Universidade de São Paulo
Escola de Engenharia de São Carlos

Análise de curvas de *trade-off* baseada em teoria de rede de filas para o projeto e planejamento de sistemas discretos de manufatura

Reinaldo Morabito

Tese apresentada à Escola de Engenharia de São Carlos, Universidade de São Paulo, para Concurso de Livre-Docência junto ao Departamento de Engenharia Mecânica, na área de conhecimento: Pesquisa Operacional.

São Carlos - SP
Março de 1998

para Julia e Cláudia

Agradecimentos

Gostaria de agradecer principalmente ao Prof. Gabriel R. Bitran, Sloan School of Management, M.I.T., por ter me introduzido e motivado para o estudo de teoria de rede de filas aplicada a sistemas de manufatura. Sua experiência e interesse nesse tópico foram contribuições importantes para a realização desta pesquisa. Sou muito grato à sua dedicação e ao seu incentivo, especialmente durante o período de agosto de 1993 a dezembro de 1994, quando estivemos trabalhando juntos na Sloan School em Cambridge, MA.

Também gostaria de agradecer ao Dr. Deb Sarkar, AT&T Bell Laboratories, ao Prof. Devanath Tirupati, Universidade do Texas em Austin, e aos doutorandos da Sloan School, Luis A. C. Pedrosa e Amit Dhadwal, pelos seus úteis comentários e sugestões na revisão dos artigos que serviram de base para esta tese. Também sou grato ao Prof. Flávio C. F. Fernandes, UFSCar, à pesquisadora Gisele C. Fontanella Pileggi, CNPq, e à doutoranda Renata Algisi, USP, pela ajuda na revisão final e formatação do texto.

Finalmente, agradeço à FAPESP e ao M.I.T. pelo apoio ao meu programa de pós-doutorado, cujo projeto de pesquisa deu origem à esta tese, e às visitas posteriores a Cambridge, para discutir resultados da pesquisa com o Prof. Bitran. Fico ainda em débito com meus pais, sempre presentes, e em particular com minha esposa, pelo carinho e paciência durante os meses em que passei escrevendo este texto. Aproveito para manifestar que o presente trabalho foi um motivo de grande satisfação pessoal.

Análise de curvas de *trade-off* baseada em teoria de rede de filas para o projeto e planejamento de sistemas discretos de manufatura

Sumário

Lista de figuras

Lista de tabelas

Lista de principais símbolos e siglas

1{.} função indicadora que resulta 1 se a expressão {.} é verdadeira e 0 caso contrário
a_0 intervalo de tempo entre chegadas externas (i.e., da estação 0) na rede
a_{0j} intervalo de tempo entre chegadas externas na estação j
a_j intervalo de tempo entre chegadas na estação j
a_{ji} intervalo de tempo entre chegadas na estação j da estação i
$a_{j,q}$ intervalo de tempo entre a (q-1)-ésima e a q-ésima chegada na estação j
a'_k intervalo de tempo entre chegadas externas da classe k
a_{kl} intervalo de tempo entre chegadas da classe k para a operação l na estação n_{kl}
a'_{kl} intervalo de tempo entre chegadas da agregação de todas as classes que chegam entre duas chegadas sucessivas da classe k para a operação l na estação n_{kl}
a^p_j intervalo de tempo entre chegadas na estação j na p-ésima iteração dos algoritmos 3a, 4a e 5a.
α_j coeficiente da função custo de capacidade F_j
β_j coeficiente da função custo de capacidade F_j
c_j perda financeira na transação de venda de capacidade da estação j
cx coeficiente quadrático de variação (*scv*) da variável aleatória x, definido por: $cx = V(x) / E(x)^2$
$Cov(x,y)$ covariância entre as variáveis aleatórias x e y.
CQN rede de filas fechada (*closed queueing network*)
χ_j coeficiente da função custo de capacidade F_j
D atraso em fila, dado que ele é positivo: ($Wq \mid Wq > 0$). No caso de um sistema de fila, indica que o processo de chegada ou serviço é determinístico.
d_j intervalo de tempo entre partidas da estação j
d_{ji} intervalo de tempo entre partidas da estação j para a estação i
$d_{j,u}$ intervalo de tempo entre a (u-1)-ésima e a u-ésima partida da estação j
d_{kl} intervalo de tempo entre partidas da classe k após a operação l na estação n_{kl}
Δ incremento de capacidade em cada iteração dos algoritmos 3 e 4
e_j intervalo de tempo entre chegadas externas das classes na estação j, levando em conta o número de visitas de cada classe nesta estação
$\mathbf{e}_i$ vetor (e_1, e_2, ..., e_n) com $e_i = 1$ e $e_j = 0$, $j = 1, ..., n, j \neq i$
$E(x)$ (ou $\bar{x}$) valor esperado da variável aleatória x
$E(x)_{GI/G/m}$ valor esperado da variável aleatória x no sistema de fila *GI/G/m*
f_j custo unitário de capacidade na estação j
F custo (ou investimento) de capacidade da rede
F_j custo de capacidade da estação j
F_{ji} custo de capacidade da estação j na alternativa i
F_T limitante superior para o custo de capacidade da rede, F
FCFS primeiro a chegar, primeiro a ser servido (*first come, first served*)
FMS sistema flexível de manufatura (*flexible manufacturing system*)

φ_j nível de serviço na estação j
G processo de chegada ou serviço genérico num sistema de filas
GI processo de chegada genérico independente (processo de renovação) num sistema de filas
GAMS *General Algebraic Modeling System*
γ_j fator multiplicador para criação e combinação de *jobs* na estação j
i, j índices em geral utilizados para indicar estações
iid independente e identicamente distribuído
k índice em geral utilizado para indicar classes
l índice em geral utilizado para indicar operações
L *WIP* da rede
$\mathbf{L}$ vetor $(L_1, L_2, ..., L_n)$ ou matriz $(\mathbf{L}_1, \mathbf{L}_2, ..., \mathbf{L}_n)$ representando o estado da rede
$\mathbf{L}_j$ vetor $(L^1_j, L^2_j, ..., L^r_j)$ representando o estado da estação j
L_j número de *jobs* na estação j
$\overline{L}_j$ igual a $E(L_j)$, número médio de *jobs* na estação j
L_{ji} número de *jobs* na estação j na alternativa i
$\overline{L}_{ji}$ igual a $E(L_{ji})$, número médio de *jobs* na estação j na alternativa i
L^k_j número de *jobs* da classe k na estação j
Lq_j número de *jobs* na fila da estação j
L_T limitante superior para o *WIP* da rede, L
λ_0 taxa média de chegada ou partida externa na rede (ou taxa média de produção da rede)
λ^*_0 capacidade da rede (limitante superior para a taxa média de produção da rede, λ_0)
λ_{0j} taxa média de chegada externa (estação 0) na estação j
λ_j taxa média de chegada na estação j
λ_{j0} taxa média de partida externa da estação j
λ_{ij} taxa média de chegada na estação j da estação i
λ'_k taxa média de chegada externa da classe k
λ^k_j taxa média de chegada da classe k na estação j
λ^k_{0j} taxa média de chegada externa da classe k na estação j
λ^k_{ij} taxa média de chegada da classe k na estação j da estação i
m número de máquinas paralelas e idênticas num sistema de filas de único estágio
$\mathbf{m}$ vetor $(m_1, m_2, ..., m_n)$ do número de máquinas em cada estação da rede
M o número de máquinas disponíveis na rede. No caso de um sistema de filas, indica que o processo de chegada ou serviço é Markoviano (sem memória).
m_j número de máquinas paralelas e idênticas na estação j
m^0_j limitante inferior para o número de máquinas na estação j, m_j
m_{ji} número de máquinas paralelas e idênticas da estação j na alternativa i
$\boldsymbol{\mu}$ vetor $(\mu_1, \mu_2, ..., \mu_n)$ da taxa média de processamento em cada estação da rede
μ_j taxa média de processamento de cada máquina na estação j
μ^0_j taxa média de processamento inicial de cada máquina na estação j
μ^p_j taxa média de processamento na estação j na p-ésima iteração dos algoritmos 3a, 4a e 5a (μ^1_j também pode denotar a taxa média de processamento não transferível da estação j)
μ_{ji} taxa média de processamento de cada máquina da estação j na alternativa i
μ_{kl} taxa média de processamento da operação l do roteiro da classe k
n número de estações internas na rede
n_k número de operações no roteiro da classe k
n_{kl} estação visitada para a operação l do roteiro da classe k
$N(t)$ número de *jobs* que chega na rede (da estação 0) durante o intervalo de tempo $(0,t]$
$N_j(t)$ número de *jobs* que chega na estação j (da estação 0) durante o intervalo de tempo $(0,t]$

$N_k(t)$ número de *jobs* da classe k que chega na rede (da estação 0) durante o intervalo de tempo $(0,t]$

η_j nível de serviço na estação j (diferente de φ_j)

OQN rede de filas aberta (*open queueing network*)

$P(A)$ probabilidade do evento A ocorrer

PI_j índice de prioridade da estação j

π distribuição de equilíbrio da rede

π_j distribuição de equilíbrio da estação j

$\boldsymbol{Q}$ matriz de transição sub-estocástica $\{q_{ij}, i, j = 1, ..., n\}$

q_{0i} probabilidade de um *job* externo entrar na rede pela estação i

q_{i0} probabilidade de um *job*, após ser atendido na estação i, sair da rede.

q_{ij} probabilidade de um *job*, após ser atendido na estação i, seguir para a estação j

q^k_{ij} probabilidade de um *job* da classe k, após ser atendido na estação i, seguir para a estação j

$q_{ij}^{kk'}$ probabilidade de um *job*, após ser atendido na estação i como um *job* da classe k, seguir para a estação j como um *job* da classe k'

q_{kl} proporção de *jobs* da classe k para operação l, dentre os *jobs* na estação n_{kl}

r número de classes na rede

$\boldsymbol{R}$ matriz de transição estocástica $\{r_{ij}, i, j = 0, ..., n\}$

r_{ij} probabilidade de um *job*, após ser atendido na estação i, seguir para a estação j

$r_j(L_j)$ número de máquinas na estação j, em função do estado L_j

ρ_j utilização média da estação j

ρ_{ji} utilização média da estação j na alternativa i

s_j tempo de processamento na estação j

$s_{j,u}$ tempo de processamento do u-ésimo *job* na estação j

$S_{j,p}$ tempo decorrido até ocorrer a p-ésima chegada na estação j

s_{kl} tempo de processamento da operação l de um *job* da classe k

$s_{kl,u}$ tempo de processamento da operação l do u-ésimo *job* da classe k

s^k_j tempo de processamento da classe k na estação j

scv coeficiente quadrático de variação (*squared coefficient of variation*)

SPT menor tempo de processamento (*shortest processing time*)

T *leadtime* de produção da rede

T_k *leadtime* de produção da classe k

v_j valor monetário médio de um *job* na estação j, independente de sua classe

V_j número de visitas na estação j

$V(x)$ variância da variável aleatória x

W_j tempo de espera em fila e serviço na estação j

$\overline{W_j}$ igual a $E(W_j)$, tempo médio de espera em fila e serviço na estação j

Wq_j tempo de espera em fila na estação j

WIP valor esperado do estoque em processo (*expected work in process*)

Resumo

Morabito, R. (1998). "Análise de curvas de *trade-off* baseada em teoria de rede de filas para o projeto e planejamento de sistemas discretos de manufatura". *Tese de Livre-Docência*, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 137p.

Esta tese explora e evidencia o potencial da análise de curvas de *trade-off*, baseada em modelos de redes de filas, para o projeto e planejamento de sistemas discretos de manufatura. Mostra-se como estas curvas podem desempenhar um papel importante para a tomada de decisões com respeito à quantidade e tipo de capacidade necessária para gerir o sistema eficientemente, para avaliar o impacto de incertezas na chegada e processamento de *jobs*, assim como as consequências de mudanças nas taxas médias de produção e no *mix* de produtos. O enfoque é no *trade-off* entre o nível médio de estoque em processo e os custos de capacidade em sistemas *job-shops*. As curvas também podem ser facilmente adaptadas para refletir *leadtimes* de produção, ao invés de estoque em processo. Para gerar as curvas, revisa-se o estado da arte dos modelos de otimização e avaliação de desempenho para redes de filas abertas. Uma questão central nesta tese é a seleção entre as várias configurações para a rede ou, mais especificamente, como os recursos financeiros devem ser adequadamente distribuídos para alocar capacidade nas várias estações. Diversos algoritmos para selecionar a alocação ótima e derivar as curvas são analisados. A metodologia é ilustrada com um exemplo de uma aplicação real numa fábrica de semicondutores.

Palavras-chave: análise de curvas de *trade-off*, projeto de sistema de manufatura, redes de filas abertas, otimização e avaliação de desempenho.

Abstract

Morabito, R. (1998). "*Trade-off curve analysis based on queueing network theory for the design and planning of discrete manufacturing systems*". *Tese de Livre-Docência*, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 137p.

This thesis explores and highlights the potential of trade-off curve analysis, based on queueing network models, for the design and planning of discrete manufacturing systems. It is shown how these curves can perform an important role to support system decisions concerning the amount and type of capacity necessary to manage the system efficiently, to assess the impact of product arrival and processing uncertainties, as well as the consequences of changes in throughput and product mix. The focus is on the trade-off between expected work in process and capacity costs in *job*-shop systems. The curves can be easily adapted to reflect product leadtimes instead of work in process. To generate the curves, the state of the art of optimization and performance evaluation models for open queueing networks is reviewed. A central issue addressed in the thesis is that of selecting an appropriate design for the network, more specifically, how financial resources should be distributed to allocate capacity at the various stations. Algorithms for selecting the optimal allocation and deriving the curves are analyzed. The methodology is illustrated with an example from an actual application in the semiconductor industry.

Keywords: *trade-off curve analysis, manufacturing system design, open queueing networks, optimization and performance evaluation*.

1. Introdução

Grande parte dos produtos que consumimos é produzida em sistemas discretos de manufatura onde os itens são processados individualmente ou em lotes. Um exemplo de sistema discreto é o sistema *job-shop*, que opera com grande diversidade de produtos ou *jobs*, porém em pequenos lotes. O resultado, em geral, é a existência de fluxos complexos de *jobs* ao longo dos *shops* (estações), e longas filas de espera na frente das máquinas. Tais sistemas podem ser freqüentemente representados por modelos de redes de filas, onde os nós correspondem às estações de trabalho e os arcos ligando os nós, aos fluxos de *jobs* entre as estações.

Redes de filas (de manufatura) são em geral difíceis de serem geridas por processarem diversos produtos que têm características diferentes, que partilham os mesmos recursos, e que são afetados por um conjunto de fontes de incerteza tais como variabilidade na demanda, confiabilidade de fornecedores e processo. A complexidade de tais sistemas pode ser reduzida compreendendo melhor os *trade-offs* entre medidas de desempenho e alocação de recursos, no momento em que esses sistemas são projetados ou modificados (o conceito de *trade-off* é aqui entendido como a troca de um benefício por um outro visto como mais vantajoso). Desde Skinner (1974), há mais de duas décadas, autores têm apontado a importância de se entender melhor esses *trade-offs*, entretanto, muitas das análises são essencialmente qualitativas e não fornecem mecanismos para quantificar as relações entre níveis médios de estoque em processo (*expected work-in-process - WIP*), *leadtimes* de produção de produtos (tempo total gasto pelo sistema para fabricar ou montar um produto), taxas médias de produção (*throughput*), carga de trabalho (*workload*), *mix* de produtos, tecnologia, custos e investimentos em capacidade.

Vários artigos procuraram analisar em termos quantitativos os *trade-offs* entre medidas de desempenho e alocação de recursos, muitos deles modelando os sistemas de manufatura como redes de filas *fechadas* (i.e., redes que mantém constante o número de *jobs* circulando). Por exemplo, Shanthikumar e Yao (1987, 1988) e Schweitzer e Seidmann (1991) estudaram o problema de como alocar servidores num *sistema flexível de manufatura* (*FMS*), de maneira a maximizar a taxa média de produção. Stecke e Solberg (1985), Dallery e Stecke (1990) e Stecke e Raman (1994) investigaram a relação ótima entre a alocação de servidores (ou ferramentas em máquinas) e a carga de trabalho em *FMS*. Askin e Krisht (1994) aplicaram procedimentos de otimização para analisar o *trade-off* entre *WIP* e taxas médias de processamento em *FMS*. Vinod e Solbert (1991) e Kouvelis e Lee (1995) analisaram a relação ótima entre o número de servidores e o número de *jobs* circulando nas estações de um *FMS*, de maneira a atender uma taxa mínima de produção.

Outros autores analisaram sistemas de manufatura que podem ser modelados como redes de filas *abertas* (redes onde o número de *jobs* circulando pode variar a cada instante). Por exemplo, Calabrese (1992) estudou a alocação ótima da carga de trabalho em *job-shops* e *FMS* representados por *redes de Jackson* (redes abertas com distribuições exponenciais dos tempos entre chegadas e processamento de *jobs*), para analisar o *trade-off* entre a taxa média de

produção e o nível de congestão. Bitran e Tirupati (1989a, 1989b) discutiram noções de curvas de *trade-off* e desenvolveram procedimentos de otimização para estudar as relações entre *WIP* e custos de capacidade em *redes de Jackson generalizadas* (redes abertas com distribuições genéricas dos tempos entre chegadas e processamento). Estes procedimentos foram refinados em Bitran e Sarkar (1994b) e Bitran e Morabito (1995d, 1997), e estendidos para análise de taxa média de produção em Bitran e Sarkar (1994a). Boxma *et al.* (1990), Frenk *et al.* (1994), Sundarraj *et al.* (1994) e Bretthauer (1996) estudaram a alocação ótima de servidores a estações em redes de Jackson, para analisar o *trade-off* entre *WIP* e custos de capacidade. Van Vliet e Rinnooy Kan (1991) estenderam a análise em Boxma *et al.* (1990) para redes de Jackson generalizadas.

Objetivo e metodologia

O objetivo principal desta tese é explorar e evidenciar o potencial da análise de curvas de *trade-off* para o projeto e planejamento de sistemas discretos de manufatura. Mostra-se como estas curvas podem desempenhar um papel importante para a tomada de decisões nesses sistemas. O enfoque é no *trade-off* entre *WIP* e custos de capacidade em *job-shops* que podem ser modelados como redes de Jackson generalizadas (estas curvas podem ser facilmente adaptadas para refletir *leadtimes* de produção, ao invés de *WIP*). Uma alocação insuficiente de capacidade nesses sistemas pode causar altos níveis de *WIP* e longos *leadtimes*, por outro lado, o excesso de capacidade pode resultar em desperdício de recursos onerosos, devido aos baixos níveis de utilização. Assim, uma questão central nesta tese é a seleção entre as várias configurações para a rede ou, mais especificamente, como os recursos financeiros devem ser adequadamente distribuídos para alocar capacidade nas várias estações. Para isso, são analisados diversos algoritmos para selecionar a alocação ótima e derivar as curvas.

Curvas de *trade-off* são curvas de fronteira ótima que indicam as vantagens (e desvantagens) da troca de um ponto por outro da fronteira. No caso de uma curva de *trade-off* entre os custos de *WIP* e capacidade, para cada custo de *WIP*, a curva indica o ponto com mínimo custo de recursos. A figura 1 ilustra um exemplo de tal curva: Uma empresa competindo na base de baixo *WIP* (ou baixo *leadtime*) pode escolher, por exemplo, o ponto A. Por outro lado, se ela deseja competir na base de baixo investimento em capacidade (recursos), sua escolha pode ser, por exemplo, o ponto B, porém com maior *WIP*. Conforme mostrado no capítulo 5, dados a taxa média de produção, o *mix* de produtos, e as incertezas nos processos de chegada e processamento dos *jobs*, a curva da figura 1 corresponde à fronteira eficiente para o sistema em consideração.

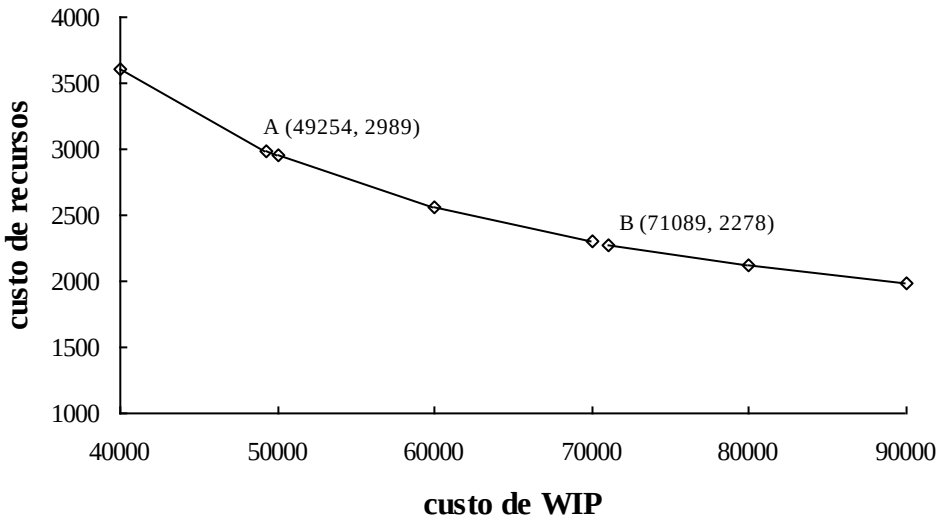


Figura 1 - Curva de *trade-off* entre o custo de recursos e o custo de *WIP*

Para gerar curvas de *trade-off*, revisa-se o estado da arte dos modelos de avaliação de desempenho e otimização em redes de filas abertas. Basicamente, a diferença entre um modelo de avaliação de desempenho e um modelo de otimização é o fato de que o primeiro é *descritivo*, isto é, por meio de medidas de desempenho, ele proporciona ao usuário importantes *insights* sobre a operação do sistema numa dada configuração, enquanto o segundo é *prescritivo*, isto é, ele determina a configuração *ótima* do sistema que otimiza certo critério e satisfaz certas restrições impostos pelo usuário. Uma contribuição desta tese é fornecer uma revisão crítica destes modelos e suas aplicações em sistemas discretos de manufatura. Em particular, poucos autores apresentaram exames dos modelos de otimização de redes de filas abertas aplicados a *job-shops* e, portanto, a presente revisão pretende ser uma contribuição útil para a literatura.

Decisões em manufatura diferem substancialmente nos seus *horizontes de tempo*, isto é, no período em que elas terão impacto. A aquisição de uma nova planta pode ter um horizonte de tempo de 10 ou 20 anos, enquanto que a escolha do próximo *job* a ser processado numa máquina pode ter um horizonte de apenas algumas horas. Muitas decisões na gestão da produção referem-se ao planejamento da capacidade e da carga de trabalho, com horizonte de tempo de 3 a 6 meses ou menos, mas também podem ter horizontes bem maiores, como por exemplo na contratação e treinamento de mão-de-obra especializada. Com base no horizonte de tempo e na extensão do impacto da decisão, é usual classificar as decisões em *estratégicas*, *táticas* e *operacionais*.

Decisões estratégicas são aquelas com longo horizonte de tempo (mais de 1 ou 2 anos) e em geral referem-se ao tamanho e local das plantas e principais instalações, diversidade de produtos, escolha de tecnologia, e grau de automação. Decisões táticas têm um horizonte de tempo intermediário (entre 3 e 18 meses) e determinam o tamanho da força de trabalho, as taxas de produção, a capacidade de espaço para os estoques em processo, o *layout* das plantas. Decisões operacionais são tipicamente aquelas do dia a dia, tais como ordens de produção de *jobs*, e alocação de *jobs* e trabalhadores em máquinas.

Nesta tese a atenção é dirigida principalmente para as decisões de longo e médio prazo, como por exemplo no projeto e planejamento de *job-shops*, e não em aspectos mais operacionais do sistema, como na programação e controle da produção. Decisões de projeto devem considerar os *trade-offs* entre as medidas de desempenho para diferentes configurações do sistema. Uma maneira efetiva de descrever esses *trade-offs* é por meio das curvas de *trade-off*, conforme é aqui explorado. Exemplos das decisões envolvidas são: seleção de produtos e tecnologia, escolha de equipamentos e capacidade, e alocação de produtos a plantas. Para o propósito desta tese, os problemas de projeto são agrupados em três classes propostas em Bitran e Dasu (1992):
(i) *desempenho desejado do sistema* (*SP1 - Strategical Problem* 1)
(ii) *desempenho ótimo do sistema* (*SP2*)
(iii) *partição da instalação* (*SP3*).

