>J. Ind. Eng.</i>	Journal of Industrial Engineers
<i>J. Opt. Theory Appl.</i>	Journal of Optimization Theory and Applications
<i>J. ACM</i>	Journal of ACM
<i>J. Comput. Syst. Sci</i>	Journal of Computer and Systems Science

<i>J. Res. Nat. Bur. Std.</i>	Journal of Research of the National Bureau of Standards
<i>J. Opns. Res. Soc.</i>	Journal of Operations Research Society
<i>J. Opns. Res. Soc. of Japan</i>	Journal of Operations Research Society of Japan
<i>J. Polit. Economy</i>	Journal of Political and Economy
<i>Man. Sci.</i>	Management Science
<i>Math. Prog.</i>	Mathematical Programming
<i>Math. Prog. Study</i>	Mathematical Programming Study
<i>Math. Comput. Modelling</i>	Mathematical and Computer Modelling
<i>Math. Slovaca</i>	Matematical Slovaca
<i>Math. Japonica</i>	Matematical Japonica
<i>MOR</i>	Mathematics of Operations Research
<i>Numerische Math.</i>	Numerische Mathematics
<i>Nav. Res. Log. Q.</i>	Naval Research Logistics Quarterly
<i>NRL</i>	Naval Research Logistics
<i>Problem. Mangement Sci.</i>	
<i>Proc IEEE</i>	Proceedings IEEE
<i>Proc. Amer. Math. Soc.</i>	Proceedings of American Mathematical Society
<i>Quart. Appl. Math.</i>	Quartely Applied Mathematics
<i>RAIRO – Operations Research Res. Rept.</i>	Reveu d'Automatique d'Informatique et de Recherche Operationnelle – Operations Reserach.
<i>SIAM – J. Appl. Math.</i>	Society for Industrial and Applied Mathematics Journal on Applied Mathematics
<i>SIAM – J. Comp.</i>	Society for Industrial and Applied Mathematics Journal on Computing
<i>SIAM J. Opt.</i>	Society for Industrial and Applied Mathematics Journal on Optimization
<i>SIAM Review</i>	Society for Industrial and Applied Mathematics Review
<i>Soviet Math. Dokl</i>	Soviet Mathematics Doklay
<i>Opns. Res. Soc.</i>	Operations Research Society
<i>Opns. Res</i>	Operations Reseach
<i>ORL</i>	Operations Reserch Letters
<i>OR Spektrum</i>	Operations Reserch Spektrum
<i>Opns. Res. Quart.</i>	Operations Research Quartely
<i>Telecomm Sys.</i>	Telecommunications Systems
<i>Transactions Sys. Man. Cyb.</i>	Transactions on Systems Management and Cybernetics
<i>Trans. Sci.</i>	Transportation Science
<i>YUJOR</i>	Yugoslav Jornal of Operations Research
<i>Zastos Mat</i>	Zastos Mathematic

PUBLICAÇÕES CIENTÍFICAS CITADAS SEM ABREVIATURAS

Aerospace Science and Technology
ACM Computing Surveys
Acta Astronautica
Acta Informatica
Acta Mathematica Universitatis Comenianae
Advances Applied Probability
Advances in Complex Systems
Algorithmica
American Mathematical Society
American Journal of Mathematical and Management Sciences
Analytica Chimica Acta
Annals of Operations Research
Annual Review of Ecology and Systematics
Applied Statistics
Applied Intelligence
Applied Thermal Engineering
Artificial Intelligence in Engineering
Artificial Intelligence Review
Asia – Pacific Journal of Operational Research
Blackwell Synergy
Belgian Journal of Operations Research, Estatistic, and Computer Science
Biometrika
Biological Cybernetics
BT Technology Journal
Building and Environment
Bulletin of the Polish Academy of Science, Technical Science
Canadian Journal of Civil Engineering
Cashier Cent. d'Etudes Recherche Operationelle
Cement and Concrete Research
Central European Journal for Operations Research and Economics
Chaos, Solitons and Fractals
Combinatorial Optimization
Communications of the ACM
Complex Systems
Composite Structures
Computational Optimization and Applications
Combinatorial Optimization: Theory and Practice
Computers and Chemical Engineering
Computers and Industrial Engineering
Computing
Decisions and Control in Management Science
DIMACS Series on Discrete Mathematics and Theoretical Computer Science
Documenta Mathematica
Electric Power Systems Research
Engineering Applications of Artificial Intelligence
Engineering Design & Automation
Engineering Optimization
Environmental Modeling & Assessment
Evolutionary Computing
Fuzzy Sets and Systems
Future Generation Computer Systems