Apresenta-se problemas das classes *SP1*, *SP2* e *SP3* formulados como programas de otimização. Na classe *SP1*, o objetivo é minimizar o investimento no sistema sujeito às restrições dos desempenhos desejados para o sistema. Típicas medidas de desempenho podem ser: nível médio de *WIP*, *leadtime* médio de produtos, taxa média de produção, e utilização média (intensidade de

tráfego) de equipamentos. A seguir escolhe-se o nível médio de *WIP* (ou simplesmente *WIP*) como medida de desempenho. Um exemplo da classe *SP1* é dado por:

(*SP1.1*) *WIP desejado:*
- *Objetivo:* minimizar o custo de aquisição de equipamentos
- *Variáveis de decisão:* capacidade de cada estação, tecnologia
- *Restrições:* limitante superior para o *WIP*.

Na classe *SP2*, deseja-se otimizar o desempenho do sistema sujeito às limitações de orçamento para investir no sistema. Um exemplo da classe *SP2* é dado abaixo:
(*SP2.1*) *WIP ótimo:*
- *Objetivo:* minimizar o *WIP*
- *Variáveis de decisão:* capacidade de cada estação, tecnologia
- *Restrições:* limitante superior para o custo de aquisição de equipamentos.

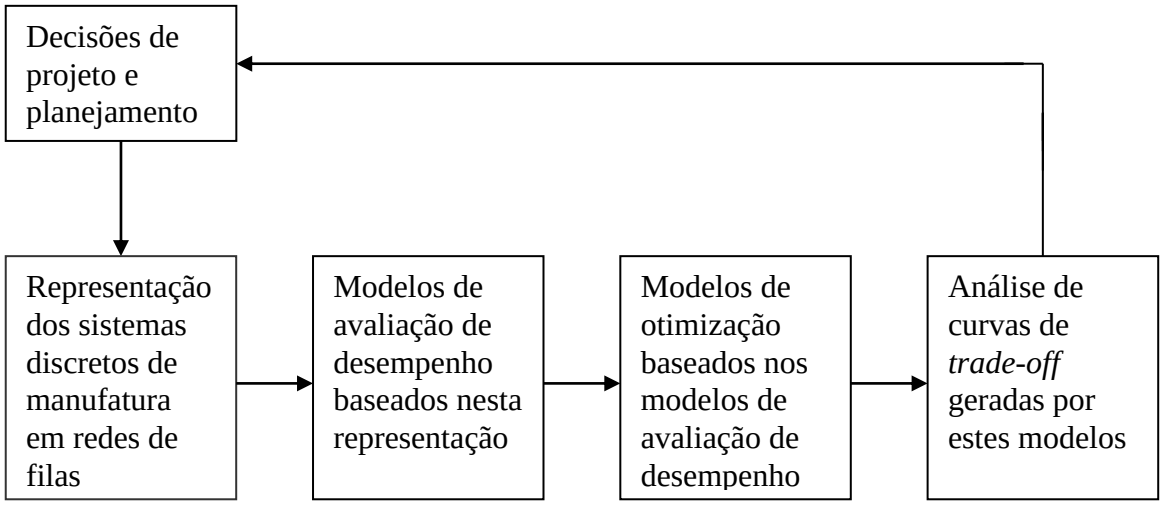
Note que *SP1.1* e *SP2.1* envolvem um *trade-off* entre o capital de investimento e o capital de trabalho. Finalmente, na classe *SP3* procura-se subdividir o sistema de manufatura em unidades de produção (que podem ser vistas como plantas dentro da planta) para melhorar o desempenho global. Entretanto, essa partição pode requerer duplicação de equipamentos e recursos. Considere o seguinte exemplo da classe *SP3*:

(*SP3.1*) *Número de produtos e WIP desejados em cada unidade de produção:*
- *Objetivo:* minimizar o custo de aquisição de equipamentos
- *Variáveis de decisão:* número de unidades de produção, *mix* de produtos em cada unidade, e capacidade de cada estação
- *Restrições:* limitante superior para o número de produtos e o *WIP* em cada unidade.

Note que *SP3.1* também envolve um *trade-off* entre o custo de adicionar capacidade e a redução da complexidade gerencial do sistema. Ele pode ser visto como um caso especial da classe *SP1*. As decisões envolvidas são: número de unidades de produção em que o sistema original é subdividido, alocação de produtos às unidades de produção, e escolha da capacidade em cada unidade. De fato, os problemas da classe *SP3* são casos especiais de ambas as classes *SP1* e *SP2*. Considera-se problemas de partição separadamente com a finalidade de enfatizar sua importância no projeto de sistemas de produção, entretanto, nesta tese eles são explorados apenas como perspectivas para pesquisa futura.

Estrutura

Esta tese está organizada conforme o esquema da figura 2.



(cap.2) (cap.3) (cap.4) (cap.5)

Figura 2 - Organização dos capítulos da tese

O capítulo 2 discute como um sistema de manufatura pode ser representado como uma rede de filas. A seção 2.1 define sistemas discretos, a seção 2.2 analisa como modelá-los por meio de redes de filas abertas (com ênfase em *job-shops*), e a seção 3.3 revisa a literatura relacionada.

O capítulo 3 analisa modelos de avaliação de desempenho para redes de filas abertas. A seção 3.1 apresenta métodos de decomposição para as redes de Jackson (classe única e múltiplas classes de produtos), e a seção 3.2, métodos de decomposição aproximados para as redes de Jackson generalizadas (classe única e múltiplas classes, com atenção especial para os casos com roteiros determinísticos). Uma das vantagens de se utilizar métodos de decomposição em redes de Jackson generalizadas é que eles produzem resultados razoavelmente precisos, demandam pouco esforço computacional, e requerem poucos dados de entrada (basicamente, apenas os dois primeiros momentos das distribuições de probabilidade dos processos de chegada e processamento de *jobs*). A seção 3.3 apresenta os resultados computacionais ao aplicar esses métodos num exemplo real de uma rede *job-shop* derivada de uma fábrica de semicondutores.

O capítulo 4 analisa modelos de otimização, baseados em programação matemática e nos modelos de avaliação de desempenho do capítulo 3, para os problemas em *SP1* e *SP2*. A seção 4.1 apresenta modelos para as redes de Jackson e a seção 4.2, para as redes de Jackson generalizadas. As variáveis de decisão podem ser a taxa média de processamento ou o número de máquinas na estação. Em alguns casos, estas variáveis estão limitadas a um conjunto de alternativas discretas para escolha de capacidade em cada estação. Diversos algoritmos da literatura são revisados e apresentados em detalhes. Em alguns casos, são propostos algoritmos alternativos mais precisos que os conhecidos da literatura, como por exemplo os algoritmos 4a e 5a, porém, demandando maiores esforços computacionais. Conforme mencionado anteriormente, esta revisão do estado da arte dos modelos de otimização em redes de filas abertas, e suas aplicações nos problemas em *SP1* e *SP2*, pretende ser uma contribuição útil para a literatura.

O capítulo 5 é o mais importante para o propósito principal desta tese. Inicialmente mostra-se como usar os métodos dos capítulos 3 e 4 para gerar as curvas de *trade-off*. As seções 5.1 e 5.2 apresentam, respectivamente, o problema de minimizar o *WIP* sem adicionar recursos no sistema, e o problema de minimizar os recursos sem aumentar o *WIP*. Na seção 5.3, as soluções destes problemas são usadas para gerar curvas de *trade-off*. A seção 5.4 apresenta outras curvas de *trade-off* para analisar os efeitos de redução de incertezas na rede, e de mudanças na taxa média de produção e no *mix* de produtos. As seções 5.5 e 5.6 estendem esta análise para o caso em que se tem um conjunto finito de alternativas discretas para mudanças de capacidade nas estações, e o caso em que não se pode aproximar cada estação como uma única máquina. Para ilustrar a apresentação dos tópicos do capítulo 5, utiliza-se o exemplo de uma aplicação real num sistema *job-shop* com 10 classes de produtos e 13 estações. Diversas curvas de *trade-off* são geradas para este exemplo, para mostrar o potencial da análise para as decisões de projeto e planejamento do sistema.

Finalmente, o capítulo 6 apresenta as conclusões desta tese, e algumas perspectivas para pesquisa futura.

2. O sistema de manufatura como uma rede de filas

Este capítulo mostra como os sistemas de manufatura podem ser representados por redes de filas abertas. Inicialmente define-se sistemas de manufatura discretos (seção 2.1), em seguida, discute-se como representá-los por meio de redes de filas (seção 2.2), e o capítulo termina com uma breve revisão dos exames da literatura (seção 2.3).

2.1 Sistemas discretos

Conforme Buzacott e Shanthikumar (1993), *sistemas de manufatura* consistem basicamente de máquinas e estações de trabalho onde operações são realizadas sobre partes, itens, submontagens e montagens, para criar produtos que serão distribuídos para clientes. Componentes adicionais desses sistemas são a movimentação de materiais e os dispositivos de estocagem. Eles permitem que itens se movam de estação para estação, que partes apropriadas estejam disponíveis para montagem, e que o trabalho seja mantido até poder entrar nas estações para processamento.

O foco desta tese são os sistemas *discretos* de manufatura, onde cada item processado é distinto. Tais sistemas aparecem principalmente nas indústrias mecânicas, elétricas e eletrônicas, produzindo, por exemplo, carros, refrigeradores, geradores elétricos, ou computadores. Sistemas que processam fluídos, como os encontrados por exemplo nas indústrias químicas e metalúrgicas, não são aqui considerados, embora algumas vezes esses fluídos sejam processados em lotes, e se cada lote for tomado como uma unidade de manufatura, então o sistema estará processando partes "discretas". Por simplicidade, um item, parte, submontagem ou montagem processada por uma máquina ou estação é chamado simplesmente de *job.*

Sistemas de manufatura discretos (ou, simplesmente, sistemas de manufatura) podem ser classificados em função do volume e variedade dos *jobs*. Tanto a variedade dos tipos de *jobs* manufaturados no sistema (i.e., o escopo), quanto o volume de cada tipo de *jobs* produzido (escala), interferem diretamente no projeto e operação do sistema. Embora um sistema de manufatura com capacidade para grande escopo e grande escala pareça ser ideal para enfrentar mudanças nas necessidades dos clientes, no projeto dos produtos, e nos processos de manufatura, tal sistema tenderia a ser de difícil controle e pouco econômico para ser operado. Duas formas tradicionais de organizar um sistema de manufatura são o *job-shop* e o *flow-shop*.

Tipos de sistemas

***Job-shop*:** Tem sido estimado que mais de 75% da manufatura ocorre em lotes de menos de 50 *jobs* (Askin e Standridge, 1993) e, nesses casos, as máquinas precisam ser capazes de desempenhar uma variedade de operações em diferentes tipos de *jobs.* A resposta tradicional

tem sido o *job-shop*, que consiste de uma variedade de tipos de máquinas agrupadas por similaridade de processo (*layout por processo*), algumas das quais podendo desempenhar operações em diferentes tipos de *jobs*, embora isto possa requerer algum tempo de preparação (*set-up*). A movimentação de materiais é tal que diferentes tipos de *jobs* podem visitar máquinas em diferentes seqüências. Departamentos são compostos de máquinas do mesmo tipo, localizadas juntas e desempenhando funções similares (figura 3), e porisso esse *layout* por processo também é chamado *funcional*. Além das máquinas e do sistema de movimentação de materiais, há também o espaço necessário para o estoque de *jobs* em processo. Este espaço em geral é próximo das máquinas, mas também pode estar longe delas ou nos pátios da fábrica. A operação do *job-shop* é quase sempre baseada na premissa de que esse espaço pode ser sempre encontrado em algum lugar, para que as máquinas nunca sejam bloqueadas.

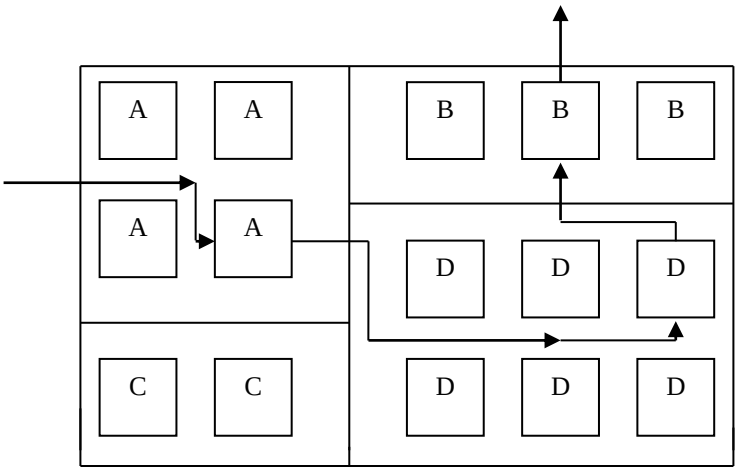


Figura 3 - Sistema *job-shop: layout* por processo (funcional)

O sistema *job-shop* tem capacidade de escopo, mas os problemas associados ao controle do movimento dos *jobs* e ao tempo de preparação das máquinas limitam sua capacidade de escala, isto é, de produzir eficientemente e economicamente grandes volumes. O sistema é caracterizado por longos *leadtimes* de produtos e altos níveis de *WIP*.

***Flow-shop*:** Por outro lado, no sistema *flow-shop* (*linha de fluxo* ou *linha de produção*) tem-se o *layout por produto*. Todos os *jobs* visitam máquinas e estações na mesma seqüência, simplificando assim a movimentação de materiais, mas limitando o escopo do sistema de manufatura. Isso facilita o controle do fluxo de trabalho e a instrução de máquinas e operadores nas suas tarefas, permitindo que grandes volumes possam ser produzidos eficientemente e economicamente. Os sistemas *flow-shops* são assim chamados porque as máquinas são alinhadas de maneira que os fluxos de produto vão da primeira máquina até a segunda, da segunda até a terceira, e assim por diante, até a última máquina da linha (figura 4). Note que os *flow-shops*, ao contrário dos *job-shops*, têm escala mas não têm escopo. Eles são caracterizados por baixos *leadtimes* de produtos e baixos níveis de *WIP*.

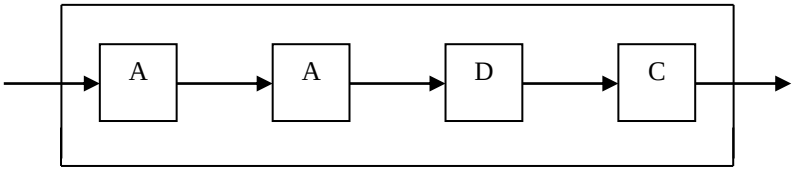


Figura 4 – Sistema *flow-shop*: *layout* por produto

Manufatura celular: Outro tipo de sistema é a *manufatura celular* (aplicação de *tecnologia de grupo*), que pode ser usado para converter um sistema de *layout* por processo num pseudo ambiente de *layout* por produto. *Jobs* similares são agrupados em quantidade suficiente para justificar sua produção em uma instalação exclusiva, a célula de manufatura, organizada para produzir apenas esse conjunto de *jobs* (figura 5). Pode não ser possível arranjar máquinas numa célula de tal modo que todos os *jobs* sigam a mesma seqüência de máquinas visitadas. De qualquer forma, o uso de células para processar um conjunto específico de *jobs* simplifica a programação e o controle, e reduz substancialmente o tempo de preparação, a movimentação de materiais, o *leadtime* de produção e o nível de *WIP*.

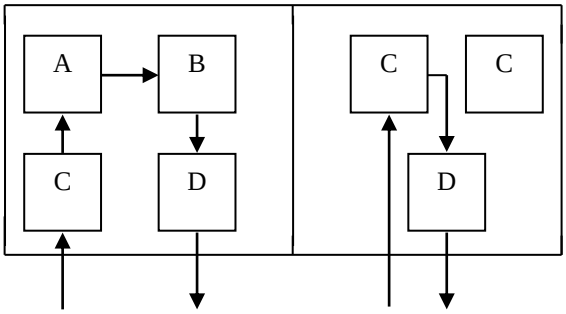


Figura 5 - Manufatura celular (tecnologia de grupo)

A figura 6 relaciona volume e variedade (i.e., a combinação escala-escopo) dos *layout*s por processo, produto e celular. Algumas características desses *layout*s são comparadas na tabela 1.

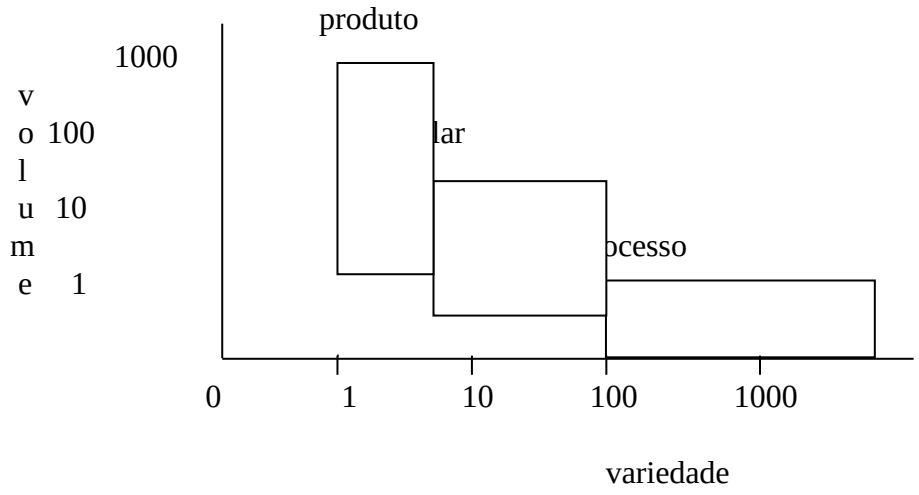


Figura 6 - Combinação volume (partes por hora) e variedade (número de tipos de partes) (Askin e Standridge, 1993, p.11)

Tabela 1 - Comparação das características dos *layouts* de processo, produto e celular

Característica	Processo	Produto	Celular
leadtime de produto	alto	baixo	baixo
WIP	alto	baixo	baixo

flexibilidade de produto	alta	baixa	média
flexibilidade de demanda	alta	média	média
utilização de máquinas	baixa	alta	média
custo unitário de produção	alto	baixo	médio

Outros tipos de sistemas de manufatura são as *linhas de transferência*, os *sistemas flexíveis de manufatura* e os *flow-shops flexíveis*.

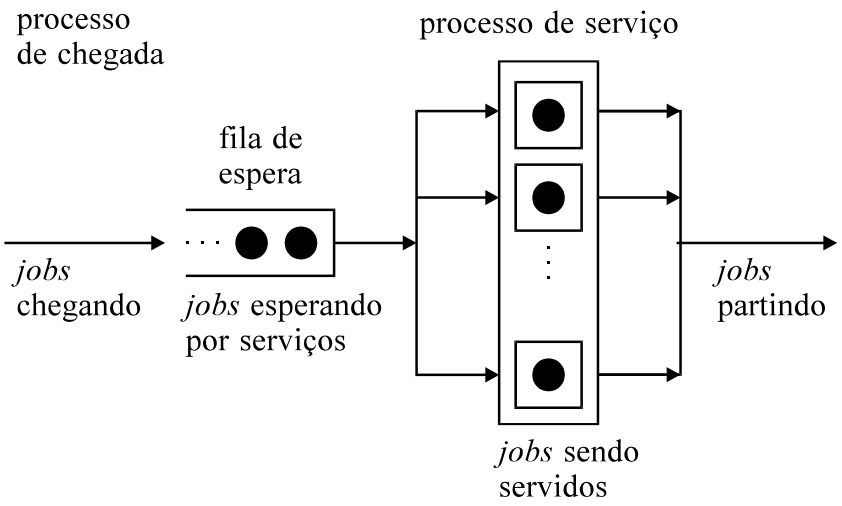
Linhas de transferência: Muitos sistemas *flow-shop* produzem um único tipo de produto, e com o aumento do volume, passa a ser atraente automatizar máquinas individuais e substituir operadores por dispositivos e máquinas automáticos. Esses sistemas são chamados *linhas de transferência automáticas* se, não apenas algumas máquinas e a movimentação de materiais entre elas, mas todas as máquinas estiverem ligadas de maneira a começarem suas tarefas simultaneamente, tal que o movimento de material seja sincronizado. Neste caso o número de *jobs* em processo pode ser mantido pequeno, enquanto a produtividade resulta extremamente alta. Assim, uma linha de transferência automática é caracterizada por escala muito elevada (maior do que o *flow-shop*), mas escopo muito reduzido (menor do que o *flow-shop*).

Sistemas flexíveis de manufatura: Linhas de transferência são essencialmente linhas de produção automáticas e, portanto, todos os *jobs* visitam as máquinas na mesma seqüência (padrão de fluxo *flow-shop*). *Sistemas flexíveis de manufatura* (*flexible manufacturing systems - FMS*) podem ser vistos como células de manufatura providas com alto grau de automação flexível, sendo que, como nas células, os *jobs* também podem visitar as máquinas em diferentes seqüências (padrão de fluxo *job-shop*). Máquinas de controle numérico, computadores de controle, dispositivos eletrônicos para armazenagem e processamento de informação, redes locais, robôs industriais, veículos controlados por computador (*auto-guided vehicles - AGV*) e determinados *software* são os principais elementos usados para se obter a automação flexível. Em geral, os *FMS* possuem menor escopo do que as células de manufatura.

Sistemas *flow-shop* flexíveis: Da mesma maneira que, ao introduzir a automação flexível nas células de manufatura obtém-se os *FMS*, ao introduzir automação flexível nos sistemas *flow-shop* obtém-se os chamados sistemas *flow-shop flexíveis*. Neste caso é essencial que haja espaço para estocagem entre as máquinas, já que é comum *jobs* saltarem uma ou mais máquinas da linha de produção e, com isso, perde-se a sincronização do movimento cadenciado. Os *flow-shops* flexíveis têm maior escopo do que o *flow-shops* tradicionais, no sentido de que eles podem produzir um maior número de variações de um produto.

Para maiores detalhes destes e outros tipos de sistemas de manufatura, veja, por exemplo, Krajewski e Ritzman (1990), Fernandes (1991), Chase e Aquilano (1992), Askin e Standridge (1993) e Buzacott e Shanthikumar (1993).

Nesta tese há um interesse particular nos sistemas *job-shops*, caracterizados por longos *leadtimes* de produtos e altos níveis de *WIP*. Para o propósito de avaliação de desempenho, *job-shops* podem ser adequadamente descritos como sistemas dinâmicos de eventos discretos, isto é, sistemas caracterizados por uma seqüência de eventos discretos ao longo do tempo (Leung e Suri, 1990). Exemplos de tais eventos são: chegada de um *job* na fila de espera de uma máquina, término do processamento de um *job* numa máquina, falha de uma certa máquina. Note que as medidas de desempenho desses sistemas dependem do instante de ocorrência dos eventos discretos.



A seguir, discute-se como *job-shops* podem ser representados por modelos de rede de filas. Redes de filas também têm sido empregadas para modelar outros sistemas de manufatura, como *flow-shops*, linhas de transferência, *FMS*, e sistemas celulares, mas isso está fora do escopo deste trabalho. Para exames destes modelos, o leitor pode consultar a literatura discutida na seção 2.3.

Figura 7 - Um sistema de fila de único estágio

2.2 Redes de filas

O estudo de redes de filas começou basicamente com o trabalho de Erlang (1917) em telefonia. Desde então, aplicações têm aparecido em diversas áreas; por exemplo, comunicação, computação, transporte, produção, manutenção, biologia (redes neurais), saúde (modelos comportamentais), química e materiais (polimerização), entre outras; veja Disney e Konig (1985). Hsu *et al.* (1993) e Suri *et al.* (1993) forneceram uma descrição abrangente do uso de redes de filas para representar sistemas de manufatura. Cada nó (estação) contém os seguintes elementos:
(i) *processo de chegada*
(ii) *processo de serviço*
(iii) *fila de espera.*

A figura 7 ilustra essa representação. Convém salientar que os modelos de filas são motivados por casos em que os processos de chegada ou serviço, ou ambos, são probabilísticos, resultando possivelmente numa fila de espera de *jobs* (indicada na figura, na frente da estação).

Processo de chegada

O *processo de chegada* na estação é descrito pelo intervalo de tempo entre chegadas de *jobs*, que pode ser determinístico (*D*) ou probabilístico. Se o processo de chegada for probabilístico, ele pode ser dependente dos outros intervalos de tempo entre chegadas e/ou do processo de serviço, ou consistir de intervalos entre chegadas *independentes e identicamente distribuídos* (*iid*).

O primeiro caso é chamado de processo de chegada *G* (genérico) e o segundo, processo de chegada *GI* (genérico independente) ou *processo de renovação*. Um exemplo de um processo de chegada *G* dependente do processo de serviço ocorre quando um *job*, ao chegar, desiste de entrar na fila caso ela seja muito longa, ou quando um *job* é removido da fila após esperar por um longo período de tempo. Um exemplo particular de um processo *GI* é aquele em que o número de chegadas ao longo do tempo ocorre conforme um processo de Poisson, isto é, os intervalos de tempo entre chegadas são independentes e exponencialmente distribuídos (processo Markoviano, ou *sem memória*, *M*).

Pode-se ter todos os *jobs* pertencendo a uma única classe ou família, ou agrupados em múltiplas classes diferentes. Todos os *jobs* de uma mesma classe são supostos idênticos estatisticamente. A chegada de *jobs* na estação pode ser individual ou em lotes. Em certos casos, o lote pode ser considerado como um único *job*.

A agregação de dados é comumente necessária, dado que um ambiente típico de manufatura pode envolver centenas de máquinas e milhares de produtos. Em geral, produtos com roteiros e necessidades de processamento similares são agrupados em classes, e máquinas com características de processamento equivalentes são agrupadas em estações. Há várias razões para a agregação de dados, talvez a mais óbvia seja reduzir a complexidade computacional dos modelos, que depende do número de classes e estações. Mas há também a preocupação de reduzir a complexidade gerencial: um gerente diante de milhares de variáveis provavelmente terá dificuldades para identificar os processos dominantes que governam o comportamento do sistema.

Processo de serviço

O *processo de serviço* na estação é descrito pelo tempo de processamento de *jobs*, que pode ser determinístico (*D*) ou probabilístico. Se o processo de serviço for probabilístico, ele pode depender de outros tempos de processamento e/ou do processo de chegada, ou consistir de tempos de processamento *iid*. Alguns autores têm chamado o primeiro caso de processo de serviço *G* e o segundo caso de processo de serviço *GI* (Disney e Konig, 1985). Aqui, ambos os casos são referidos apenas como processo de serviço *G*, uma vez que é mais usual na literatura.

Um exemplo de processo de serviço *G* dependente do processo de chegada é aquele em que o tempo de processamento varia de acordo com o número de *jobs* na fila. Similarmente aos intervalos de tempo entre chegadas, os tempos de processamento podem ser variáveis aleatórias independentes e exponencialmente distribuídas (processo de serviço *M*). Estações podem ter uma única máquina (servidor), ou várias máquinas. Uma máquina pode executar uma operação em um *job* individual ou em lotes de *jobs*, e eventualmente pode falhar. Em certos casos, o lote pode ser considerado como um único *job*. Cada máquina pode representar um conjunto de recursos, como: várias máquinas, operadores, ferramentas, etc. A notação *GI/G/m* é usada para indicar um sistema de fila de um estágio, no qual o processo de chegada é um processo de renovação (*GI*), os tempos de processamento são variáveis aleatórias *iid* (*G*), e o número de servidores no sistema é *m*.

Fila de espera

A *fila de espera* na estação pode ter capacidade limitada ou ilimitada para o máximo número de *jobs* na fila, geralmente determinada pelo espaço físico disponível. Uma vez alcançada essa capacidade, a entrada de novos *jobs* na fila é bloqueada. A fila tem uma disciplina ou regra para ordenar os *jobs* esperando por processamento. Exemplos de disciplinas são: primeiro a chegar,

primeiro a ser servido (*first come, first served - FCFS*), fila com prioridades, primeiro o de menor tempo de processamento (*shortest processing time first - SPT*), e primeiro o de maior tempo de processamento (*largest processing time first*). No caso de fila com prioridades, pode-se ter o caso preemptivo e o não-preemptivo. No caso preemptivo, o *job* com maior prioridade entra em serviço assim que chegar na fila, mesmo que um *job* com menor prioridade já esteja em serviço. No caso não-preemptivo, um *job* já iniciado não pode ser interrompido enquanto não for completado.

O conjunto de nós, arcos e *jobs* compõe a rede de filas com as seguintes características:
(i) número de estações (nós)
(ii) seqüência de operações (roteiros)
(iii) tipo de rede de filas: aberta, fechada e mista.

O número de nós na rede, maior ou igual a 1, corresponde ao número de estações. Cada estação pode ser visitada pelo mesmo *job* mais de uma vez, e em cada visita pode realizar uma operação diferente. A seqüência de visitas (ou roteiro) ao longo das estações, determinística ou probabilística, pode ser seqüencial, seqüencial com *realimentação* (*feedback*), de montagem, arborescente, acíclica, e cíclica. Arcos com realimentação podem ser utilizados para representar, por exemplo, retrabalho, e roteiros probabilísticos podem modelar, por exemplo, falha de máquinas.

Uma rede de filas pode ser aberta, fechada ou mista. Numa redes de filas *aberta* (*open queueing networks - OQN*), os *jobs* assim que chegam entram na rede, recebem processamento em um ou mais nós, e eventualmente saem da rede. A figura 8 ilustra uma rede aberta onde *jobs* entram na rede pela estação do lado esquerdo da figura, percorrem roteiros seqüenciais com realimentação ao longo das estações (conforme indicado pelos arcos da figura), eventualmente aguardam em fila defronte às estações visitadas, e deixam a rede pela estação do lado direito da figura. O número de *jobs* circulando na rede é uma variável aleatória.

Numa rede de filas *fechada* (*closed queueing network - CQN*), ao contrário, não há chegadas ou partidas externas, o número de *jobs* circulando na rede é mantido fixo, enquanto a taxa de partidas internas de cada nó é uma variável aleatória. Entretanto, pode-se artificialmente representar chegadas e partidas externas de *jobs* numa *CQN*, ao se definir uma estação de carga e descarga instantânea. Para isso, para cada partida externa de um *job* desta estação, há uma chegada externa de outro *job* nesta estação e, portanto, o número de *jobs* na *CQN* é estritamente controlado para que se mantenha constante. Note que, desta maneira, um *FMS* pode ser representado por meio de uma *CQN*. Se a rede tiver múltiplas classes de *jobs*, pode-se redefinir sub-redes abertas para algumas classes e sub-redes fechadas para outras. Neste caso, a rede de filas é chamada *mista*.

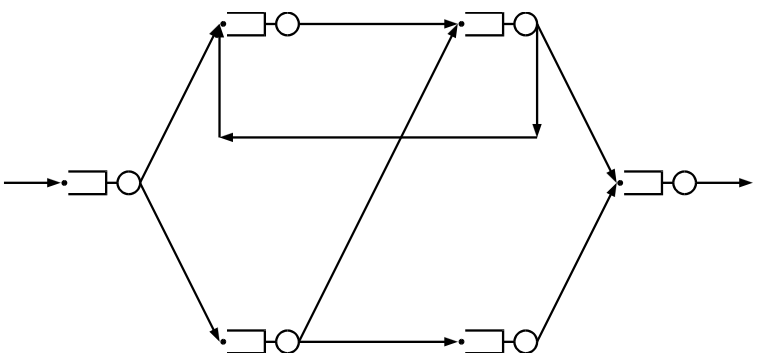


Figura 8 - Rede de filas aberta (*OQN*)

Os modelos de redes de filas analisados nesta tese referem-se a *OQN* (modelos de *OQN*, além de serem analiticamente mais tratáveis do que modelos de *CQN*, podem ser utilizados para aproximá-los; veja Whitt (1984) e Calabrese (1992)). Os modelos pressupõem que o sistema atinja o *estado de equilíbrio* ou *regime permanente* (*steady state*), isto é, baseiam-se no comportamento do padrão típico do sistema num longo período de tempo. Em outras palavras, os resultados dos modelos descrevem o comportamento médio de longo prazo de sistemas "estáticos", ou seja, sistemas em que seja razoável admitir que parâmetros como a taxa *média* de chegada de *jobs* ou o tempo *médio* de processamento de *jobs* não variam ao longo do horizonte de análise (processos de chegada e serviço estacionários). Também admite-se que os sistemas sejam *estáveis* no sentido de que suas capacidades, medidas pela taxa máxima de produção, excedam a demanda média, de maneira que os níveis médios de *WIP* ou os tempos médios de espera sejam finitos. A seguir são relacionadas as principais hipóteses sobre os *jobs* e as estações, admitidas nos modelos:

Hipóteses sobre *jobs:* Cada *job*, ao chegar no sistema, segue imediatamente para a estação da sua primeira operação. As características de um *job* são admitidas estatisticamente independentes das características dos demais. Os processos de chegada externa de *jobs* são considerados probabilísticos com intervalos de tempo entre chegadas *iid* nas estações. Os *jobs* pertencem (ou podem ser agregados) a uma classe única ou a múltiplas classes, e chegam individualmente nas estações. Eles podem ser criados ou combinados nas estações, por exemplo, uma única partida de um *job* de uma estação pode resultar em várias chegadas de *jobs* na próxima estação a ser visitada (criação), ou várias partidas consecutivas podem resultar numa única chegada na próxima estação (combinação). Em geral cada *job* tem uma seqüência determinada de estações que deverá visitar, mas roteiros probabilísticos também são permitidos. Os *jobs* eventualmente têm que esperar em fila para receber serviço nas estações e, para isso, estoques em processo são permitidos, sem limitação de espaço. Uma vez aceito, o *job* é processado até terminar sua operação, isto é, não são permitidos cancelamentos ou interrupções.

Hipóteses sobre estações: Cada estação pode ter uma ou várias máquinas idênticas, que processam, cada uma, apenas um *job* por vez. A disciplina de atendimento em geral é *FCFS*. Cada máquina opera independente das demais e assim, é capaz de produzir na sua capacidade máxima. Os processos de serviço são considerados probabilísticos com tempos de processamento *iid* em cada estação (em alguns casos esses processos podem ser dependentes do estado do sistema, mas isso é evidenciado no texto quando for o caso). Admite-se que as máquinas estejam continuamente disponíveis para processar *jobs*, e que não haja interrupções devido a falhas, manutenção ou outras causas. Em alguns casos essas interrupções podem ser modeladas definindo uma classe artificial de *jobs*. Cada estação possui espaço suficiente para a fila de espera de *jobs* aguardando processamento, também possui espaço suficiente para estocar os *jobs* processados, até que eles se movam para a próxima estação ou para fora do sistema.

Os modelos aqui estudados podem ser estendidos para tratar processamento em lotes e horas extras, falha de máquinas, limitação da capacidade da fila de espera, e outras disciplinas além de *FCFS*, conforme observado em Segal e Whitt (1989), Bitran e Tirupati (1989c, 1991), Kouvelis e Tirupati (1991), e Buzacott e Shanthikumar (1993). Aqui, processamento em lotes significa que *jobs* são reunidos (combinados) para formar um lote antes de serem processados, ou que *jobs* são separados (criados) após o processamento, para retornarem à forma original. Outros modelos para os vários casos não considerados nesta tese podem ser encontrados nos exames da literatura discutidos a seguir, e nas referências lá citadas.

2.3 Exames da literatura

Disney e Konig (1985) apresentaram um extenso exame da teoria de redes de filas, desde os trabalhos pioneiros de Jackson (1957, 1963) e as extensões de Kelly (1975, 1979) e de Baskett, Chandy, Muntz e Palacios (1975), incluindo uma bibliografia com mais de 300 referências. Outros exames aparecem em Lemoine (1977), Koenigsberg (1982), Askin e Standridge (1993), Gershwin (1994) e Papadopoulos e Heavey (1996). Estes dois últimos analisaram particularmente modelos para linhas de produção e transferência. Buzacott e Yao (1986) discutiram os desenvolvimentos em *CQN* antes de 1986 (com ênfase na aplicação em *FMS*), e classificaram as abordagens baseando-se nos diferentes grupos de pesquisa. Suri *et al.* (1993) examinaram modelos de avaliação de desempenho para diversos sistemas de manufatura, como sistemas com um único estágio (fila simples), linhas de produção (filas em série), linhas de montagem (filas em árvore), *job-shops* (*OQN*) e *FMS* (*CQN*). Suri *et al.* comentaram o uso de teoria de redes de filas em tópicos como *MRP II* (*material resource planning*), *just-in-time*, *Kanban*, e sugeriram abordagens alternativas, como análise de sensibilidade em simulação, modelos baseados em redes de Petri, e redes de filas hierárquicas. Para referências mais recentes sobre aplicação de modelos de rede de filas em sistemas *Kanban*, veja Mascolo *et al.* (1996).

Buzacott e Shanthikumar (1992, 1993), Hsu *et al.* (1993), Kouvelis e Tirupati (1991), Bitran e Dasu (1992), e Bitran e Morabito (1996) examinaram ambos os modelos de avaliação de desempenho e os modelos de otimização de redes de filas. Buzacott e Shanthikumar (1993) realizaram uma extensa análise orientada para o projeto de diferentes sistemas de manufatura, tais como linhas de produção, linhas de transferência automatizadas, *job shops*, *FMS* e sistemas multicelulares. Eles analisaram problemas de projeto ótimo e em particular, consideraram alguns modelos de otimização em *job shops* que não serão aqui cobertos, tais como: alocação ótima de *jobs* nas estações, e análise dos efeitos da diversidade de roteiros e tempos de processamento de *jobs*. Hsu *et al.* (1993) examinaram modelos de otimização para *FMS* baseados em *CQN*; eles também sugeriram o uso de técnicas alternativas, como álgebra max-plus, conjuntos nebulosos e sistemas especialistas.

Kouvelis e Tirupati (1991) revisaram modelos de *OQN* e *CQN* no contexto de projeto e planejamento, com foco nos modelos de instalações de manufatura, avaliação de desempenho e otimização. Eles observaram que um número de suposições e aproximações é geralmente necessário para tornar os problemas práticos tratáveis, entretanto, o efeito acumulativo na qualidade das soluções não tem sido sistematicamente analisado, e permanece como um tópico para pesquisa futura. Bitran e Dasu (1992) discutiram problemas estratégicos, táticos e operacionais de sistemas de manufatura, com uma atenção especial para os modelos de projeto e planejamento de *job-shops*. Esta orientação foi atualizada e estendida em Bitran e Morabito (1996), com um enfoque mais quantitativo, onde foram apresentados modelos com múltiplas classes de produtos enfatizando a importância da interferência entre as classes e das aproximações de tráfego leve. Também foram incluídos desenvolvimentos mais recentes, como a partição de produtos. A análise de modelos e algoritmos dos capítulos 3 e 4 baseia-se em Bitran e Morabito (1995d, 1996).

Muitas abordagens para os modelos de otimização utilizam os *métodos de decomposição* (veja capítulo 3) para avaliar medidas de desempenho de uma *OQN*. Abordagens alternativas também têm sido exploradas (método *QNET*), baseadas em modelos Brownianos e nos teoremas do limite de tráfego pesado. No capítulo 4 apresenta-se um exemplo destas abordagens (Wein,

1990a), sem explorar desenvolvimentos adicionais neste tópico. Para um exame do estado da arte dos modelos Brownianos para *OQN* com múltiplas classes, veja Harrison e Nguyen (1993). Referências mais recentes em modelos Brownianos são encontradas em Harrison e Pich (1996) e Dai *et al.* (1997).

3. Modelos de avaliação de desempenho

Este capítulo analisa os chamados *modelos de avaliação de desempenho* do sistema, que auxiliam a estimar medidas como: média e variância do número de *jobs* no sistema e em cada estação, média e variância do *leadtime* de produtos no sistema e em cada estação, taxa média de produção do sistema, utilização média de equipamentos em cada estação. Os modelos baseiam-se na representação do sistema por meio de uma rede de filas aberta, conforme o capítulo 2. As medidas avaliadas por esses modelos são utilizadas no próximo capítulo, nos *modelos de otimização* de *OQN*.

Modelos descritivos e prescritivos: Conforme mencionado no capítulo 1, a diferença básica entre um modelo de avaliação de desempenho e um modelo de otimização é o fato de que o primeiro é *descritivo*, isto é, através das medidas de desempenho ele proporciona ao usuário importantes *insights* sobre a operação do sistema numa dada configuração, enquanto o segundo é *prescritivo*, isto é, ele determina a configuração *ótima* do sistema, que otimiza certo critério e satisfaz certas restrições impostos pelo usuário (figura 9). Alguns autores denominam os modelos de avaliação de desempenho como *avaliativos*, e os modelos de otimização como *gerativos* (Papadopoulos e Heavey, 1996).

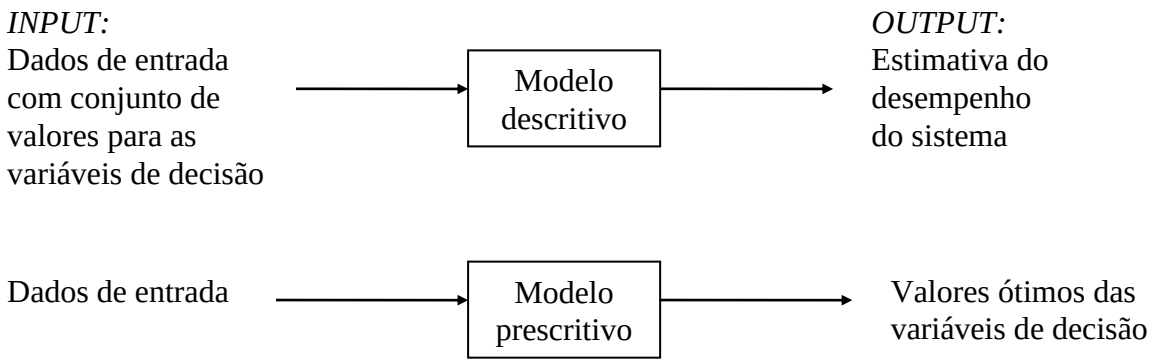


Figura 9 - Modelos descritivos e prescritivos

Métodos analíticos e experimentais: Modelos de avaliação de desempenho têm sido aplicados utilizando basicamente *métodos analíticos* (exatos e aproximados), e *simulação e técnicas relacionadas*. Em alguns casos estes métodos são empregados de maneira integrada, primeiro, aplicando um método analítico para reduzir o conjunto de possíveis configurações do sistema e identificar os parâmetros que mais afetam o seu desempenho e, depois, utilizando simulação detalhada para selecionar a melhor configuração entre as candidatas (Leung e Suri, 1990). Um exemplo de uma abordagem integrada aparece em Deuermeyer *et al.* (1993). Métodos analíticos

exatos existem para as chamadas *redes de Jackson*, discutidas a seguir (seção 3.1), onde os processos de chegada e serviço são supostos de Poisson. Entretanto, em muitos sistemas de manufatura esses processos são geralmente menos variáveis do que o processo de Poisson, o que limita a aplicação das redes de Jackson.

Na falta de métodos analíticos exatos para *OQN* genéricas, pode-se usar métodos experimentais, como *simulação e técnicas relacionadas* (Law e Haider, 1989, Law e McComas, 1989, Schriber, 1991). Estas abordagens são as mais comumente utilizadas para avaliação de desempenho e permitem considerações bem próximas das situações reais, entretanto, podem requerer grandes quantidades de dados, em comparação aos métodos analíticos, e longos tempos para desenvolvimento e verificação dos modelos de computador. Outra desvantagem são os requisitos computacionais que, em geral, limitam o número de alternativas a serem consideradas, principalmente se uma variedade de mudanças nos valores dos parâmetros precisa ser explorada (Leung e Suri, 1990). Técnicas como análise de perturbação indicam possíveis caminhos para reduzir esses esforços computacionais. Essas técnicas, baseadas na análise de sensibilidade de parâmetros ao se observar uma única replicação simulada, estão além dos objetivos desta tese. Para detalhes, veja Ho e Cao (1983), Ho (1987), Suri (1989) e Cao (1994).

As limitações impostas pelos métodos analíticos exatos e pela simulação levaram autores a desenvolver métodos analíticos *aproximados*, que podem ser classificados em cinco categorias:
(i) *aproximações de difusão*
(ii) *análise de valor médio*
(iii) *análise operacional*
(iv) *aproximação exponencial*
(v) *métodos de decomposição.*

As *aproximações de difusão* são motivadas pelos teoremas do limite de tráfego pesado, e têm gerado novos métodos de solução para as *OQN* (Reiman, 1990, Harrison e Nguyen, 1990, Wein, 1990b, Harrison e Pich, 1996, Dai *et al.*, 1997). Elas têm sido aplicadas principalmente para problemas de programação e controle operacional. *Análise de valor médio* (Seidmann *et al.*, 1987, Suri *et al.*, 1993, Tetzlaff, 1996), *análise operacional* (Denning e Buzen, 1978, Dallery e David, 1986), e *aproximação exponencial* (Yao e Buzacott, 1986, Hsu *et al.*, 1993) têm sido utilizadas basicamente para analisar *CQN*. Os métodos analíticos mais freqüentemente utilizados para analisar modelos de *OQN* em *job-shops* têm sido os *métodos de decomposição*.

Nesta tese utiliza-se apenas os métodos de decomposição. A seção 3.1 examina brevemente métodos exatos para redes de Jackson com filas *M/M/m* de classe única (seção 3.1.1) e múltiplas classes (seção 3.1.2). A seção 3.2 apresenta métodos de decomposição para redes de Jackson generalizadas com filas *GI/G/1* de classe única (seção 3.2.1), filas *GI/G/m* de classe única (seção 3.2.2) e filas *GI/G/m* de múltiplas classes (seção 3.2.3).

No texto que segue, os índices *i* e *j* em geral indicam a estação, o índice *k* indica a classe de produtos, e o índice *l* indica a operação da classe numa estação. As notações $E(x)$, $V(x)$ e cx denotam respectivamente a média, a variância, e o coeficiente quadrático de variação (*squared coefficient of variation - scv*) da variável aleatória *x*. O *scv* de *x* é um parâmetro adimensional da variabilidade de *x*, definido por: $cx = V(x)/E(x)^2$. Vetores e matrizes aparecem no texto em negrito.

3.1 Redes de Jackson (Métodos exatos de decomposição)

Considere uma rede de filas aberta com *n* estações, cada uma com uma ou várias máquinas paralelas e idênticas e capacidade de espera infinita. As estações 1, 2, ..., *n* são estações internas e é conveniente definir a estação 0 como uma estação externa do sistema. Sejam $N(t)$ o número de *jobs* que chegam na rede da estação 0 durante o intervalo de tempo $(0,t]$, e $s_{j,u}$ o tempo de processamento do *u*-ésimo *job* servido na estação interna *j* (sem incluir o tempo em fila). Observe que $\{N(t), t \geq 0\}$ e $\{s_{j,u}, u = 1,2,\ldots\}$ são processos estocásticos. Admitindo-se que ambos sejam *iid* e que o sistema atinja equilíbrio, então:

$$\lambda_0 = \lim_{t\to\infty} \frac{N(t)}{t} \tag{3.1}$$

$$E(s_j) = \lim_{u\to\infty} \frac{\sum_{v=1}^{u} s_{j,v}}{u} \tag{3.2}$$

Note que λ_0 é a taxa média de chegadas (ou partidas) externas da rede e, portanto, $E(a_0) = 1/\lambda_0$ é o valor esperado do intervalo de tempo entre chegadas externas, a_0 Similarmente, $E(s_j)$ é o valor esperado do tempo de serviço (carga de trabalho) de um *job* na estação *j* e $\mu_j = 1/E(s_j)$ é a taxa média de serviço para cada máquina na estação *j*. Na prática, pode-se obter λ_0 e μ_j aplicando as expressões acima para *t* e *u* suficientemente grandes.

Após serem processados na estação *j*, *jobs* deixam a estação com intervalo de tempo entre partidas d_j e vão para a estação *i*, *i* = 0, ..., *n*, com probabilidade de transição r_{ji} definida por uma cadeia de Markov (isso significa que a próxima estação no roteiro do *job* depende apenas da estação atual, e não da história de processamento já passada pelo *job*). Assume-se que qualquer seqüência de intervalos de tempo entre chegadas externas, tempos de serviço e decisões de roteamento sejam mutuamente independentes, com *jobs* sendo processados em cada estação conforme a disciplina *FCFS*.

Se os intervalos de tempo entre chegadas externas a_0 e os tempos de serviço s_j forem exponencialmente distribuídos (processos de Poisson), então a *OQN* acima é referida como uma *rede de Jackson*, caso contrário, como uma *rede de Jackson generalizada* (ou, simplesmente, uma *OQN* genérica). Redes de Jackson possuem soluções exatas na forma de produto (elegantes do ponto de vista matemático) demonstradas por Jackson (1957, 1963), como é visto a seguir. O resultado principal é que a distribuição de equilíbrio do número de *jobs* na rede, se existir, pode ser definida em forma de produto, e cada estação pode ser analisada individualmente como um sistema de fila *M/M/m* estocasticamente independente.