Genetic Programming and Evolvable Machines
Geoinformatica
IEE Electronic Letters
IEEE Journal of Oceanic Engineering
IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems
IEEE Transactions on Evolutionary Computation
IEEE Transactions on Image Processing
IEEE Transactions on Magnetics
IEEE Transaction on Neural Networks
IEEE Transaction on Systems Science and Cybernetics
Information and Control
Information Processing Letters
Information Sciences
Information Technology and Tourism
Interfaces
INFOR
INFORMS Journal on Computing
The International Journal of Advanced Manufacturing Technology
International Journal of Pattern Recognition and Artificial Intelligence
International Journal of Physical Distribution & Logistics Management
International Journal of Production Economics
International Transactions in Operational Research
Investigacion Operativa
Journal de Physique
Journal of Advanced Transportations
Journal of Applied System Sciences
Journal of Artificial Intelligence Research
Journal of Business
Journal of Business Logistics
Journal Chemical and Physical
Journal of Combinatorial Optimization
Journal of Computational Physics
Journal of Computer and System Sciences
Journal of Control and Cybernetics
Journal of Crystal Growth
Journal of Economic Dynamics and Control
Journal of Electroanalytical Chemistry
Journal of Experimental Algorithmics
Journal of food engineering
Journal of Forecasting
Journal of Global Optimization
Journal of Intelligent Manufacturing
Journal of Heuristics
Journal of KISS
Journal of Manufacturing Systems
Journal of Optimization Theory and Applications
Journal of Power Sources
Journal of Quantitative Spectroscopy and Radiative Transfer
Journal of Regional Science
Journal of Scheduling
Journal of Sound and Vibration
Lectures Notes in Computer Science
Materials Science and Engineering
Mathware and Soft Computing
Microeconomie
Microprocessors and Microsystems
Mineral and Metallurgical Processing
Natural Computing
Networks
New Scientist
Nucleic Acids Research
Optics and Lasers in Engineering
Optics & Laser Technology



Optimization and Engineering
Omega
ORSA, Artificial Intelligence Newsletter
Pacific Journal of Mathematics
Parallel Computing
Pattern Recognition
Pesquisa Naval
Pesquisa Operacional
Physical Review
Production and Manufacturing systems
Production Planning and Control
Revista Militar de Ciência e Tecnologia
Revue Francaise D Informatique De Recherche Operationnelle
Ricerca Operativa
Robotics and Autonomous Systems
Statistics and Computing
Socio-Economic Planning Science
Science
Science and Technology
Studies in Locational Analysis
TAPPI Journal
Technique et Sciences Informatiques
Technometrzcs
Telecommunications Systems
The International Journal of Computer and Telecommunications Networking
Theoretical Computer Science
Transactions of Science
Transportation Research
Transportation Strategy
Zeitschrift für Operations Research

BIBLIOGRAFIA

Para obter a bibliografia deste livro, consulte nosso site *www.elsevier.com.br*.

ÍNDICE

A

Algoritmo

- Aproximativo, 196, 248, 351
- Busca Geral em Grafos, 230
- Carteiro Chinês, 377, 378, 380
- Cobertura Prime, 433
- de Christofides, 172, 261, 351, 383, 410, 441
- de Circulação, 304, 305, 309
- de Clark e Wright, 367, 400
- de Dijkstra, 187, 234, 235, 236, 238, 270
- de Gillet e Miller, 404
- de Kruskal, 241, 244, 245, 473, 506
- de Malhotra, Pramodh-Kamar e Maheshwari, 311
- de Minoux, 240, 246, 410
- de Mole e Lameson, 402
- de Recobrimento em Arestas, 263
- de Roy, 231
- de Sleator e Tarjan, 241, 453, 454
- Ford-Fulkerson, 238, 309, 372
- Ford-Moore-Bellman, 236, 237
- Genético, 351, 354, 436
- GRASP, 436, 437
- Guloso para o PK, 186, 437, 438, 506
- Húngaro, 294, 298-299
- K-Dispersão de Erkut, 438-439, 440, 441, 442, 443, 444, 445
- K-Separação, 451, 452
- M³S Busca, 239, 240, 246, 247, 340
- Prim Colorido, 241-244
- Prim, 241, 242, 506
- Simplex para o PFCM, 316, 317
- Simplex, 96, 100, 106, 115, 119-120, 1'25, 138-140, 141, 275, 302, 317, 463
- Árvore Geradora Mínima, 241, 242, 349, 350, 361, 449, 450
- Árvore, 172, 182-185, 195, 196, 229, 239, 242, 243-247, 259, 261-263, 282, 283, 288, 324, 349, 350, 361