3.1.1 Rede de filas *M/M/m* de classe única

Suponha inicialmente que todos os *jobs* pertençam a uma mesma classe. Considere a seguinte notação para os dados de entrada:

- n número de estações internas na rede
- λ_0 taxa média de chegada externa na rede ($\lambda_0 = 1/E(a_0)$).

Para cada estação *j* = 1, 2, ..., *n*:

- m_j número de máquinas na estação *j*, $m_j \geq 1$,
- μ_j taxa média de serviço para cada máquina na estação *j* ($\mu_j = 1/E(s_j)$).

Para cada par (i,j), *i* = 0, 1, ..., *n*, *j* = 0, 1, ..., *n*:

r_{ij} probabilidade de um *job*, após ser atendido na estação *i*, seguir para a estação *j*.

Ao todo são $(n+1)^2 + 2n$ dados de entrada. Os fluxos de *jobs* entre as estações são descritos pela cadeia de Markov com matriz de transição $\boldsymbol{R} = \{r_{ij},\ 0 \le r_{ij} \le 1,\ i = 0, ..., n,\ j = 0, ..., n\}$, onde $\sum_{j=0}^{n} r_{ij} = 1$, $i = 0, ..., n$, e $r_{00} = 0$ por definição. A hipótese de roteiro Markoviano implica que, após completar a operação na estação *i*, a próxima estação *j* a ser visitada pelo *job* deve ser uma função apenas da estação *i*, e independente das demais estações visitadas anteriormente. Cada estação interna *j* pode ser descrita por 3 parâmetros $\{m_j, \lambda_{0j}, \mu_j\}$, onde:

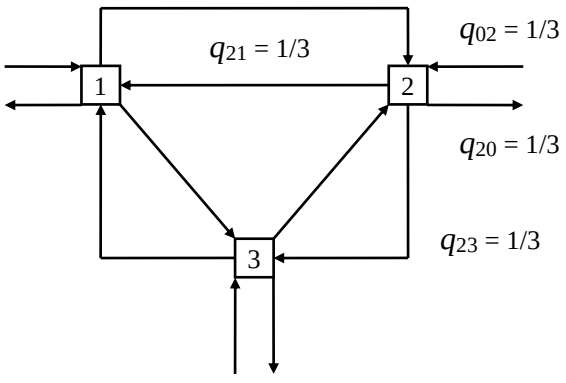
$$\lambda_{0j} = r_{0j}\lambda_0 \qquad (3.3)$$

é a taxa média de chegada externa na estação *j* (lembre-se que $\sum_{j=0}^{n} r_{0j} = 1$), e portanto, $E(a_{0j}) = 1/\lambda_{0j}$ é o valor esperado do intervalo de tempo entre chegadas externas na estação *j*, a_{0j}. Na prática, às vezes é mais conveniente obter λ_{0j} por meio de $\lambda_{0j} = \lim_{t\to\infty} N_j(t)/t$, ao invés de (3.3), onde $N_j(t)$ é o número de *jobs* que chegam da estação 0 na estação *j* durante $(0,t]$. Note que a combinação de *n* processos de Poisson independentes $N_j(t)$ com taxa λ_{0j} resulta num processo de Poisson com taxa λ_0. Inversamente, também pode ser mostrado que se um processo de Poisson $N(t)$ com taxa λ_0 for decomposto em *n* processos $N_j(t)$ independentes, tal que cada chegada tem probabilidade r_{0j} de pertencer a $N_j(t)$, então cada processo $N_j(t)$ também é Poisson com taxa λ_{0j} conforme (3.3) (veja p.e. Ross (1993) e Kleinrock (1975)).

Por simplicidade, define-se $\boldsymbol{Q} = \{q_{ij},\ i = 1, ..., n,\ j = 1, ..., n\}$ e $q_{i0} = 1 - \sum_{j=1}^{n} q_{ij}$. $\boldsymbol{Q}$ é a matriz $\boldsymbol{R}$ sem a linha 0 (probabilidade de um *job* entrar no sistema pela estação *j*) e sem a coluna 0 (probabilidade de um *job* deixar o sistema a partir da estação *i*). Similarmente, q_{i0}, $i = 1, ..., n$, é a coluna 0 da matriz $\boldsymbol{R}$ sem o elemento da linha 0. Desta maneira, os dados de entrada são reduzidos para $n^2 + 3n$, isto é, a matriz $\boldsymbol{Q}$ e os parâmetros $\{m_j, \lambda_{0j}, \mu_j\}$. A taxa média de chegada na rede, λ_0, é obtida simplesmente por $\sum_{j=1}^{n} \lambda_{0j}$.

Pode-se descrever roteiros determinísticos (Markovianos) por meio de $\boldsymbol{Q}$, uma vez que são casos particulares de roteiros probabilísticos Markovianos. Se $q_{jj} > 0$, então ocorre uma *realimentação imediata* na estação *j*. Para ilustrar, considere um *job-shop* simétrico (Shanthikumar e Buzacott, 1981) onde um *job*, ao partir de uma estação, tem a mesma chance de seguir para qualquer outra estação, incluindo a estação 0, ou seja, $\boldsymbol{Q} = \{q_{ij} = 1/n$ e $q_{ii} = 0,\ i \ne j,\ i = 1, ..., n,\ j = 1, ..., n\}$ e $q_{i0} = 1/n$, $i = 1, ..., n$. Considere agora um *flow-shop* uniforme onde todos os *jobs* seguem as mesmas estações em série na ordem 1, 2, ..., *n*, ou seja, $\boldsymbol{Q} = \{q_{i,i+1} = 1,\ i = 1, ..., n - 1$, e $q_{ij} = 0$, caso contrário$\}$, $q_{i0} = 0$, $i = 1, ..., n - 1$ e $q_{n0} = 1$. A figura 10 ilustra um *job-shop* simétrico e um *flow-shop* uniforme, cada um com $n = 3$ estações (a estação 0 não aparece na figura). Note que em ambos os exemplos não ocorre realimentação imediata.

(a)



(b)

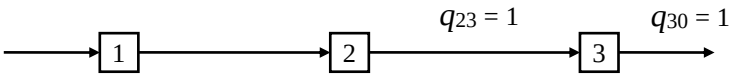


Figura 10 - (a) *Job-shop* simétrico e (b) *flow-shop* uniforme, ambos com $n = 3$ estações

Equações de taxa de tráfego: As equações de taxa de tráfego fornecem a taxa média de chegada dos fluxos em cada estação. Seja λ_j a taxa média de chegada na estação j, definida como $\lambda_j = 1/E(a_j)$, onde a_j é o intervalo de tempo entre chegadas (externas e internas) na estação j. Admitindo-se que o sistema atinja o estado de equilíbrio, λ_j é obtido através do seguinte sistema de equações lineares:

$$\lambda_j = \lambda_{0j} + \sum_{i=1}^{n} q_{ij}\lambda_i \qquad \text{para } j = 1, 2, ..., n \tag{3.4}$$

(obviamente, se a rede for acíclica, cada λ_j pode ser computado sem precisar resolver (3.4)). Dado que a rede é aberta, segue que $q_{i0} > 0$ para pelo menos algum i, $i = 1, ..., n$. Assim, a matriz $\boldsymbol{Q}$ é sub-estocástica (i.e., $\sum_{j=1}^{n} q_{ij} < 1$ para pelo menos algum i) e, com isso, pode ser mostrado que (3.4) tem uma solução única satisfazendo $\lambda_j \geq 0$ para todo j. Usando essa solução, pode-se calcular a *utilização média* ρ_j (ou intensidade de tráfego) da estação j, definida por:

$$\rho_j = \frac{\lambda_j}{\mu_j m_j} \qquad \text{com } 0 \leq \rho_j < 1 \tag{3.5}$$

A razão $\lambda_j / \mu_j = \lambda_j E(s_j)$ em (3.5) é chamada *carga ofertada*; note que ela corresponde ao número médio de máquinas ocupadas na estação j, $\rho_j m_j$. A taxa média de chegada na estação j a partir da estação i é dada por:

$$\lambda_{ij} = \lambda_i q_{ij} \tag{3.6}$$

e a taxa média de partida externa (para a estação 0) da estação j é dada por:

$$\lambda_{0j} = \lambda_j (1 - \sum_{i=1}^{n} q_{ji}) \tag{3.7}$$

Como foi admitido que o sistema esteja em equilíbrio, a soma em j de λ_{j0} (ou λ_{0j}) resulta igual a λ_0, a *taxa média de produção* da rede (*throughput*). O número médio de visitas $E(V_j)$ de um *job* arbitrário na estação j é definido por:

$$E(V_j) = \frac{\lambda_j}{\lambda_0} \tag{3.8}$$

Uma outra medida de interesse é o máximo valor da taxa média de produção da rede, diga-se λ^*_0, tal que, para qualquer valor acima de λ^*_0, o número de *jobs* no sistema cresce sem limite (alguns autores têm chamado essa taxa de *capacidade do sistema*). Note que, enquanto $\rho_j < 1$, o número de *jobs* na estação *j* é finito. O valor λ^*_0 é obtido assim que $\rho_j = 1$ para alguma estação *j*. Pode ser mostrado, substituindo (3.3) e (3.8) em (3.4), que (Shanthikumar e Buzacott, 1993):

$$\lambda_0^* = \min_{j=1,..,n}\left\{\frac{m_j \mu_j}{E(V_j)}\right\} = \min_{j=1,..,n}\left\{\frac{\lambda_0}{\rho_j}\right\}$$

Teorema de Jackson: Seja o vetor **L** = (L_1, L_2, ..., L_n) definindo o estado do sistema, onde cada L_j corresponde ao número de *jobs* em fila e em serviço na estação *j*. Jackson (1957, 1963) mostrou que se $\rho_j < 1$ para todo *j*, então a probabilidade do sistema estar em equilíbrio no estado **L**, π(**L**), é dada pela seguinte forma de produto:

$$\pi(\mathbf{L}) = \prod_{j=1}^{n} \pi_j(L_j) \tag{3.9}$$

com: $\pi_j(L_j) = f_j(L_j)\pi_j(0)$ para $L_j = 0, 1, ...$

$$f_j(L_j) = f_j(L_j - 1)\frac{\lambda_j}{\mu_j r_j(L_j)} \quad \text{para } L_j = 1, 2, ...$$

$$f_j(0) = 1$$

$$\pi_j(0) = \frac{1}{\sum_{L_j=0}^{\infty} f_j(L_j)}$$

Observe que π é a distribuição conjunta de (L_1, L_2, ..., L_n). O denominador $\mu_j\, r_j(L_j)$ define a taxa de serviço da estação *j* quando o sistema tem L_j *jobs* (note que essa taxa pode variar em função do estado do sistema). No caso particular de $r_j(L_j) = \min\{L_j, m_j\}$, então pode-se rescrever $\pi_j(L_j)$ e $\pi_j(0)$ simplesmente como:

$$\pi_j(L_j) = \begin{cases} \dfrac{(\lambda_j / \mu_j)^{L_j}}{L_j!}\pi_j(0), & se \quad L_j \le m_j \\ \dfrac{(\lambda_j / \mu_j)^{L_j}}{m_j{}^{L_j - m_j} m_j!}\pi_j(0), & se \quad L_j > m_j \end{cases} \tag{3.10}$$

$$\pi_j(0) = \left\{\sum_{t=0}^{m_j-1}\frac{(\lambda_j / \mu_j)^t}{t!} + \frac{(\lambda_j / \mu_j)^{m_j}}{(1-\rho_j)m_j!}\right\}^{-1}$$

onde ρ_j foi definido em (3.5), $\pi_j(L_j)$ é a probabilidade da estação *j* ter L_j *jobs* ($L_j = 0, 1, ...$), e $\pi_j(0)$ é uma constante normalizadora. A prova deste resultado baseia-se em mostrar que a solução em forma de produto em (3.9) satisfaz as equações de balanço do sistema, isto é, a taxa média de

entrada no estado **L** (lado esquerdo da expressão (3.11) a seguir) é igual à taxa média de saída do estado **L** (lado direito de (3.11)):

$$\sum_{i=1}^{n} \lambda_{0i}\pi(\mathbf{L}-\mathbf{e}_i) + \sum_{i=1}^{n} \mu_i r_i(L_i+1) q_{i0}\pi(\mathbf{L}+\mathbf{e}_i) + \sum_{i=1}^{n}\sum_{\substack{j=1 \\ j\neq i}}^{n} \mu_i r_i(L_i+1) q_{ij}\pi(\mathbf{L}+\mathbf{e}_i-\mathbf{e}_j)$$

$$= \sum_{i=1}^{n} [\lambda_{0i} + \mu_i r_i(L_i)(1-q_{ii})]\pi(\mathbf{L}) \qquad \text{para } \mathbf{L} \in N_+^n \qquad \textbf{(}3.11)$$

($\mathbf{e}_i$ é o vetor de dimensão n com 1 na posição i e 0 nas demais posições), junto com a equação normalizadora:

$$\sum_{\mathbf{L}\in N_+^n} \pi(\mathbf{L}) = 1$$

Conforme ilustrado na figura 11, o primeiro termo do lado esquerdo de (3.11) corresponde à taxa de transição do estado **L** - $\mathbf{e}_i$ para o estado **L**, ao chegar um *job* de fora do sistema na estação *i*, o segundo termo corresponde à taxa de transição do estado **L** + $\mathbf{e}_i$ para o estado **L**, ao completar um *job* na estação *i* que sai do sistema, e o terceiro termo corresponde à taxa de transição do estado **L** + $\mathbf{e}_i$ - $\mathbf{e}_j$ para o estado **L**, ao completar um *job* na estação *i* que segue para a estação *j*. O lado direito de (3.11) refere-se as taxas de transição do estado **L** para outros estados (veja figura 11), ao chegar um *job* de fora do sistema na estação *i*, ou ao completar um *job* na estação *i* que não retorna imediatamente para a estação *i*.

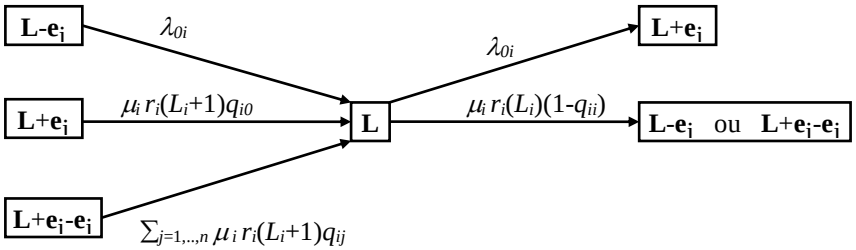


Figura 11 - Diagrama das taxas de transição de estados na expressão (3.11)

O resultado do teorema de Jackson implica que, para calcular a probabilidade de equilíbrio para um dado estado **L**, pode-se considerar cada estação como se fosse independente das demais (note em (3.9) que $\pi(\mathbf{L})$ é o produto das probabilidades $\pi_j(L_j)$ em (3.10), para cada fila $M/M/m_j$ da rede). Assim, partindo dos parâmetros iniciais $\{m_j, \lambda_{0j}, \mu_j\}$ e da matriz $\boldsymbol{Q}$, após aplicar o sistema linear (3.4) para determinar cada λ_j, pode-se decompor a rede em n estações individuais $M/M/m_j$, cada uma descrita por $\{m_j, \lambda_j, \mu_j\}$.

Para calcular as medidas de desempenho, considera-se cada estação separadamente das demais. Por exemplo, o *número médio de jobs* $E(L_j)$ na estação j é dado por (Tijms, 1986):

$$E(L_j) = \sum_{L_j=0}^{\infty} L_j \pi_j(L_j) = \frac{\rho_j(\lambda_j/\mu_j)^{m_j}\pi_j(0)}{(1-\rho_j)^2 m_j!} + \frac{\lambda_j}{\mu_j} \qquad (3.12)$$

onde $\pi_j(0)$ foi definido em (3.10). A partir de $E(L_j)$, pode-se calcular o *número médio de jobs* em fila $E(Lq_j)$, o *tempo médio de espera* em fila $E(Wq_j)$ e o *tempo médio de permanência* em fila e em serviço $E(W_j)$ na estação j, por meio das expressões:

$$E(L_j) = E(Lq_j) + \frac{\lambda_j}{\mu_j} \tag{3.13a}$$

$$E(W_j) = E(Wq_j) + E(s_j) \tag{3.13b}$$

$$E(L_j) = \lambda_j E(W_j) \tag{3.13c}$$

onde (3.13c) corresponde à *lei de Little* (veja p.e. Tijms, 1986). Por exemplo, substituindo (3.12) e (3.13c) em (3.13b), o atraso médio $E(Wq_j)$ na fila $M/M/m_j$ da estação j resulta em:

$$E(Wq_j) = \frac{(\lambda_j / \mu_j)^{m_j} \pi_j(0)}{\mu_j m_j (1-\rho_j)^2 m_j!} \tag{3.14}$$

Também pode-se calcular variâncias das medidas de desempenho, por exemplo, a variância de L_j pode ser calculada lembrando-se que $V(L_j) = E(L_j^2) - E(L_j)^2$ e $E(L_j{}^2) = \sum_{L_j=0}^{\infty} L_j{}^2 \pi_j(L_j)$. As expressões para $V(L_j)$ e $V(Wq_j)$ resultam em (Whitt, 1983a, 1993):

$$V(L_j) = E(L_j)^2 c_{L_j} \qquad \text{e} \qquad V(Wq_j) = E(Wq_j)^2 c_{Wq_j} \tag{3.15}$$

onde:
$$c_{L_j} = \frac{\frac{\lambda_j}{\mu_j}\Big(1 + P(L_j \geq m_j)\Big) + \Big[P(L_j \geq m_j)\rho_j + P(L_j \geq m_j)\Big(1 - P(L_j \geq m_j)\Big)\Big]\frac{\rho_j{}^2}{(1-\rho_j)^2}}{\left[\frac{\lambda_j}{\mu_j} + P(L_j \geq m_j)\frac{\rho_j}{(1-\rho_j)}\right]^2}$$

$$c_{Wq_j} = \frac{2 - P(Wq_j > 0)}{P(Wq_j > 0)}$$

$$P(L_j \geq m_j) = P(Wq_j > 0) = \frac{(\lambda_j / \mu_j)^{m_j} \pi_j(0)}{(1-\rho_j) m_j!}$$

Note que se $m_j = 1$ (sistema *M/M/1*), então $\pi_j(0) = 1 - \rho_j$, $P(L_j \geq m_j) = P(Wq_j > 0) = \rho_j$, e as expressões em (3.12), (3.14) e (3.15) se reduzem a:

$$E(L_j) = \frac{\rho_j}{1-\rho_j}\ , \quad E(Wq_j) = \frac{\rho_j}{\mu_j(1-\rho_j)}\ , \quad V(L_j) = \frac{\rho_j}{(1-\rho_j)^2}\ , \quad V(Wq_j) = \frac{2\rho_j - \rho_j{}^2}{\mu_j{}^2(1-\rho_j)^2} \tag{3.16}$$

Pode-se também calcular medidas de desempenho da rede, por exemplo, o número médio de *jobs* na rede é $\sum_{j=1}^{n} E(L_j)$ e, dado que L_1, L_2, ..., L_n são estatisticamente independentes conforme (3.9), a variância do número de *jobs* na rede é simplesmente $\sum_{j=1}^{n} V(L_j)$. O *leadtime médio* $E(T)$ para um *job* arbitrário (tempo médio total gasto pelo sistema para processar o *job*), incluindo os

tempos de espera e de processamento gastos desde sua chegada externa até sua partida externa da rede, é dado por:

$$E(T) = \sum_{j=1}^{n} E(V_j)E(W_j) = \sum_{j=1}^{n} E(V_j)[E(Wq_j) + E(s_j)] \qquad (3.17)$$

onde $E(V_j)$ foi definido em (3.8). É interessante observar que, embora os números de *jobs* nas estações sejam independentes entre si num dado instante de tempo, os tempos de espera Wq_1, Wq_2, ..., Wq_n nas diferentes estações são, em geral, variáveis aleatórias dependentes. A variância do *leadtime* na rede $V(T)$ é usualmente aproximada ignorando-se a correlação entre os tempos de espera (Shanthikumar e Buzacott, 1984). Buzacott e Shanthikumar (1993) forneceram uma análise das redes de Jackson considerando a correlação acima.

Agregação de *jobs* numa única classe

Na prática, os sistemas *job-shops* em geral possuem *jobs* de várias classes k, $k = 1, ..., r$, com roteiros determinísticos. O roteiro da classe k define a seqüência de estações a ser visitada pelos *jobs* desta classe, e cada visita pode corresponder a uma operação diferente. Por exemplo, a seqüência (2, 3, 1, 3, 5) define um roteiro que primeiro visita a estação 2, depois a estação 3, e assim por diante. Este exemplo ilustra que as estações podem aparecer em qualquer ordem no roteiro, que nem todas as estações da rede precisam fazer parte do roteiro (p.e., a estação 4), e que pode-se ter estações sendo visitadas mais de uma vez no mesmo roteiro (estação 3). Além disso, as duas operações produzidas na estação 3 podem ser diferentes, cada uma com uma distribuição de probabilidade diferente do tempo de serviço. Pode-se encontrar desde situações com muitas classes (100 a 1000), cada uma seguindo um roteiro diferente, como no caso da montagem de placas de circuito impresso, até situações com poucas classes (1 a 10), mas com longos roteiros (30 a 300 visitas) ao longo de relativamente poucas estações (5 a 50), como no caso da fabricação de circuitos integrados (Segal e Whitt, 1989).

Considere a seguinte notação para os dados de entrada:

- n número de estações internas na rede
- r número de classes na rede.

Para cada estação $j = 1, 2, ..., n$:

- m_j número de máquinas na estação j.

Para cada classe $k = 1, 2, ..., r$:

- n_k número de operações no roteiro da classe k
- λ_k' taxa média de chegada externa da classe k.

Para cada classe $k = 1, 2, ..., r$, e para cada operação $l = 1, 2, ..., n_k$ do roteiro da classe k:

- n_{kl} estação visitada para a operação l do roteiro da classe k
- $E(s_{kl})$ tempo médio de serviço da operação l do roteiro da classe k
- ou μ_{kl} taxa média de serviço da operação l do roteiro da classe k (i.e., $\mu_{kl} = 1/E(s_{kl})$).

Similarmente a λ_0 e $E(s_j)$ em (3.1) e (3.2), λ'_k e $E(s_{kl})$ são definidos como $\lambda_k' = \lim_{t\to\infty} N_k(t)/t$ e $E(s_{kl}) = \lim_{u\to\infty} \sum_{v=1}^{u} s_{kl,v}/u$, onde $N_k(t)$ e $s_{kl,u}$ são respectivamente o número de *jobs* da classe k que chegam na rede da estação 0 durante $(0,t]$, e o tempo de processamento do u-ésimo *job* da classe k requerendo a operação l. Note agora que o roteiro de cada classe k é descrito por meio de n_k e n_{kl} (ao invés da matriz $\boldsymbol{Q}$).

Devido ao fato do número de classes *r* poder ser muito grande, pode ser conveniente tratá-las agregadas numa única classe. Após esta *classe agregada* ter sido analisada, pode-se retornar à rede original e estimar as medidas de desempenho para cada classe individualmente. Este procedimento é descrito a seguir. Primeiramente, deseja-se obter os parâmetros da classe agregada em cada estação *j*, *j* = 1, ..., *n*. Seja 1{.} uma função indicadora que resulta 1 se a expressão {.} é verdadeira e 0 caso contrário. Cada λ_{0j} é calculado simplesmente somando-se as taxas médias de chegadas externas de todas as classes cuja primeira operação ocorra na estação *j*, ou seja,

$$\lambda_{0j} = \sum_{k=1}^{r} \lambda_k' 1\{n_{k1} = j\} \qquad (3.18)$$

Similarmente, $\lambda_{j0} = \sum_{k=1}^{r} \lambda_k' 1\{n_{kn_k} = j\}$. Assim como antes, a taxa média de produção da rede λ_0 é obtida simplesmente somando-se λ_{0j} (ou λ_{j0}) em j. As taxas agregadas λ_j e λ_{ij} agora são facilmente computadas por:

$$\lambda_j = \sum_{k=1}^{r} \sum_{l=1}^{n_k} \lambda_k' 1\{n_{kl} = j\} \qquad (3.19a)$$

$$\lambda_{ij} = \sum_{k=1}^{r} \sum_{l=1}^{n_k - 1} \lambda_k' 1\{n_{kl} = i, n_{k,l+1} = j\} \qquad (3.19b)$$

ao invés de se ter que resolver um sistema linear como em (3.4). Cada elemento da matriz de transição $\boldsymbol{Q}$ é dado simplesmente por $q_{ij} = \lambda_{ij} / \lambda_i$, conforme (3.6).