B

Branch-and-Bound, 180

Busca

- em Largura, 183
- em Profundidade, 183, 184, 186, 231

C

Caminho

- Alternante, 250-251
- Caminho Aumentante, 247-250
- Caminho de aumento de fluxo, 307

Carteiro Chinês, 377, 378

Ciclagem, 119, 120

Circuitos Eulerianos, 233

Circuitos Hamiltonianos, 233, 366, 505

Clique, 486

Competitividade, 4

Complexidade Local Assintótica, 502, 503

Conexão, 229, 336, 430, 450

Conjunto convexo, 94, 95, 96, 475, 476, 478-479

Conjunto Gerador, 471-472

Corte s-t, 306

Problemas de Coberturas Capacitadas, 428

Cromossomo, 351-358, 360-361

Custos Reduzidos, 100, 117, 125, 139, 152, 158, 285, 324

D

Degeneração, 119, 120

Dependência Linear, 103, 467, 468

Determinante, 98, 275, 462, 465, 466, 472, 473

Direções Extremas, 478

Dispersão, 441, 442, 445-447, 448, 449

Dualidade, 129, 130, 131, 133, 137, 165, 435, 445

E

Emparelhamentos, 229, 249, 250, 251, 445
 Engenharia de Sistemas, 3
 Envoltória convexa, 92, 120, 180, 479, 480
 Equações Lineares, 93, 182, 479, 480
 Espaço Vetorial, 471, 472, 477
 Estratégias
 Dinâmicas, 184

F

Fluxo de Custo Mínimo, 275, 276
 Fluxo Máximo, 274, 302, 306-307, 309-310, 316
 Fluxos aumentante, 307
 Folgas Complementares, 132, 136, 137, 290
 Formulação
 Clássica do Packing (PK), 412
 Clássica do Particionamento (PP), 412
 Clássica para o Problema de Transporte, 278
 de Claus (C), 335
 de Fox-Gavish-Graves (FGG), 334
 de Miller-Tucker-Zemlin (MTZ), 333
 de Produto Único (FPU), 259
 Matemática do (PCCS), 301, 422
 Multiproduto (FPM), 260
 Restrita para o Problema de Transporte, 278

G

Geni and Genius, 365
 Gestão, 17, 18, 21
 Grafo
 Bipartido, 277, 297, 486
 Completo, 361, 367, 486
 de Aumento de Fluxo, 307
 Direcionado, 175, 485, 494
 Regular, 487
 GRASP, 436, 439-440

H

Heurística
 de "Saving", 367
 de Backer, 436
 de Balas e Ho, 435
 de Bellmore e Nemhauser, 361, 362, 369
 de Chvatal, 169, 435
 de Erkut, 442, 443, 444
 de Gillet e Miller, 403, 404
 Híbridas, 183, 435
 de Inserção, 363-365
 de K-Substituições ou k-Opt, 365
 de Mole e Jameson, 401, 402
 de Separação por Grupamentos, 445

de Vasko e Wilson, 435
 Dynasearch, 365
 L&K, 366
 Primais, 433

I

Implantes Radioativos, 440
 Independência Linear, 103, 285
 Inteligência Artificial, 16, 17
 Interpretação
 Econômica, 1, 140-141, 142, 145, 151, 258
 Gráfica, 125, 483
 Inversão de Matriz, 108, 462, 463

L

Limites, 14, 15, 17-18, 133-134, 141, 253, 276, 294, 302,
 304-305, 338, 349-351, 361, 366, 413, 418, 426
 Lista de adjacências, 360
 Lista Ordinal, 360

M

Matriz
 de Adjacência, 494
 de Incidência, 318, 324, 494
 Matriz Triangularizada, 319
 Meta-heurísticas, 436
 Método
 das duas fases, 115, 124, 127
 de Vogel, 183, 184
 do Canto Noroeste, 286
 Modelos
 Analógicos, 7
 Icônicos, 7
 Simbólicos, 7

N

Notação O, 500, 501
 Núcleo, 1, 351
 Número
 de Absorção, 415
 de Estabilidade, 416, 196
 de Independência, 416, 496

O

Operação Elementares com Matrizes, 108, 457
 Operadores dos Algoritmos Genéticos, 351, 354, 355
 Otimização Combinatória, 239