Para obter o tempo médio de serviço agregado $E(s_j)$ na estação *j*, observe que:

$s_j = s_{kl}$ com probabilidade: $\dfrac{\lambda_k' \{n_{kl} = j\}}{\lambda_j}$

e portanto,

$$E(s_j) = \frac{\sum_{k=1}^{r} \sum_{l=1}^{n_k} E(s_{kl}) \lambda_k' \{n_{kl} = j\}}{\lambda_j} \qquad (3.20)$$

Desta forma, tem-se cada estação *j* descrita por $\{m_j, \lambda_j, \mu_j = 1/E(s_j)\}$. As medidas ρ_j, λ_{j0}, $E(V_j)$, $E(L_j)$, $E(Wq_j)$, $E(T)$ são definidas conforme as equações (3.5), (3.7), (3.8), (3.12), (3.14) e (3.17).

Ao retornar à rede original, pode-se obter medidas de desempenho para cada classe *k* individualmente, a partir da classe agregada. Por exemplo, o *leadtime* médio $E(T_k)$ para um *job* arbitrário da classe *k* é dado por:

$$E(T_k) = \sum_{l=1}^{n_k} E(Wq_{n_{kl}}) + \sum_{l=1}^{n_k} E(s_{kl}) \qquad (3.21)$$

onde $E(Wq_{n_{kl}})$ é o tempo médio de espera em fila da classe agregada na estação n_{kl} (i.e., a estação relativa à *l*-ésima operação do roteiro da classe *k*), obtido conforme (3.14). Note que o primeiro termo do lado direito de (3.21) corresponde ao tempo médio total de espera em fila na

rede para um *job* da classe *k*, enquanto o segundo termo corresponde ao tempo médio total de serviço na rede para um *job* da classe *k*.

Convém observar que em redes onde *jobs* visitam várias vezes a mesma estação *j*, se em cada visita eles recebem uma operação *l* diferente (i.e. as taxas médias de processamento na estação *j* variam com a operação *l*), então a rede sob certas condições pode resultar instável, mesmo que os processos de chegada e processamento sejam Poisson, a disciplina seja *FCFS*, e as utilizações médias sejam menores do que 1 (veja a discussão em Bramson (1994) sobre uma rede de classe única com roteiro (1, 2, 2, ..., 2, 1)).

Agregação com roteiros probabilísticos

Se o roteiro da classe *k* for probabilístico, então o procedimento de agregação acima precisa ser estendido. Por exemplo, após completarem a operação numa estação, em média 5% dos *jobs* têm de ser descartados e outros 10% requerem retrabalho. Considere inicialmente o caso em que este roteiro não envolve mais de uma visita à mesma estação ou, se envolver, a revisita é uma operação de retrabalho (com mesmo tempo médio de processamento), e os *jobs*, após o retrabalho, voltam ao roteiro original (lembre-se que o retrabalho é representado por arcos com realimentação, conforme seção 2.2). Note que este roteiro ainda é Markoviano, pois, a probabilidade q^k_{ij} da classe *k* visitar a estação *j* após a estação *i* não depende das estações visitadas antes da estação *i*. Neste caso, basta resolver o sistema linear (3.4) para esta classe *k*, isto é,

$$\lambda_j^k = \lambda_{0j}^k + \sum_{i=1}^{n} q_{ij}^k \lambda_i^k \qquad \text{para } j = 1, ..., n. \tag{3.22}$$

onde λ_j^k denota a taxa média de chegadas da classe *k* na estação *j* (similarmente para λ_{0j}^k). Para ilustrar, considere o exemplo de rede de Jackson apresentado em Askin e Standridge (1993), com 4 classes de *jobs* e 4 estações (figura 12). Cada estação tem respectivamente 3, 3, 1 e 1 máquinas, indicadas entre parênteses dentro de cada estação da figura (por simplicidade, a figura não inclui a estação 0). Os dados de cada classe aparecem na tabela 2. Os números entre parênteses na coluna n_{kl} indicam $q^k_{ij\ j}$, $i = n_{k,l-1}$, $j = n_{kl}$, isto é, a probabilidade dos *jobs* da classe *k* receberem a operação *l* na estação *j*, após terem recebido a operação *l*-1 na estação *i*. Por exemplo, apenas 50% dos *jobs* da classe 2 seguem para a operação 3 na estação 4, após a operação 2 na estação 2 (veja também a indicação entre parênteses nos arcos da figura).

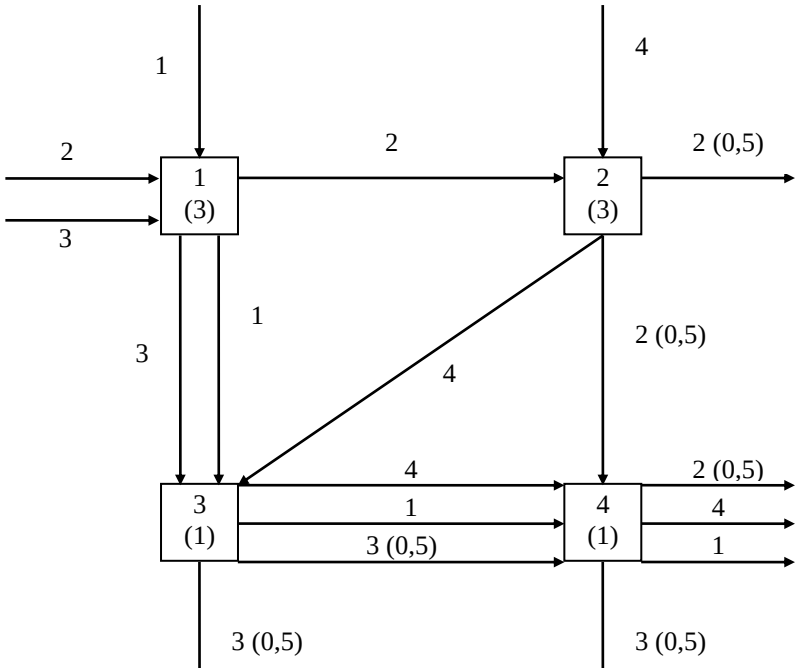


Figura 12 - Roteiros das 4 classes de *jobs* ao longo das 4 estações do exemplo de Askin e Standridge (1993)

Tabela 2 - Dados de entrada das classes de produto no exemplo de Askin e Standridge (1993)

Classe k	λ_k' (*jobs* por h)	n_k	n_{kl} (q_{ij}^k , $i = n_{k,l-1}$, $j = n_{kl}$)	$E(s_{kl})$ (h por *job*)
1	2	3	1, 3 (1), 4 (1)	0,05, 0,025, 0,05
2	10	3	1, 2 (1), 4 (0,5)	0,1, 0,125, 0,025
3	6	3	1, 3 (1), 4 (0,5)	0,25, 0,025, 0,05
4	3	3	2, 3 (1), 4 (1)	0,25, 0,05, 0,05
Total:	21			

A tabela 3 apresenta os valores de λ_j^k obtidos ao resolver o sistema em (3.22) para cada classe *k*, após utilizar (3.18) para determinar λ_{0j}^k. Também pode-se obter λ_{ij}^k usando (3.6). Note que no caso das classes 1 e 4 (com roteiros determinísticos), λ_j e λ_{ij}^k também poderiam ter sido encontrados por meio de (3.19) e (3.20). A coluna λ_j da tabela 3 apresenta a taxa média de chegada agregada em cada estação *j*, determinada simplesmente por:

$$\lambda_j = \sum_{k=1}^{r} \lambda_j^k \tag{3.23}$$

Tabela 3 - Valores obtidos para cada estação *j* no exemplo de Askin e Standridge (1993)

j	λ_j^1	λ_j^2	λ_j^3	λ_j^4	λ_j	$E(s_j)$	μ_j	ρ_j	$E(L_j)$	$E(Wq_j)$
1	2	10	6	0	18	0,144	6,925	0,87	7,52	0,273
2	0	10	0	3	13	0,154	6,502	0,67	2,89	0,068
3	2	0	6	3	11	0,032	31,427	0,35	0,54	0,017
4	2	5	3	3	13	0,040	24,765	0,52	1,10	0,045
$\sum$									12,05	

Seja s_j^k o tempo de processamento da classe *k* na estação *j*. Lembre-se que foi assumido que cada roteiro probabilístico *k* não envolve mais de uma visita à mesma estação *j* ou, se envolver, trata-se de uma operação de retrabalho com o mesmo tempo de processamento s_j^k. Neste caso, após receber a operação na estação *j*, o *job* da classe *k* segue para as demais estações conforme o roteiro original. Note que:

$s_j = s_j^k$ com probabilidade: $\frac{\lambda_j^k}{\lambda_j}$

onde $s_j^k = \sum_{l=1}^{n_k} s_{kl} 1\{n_{kl} = j\}$, e portanto,

$$E(s_j) = \frac{\sum_{k=1}^{r} E(s_j^k)\lambda_j^k}{\lambda_j} = \frac{\sum_{k=1}^{r}\sum_{l=1}^{n_k} E(s_{kl})1\{n_{kl} = j\}\lambda_j^k}{\lambda_j}$$

As demais colunas da tabela 3 apresentam os valores de $E(s_j)$, μ_j, ρ_j, $E(L_j)$ e $E(Wq_j)$ para a classe agregada. Também pode-se calcular o *leadtime* médio para cada classe *k*, dado por (compare com (3.17) e (3.21)):

$$E(T_k) = \sum_{j=1}^{n} \frac{\lambda_j^k}{\lambda_j}[E(Wq_j) + \sum_{l=1}^{n_k} E(s_{kl})1\{n_{kl} = j\}]$$

No caso mais geral de se ter classes com roteiros probabilísticos fazendo mais de uma visita à mesma estação, se estas visitas envolverem operações diferentes ou, se a partir de uma revisita, os *jobs* seguirem roteiros diferentes do roteiro original (i.e., não trata de um simples retrabalho), então o sistema (3.22) não é mais válido. Para ver isso, defina uma nova classe no exemplo anterior (classe 5) com $\lambda_5^{'}$ = 10 e roteiro (1, 2, 1 (0,2)), isto é, a operação 1 ocorre na estação 1, em seguida os *jobs* seguem para a operação 2 na estação 2 e apenas 20% deles voltam para a estação 1 para a operação 3 (diferente da operação 1), antes de saírem da rede. A figura 13a ilustra esse roteiro (não confundir com o retrabalho ilustrado na figura 13b por meio de um arco de realimentação).

Figura 13 – (a) Roteiro da classe 5, sem retrabalho, (b) roteiro com retrabalho

Note que, em equilíbrio, $\lambda^5{}_1$ = 12, $\lambda^5{}_2$ = 10, entretanto, o sistema (3.22) resulta em:

λ_1^5 = 10 + 0,2 λ_2^5
λ_2^5 = 0 + λ_1^5

o que implica incorretamente em $\lambda_1^5 = \lambda_2^5$ = 12,5. Buzacott e Shanthikumar (1993) propuseram um procedimento que, primeiro, cria classes artificiais para eliminar roteiros com mais de uma visita à mesma estação e, depois, resolve o seguinte sistema linear em λ_j^k admitindo que *jobs* possam trocar de classe:

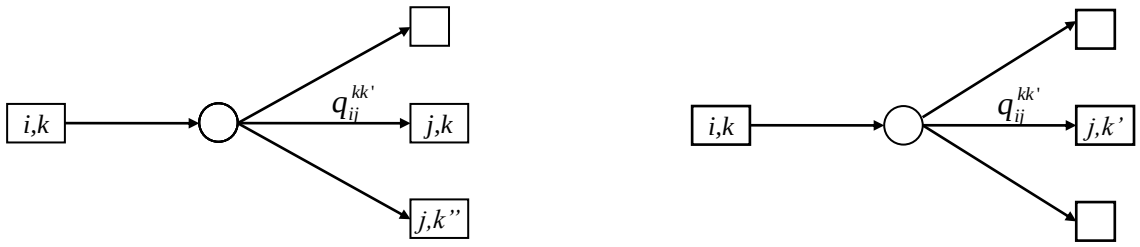
$$\lambda_j^k = \lambda_{0j}^k + \sum_{i=1}^{n}\sum_{k'=1}^{r} q_{ij}^{k'k}\lambda_i^{k'} \qquad \text{para } j = 1, ..., n,\ k = 1, ..., r \qquad (3.24)$$

onde $\lambda^k{}_{0j}$ é definido conforme (3.18) e $q_{ij}^{k'k}$ é a probabilidade de um *job* saindo da estação *i* como um *job* da classe *k*', seguir para a estação *j* como um *job* da classe *k*. Note que, desta maneira, o roteiro de cada classe *k* (incluindo as classes artificiais) é Markoviano. Cada probabilidade $q_{ij}^{kk'}$ é definida por:

$$q_{ij}^{kk'} = \begin{cases} \sum_{l=1}^{n_k-1} 1\{n_{kl} = i, n_{k,l+1} = j\} - \sum_{\substack{k''=1 \\ k''\neq k}}^{r}\sum_{l=1}^{n_k}\sum_{l''=1}^{n_{k''}} p_{ll''}^{kk''} 1\{n_{kl} = i, n_{k''l''} = j\}, & se \quad k = k' \\ \sum_{l=1}^{n_k}\sum_{l'=1}^{n_{k'}} p_{ll'}^{kk'} 1\{n_{kl} = i, n_{k'l'} = j\}, & se \quad k \neq k' \end{cases} \qquad (3.25)$$

onde $p_{ll'}^{kk'}$ é a probabilidade de um *job* da classe *k*, após a operação *l* na estação n_{kl}, seguir como um *job* da classe *k*' para a operação *l'* na estação $n_{k'l'}$. Note em (3.25) que:

- se *k* = *k*' e (*i*,*j*) pertence ao roteiro da classe *k*, então $q_{ij}^{kk'}$ corresponde a 1 menos a probabilidade de que *jobs* saindo da estação *i* como *jobs* da classe *k* (em alguma operação *l* da classe *k*) sigam para a estação *j* como *jobs* de qualquer outra classe *k*" diferente de *k* (em alguma operação *l"* dessa classe *k*") (veja figura 14a)
- se *k* ≠ *k*', então $q_{ij}^{kk'}$ corresponde simplesmente à probabilidade de que *jobs* saindo da estação *i* como *jobs* da classe *k* (em alguma operação *l* da classe *k*) sigam para a estação *j* como *jobs* da classe *k*' (em alguma operação *l'* dessa classe *k*') (veja figura 14b).



(a) se *k* = *k*' (b) se *k* ≠ *k*'

Figura 14 - $q_{ij}^{kk'}$ na expressão (3.25): (a) se *k* = *k*' e (b) se *k* ≠*k*'

Por meio de $p_{ll'}^{kk'}$, pode-se definir classes artificiais para evitar várias visitas de uma mesma classe à mesma estação. Por exemplo, a classe 5, definida acima com roteiro (1, 2, 1 (0,2)), pode ser substituída por duas classes artificiais, diga-se 5' e 5", a primeira percorrendo o roteiro (1, 2) (com duas operações), e a segunda percorrendo o roteiro (1 (0,2)) (com apenas uma operação). Ao terminar a operação 2 na estação 2, 20% dos *jobs* da classe 5' mudam para a classe 5", e assim, seguem para a estação 1 para obter a operação 1 dessa classe. Note que para descrever essa mudança, deve-se definir $p_{ll'}^{kk'} = p_{21}^{5'5"} = 0{,}2$, o que resulta em $q_{ij}^{kk'} = q_{21}^{5'5"} = 0{,}2$. O sistema (3.24) resulta em:

$\lambda_1^{5'} = 10 + 0$

$\lambda_2^{5'} = 0 + \lambda_1^{5'}$

$\lambda_1^{5''} = 0 + 0{,}2\ \lambda_2^{5'}$

o que produz corretamente $\lambda_1^{5'} = 10$, $\lambda_2^{5'} = 10$ e $\lambda_1^{5''} = 2$, ou seja, $\lambda_1^{5} = 12$, $\lambda_2^{5} = 10$, dado que as classes 5' e 5" correspondem originalmente à classe 5. Os passos seguintes são similares aos do procedimento anterior. Para maiores detalhes, veja Buzacott e Shanthikumar (1993).

Criação e combinação de *jobs*

Para representar a criação ou combinação de *jobs* após o término de serviço na estação *j*, pode-se definir o fator multiplicador γ_j. Para uma dada taxa média de chegada λ_j, a taxa média de partida da estação *j* é simplesmente $\gamma_j \lambda_j$, onde $\gamma_j > 1$ representa criação de *jobs* e $\gamma_j < 1$, combinação de *jobs*, após o serviço na estação *j*. Desta maneira, pode-se representar, por exemplo, situações em que uma única partida dispara γ_j, $\gamma_j > 1$, chegadas na próxima estação a ser visitada (criação), ou situações em que $1/\gamma_j$, $\gamma_j < 1$, partidas resultam em uma única chegada na próxima estação (combinação). Para isso, as equações de taxa de tráfego em (3.4) (ou em (3.22)) devem ser substituídas por:

$$\lambda_j = \lambda_{0j} + \sum_{i=1}^{n} \gamma_i q_{ij} \lambda_i$$

No caso de $\gamma_j > 1$, condições adicionais têm de ser satisfeitas para garantir que este sistema tenha solução (Whitt, 1983a). Note que, se $\gamma_j \neq 1$, então as taxas médias de chegadas e partidas de *jobs* da rede não precisam mais coincidir sob condição de equilíbrio. Na prática, este procedimento pode ser utilizado para representar operações de montagem, desmontagem, alterações do tamanho dos lotes, processamento em lotes, entre outros (Whitt, 1983a, Segal e Whitt, 1989).

Sistema de movimentação de materiais

Em *job-shops* com sistema de movimentação de materiais lento ou relativamente custoso, pode ser necessário ter que incorporá-lo no modelo de rede de filas para permitir uma análise mais adequada. Se o sistema contiver transportadores *dedicados* para fazer a movimentação de *jobs* entre as estações *i* e *j*, $i \neq j$, pode-se representar esses transportadores como estações artificiais (*i*,*j*), entre as estações *i* e *j*, e simplesmente incorporá-las na rede.

Para isso, deve-se definir o tempo médio de processamento (i.e., transporte) $E(s_{(i,j)})$ e o número de máquinas (transportadores) $m_{(i,j)}$ de cada estação (*i*,*j*). Eventualmente pode-se fazer $m_{(i,j)} \to \infty$ para refletir o tempo em fila tendendo a zero em certos sistemas de movimentação como, por exemplo, esteiras rolantes. Além disso, deve-se modificar a matriz de transição $\boldsymbol{Q}$ para $\boldsymbol{Q}$', onde $q'_{i,(i,j)} = q_{ij}$ e $q'_{(i,j),j} = 1$ indicam que o fluxo de saída da estação *i* com destino a *j* deve passar pela estação (*i*,*j*) antes de chegar em *j*. Se for o caso, também pode-se incluir estações (0,*j*) e (*j*,0) entre a estações 0 (externa) e *j* (interna), para representar o transporte para dentro e fora da rede.

Se o sistema de movimentação de materiais for *centralizado*, pode-se definir uma única estação *n*+1 para representá-lo, e modificar a matriz $\boldsymbol{Q}$ conforme acima, para refletir os fluxos entre a estação *n*+1 e as demais estações da rede. Em certos casos, entretanto, os tempos médios de transporte podem variar muito entre duas estações *i* e *j* passando através da estação *n*+1, prejudicando assim o cálculo de $E(s_{n+1})$. Uma alternativa é definir classes artificiais (*i*,*j*) para

representar *jobs* em trânsito entre as estações *i* e *j*, e depois agregar estas classes conforme os procedimentos anteriores. Para maiores detalhes de como representar o sistema de movimentação nas redes de filas, veja p.e. Askin e Standridge (1993) e Buzacott e Shanthikumar (1993).

3.1.2 Rede de filas *M/M/m* de múltiplas classes

Se os roteiros (determinísticos ou probabilísticos) e suas necessidades de processamento em cada estação forem significativamente diferentes entre as classes, ou se a disciplina de atendimento nas filas for com prioridade, então pode ser mais conveniente utilizar modelos de múltiplas classes, ao invés de agregar todas as classes em uma única classe e tratá-la pelo modelo da seção 3.1.1. Kelly (1975, 1979) e Baskett *et al.* (1975) estenderam as soluções em forma de produto de Jackson para redes de filas com múltiplas classes de *jobs*.

Seja a matriz $\mathbf{L} = (\mathbf{L}_1, \mathbf{L}_2, ..., \mathbf{L}_n)$ definindo o estado do sistema, onde cada vetor $\mathbf{L}_j = (L_j^1, L_j^2, ..., L_j^r)$ denota o estado da estação *j*, e cada L_j^k indica o número de *jobs* da classe *k* na estação *j*. Define-se $L_j = \sum_{k=1}^{r} L_j^k$ como o número de *jobs* na estação *j* (não confundir com o vetor $\mathbf{L}_j$). Pode ser mostrado que a probabilidade de equilíbrio $\pi(\mathbf{L})$ tem a seguinte forma de produto (veja p.e. Buzacott e Shanthikumar, 1993):

$$\pi(\mathbf{L}) = \prod_{j=1}^{n} \pi_j(\mathbf{L}_j) \qquad (3.26)$$

com: $\pi_j(\mathbf{L}_j) = \binom{L_j}{L_j^1, L_j^2, ..., L_j^r} \prod_{k=1}^{r} \left(\frac{\lambda_j^k}{\lambda_j}\right)^{L_j^k} \frac{f_j(L_j)}{\sum_{i=0}^{\infty} f_j(i)}$ para $\mathbf{L}_j \in N_+^r$

$f_j(L_j) = f_j(L_j - 1) \frac{\lambda_j}{\mu_j r_j(L_j)}$ para $L_j = 1, 2, ...$

$f_j(0) = 1$

onde $\binom{L_j}{L_j^1, L_j^2, ..., L_j^r}$ denota a combinação multinomial de L_j em $(L_j^1, L_j^2, ..., L_j^r)$, e λ_j e λ_j^k são definidos conforme (3.23) e (3.24). Similarmente ao teorema de Jackson da seção 3.1.1, o denominador $\mu_j r_j(L_j)$ define a taxa de serviço da estação *j* quando o sistema tem L_j *jobs* (note que a taxa pode ser dependente do estado do sistema). No caso particular de $r_j(L_j) = 1$ (i.e., uma única máquina na estação *j*), então pode-se rescrever $\pi_j(\mathbf{L}_j)$ simplesmente como:

$$\pi_j(\mathbf{L}_j) = \binom{L_j}{L_j^1, L_j^2, ..., L_j^r} \prod_{k=1}^{r} \left(\frac{\lambda_j^k}{\lambda_j}\right)^{L_j^k} (1 - \rho_j) \rho_j^{L_j} \qquad \text{para } \mathbf{L}_j \in N_+^r$$

Estas expressões ainda podem ser estendidas para permitir transições de *jobs* entre as classes e outras disciplinas além de *FCFS* (p.e., fila com prioridades), conforme Baskett *et al.* (1975). Embora estes resultados sejam interessantes, eles são difíceis de serem implementados na prática devido à elevada dimensão do espaço de estados **L** em (3.26). Além disto, as suposições feitas nas redes de Jackson são muito restritivas em *job-shops* genéricos e outros sistemas de manufatura. Por exemplo, Bitran e Tirupati (1988, 1989a) sugeriram que distribuições exponenciais superestimam a variabilidade dos tempos de processamento encontrados em muitas operações de manufatura (veja, p.e., o estudo de caso da fabricação de circuitos integrados em

Lynes e Miltenburg, 1994). Maiores detalhes das redes de Jackson podem ser encontrados nos exames de Disney e Konig (1985), Walrand (1990), Suri *et al.* (1993), Buzacott e Shanthikumar (1993), Gershwin (1994), e nas referências neles encontradas.

3.2 Redes de Jackson generalizadas (Métodos aproximados de decomposição)

Métodos de decomposição *paramétricos* podem ser vistos como esforços para estender para as redes de Jackson generalizadas, a solução em forma de produto e a "independência" entre as estações das redes de Jackson. Os processos de chegada e serviço são aproximados por processos de renovação, e as estações são tratadas (aproximadamente) como se fossem estocasticamente independentes. Cada estação é então analisada separadamente como um sistema de fila *GI/G/m*, onde cada processo de chegada e serviço é descrito apenas por dois *parâmetros*: a média e o *scv*. Note que admitir que o processo de chegada numa estação é de renovação não implica em ignorar a eventual dependência entre os intervalos de tempo entre chegadas sucessivas nesta estação. Conforme é visto na seção 3.1.1, as aproximações tentam capturar a dependência entre estes intervalos por meio dos *scv* dos processos de chegada nas estações.

O procedimento completo de decomposição pode ser descrito essencialmente em três passos:

- *Passo 1*: análise das interações entre as estações da *OQN*
- *Passo 2*: avaliação das medidas de desempenho para cada estação
- *Passo 3*: avaliação das medidas de desempenho da rede.

O passo 1 determina os fluxos internos de chegada para cada estação. O passo 2 decompõe a *OQN* num sistema de estações individuais para computar, separadamente, as medidas de desempenho para cada estação. O passo 3 recompõe os resultados da decomposição dos passos 1 e 2, para computar as medidas de desempenho da rede como um todo. O passo 1 é fundamental neste procedimento e envolve três processos básicos:

(i) *superposição de chegadas*
(ii) *partidas*
(iii) *separação de partidas.*

A figura 15 ilustra cada um desses processos. O *processo de superposição* combina os diversos fluxos individuais de chegada numa estação, vindos de outras estações (inclusive a estação externa), resultando num único fluxo superposto de chegada na estação. O *processo de partidas* é o resultado da combinação do processo de chegada agregado e do processo de serviço na estação. Finalmente, o *processo de separação* decompõe o fluxo superposto de partida da estação nos diversos fluxos individuais de partida dessa estação, a caminho das outras estações (inclusive a estação externa).

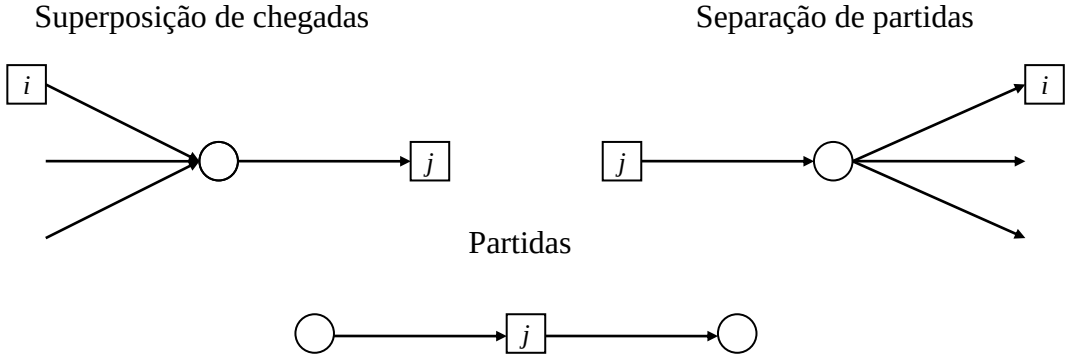


Figura 15 - Superposição de chegadas, partidas e separação de partidas

Em geral, apenas a média e o *scv* dos processos de chegada e serviço são suficientes para fornecer boas aproximações, e eles têm sido freqüentemente utilizados para descrever os fluxos acima. Esta abordagem foi inicialmente proposta por Reiser e Kobayashi (1974) e Kobayashi (1974), e melhorada por Sevcik *et al.* (1977), Kuehn (1979), Shanthikumar e Buzacott (1981), Albin (1984), Whitt (1983a), Bitran e Tirupati (1988), Segal e Whitt (1989), e Whitt (1994, 1995), entre outros. Shanthikumar e Buzacott (1981) foram os primeiros a aplicar este método para sistemas de manufatura. A seção 3.2.1 apresenta os passos 1, 2 e 3 para rede de filas *GI/G/1* de classe única. A seção 3.2.2 estende estes passos para rede de filas *GI/G/m* e a seção 3.2.3, para rede de filas *GI/G/m* com múltiplas classes, com uma atenção especial para roteiros determinísticos. Este último caso tem sido considerado na prática para modelar *job-shops*.

3.2.1 Rede de filas *GI/G/1* de classe única

Esta seção supõe que todos os *jobs* pertençam (ou possam ser adequadamente agregados) a uma única classe, e se movam pelas estações de acordo com um roteiro Markoviano. Assim como nas redes de Jackson, tanto os intervalos de tempo entre chegadas externas a_{0j}, como os tempos de serviço s_j em cada estação, são supostos *iid*, porém, considera-se agora distribuições genéricas. Inicialmente analisa-se o caso em que as estações têm uma única máquina. Seja a seguinte notação para os dados de entrada (compare com a da seção 3.1.1):

n número de estações internas na rede.

Para cada estação j = 1, 2, ..., n:

λ_{0j} taxa média de chegada externa na estação j ($\lambda_{0j} = 1/E(a_{0j})$)

ca_{0j} *scv* ou parâmetro de variabilidade do intervalo de tempo entre chegadas externas na estação j ($ca_{0j} = V(a_{0j})/E(a_{0j})^2$)

μ_j taxa média de serviço na estação j ($\mu_j = 1/E(s_j)$)

cs_j *scv* ou parâmetro de variabilidade do tempo de serviço na estação j ($cs_j = V(s_j)/E(s_j)^2$).

Para cada par (i,j), i = 1, 2, ..., n; j = 1, 2, ..., n:

q_{ij} probabilidade de um *job*, após ser atendido na estação i, seguir para a estação j.

Assim, os dados de entrada têm $n^2 + 4n$ números e cada estação interna j é descrita pelos 4 parâmetros: $\{\lambda_{0j}, ca_{0j}, \mu_j, cs_j\}$ e os fluxos de *jobs* entre as estações, pela matriz de transição $\boldsymbol{Q} = \{q_{ij}, i = 1, ..., n, j = 1, ..., n\}$. Na prática, os *scv* ca_j e cs_j podem ser obtidos coletando-se amostras, similarmente à discussão para λ_{0j} e $E(s_j)$ (veja p.e. (3.2)). Cada *scv* pode assumir qualquer valor não-negativo, por exemplo, 0 para uma distribuição puramente determinística, 1/p para uma distribuição Erlang de ordem p, 1 para uma distribuição exponencial, ou um valor maior que 1 para uma distribuição hiperexponencial.

Convém lembrar que $\boldsymbol{Q}$ também pode descrever roteiros determinísticos (Markovianos), uma vez que estes são casos particulares de roteiros probabilísticos Markovianos. Se $q_{jj} > 0$, então a estação j tem uma realimentação imediata. No texto que segue, assume-se que não ocorram realimentações imediatas. No caso de uma rede contê-las originalmente, pode-se facilmente removê-las com alguns ajustes nos parâmetros iniciais, conforme mostrado no anexo 4. Este procedimento melhora a qualidade das aproximações (Kuehn, 1979, Whitt, 1983a). Todas as suposições feitas nas redes de Jackson também são aqui assumidas exceto, é claro, distribuições

exponenciais para os processos de chegada externa e serviço (que resultariam em $ca_{0j} = 1$ e $cs_j = 1$ para cada estação j).

Passo 1

No passo 1 deseja-se determinar, para cada estação j, dois parâmetros:
(i) a taxa média de chegada λ_j, definida como $\lambda_j = 1 / E(a_j)$, onde a_j é o intervalo de tempo entre chegadas na estação j, conforme seção 3.1.
(ii) o *scv* ou parâmetro de variabilidade do intervalo de tempo entre chegadas ca_j (note que nas redes de Jackson tem-se $V(a_j) = E(a_j)^2$ devido ao processo de Poisson e, portanto, $ca_j = 1$).

Em outras palavras, partindo dos parâmetros iniciais $\{\lambda_{0j}, ca_{0j}, \mu_j, cs_j\}$ e da matriz $\boldsymbol{Q}$, deseja-se descrever cada estação j pelos parâmetros $\{\lambda_j, ca_j, \mu_j, cs_j\}$. Os dois parâmetros λ_j e ca_j são, respectivamente, as soluções de dois sistemas lineares: as equações de taxa de tráfego (3.4) e as equações de variabilidade de tráfego (definidas a seguir). Inicialmente, obtém-se taxas médias de chegada *exatas* por meio do primeiro sistema linear, similarmente às redes de Jackson. Estas taxas são então utilizadas para obter-se parâmetros de variabilidade *aproximados* por meio do segundo sistema linear. Pode ser mostrado que, dado que $\boldsymbol{Q}$ é sub-estocástica, ambos os sistemas têm solução única não-negativa.

Equações de variabilidade de tráfego

As equações de variabilidade de tráfego envolvem os três processos discutidos anteriormente: superposição de chegadas, partidas, e separação de partidas. Elas fornecem aproximações para o *scv* do intervalo de tempo entre chegadas ca_j em cada estação j. Estas aproximações combinam dois métodos básicos: o *método assintótico* e o *método dos intervalos estacionários*, conforme discussão a seguir.

Superposição na chegada

No processo de superposição (figura 15a), as taxas médias de chegadas e os *scv* do intervalo de tempo entre chegadas na estação j são combinados, resultando na taxa média de chegada superposta λ_j, obtida por (3.4), e no *scv* do intervalo de tempo entre chegadas superposto ca_j (lembre-se que $\lambda_j = 1 / E(a_j)$ e $ca_j = V(a_j) / E(a_j)^2$). O método assintótico e o método dos intervalos estacionários podem ser usados para determinar ca_j (ou $V(a_j)$). Estes métodos são também chamados *macro* e *micro*, respectivamente, devido à visão macroscópica e microscópica no processo de chegadas (Whitt, 1982, 1983a).

Suponha que chegadas venham ocorrendo numa estação j desde $t = -\infty$, e que uma nova chegada ocorra em $t = 0$. Seja $S_{j,p}$ o tempo decorrido até ocorrer a p-ésima chegada na estação j a partir de $t = 0$. Note que $S_{j,p} = a_{j,1} + a_{j,2} + ... + a_{j,p}$, onde $a_{j,q}$ é o intervalo de tempo entre a (q-1)-ésima e a q-ésima chegada na estação j. O método assintótico, numa tentativa de levar em conta a dependência entre os intervalos de tempo entre chegadas sucessivas, toma uma visão macroscópica do processo de chegada e tenta descrever seu comportamento num intervalo de tempo relativamente longo, produzindo:

$$E(a_j) = \lim_{p\to\infty} \frac{E(S_{j,p})}{p} = \lim_{p\to\infty} \frac{\sum_{q=1}^{p} E(a_{j,q})}{p} = \frac{pE(a_{j,1})}{p} = E(a_{j,1})$$

$$V(a_j) = \lim_{p\to\infty} \frac{V(S_{j,p})}{p} = \lim_{p\to\infty} \frac{pV(a_{j,1}) + \sum_{q=1}^{p}\sum_{\substack{r=1\\ r\neq q}}^{p} Cov(a_{j,q}, a_{j,r})}{p}$$

Este limite significa que o método assintótico incorpora a dependência das variáveis $a_{j,1}$, $a_{j,2}$, ..., $a_{j,p}$ (i.e., todos os termos de covariância). O método dos intervalos estacionários, ao contrário, ignora qualquer dependência no processo de chegada, e toma uma visão microscópica para descrever o comportamento do processo num intervalo de tempo relativamente pequeno, produzindo:

$$E(a_j) = E(a_{j,1}) \qquad \text{e} \qquad V(a_j) = V(S_{j,1}) = V(a_{j,1})$$

onde $S_{j,1} = a_{j,1}$ é referido como o *intervalo estacionário*. Note que ambos os métodos resultam corretamente no mesmo intervalo médio $E(a_j) = E(a_{j,1}) = 1 / \lambda_j$, mas podem resultar em variâncias muito diferentes. Se o processo de chegada for de renovação, então $a_{j,1}$, $a_{j,2}$, ..., $a_{j,p}$ são *iid*, resultando em $Cov(a_{j,q}, q_{j,r}) = 0$ para todo $q \neq r$ e, desta maneira, os dois métodos produzem a mesma variância $V(a_j) = V(a_{j,1})$. Uma maneira de detectar dependência entre os intervalos de tempo entre chegadas é verificar se existe diferença significativa entre as aproximações fornecidas pelos dois métodos.

Pode ser mostrado (Whitt, 1982, 1995) que: (i) o método assintótico (macro) é assintoticamente correto com $\rho_j \to 1$ (intensidade de tráfego pesado), e (ii) o método dos intervalos estacionários (micro) é assintoticamente correto com o número de processos de chegada na estação *j* tendendo a infinito, quando a superposição destes processos tende ao processo de Poisson.

Seja ca_{ij} o *scv* do intervalo de tempo entre chegadas na estação *j* a partir da estação *i*. Pelo método assintótico, pode ser mostrado que a superposição ca_j (ou $V(a_j)$) é uma combinação convexa dos ca_{ij} ($V(a_{ij})$) chegando da estação *i*, dada por (Sevcik *et al.*, 1977):

$$ca_j = \frac{\lambda_{0j}}{\lambda_j} ca_{0j} + \sum_{i=1}^{n} \frac{\lambda_{ij}}{\lambda_j} ca_{ij} = \sum_{i=0}^{n} \frac{\lambda_{ij}}{\lambda_j} ca_{ij} \qquad (3.27)$$

onde λ_{ij} e λ_j são obtidos conforme (3.6) e (3.4), respectivamente. Note que se o processo de chegada em cada estação for Poisson (i.e., $ca_{ij} = 1$, $i = 0, ..., n$), então a equação (3.27) é exata e produz $ca_j = 1$. No entanto, (3.27) não reflete a convergência para o processo de Poisson a medida que o número de processos de chegada na estação *j*, procedentes de outras estações, é grande (note que para isso cada processo de chegada não precisa ser Poisson; a superposição deles é que tende a ser Poisson, se eles forem suficientemente numerosos). Pelo método dos intervalos estacionários, ca_j resulta numa função não linear (Kuehn, 1979) que se deteriora com $\rho_j \to 1$ (Whitt, 1982, 1983b).

Albin (1982, 1984) sugeriu uma aproximação mais refinada para ca_j com um erro relativo em torno de 3% em comparação com simulação. Esta aproximação é baseada na combinação convexa entre o valor obtido em (3.27) e o valor obtido pelo método dos intervalos estacionários. Whitt (1983b) simplificou o refinamento de Albin, substituindo o método dos intervalos estacionários pelo processo de Poisson, e obteve:

$$ca_j = w_j \sum_{i=0}^{n} \frac{\lambda_{ij}}{\lambda_j} ca_{ij} + 1 - w_j \qquad (3.28)$$

onde: $w_j = \dfrac{1}{1 + 4(1-\rho_j)^2 (w_j' - 1)}$ e $w_j' = \dfrac{1}{\sum_{i=0}^{n} (\lambda_{ij} / \lambda_j)^2}$.

A aproximação (3.28) produz resultados muito próximos dos da aproximação híbrida de Albin (Albin, 1982).

Processo de partidas

No processo de partida (figura 15b), a taxa média de chegada superposta λ_j e o *scv* do intervalo de tempo entre chegadas superposto ca_j na estação *j*, juntos com o *scv* do tempo de serviço cs_j, são utilizados para determinar a taxa média de partida superposta e o *scv* do intervalo de tempo entre partidas superposto da estação *j*. Se a estação *j* não está saturada (i.e., $\rho_j < 1$) e está em equilíbrio, então a taxa média de partida é igual a taxa média de chegada. Porém, a determinação do *scv* do intervalo de tempo entre partidas envolve maiores dificuldades.

Seja cd_j o *scv* do intervalo de tempo entre partidas da estação *j*. Conforme mostrado no anexo 1, pelo método dos intervalos estacionários obtém-se (Kuehn, 1979, Buzacott e Shanthikumar, 1993):

$$cd_j = ca_j + 2\rho_j^2 cs_j - 2\rho_j (1-\rho_j) \mu_j E(Wq_j) \qquad (3.29)$$

onde $E(Wq_j)$ é o tempo médio de espera na fila da estação *j*. Conforme Whitt (1983a), ao se substituir em (3.29) a fórmula de Kraemer e Lagenbach-Belz para $E(Wq_j)$, com $g(\lambda_j, ca_j, cs_j) = 1$ (veja equação (3.34) abaixo), obtém-se uma aproximação razoável para cd_j, dada por:

$$cd_j = \rho_j^2 cs_j + (1-\rho_j^2) ca_j \qquad (3.30)$$

onde ρ_j é obtido conforme (3.5). Note que (3.30) é uma combinação convexa de ca_j e cs_j, e que se os processos de chegada e de serviço forem Poisson (i.e., $ca_j = cs_j = 1$), ela é exata e produz $cd_j = 1$ (lembre-se que o processo de partida de filas *M/M/1* ou *M/M/m*$_j$ com capacidades de fila infinitas é Poisson; veja p.e. Ross (1993) ou Kleinrock (1975)). Note também que se $\rho_j \to 1$, então $cd_j \to cs_j$, sugerindo que o *scv* do intervalo de tempo entre partidas tende ao *scv* do tempo de serviço quando a utilização média da estação *j* é muito alta (ou seja, longas filas na estação *j* tendem a diminuir o efeito do *scv* do intervalo de tempo entre chegadas). Por outro lado, se $\rho_j \to 0$, então $cd_j \to ca_j$, sugerindo que o *scv* do intervalo de tempo entre partidas tende ao *scv* do intervalo de tempo entre chegadas quando a utilização média é muito baixa, e filas não são esperadas na estação *j*. Pode ser mostrado que a expressão (3.30) é sempre uma subestimativa de cd_j (veja o anexo 2).

O método assintótico, entretanto, resulta numa aproximação mais elementar para cd_j, dada apenas por:

$$cd_j = ca_j \qquad (3.31)$$

A aproximação (3.31) também é exata se os processos de chegada e serviço forem Poisson. Além disto, torna-se mais precisa na estação *j* à medida que a utilização média cresce nas estações subseqüentes à estação *j*. Por exemplo, considere uma rede composta de duas estações 1 e 2 em série, com parâmetros $\{\lambda_{01}, ca_{01}, \mu_1, cs_1\}$ e $\{0, 0, \mu_2, cs_2\}$, respectivamente, e com $q_{12} = 1$ e $q_{11} = q_{22} = q_{21} = 0$ (figura 16). Usando (3.4) obtém-se $\lambda_1 = \lambda_2 = \lambda_{01}$. Além disto, se $\mu_2 \rightarrow \lambda_2$ e μ_1 for constante, então obtém-se $\rho_2 \rightarrow 1$. A partir dos teoremas de limite de tráfego pesado, Whitt (1983a) observou que as medidas de desempenho na estação 2 são assintoticamente as mesmas medidas obtidas ao se remover a estação 1 (i.e., se $1/\mu_1 = 0$). Em outras palavras, o processo de chegada da estação 2 é o mesmo processo de chegada da estação 1. Sob essas condições, (3.31) é assintoticamente correta para a estação 1, produzindo $ca_1 = cd_1 = ca_2$, enquanto ocorre na estação 2 o fenômeno do *gargalo de tráfego pesado.*

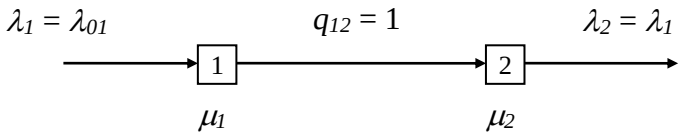


Figura 16 - Rede com estações 1 e 2 em série do exemplo de Whitt (1983a)

Um possível refinamento seria combinar as aproximações dos dois métodos acima, assim como foi feito no processo de superposição. Whitt (1983a) observou, no entanto, que este refinamento não é tão crítico como para o caso de superposição, e sugeriu apenas o uso de (3.30). Mais tarde, Suresh e Whitt (1990) observaram que o fenômeno do gargalo de tráfego pesado pode ocorrer na prática em níveis razoáveis da utilização média. Experimentos com várias estações em série e diversos parâmetros revelaram limitações no uso das aproximações (3.30) e (3.31) separadamente. Suresh e Whitt sugeriram que poderia ser apropriado considerar aproximações híbridas também para o processo de partida, combinando o método dos intervalos estacionários e o método assintótico. Eles observaram que o tempo médio de espera $E(Wq_j)$ na estação *j* não reflete o fenômeno de tráfego pesado porque o parâmetro ca_j é suposto totalmente independente de ρ_j (veja (3.34) abaixo). Assim, eles sugeriram que ca_j deveria ser uma função de $ca_1, cs_1, cs_2, ..., cs_{j-1}$ e $\rho_1, \rho_2, ..., \rho_j$. Por exemplo, ca_j poderia ser uma combinação convexa de $ca_1, cs_1, cs_2, ..., cs_{j-1}$, com pesos definidos por funções contínuas de $\rho_1, \rho_2, ..., \rho_j$.

Mais recentemente, Whitt (1995) propôs *funções de variabilidade* $ca_j(\rho_j)$, em função da utilização média na estação *j*, para substituir ca_j nas equações de variabilidade de tráfego. Se o processo de chegada na estação *j* for de renovação, então a função $ca_j(\rho_j)$ retorna corretamente o *scv* deste processo. Por outro lado, se o processo de chegada não for próximo de um processo de renovação, então $ca_j(\rho_j)$ retorna um parâmetro de variabilidade mais apropriado (do que o *scv* de um suposto processo de renovação), que leva em conta a utilização média na estação *j*. Embora interessante, esta alternativa não será explorada nesta tese.

Separação da partida

No processo de separação (figura 15c), a taxa média de partida superposta e o *scv* do intervalo de tempo entre partidas superposto da estação *j* são decompostos, produzindo as taxas médias λ_{ji} conforme (3.6). O *scv* de d_{ji}, o intervalo de tempo entre partidas de *jobs* da estação *j* para a estação *i*, é definido a seguir, em função de cd_j (Sevcik *et al.*, 1977, Kuehn, 1979).

Seja z_{ji} o número de *jobs* que partem da estação *j* durante um intervalo de tempo d_{ji}. Note que d_{ji} pode ser descrito como a soma de z_{ji} + 1 variáveis aleatórias d_j *iid*:

$$d_{ij} = d_{j,1} + d_{j,2} + \ ... \ + d_{j,z_{ij}} + d_{j,z_{ij}+1}$$

onde $d_{j,1}, d_{j,2}, ..., d_{j,z_{ij}}$ são os intervalos de tempo entre partidas das z_{ji} primeiras partidas da estação *j*, todas seguindo para estações diferentes de *i*, e $d_{j,z_{ij}+1}$ é o intervalo de tempo entre a z_{ji}-ésima e a (z_{ji}+1)-ésima partida da estação *j*, esta última seguindo para a estação *i*. Por conveniência, defina $z'_{ji} = z_{ji} + 1$. Note que z'_{ji} tem distribuição geométrica com probabilidade q_{ji}, logo, $E(z'_{ji}) = 1 / q_{ji}$ e $V(z'_{ji}) = (1 - q_{ji}) / q_{ji}^2$. Dado que a média e a variância da soma de z'_{ji} variáveis aleatórias *iid* são dadas por:

$$E(d_{ji}) = E(z'_{ji})E(d_j) = \frac{1}{q_{ji}} E(d_j)$$

$$V(d_{ji}) = E(z'_{ji})V(d_j) + V(z'_{ji})E(d_j)^2 = \frac{1}{q_{ji}} V(d_j) + \frac{(1-q_{ji})}{q_{ji}^2} E(d_j)^2$$

e que $cd_{ji} = V(d_{ji}) / E(d_{ji})^2$, segue que:

$$cd_{ji} = q_{ji} cd_j + (1 - q_{ji}) \qquad (3.32)$$

Se o processo de partida for Poisson (i.e., cd_j = 1), então (3.32) é exata e retorna cd_{ji} = 1. Note que se $q_{ji} \to 1$, então (3.32) resulta em $cd_{ji} \to cd_j$. Ou seja, à medida que a taxa média de partida da estação *j* para a estação *i* tende a taxa média de partida superposta da estação *j*, o *scv* do intervalo de tempo entre partidas da estação *j* para a estação *i* também tende ao *scv* do intervalo de tempo entre partidas superposto da estação *j*. Além disso, se $q_{ji} \to 0$, então (3.32) resulta em $cd_{ji} \to 1$, indicando que se a proporção do fluxo entre as estações *j* e *i* tende a zero, então o processo de partida da estação *j* para *i* tende a um processo de Poisson. Note também que cd_{ji} em (3.32) é igual a ca_{ji} em (3.28) para *i*, *j* = 1, ..., *n*, isto é, o *scv* do intervalo de tempo entre partidas da estação *j* para a estação *i* é exatamente o mesmo *scv* do intervalo de tempo entre chegadas na estação *i* a partir da estação *j*. Admitindo que o processo de partida seja um processo de renovação e que q_{ij}, *i* = 1, ..., *n*, representem eventos independentes (roteiro Markoviano), então (3.32) é exata e os métodos dos intervalos estacionários e assintótico coincidem.