P**PCV**

- com Backhauls, 337
- com Bônus, 338, 339
- com Clientes Estocásticos, 340
- com Gargalo, 338
- com Janela de Tempo, 369
- com tempo de Viagem Estocástico, 340
- Estocástico, 339
- Generalizado, 336
- Min-Max-Min-Sum, 340
- Múltiplo com Tempo de Viagem Estocástico, 340
- Múltiplo, 340
- Simétrico, 336, 365, 368

Pesquisa Operacional, 10, 12, 13, 411

Pivoteamento, 97, 108-109, 111, 139, 140, 145, 156-159

Pontos Extremos, 96, 99, 100, 120, 122, 125, 475, 476

Princípio de Bellman, 170

Problema

- da Cobertura Maximal em Caminho Mais Curto, 428
- da Cobertura Sobre o Contínuo, 417
- da K-Dispersão Discreta, 438
- da Mochila, 170-178, 195, 213, 214
- de Coberturas Capacitadas, 428
- de Designação, 227, 228, 262, 289-290, 292-295, 297, 298, 422, 424
- de Emparelhamento, 261, 273, 277, 278, 279, 281-283, 285-287, 288, 317, 336, 428
- de Fluxo à Custo Mínimo, 275
- de Fluxo Máximo, 274, 309, 310, 311, 316
- de Maximal Covering Location Planning, 424
- de Packing, 262, 413-415, 416, 417
- de Particionamento, 120, 411, 412, 413, 414, 415, 423, 431, 460
- de Recobrimento, 261, 331, 432
- de Roteamento e Designação, 229, 230, 248-252, 350
- de Roteamento, 177, 196, 233, 240, 262, 336, 361, 375-377, 385-393, 398-399
- de Steiner, 253-263
- de Transporte, 193, 214, 261, 271, 277-279, 280, 281, 282-286, 288
- do Caixeiro Rural, 384
- do Caminho mais Curto, 232, 233
- do Carteiro Chinês Capacitado, 382
- do Carteiro Chinês, 377, 378, 381-382, 383
- do Transbordo, 277, 295, 296, 301

Programação

- Dinâmica, 170
- Inteira, 11, 17, 169, 170
- Linear, 111, 16, 92-93, 97, 110, 119-120, 124-125, 131, 135, 137, 140, 170, 350, 411, 412, 476

Q

Quadro de Transporte, 281, 282, 283, 285, 286

Qualidade, 19, 140, 183, 197, 198, 218, 282, 350, 361, 365, 369, 370

R**Rede**

- definição, 488
- de Abastecimento, 217, 218
- de Lanchonetes, 219
- de Oferta x Demanda, 278
- de Petri, 7
- de Televisão, 225, 226
- PERT, 326, 410
- PERT/CPM, 326, 327
- de Água, 270
- de Comunicações, 256, 261, 272, 425, 426, 441
- de Distribuição, 425
- de Drenagem, 257
- de Energia Elétrica, 430
- de Esgoto, 270
- Locais, 261

Relaxação

- Lagrangeana, 184, 254, 431
- Linear, 174, 184, 431

Roteamento de Veículos, 240, 412

Rotulação, 232, 234, 235, 250, 253, 294, 300, 301, 304, 305, 310

S

Simulated Annealing, 203, 432

Sinergia, 4

Sistemas

- Biológicos, 351
 - conceito, 3, 4
 - de Atendimento de Emergência, 447
 - de Comunicação, 273
 - de Equações, 92, 93, 96, 154, 280, 287, 319, 320, 460-461, 465-473
 - de Exame de Sangue, 455
 - de Independência, 506
 - de Manufatura Flexível, 448
 - de Saúde, 450
 - de Tomada de Decisão, 16
 - de Transporte, 271
 - Indeterminado, 474
 - Lineares, 461, 465-473
 - Submarinos de Exploração de Petróleo, 418
- Solução Básica, 92-96, 101-102, 105, 125, 134, 281, 285, 286
- Submatriz, 98, 280, 318, 460, 462, 470-473
- Simulação, 1, 14, 16, 17, 23

T

T-coloração, 497

Teorema

- da Existência, 133
- de Camion, 473
- de Cramer, 465

de Heller & Tompkins, 473
de Hoffman e Kruskal, 473
de Truemper, 473
Tomada de Decisão, 11, 12, 16, 17, 18, 173, 361, 365

U

Unimodularidade, 472-473

V

Variáveis

Artificiais, 114, 115, 117
Básicas, 92, 94, 95, 106, 115, 119-120, 253
de Folga, 106, 110, 113, 115, 120
Inteiras, 333
Vivacidade, 19-23