Combinando as equações (3.28), (3.30) e (3.32), obtém-se o segundo sistema linear em função das variáveis ca_j, cd_j, e ca_{ij} (ou cd_{ij}). Este sistema, conhecido como *equações de variabilidade de tráfego*, fornece uma boa aproximação para ca_j, *j* = 1, ..., *n*. Note que as soluções das equações de taxa de tráfego e variabilidade de tráfego permite descrever cada estação *j* através dos parâmetros desejados $\{\lambda_j, ca_j, \mu_j, cs_j\}$. Pode-se depois proceder aos passos 2 e 3. Se a rede for acíclica (i.e., o roteiro dos *jobs* não forma ciclos), então as estações 1, 2, ..., *n* podem ser renumeradas como j_1, j_2, ..., j_n, tal que os *jobs* visitam a estação j_i após a estação j_k para $j_i > j_k$. Uma vez que não há ciclos, os parâmetros λ_j e ca_j podem ser facilmente computados para cada estação *j* seguindo as estações na ordem crescente.

Passos 2 e 3

O passo 1 decompôs a *OQN* num conjunto de estações individuais, cada uma descrita por $\{\lambda_j, ca_j, \mu_j, cs_j\}$. No passo 2 deseja-se avaliar as medidas de desempenho para cada estação, como por exemplo, o tempo médio de espera em fila, o comprimento médio da fila, etc. Estas medidas podem ser aproximadas pela teoria de filas (e.g. Kleinrock, 1975, Tijms, 1986). Whitt (1983a) observou que, dado que o processo de chegada real geralmente não é um processo de renovação, e que apenas dois parâmetros (média e *scv*) são conhecidos para cada distribuição, ganha-se pouco ao utilizar procedimentos mais elaborados do que este.

Para ilustrar, o número médio na fila *GI/G/1* da estação *j* pode ser estimado pela fórmula de Kraemer e Lagenbach-Belz (modificada por Whitt (1983a)), dada por:

$$E(L_j) = \frac{\rho_j^2 (ca_j + cs_j) g(\rho_j, ca_j, cs_j)}{2(1-\rho_j)} + \frac{\lambda_j}{\mu_j} \tag{3.33}$$

onde: $g(\rho_j, ca_j, cs_j) = \begin{cases} \exp\left\{\dfrac{-2(1-\rho_j)(1-ca_j)^2}{3\rho_j(ca_j + cs_j)}\right\}, & se \quad ca_j < 1 \\ 1, & se \quad ca_j \geq 1 \end{cases}$

e o tempo médio de espera na fila da estação *j*, por (conforme (3.13a)-(3.13c)):

$$E(Wq_j) = \frac{\rho_j (ca_j + cs_j) g(\rho_j, ca_j, cs_j)}{2\mu_j (1-\rho_j)} \tag{3.34}$$

As variâncias de L_j e Wq_j podem ser aproximadas por (veja o anexo 2 para detalhes de como elas foram obtidas):

$$V(L_j) = \lambda_j E(Wq_j) + \rho_j + \rho_j^2 cs_j + \lambda_j^2 V(Wq_j) \tag{3.35a}$$

$$V(Wq_j) = E(Wq_j)^2 c_{Wq_j} = E(Wq_j)^2 \frac{c_{D_j} + 1 - P(Wq_j > 0)}{P(Wq_j > 0)} \tag{3.35b}$$

onde: $c_{D_j} = 2\rho_j - 1 + \dfrac{4(1-\rho_j)E(s_j^3)}{3(cs_j+1)^2 E(s_j)^3}$ (3.35c)

$$P(Wq_j > 0) = \rho_j + (ca_j - 1)\rho_j(1-\rho_j)h(\rho_j, ca_j, cs_j)$$

$$h(\rho_j, ca_j, cs_j) = \begin{cases} \dfrac{1 + ca_j + \rho_j cs_j}{1 + \rho_j(cs_j - 1) + \rho_j^2(4ca_j + cs_j)}, & se \quad ca_j < 1 \\ \dfrac{4\rho_j}{ca_j + \rho_j^2(4ca_j + cs_j)}, & se \quad ca_j \geq 1 \end{cases}$$

Para sistemas *M/G/1* ($ca_j = 1$), as expressões (3.33)-(3.35) são exatas e resultam nas equações de Pollaczek-Khinchine em (A2.9) e (A2.10), conforme mostrado no anexo 2. Em particular, para sistemas *M/M/1*, as expressões (3.33)-(3.35) resultam corretamente em (3.16). Note que, neste

caso, (3.35c) se reduz a $c_{D_j} = 1$, dado que $E(s_j^{\,3}) = 6E(s_j)^3$ quando o tempo de serviço s_j é exponencialmente distribuído. Para continuar usando aproximações baseadas apenas nos dois primeiros momentos das distribuições, o termo $E(s_j^{\,3}) / E(s_j)^3$ pode ser aproximado por (Whitt, 1983a):

$$\frac{E(s_j^3)}{E(s_j)^3} \approx \begin{cases} 3cs_j(1+cs_j), & \text{se } cs_j \geq 1 \\ (2cs_j+1)(cs_j+1), & \text{se } cs_j < 1 \end{cases}$$

utilizando as distribuições hiperexponencial de ordem 2 (para $cs_j \geq 1$) e Erlang de ordem k (para $cs_j < 1$) para o tempo de serviço s_j. Esta substituição produz bons resultados, como mostram os experimentos computacionais em Whitt (1983a, 1983b). O anexo 2 apresenta outras aproximações para $E(L_j)$ e $E(Wq_j)$, além de (3.33) e (3.34), e compara a precisão entre elas num experimento computacional. Os resultados obtidos com (3.33) e (3.34) foram muito precisos (erros menores que 2%).

Finalmente, no passo 3 deseja-se avaliar medidas de desempenho para a rede, por exemplo, o número médio de *jobs*, o *leadtime* médio de um *job*, a taxa de produção da rede, etc. O número médio de *jobs* na rede é simplesmente $\sum_{j=1}^{n} E(L_j)$ e, admitindo-se que L_1, L_2, ..., L_n sejam independentes, a variância do número de *jobs* na rede é aproximada por $\sum_{j=1}^{n} V(L_j)$, conforme seção 3.1.1. O *leadtime* médio de um *job* arbitrário, incluindo o tempo de espera e o tempo de serviço gasto na rede, é dado por (similarmente a (3.17)):

$$E(T) = \sum_{j=1}^{n} E(V_j)(E(Wq_j) + E(s_j)) \tag{3.36}$$

onde $E(V_j)$ é o número médio de visitas na estação j definido por (3.8). Assim como nas redes de Jackson, a variância do *leadtime* é usualmente aproximada ignorando-se a correlação dos tempos de espera nas várias estações. Maiores detalhes sobre os passos 2 e 3 podem ser encontrados em Whitt (1983a, 1983b) e em Suri *et al.* (1993).

3.2.2 Rede de filas *GI/G/m* de classe única

O modelo anterior considera apenas estações com uma única máquina. O caso mais geral, em que se pode ter várias máquinas paralelas e idênticas em cada estação, deriva do anterior. Seja m_j, $m_j \geq 1$, o número de máquinas disponíveis na estação j, agora definida pelos 5 parâmetros $\{m_j, \lambda_{0j}, ca_{0j}, \mu_j, cs_j\}$. No passo 1, a equação (3.30) é substituída por (Whitt, 1983a):

$$cd_j = 1 + (1-\rho_j^2)(ca_j - 1) + \frac{\rho_j^2}{\sqrt{m_j}}(cs_j - 1) \tag{3.37}$$

Note que se $m_j = 1$, então (3.37) se reduz a (3.30). Além disso, para sistemas $M/M/m_j$ ($ca_j = 1$, $cs_j = 1$) e $M/G/\infty$ ($ca_j = 1$, $m_j \to \infty$), (3.37) resulta corretamente no processo de Poisson (i.e., $cd_j = 1$). Para um sistema *M/D/1* ($ca_j = 1$, $cs_j = 0$), no entanto, (3.37) e (3.30) produzem incorretamente o *scv* do intervalo de tempo entre partidas menor do que o *scv* do intervalo de tempo entre chegadas (i.e., $cd_j = 1 - \rho_j < 1$). De fato, Shanthikumar e Buzacott (1981) não encontraram bons

resultados (relativos à simulação) ao aplicar (3.30) para sistemas *M/D/1* e *GI/D/1*. Para reduzir essa distorção, Whitt (1983a) sugeriu modificar (3.37) para:

$$cd_j = 1 + (1-\rho_j^2)(ca_j - 1) + \frac{\rho_j^2}{\sqrt{m_j}}(max\{cs_j, 0{,}2\} - 1) \tag{3.38}$$

Finalmente, combinando as equações (3.28), (3.38) e (3.32) obtém-se o seguinte sistema linear em ca_j:

$$ca_j = \alpha_j + \sum_{i=1}^{n} \beta_{ij} ca_i \qquad \text{para } j = 1, ..., n \tag{3.39}$$

onde: $\alpha_j = 1 + w_j \left\{ p_{0j} ca_{0j} - 1 + \sum_{i=1}^{n} p_{ij} \left[(1-q_{ij}) + q_{ij} \rho_i^2 x_i \right] \right\}$ e $\beta_{ij} = w_j p_{ij} q_{ij} (1-\rho_i^2)$

com w_j definido conforme em (3.28) e:

$$p_{ij} = \frac{\lambda_{ij}}{\lambda_j}, \qquad q_{ij} = \frac{\lambda_{ij}}{\lambda_i}, \qquad x_i = 1 + \frac{\max\{cs_i, 0{,}2\} - 1}{\sqrt{m_j}}$$

Os passos 2 e 3 são similares aos da seção anterior, usando as fórmulas das medidas de desempenho derivadas da teoria de filas *GI/G/m*. Por exemplo, o número médio de *jobs* na estação *j* pode ser aproximado por:

$$E(L_j) = \frac{\lambda_j (ca_j + cs_j)}{2} E(Lq_j)_{M/M/m_j} + \frac{\lambda_j}{\mu_j} \tag{3.40}$$

onde $E(Lq_j)_{M/M/m_j}$ denota o número médio de *jobs* em fila num sistema $M/M/m_j$, correspondendo ao primeiro termo do lado direito de (3.12). O tempo médio de espera na estação *j* pode ser aproximado por (usando (3.13a)-(3.13c)):

$$E(Wq_j) = \frac{(ca_j + cs_j)}{2} E(Wq_j)_{M/M/m_j} \tag{3.41}$$

onde $E(Wq_j)_{M/M/m_j}$ é o tempo médio de espera para uma fila $M/M/m_j$ definido em (3.14). Note que se os processos de chegada e serviço forem Poisson, então (3.40) e (3.41) se reduzem a (3.12) e (3.14) respectivamente. Além disso, se $m_j = 1$ e $ca_j \geq 1$, então (3.40) e (3.41) se reduzem a (3.33) e (3.34). Outras aproximações para $E(L_j)$ e $E(Wq_j)$ podem ser encontradas em Buzacott e Shanthikumar (1993), em particular,

$$E(L_j) = \frac{E(Lq_j)_{M/M/m_j}}{E(Lq_j)_{M/M/1}} E(Lq_j)_{GI/G/1} + \frac{\lambda_j}{\mu_j}$$

$$E(Wq_j) = \frac{E(Wq_j)_{M/M/m_j}}{E(Wq_j)_{M/M/1}} E(Wq_j)_{GI/G/1}$$

onde $E(Lq_j)_{GI/G/1}$ e $E(Wq_j)_{GI/G/1}$ correspondem, por exemplo, à expressão (3.33) sem a carga ofertada, e à expressão (3.34), respectivamente. As variâncias de L_j e Wq_j podem ser aproximadas por (Whitt, 1983a):

$$V(L_j) = E(L_j)^2 c_{L_j(M/M/m_j)} \qquad \text{e} \qquad V(Wq_j) = E(Wq_j)^2 c_{Wq_j(M/M/m_j)} \tag{3.42}$$

onde $c_{L_j(M/M/m_j)}$ e $c_{Wq_j(M/M/m_j)}$ correspondem aos *scv* de L_j e Wq_j definidos em (3.15) para sistemas $M/M/m_j$. As aproximações (3.40)-(3.42) podem ser refinadas, conforme discutido no anexo 3, no entanto, para o propósito das decisões envolvidas nos modelos dos próximos capítulos, estas aproximações são suficientemente precisas (erros em geral menores que 10%). O *leadtime* médio $E(T)$ de um *job* pode ser definido similarmente a (3.36).

3.2.3 Rede de filas *GI/G/m* de múltiplas classes

Esta seção modifica o modelo anterior (seções 3.2.1 e 3.2.2) para analisar *OQN* com múltiplas classes. Inicialmente considera-se que todas as classes têm roteiros determinísticos. Seja a seguinte notação para os dados de entrada (compare com a da seção 3.1.1):

- n número de estações internas na rede,
- r número de classes na rede.

Para cada estação $j = 1, 2, ..., n$

- m_j número de máquinas na estação j

Para cada classe $k = 1, 2, ..., r$:

- n_k número de operações no roteiro da classe k.
- $\lambda_k^{'}$ taxa média de chegada externa da classe k ($\lambda_k^{'} = 1 / E(a_k^{'})$)
- $ca_k^{'}$ *scv* ou parâmetro de variabilidade do intervalo de tempo entre chegadas externas da classe k $ca_k^{'} = V(a_k^{'}) / E(a_k^{'})^2$)

Para cada classe $k = 1, 2, ..., r$, e para cada operação $l = 1, 2, ..., n_k$ do roteiro da classe k:

- n_{kl} estação visitada para a operação l do roteiro da classe k,
- $E(s_{kl})$ tempo médio de serviço da operação l do roteiro da classe k,
- ou μ_{kl} taxa média de serviço da operação l do roteiro da classe k ($\mu_{kl} = 1 / E(s_{kl})$),
- cs_{kl} *scv* ou parâmetro de variabilidade do tempo de serviço da operação l do roteiro da classe k ($cs_{kl} = V(s_{kl}) / E(s_{kl})^2$).

Conforme observado na seção 3.1.1, o roteiro de cada classe k é descrito por meio de n_k e n_{kl} (ao invés da matriz $\boldsymbol{Q}$), e pode-se ter uma distribuição do tempo de serviço diferente para cada operação. Inspirado nos procedimentos de agregação de redes de Jackson, Whitt (1983a) apresentou um procedimento para agregar todas as classes numa única classe e utilizar o modelo de única classe discutido anteriormente (seções 3.2.1 e 3.2.2). Note que desta maneira a *OQN* original, com múltiplas classes, é reduzida a uma *OQN* com uma única classe *agregada*. Após esta classe agregada ter sido analisada, pode-se retornar à rede original e estimar as medidas de desempenho para cada classe individualmente. Este procedimento é descrito a seguir.

Primeiramente, obtém-se os parâmetros iniciais $\{m_j, \lambda_{0j}, ca_{0j}, \mu_j, cs_j\}$ da classe agregada em cada estação j, $j = 1, ..., n$ e, em seguida, utiliza-se o passo 1 da seção 3.2.1 para obter os parâmetros $\{m_j, \lambda_j, ca_j, \mu_j, cs_j\}$ da classe agregada. Os parâmetros iniciais λ_{0j}, $\boldsymbol{Q}$, $E(s_j)$ e μ_j, podem ser obtidos conforme seção 3.1.1 (veja expressões (3.18), (3.6), (3.20)). O *scv* do tempo de serviço agregado na estação j é estimado usando (3.20) e a propriedade de que o segundo momento da soma de distribuições independentes é igual à soma dos respectivos segundos momentos, ou seja:

$$E(s_j) = \frac{\sum_{k=1}^{r}\sum_{l=1}^{n_k} E(s_{kl}^2)\lambda_k^{'} 1\{n_{kl} = j\}}{\lambda_j}$$

Dado que $V(s_j) = E(s_j^2) - E(s_j)^2$, segue que:

$$cs_j = \frac{V(s_j)}{E(s_j)^2} = \frac{E(s_j^2)}{E(s_j)^2} - 1 = \frac{\sum_{k=1}^{r}\sum_{l=1}^{n_k} E(s_{kl}^2)\lambda_k^{'} 1\{n_{kl} = j\}}{\lambda_j E(s_j)^2} - 1$$

e usando (3.19a) e $V(s_{kl}) = E(s_{kl}^2) - E(s_{kl})^2$, obtém-se:

$$cs_j = \frac{\sum_{k=1}^{r}\sum_{l=1}^{n_k} E(s_{kl})^2 (cs_{kl} + 1)\lambda_k^{'} 1\{n_{kl} = j\}}{\sum_{k=1}^{r}\sum_{l=1}^{n_k} \lambda_k^{'} 1\{n_{kl} = j\} E(s_j)^2} - 1 \qquad (3.43)$$

Finalmente, pode-se utilizar o método híbrido (veja (3.28)) para obter o *scv* do intervalo de tempo entre chegadas externas agregado na estação *j*. Através de (3.18) e superpondo todos os *scv* dos intervalos de tempo entre chegadas externas na estação *j*, obtém-se:

$$ca_{0j} = w_j \sum_{k=1}^{r} \frac{\lambda_k^{'} 1\{n_{kl} = j\} ca_k^{'}}{\sum_{l=1}^{n_k} \lambda_k^{'} 1\{n_{k1} = j\}} + 1 - w_j \qquad (3.44)$$

onde: $w_j = \dfrac{1}{1 + 4(1-\rho_j)^2 (w_j^{'} - 1)}$ e $w_j^{'} = \dfrac{1}{\sum_{k=1}^{r} \left(\dfrac{\lambda_k^{'} 1\{n_{k1} = j\}}{\sum_{l=1}^{r} \lambda_k^{'} 1\{n_{l1} = j\}} \right)^2}$

As expressões (3.18), (3.6), (3.20), (3.43) e (3.44) produzem os parâmetros iniciais $\{m_j, \lambda_{0j}, ca_{0j}, \mu_j, cs_j\}$ para cada estação *j* e a matriz $\boldsymbol{Q}$. Desta maneira tem-se todos os dados de entrada para uma *OQN* de classe única da seção 3.2.2. O passo 1 descreve cada estação por $\{m_j, \lambda_j, ca_j, \mu_j, cs_j\}$ após resolver (3.4) (ou (3.19a)) e (3.39). Os passos 2 e 3 da seção 3.2.1 obtêm as medidas de desempenho da classe agregada para cada estação e para a rede, respectivamente. Pode-se então retornar à rede original para estimar as medidas de desempenho para cada classe individualmente, a partir da classe agregada. Por exemplo, o *leadtime* médio $E(T_k)$ para um *job* da classe *k* pode ser estimado por (3.21), onde $E(Wq_{n_{kl}})$ é aproximado por (3.41). Similarmente, a variância do *leadtime* $V(T_k)$ para um *job* da classe *k* é aproximada por:

$$V(T_k) = \sum_{l=1}^{n_k} V(Wq_{n_{kl}}) + \sum_{l=1}^{n_k} V(s_{kl}) = \sum_{l=1}^{n_k} V(Wq_{n_{kl}}) + \sum_{l=1}^{n_k} E(s_{kl})^2 cs_{kl} \qquad (3.45)$$

onde as duas somatórias correspondem às variâncias do tempo de espera e de serviço na rede, respectivamente. Cada $V(Wq_{n_{kl}})$ em (3.45) pode ser aproximado conforme (3.42).

Extensão para roteiros probabilísticos e criação e combinação de *jobs*

No caso de se ter classes com roteiros probabilísticos, o procedimento de agregação pode ser estendido, conforme foi feito com as redes de Jackson na seção 3.1.1. Para ilustrar, tome novamente o exemplo da rede com $r = 4$ classes (com roteiros probabilísticos) e $n = 4$ estações da seção 3.1.1, e considere o caso especial em que os tempos de processamento dados na tabela 3 são determinísticos (i.e., $cs_j = 0$, $j = 1, ..., n$). Note que, desta maneira, tem-se agora uma rede de Jackson generalizada. Lembre-se da figura 12 que cada classe k percorre um roteiro definido pela matriz de transição $\{ q_{ij}^k , i, j = 1, ..., n\}$. Para computar a matriz *agregada* de transição $\boldsymbol{Q}$, faz-se $q_{ij} = \lambda_{ij} / \lambda_i$ conforme (3.6), onde $\lambda_i = \sum_{k=1}^{r} \lambda_i^k$ e $\lambda_{ij} = \sum_{k=1}^{r} \lambda_{ij}^k$, $\lambda_{ij}^k = q_{ij}^k \lambda_i^k$, com cada λ_i^k dado na tabela 3. O roteiro probabilístico agregado descrito por $\boldsymbol{Q}$ aparece ilustrado na figura 17.

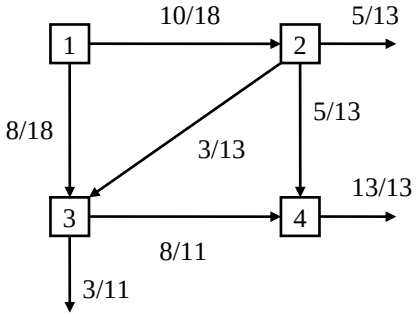


Figura 17 - Roteiro probabilístico agregado descrito pela matriz *Q* do exemplo de Askin e Standridge (1993)

Também pode-se computar λ_{0j} e ca_{0j} conforme (3.18) e (3.44). Com os parâmetros $\{m_j, \lambda_j, ca_{0j}, \mu_j, cs_j = 0\}$ e a matriz $\boldsymbol{Q}$ computados, pode-se estimar ca_j com (3.39). A tabela 4 a seguir apresenta estes valores e os de $E(L_j)$ e $E(Wq_j)$ obtidos por (3.40) e (3.41) para cada estação j. Observe que o número médio de *jobs* na rede é 8,685, ou seja, menos de 10% de desvio em relação ao valor médio 9,412 obtido por meio de simulação (para detalhes de como a simulação foi realizada, veja a seção 3.3). Ao comparar as tabelas 3 e 4, nota-se que ao eliminar a variabilidade nos tempos de processamento em cada estação, os atrasos médios $E(Wq_j)$ caem cerca de 50%.

Tabela 4 - Valores obtidos para cada estação j no exemplo de Askin e Standridge (1993) com os *scv* $cs_j = 0$

j	m_j	$E(s_j)$	μ_j	λ_{0j}	λ_j	ca_{0j}	ca_j	$E(L_j)$	$E(Wq_j)$
1	3	0,144	6,925	18	18	1	1	5,06	0,137
2	3	0,154	6,502	3	13	1	0,881	2,39	0,030
3	1	0,032	31,427	0	11	1	0,939	0,44	0,008
4	1	0,040	24,765	0	13	1	0,940	0,80	0,021
$\sum$				21				8,68	

As aproximações também podem ser estendidas para permitir criação e combinação de *jobs* nas estações, conforme discussão na seção 3.1.1. Lembre-se que, no caso de haver criação de *jobs* na estação j, basta substituir cada partida por um lote de tamanho γ_j, $\gamma_j > 1$. Similarmente, quando há combinação de *jobs* na estação j, fazemos $\gamma_j < 1$. As equações de taxa de tráfego devem ser

alteradas conforme antes, dado que a taxa média de partidas na estação *j* agora é $\gamma_j \lambda_j$. Aplicando-se o método assintótico, o *scv* do processo de partidas para criação ou combinação de *jobs* na estação *j* pode ser aproximado simplesmente por $\gamma_j\ cd_j$ (Whitt, 1983a). Procedimentos mais refinados para tratar chegadas e processamento de *jobs* em lotes podem ser encontrados em Bitran e Tirupati (1989c).

Também pode-se incorporar nos modelos o sistema de movimentação de materiais, da maneira como foi discutido na seção 3.1.1.

Interferência entre as classes

Bitran e Tirupati (1988) mostraram que a expressão (3.32) pode ser menos efetiva para o processo de separação quando tem-se *OQN* com muitas classes e roteiros determinísticos. Note em (3.32) que se $q_{ji} \rightarrow 0$, então $cd_{ji} \rightarrow 1$. Bitran e Tirupati estenderam (3.32) de maneira a representar o efeito de *interferência entre as classes*. Para cada par: classe *k* e operação *l* em uma estação, a análise é reduzida a duas classes:

(i) a classe de interesse (*k*,*l*)

(ii) a agregação de todas as outras classes (*k'*,*l'*), (*k'*,*l'*) ≠ (*k*,*l*), que chegam entre duas chegadas sucessivas da classe de interesse (*k*,*l*). Essa segunda classe é chamada *classe agregada* (não confundir com a classe agregada da seção anterior).

Seja (*k*,*l*) a classe de interesse numa certa estação n_{kl} em uma *OQN* com múltiplas classes e roteiros determinísticos. Admita que os intervalos de tempo entre chegadas e entre partidas de todas as classes na estação n_{kl} sejam *iid*. Uma vez que só há roteiros determinísticos na rede, pode-se facilmente obter $\lambda_{n_{kl}}$ por meio de (3.19a). Também é fácil obter a proporção da classe de interesse (*k*,*l*) na estação n_{kl}, $q_{kl} = \lambda_k^{'} / \lambda_{n_{kl}}$ .Seja d_{kl} o intervalo de tempo entre partidas da classe (*k*,*l*) da estação n_{kl}, $d_{n_{kl}}$ o intervalo de tempo entre partidas de todas as classes da estação n_{kl}, e z_{kl} o número de *jobs* da classe agregada que chega (ou que sai) na estação n_{kl} durante um intervalo de tempo entre chegadas (entre partidas) da classe (*k*,*l*). Note que d_{kl} é a soma de z_{kl} + 1 variáveis aleatórias *iid*. Defina $z_{kl}^{'} = z_{kl} + 1$.

Dado que a média e a variância da soma de $z_{kl}^{'}$ variáveis aleatórias *iid* são, respectivamente, $E(d_{kl}) = E(z_{kl}^{'})E(d_{n_{kl}})$ e $V(d_{kl}) = E(z_{kl}^{'})V(d_{n_{kl}}) + V(z_{kl}^{'})E(d_{n_{kl}})^2$, onde $E(d_{kl}) = 1/\lambda_k^{'}$ e $E(d_{n_{kl}}) = 1/\lambda_{n_{kl}}$, segue que $E(z_{kl}^{'}) = \lambda_{n_{kl}} / \lambda_k^{'}$ e cd_{kl} é dado por (compare com (3.32)):

$$cd_{kl} = q_{kl} cd_{n_{kl}} + cz_{kl}^{'} \qquad (3.46)$$

Bitran e Tirupati (1988) observaram que o primeiro termo do lado direito de (3.46) reflete o efeito do processo de fila na estação n_{kl}, enquanto que o segundo termo não depende do processo de serviço. Ele captura o efeito do processo de chegada da classe agregada durante o intervalo de tempo entre duas chegadas sucessivas da classe (*k*,*l*). Bitran e Tirupati propuseram duas aproximações para cz'_{kl} baseadas na hipótese de que z_{kl} tenha distribuição de Poisson e Erlang, respectivamente. Assumindo-se que z_{kl} tem distribuição de Poisson com taxa $(1-q_{kl})\lambda_{n_{kl}}$, segue que:

$$P(z_{kl} = z) = \int_0^\infty P(z_{kl} = z \mid a_{kl} = a) f_{a_{kl}}(a) da$$

onde $f_{a_{kl}}(a)$ é a função densidade de probabilidade de a_{kl}, o intervalo de tempo entre chegadas da classe de interesse (k,l) na estação n_{kl} (lembre-se que $E(a_{kl}) = 1/\lambda_k^{'}$). Pode ser mostrado que (Bitran e Tirupati, 1988):

$$E(z_{kl}) = \sum_{z=0}^{\infty} zP(z_{kl} = z) = \frac{1-q_{kl}}{q_{kl}}$$

$$E(z_{kl}^2) = \sum_{z=0}^{\infty} z^2 P(z_{kl} = z) = \frac{1-q_{kl}}{q_{kl}} + (1+ca_{kl})\left(\frac{1-q_{kl}}{q_{kl}}\right)^2$$

e portanto, $V(z_{kl}) = E(z_{kl}^{'}) - E(z_{kl})^2 = \frac{1-q_{kl}}{q_{kl}} + ca_{kl}\left(\frac{1-q_{kl}}{q_{kl}}\right)^2$. Como $z_{kl}^{'} = z_{kl} + 1$, segue que: $E(z_{kl}^{'}) = E(z_{kl}) + 1 = 1/q_{kl}$, $V(z_{kl}^{'}) = V(z_{kl})$, e $cz_{kl}^{'} = (1-q_{kl})[q_{kl} + (1-q_{kl})ca_{kl}]$. A expressão (3.46) (processo de separação) pode então ser rescrita como:

$$cd_{kl} = q_{kl}cd_{n_{kl}} + (1-q_{kl})q_{kl} + (1-q_{kl})^2 ca_{kl} \qquad (3.47)$$

Note que $ca_{kl} = cd_{k,l-1}$. Também pode-se rescrever a expressão (3.28) (processo de superposição) em função de ca_{kl}:

$$ca_j = w_j \sum_{k=1}^{r}\sum_{l=1}^{n_k} \frac{\lambda_k^{'}}{\lambda_j} ca_{kl} 1\{n_{kl} = j\} + 1 - w_j \qquad (3.48)$$

onde: $w_j = \frac{1}{1 + 4(1-\rho_j)^2(w_j^{'} - 1)}$ e $w_j^{'} = \frac{1}{\sum_{k=1}^{r}\sum_{l=1}^{n_k}\left(\lambda_k^{'}/\lambda_j\right)^2 1\{n_{kl} = j\}}$

e λ_j é obtido por meio de (3.19a). Combinando (3.48), (3.38) e (3.47) obtém-se um sistema linear alternativo, em função de ca_j, cd_j, e cd_{kl} (ou $ca_{k,l+1}$), para determinar ca_j. Após obter os parâmetros $\{m_j, \lambda_j, ca_j, \mu_j, cs_j\}$ do passo 1, prossegue-se para os passos 2 e 3 da mesma maneira que antes. Esta aproximação baseada em (3.47) produz estimativas para ca_j muito melhores do que (3.39), que é baseada em (3.32) (veja os resultados computacionais em Bitran e Tirupati (1988, 1989b); veja também os resultados da seção 3.3 a seguir).

De fato, a expressão (3.47) pode ser vista como uma generalização de (3.32). Para ver isto, considere uma situação particular na qual *jobs* da classe de interesse *k* entram na rede pela estação *j*, esperam em fila junto com *jobs* de outras classes e, após receberem serviço, somente os *jobs* da classe *k* prosseguem para uma certa estação *i*. Assim, a taxa média de partida do arco (j,i) é $\lambda_{ji} = \lambda_k^{'}$, e a proporção (ou probabilidade) de *jobs* que fluem da estação *j* para a estação *i* é $q_{ji} = \lambda_k^{'}/\lambda_j$. Seguindo os mesmos passos anteriores, pode-se definir d_{ji}, $z_{ji}, z_{ji}^{'}$ e assim por diante, e rescrever (3.47) como: $cd_{ji} = q_{ji}cd_j + cz_{ji}^{'}$. Admitindo-se que z_{ji} tenha distribuição de Poisson com taxa $(1-q_{ji})\lambda_j$, segue que $cz_{ji}^{'} = (1-q_{ji})[q_{ji} + (1-q_{ji})ca_k^{'}]$, e (3.47) pode ser rescrito como:

$$cd_{ji} = q_{ji}cd_j + (1-q_{ji})q_{ji} + (1-q_{ji})^2 ca_k^{'} \qquad (3.49)$$

Note que se o processo de chegada da classe *k* for Poisson (i.e., $ca'_k = 1$), então (3.49) se reduz a (3.32). De fato, é possível mostrar que (3.32) é o caso especial de (3.49) quando z'_{ji} é geometricamente distribuído com parâmetro q_{ji}, resultando em: $cz'_{ji} = 1 - q_{ji}$. Note também que se $q_{ji} \to 1$, então (3.32) e (3.49) levam a $cd_{ji} \to cd_j$ mas, se $q_{ji} \to 0$, então apenas (3.49) leva a $cd_{ji} \to ca'_k$. Este último resultado é assintoticamente exato (Bitran e Tirupati, 1988) e permitiu duas importantes aproximações para *OQN* com múltiplas classes e roteiros determinísticos, apresentadas em Whitt (1988). Inicialmente, considere uma certa estação da rede (Whitt, 1988, p.1335):

> *"Se a contribuição da taxa de chegada de uma certa classe numa certa visita nesta estação for uma pequena proporção da taxa total de chegada nesta estação, então o processo de partida desta classe nesta visita tende a ser quase igual ao processo de chegada desta classe nesta visita."*

Whitt observou que este princípio pode ser visto como uma aproximação de tráfego leve, onde apenas a classe precisa ter baixa utilização (i.e., a utilização total da estação não precisa ser baixa). Considere agora uma certa classe com roteiro determinístico na rede (Whitt, 1988, p.1335):

> *"Se a contribuição da taxa de chegada desta classe em cada visita em cada estação da rede for uma pequena proporção da taxa total de chegada da estação, então o processo de partida desta classe a partir de cada visita de cada estação, e portanto da rede inteira, é quase igual ao processo de chegada externo desta classe na rede."*

Baseando-se em (3.49) e nas aproximações acima, Segal e Whitt (1989) apresentaram uma expressão alternativa para o processo de separação de *OQN* com múltiplas classes e roteiros determinísticos. Seja ce_j a média entre os *scv* do intervalo de tempo entre chegadas externas das classes na estação *j*, ponderada pelo número médio de visitas de cada classe na estação *j*. Tem-se que:

$$ce_j = \frac{\sum_{k=1}^{r} \sum_{l=1}^{n_k} \lambda'_k 1\{n_{kl} = j\} ca'_k}{\sum_{k=1}^{r} \sum_{l=1}^{n_k} \lambda'_k 1\{n_{kl} = j\}} \tag{3.50}$$

Usando (3.50), o *scv* cd_{ji} entre as estações *j* e *i* é redefinido por (compare com (3.32)):

$$cd_{ji} = q_{ji} cd_j + (1 - q_{ji}) q_{ji} ca_j + (1 - q_{ji})^2 ce_j \tag{3.51}$$

Note que para $ce_j = 1$ e $ca_j = 1$, a aproximação (3.51) é equivalente a (3.32). Segal e Whitt (1989) sugeriram substituir (3.32) por (3.51) se todas as classes tiverem roteiros puramente determinísticos. No caso de também se ter classes com roteiros probabilísticos, eles sugeriram o uso de uma combinação convexa de (3.32) e (3.51), para capturar melhor o efeito dos roteiros probabilísticos. Note que ao substituir (3.32) por (3.51), deve-se modificar o sistema linear (3.39) por (3.28), (3.38) e (3.51). Os passos 2 e 3 são seguidos conforme anteriormente. Entretanto, nenhuma experiência computacional foi encontrada na literatura comparando o desempenho desta aproximação, baseada em (3.51), com a anterior, baseada em (3.47). Isto está além do escopo desta tese e é um tópico para pesquisa futura.

Mais tarde, Whitt (1994) propôs uma extensão de (3.47) definida como:

$$cd_{kl} = q_{kl} cd_{n_{kl}} + (1 - q_{kl}) q_{kl} ca'_{kl} + (1 - q_{kl})^2 ca_{kl} \qquad (3.52)$$

onde ca'_{kl} é o *scv* do intervalo de tempo entre chegadas da agregação de todas as classes que chegam entre duas chegadas sucessivas da classe *k* para a operação *l* na estação n_{kl}. Whitt (1994) apresentou resultados computacionais sugerindo que (3.52) é mais efetiva para o processo de separação do que (3.47). Note que (3.47) pode ser vista como um caso especial de (3.52) ao se assumir que o processo de chegada da classe agregada é Poisson (i.e., ca'_{kl} = 1). Whitt (1994) também propôs outras aproximações para o processo de separação, sob a hipótese de que a máquina esteja continuamente ocupada, que não serão aqui discutidas.

Conforme mostrado até agora, os métodos de decomposição são flexíveis e não é difícil estendê-los para representar situações mais complexas, incluindo processamento em lotes e horas extras (Bitran e Tirupati, 1989c, 1991), falha de máquinas, mudanças do tamanho de lotes, teste de produtos com reparo (Segal e Whitt, 1989, Kouvelis e Tirupati, 1991), outras disciplinas além de *FCFS*, como *SPT*, e espaço limitado para a formação de fila de *jobs* (Buzacott e Shanthikumar, 1993), também podem ser incorporadas a estes métodos, com pequenas modificações.

O potencial de aplicações práticas motivou o desenvolvimento de diversos pacotes de *software* baseados nos métodos de decomposição, tais como o *QNA* (*Queueing Network Analyser*) (Whitt, 1983a, 1983b; Segal e Whitt, 1989), *Q-LOTS* (Karmarkar *et al.*, 1985), *ManuPlan* (Suri *et al.*, 1986, Brown, 1988), *QNAP - Queueing Network Analysis Package* (Pujolle e Ai, 1986), *Operations Planner* (Jackman e Johnson, 1993), e *MPX* (Suri e De Treville, 1991, Suri *et al.*, 1995). Em particular, Suri e De Treville (1993) e De Treville (1992) têm apontado as vantagens de se utilizar estes métodos para responder as questões levantadas pelos paradigmas recentes em manufatura de redução de *leadtimes* e *WIP*. Note que as aproximações apresentadas neste capítulo permitem analisar grandes e complexas redes de manufatura com pouco esforço computacional, dado que as aproximações são baseadas apenas na solução de sistemas lineares. Uma lista de aplicações em grandes empresas e estudos de caso pode ser encontrada em Suri *et al.* (1993).

Entretanto, a questão da qualidade destas aproximações levantada por diversos autores, como Kouvelis e Tirupati (1991), deveria ser ainda melhor respondida. Diversas suposições foram feitas nas aproximações apresentadas neste capítulo, tais como na agregação das classes de produtos em uma única classe, na estimativa das medidas de desempenho da classe agregada e, depois, na estimativa das medidas de desempenho para cada classe individual. Além destas suposições, o próximo capítulo considera suposições adicionais para simplificar os modelos de otimização, tais como sobre o comportamento dos parâmetros de variabilidade e das funções de *WIP* e custo de capacidade, ao se variar a configuração da rede, e sobre o uso de algoritmos heurísticos de solução. Tomadas isoladamente, todas estas suposições parecem razoáveis e isso é comprovado por diversos resultados teóricos e computacionais. Entretanto, os efeitos acumulados destas suposições na qualidade das aproximações ainda são analisados na literatura apenas por limitados resultados computacionais.

3.3 Resultados computacionais

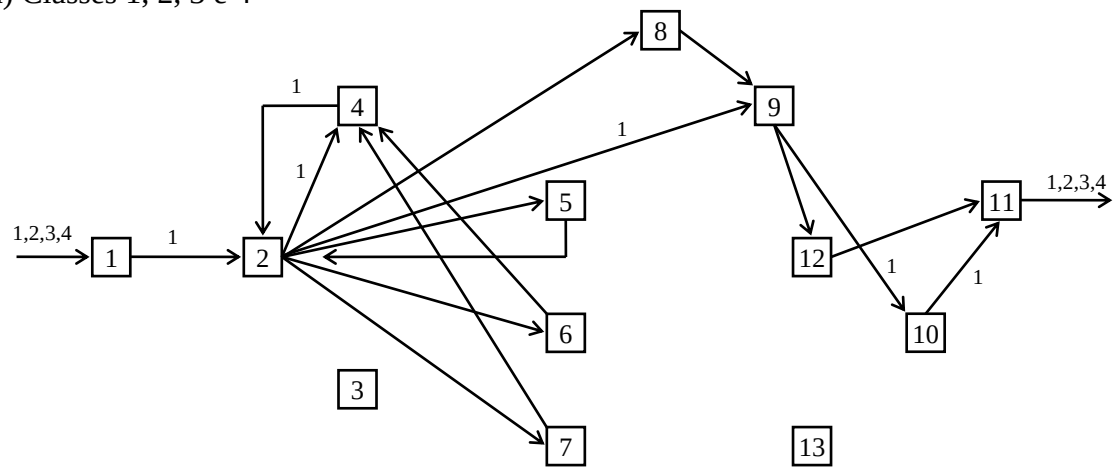
A título de ilustração, a seguir são apresentados alguns resultados computacionais tomando-se como exemplo um caso real de uma fábrica de semicondutores. Trata-se de um sistema *job-shop* analisado em Bitran e Tirupati (1988) e Bitran e Morabito (1995a), com *r* = 10 classes de

produtos e n = 13 estações de trabalho, sem limitações de capacidade de fila. Os roteiros das classes são determinísticos e os *jobs* visitam a estação de fotolitografia (estação 2) mais de uma vez. A tabela 5 apresenta, para cada classe k, k = 1, ..., 10, a distribuição do intervalo de tempo entre chegadas, a taxa média de chegada externa ($\lambda_k^{'}$), o *scv* do intervalo de tempo entre chegadas externas ($ca_k^{'}$), o número (n_k) e a seqüência (n_{kl}) de estações no roteiro determinístico da classe k (a figura 18 ilustra o roteiro de cada classe k). Alguns dados foram alterados a pedido da fábrica. Note que a taxa média de produção da rede é λ_0 = 1 produto por unidade de tempo.

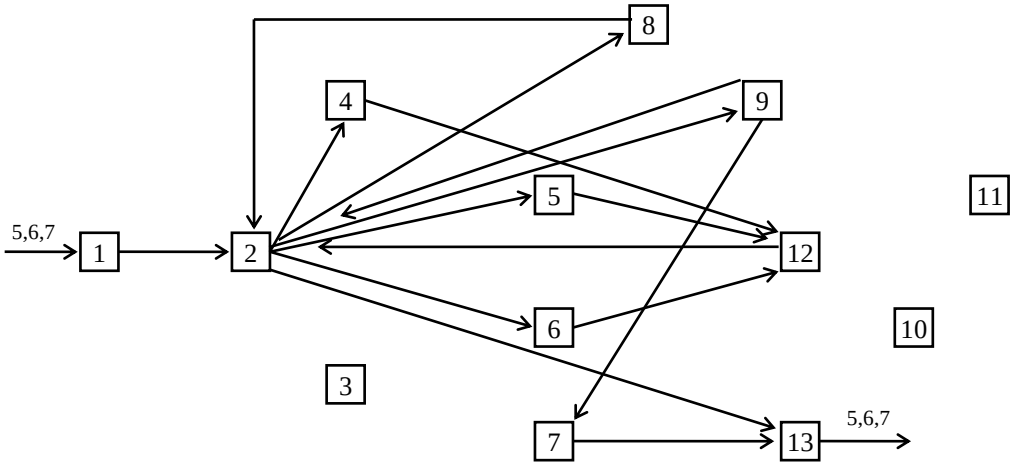
Tabela 5 - Dados de entrada das classes de produto no exemplo de Bitran e Tirupati (1988)

Classe k	Distribuição	$\lambda_k^{'}$	$ca_k^{'}$	n_k	n_{kl}
1	Erlang 3	0,1	0,333	7	1, 2, 4, 2, 9, 10, 11
2	Erlang 2	0,1	0,500	8	1, 2, 5, 2, 8, 9, 10, 11
3	uniforme	0,1	0,333	8	1, 2, 6, 4, 2, 9, 12, 11
4	Erlang 3	0,1	0,333	8	1, 2, 7, 4, 2, 9, 10, 11
5	Erlang 4	0,1	0,250	8	1, 2, 4, 12, 2, 9, 2, 13
6	Erlang 2	0,1	0,500	8	1, 2, 5, 12, 2, 9, 2, 13
7	Erlang 4	0,1	0,250	8	1, 2, 6, 12, 2, 8, 2, 13
8	uniforme	0,1	0,333	12	1, 2, 3, 7, 4, 12, 2, 8, 6, 9, 2, 13
9	Erlang 4	0,1	0,250	13	1, 2, 3, 5, 4, 6, 12, 2, 8, 2, 10, 6, 13
10	Erlang 2	0,1	0,500	13	1, 2, 3, 6, 2, 4, 12, 7, 2, 9, 11, 5, 13
Total:		1,0			

(a) Classes 1, 2, 3 e 4



(b) Classes 5, 6 e 7



(c) Classes 8, 9 e 10

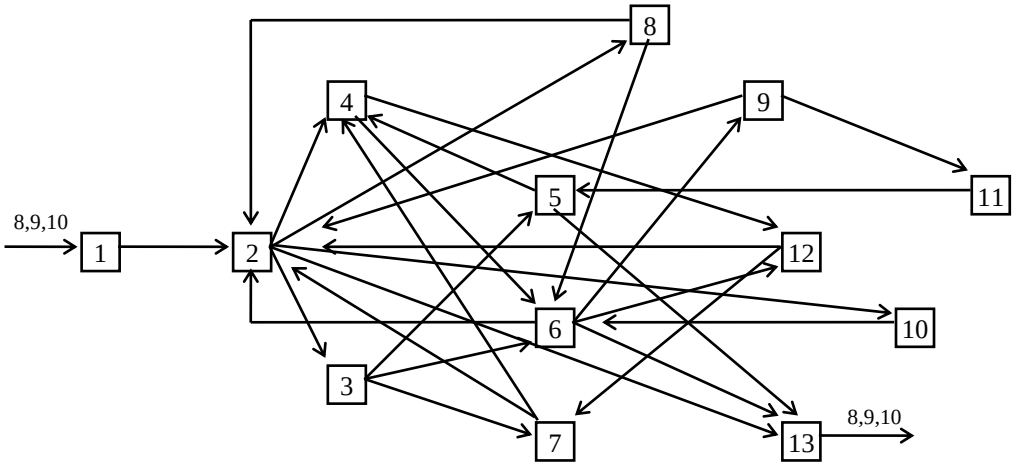


Figura 18 - Roteiros das 10 classes do exemplo de Bitran e Tirupati (1988)

A tabela 6 apresenta, para cada estação j = 1, 2, ..., 13, a distribuição do tempo de processamento, o tempo médio de processamento $E(s_j)$, o *scv* do tempo de processamento cs_j, e a taxa média de processamento ou capacidade $\mu_j = 1 / E(s_j)$. Por simplicidade, admite-se que cada estação tem

apenas uma máquina. A tabela 6 também apresenta a taxa média de chegada λ_j, e a utilização média $\rho_j = \lambda_j / \mu_j$ de cada estação j. A taxa média de chegada é obtida através da expressão (3.19a) e a utilização média, através de (3.5).

Tabela 6 - Dados de entrada e parâmetros das estações no exemplo de Bitran e Tirupati (1988)

Estação j	Distribuição	$E(s_j)$	cs_j	μ_j	λ_j	ρ_j
1	uniforme	0,780	0,333	1,282	1,0	0,78
2	uniforme	0,348	0,333	2,874	2,5	0,87
3	Erlang 2	2,667	0,500	0,375	0,3	0,80
4	exponencial	1,057	1,000	0,946	0,7	0,74
5	Erlang 3	2,000	0,333	0,500	0,4	0,80
6	Erlang 4	1,400	0,250	0,714	0,6	0,84
7	uniforme	1,775	0,333	0,563	0,4	0,71
8	uniforme	1,875	0,333	0,533	0,4	0,75
9	Erlang 2	1,175	0,500	0,851	0,8	0,94
10	Erlang 3	1,800	0,333	0,556	0,4	0,72
11	exponencial	1,440	1,000	0,694	0,5	0,72
12	Erlang 4	1,157	0,250	0,864	0,7	0,81
13	uniforme	1,450	0,333	0,690	0,6	0,87

A seção 3.2 apresentou diversas aproximações para o cálculo do *scv* do intervalo de tempo entre chegadas de cada estação j, ca_j. A tabela 7 compara o número médio de *jobs* (fila e serviço) em cada estação j, $E(L_j)$, obtido ao se utilizar tais aproximações de ca_j na fórmula de Kraemer e Lagenbach-Belz descrita em (3.33). Lembre-se da seção 3.2 que para estimar ca_j por meio do método de decomposição, não é preciso conhecer as distribuições de $a_k^{'}$ e s_j (apresentadas na segunda coluna das tabelas 5 e 6), mas apenas as médias e os *scv* de $a_k^{'}$ e s_j. Tais distribuições aparecem nas tabelas porque foram usadas no modelo de simulação discutido adiante.

As colunas $E(L_j)^1$, $E(L_j)^2$ e $E(L_j)^3$ da tabela 7 apresentam, respectivamente, os resultados obtidos com as aproximações de Whitt (1983a) (expressões (3.27), (3.38) e (3.32)), Bitran e Tirupati (1988) ((3.27), (3.38) e (3.47)) e Segal e Whitt (1989) ((3.27), (3.38) e (3.51)), ao se aproximar a superposição de chegadas pelo método assintótico expresso em (3.27). Por outro lado, as colunas $E(L_j)^4$, $E(L_j)^5$ e $E(L_j)^6$ apresentam, respectivamente, os resultados das aproximações de Whitt (expressões (3.28), (3.38) e (3.32), isto é, (3.39)), Bitran e Tirupati ((3.48), (3.38) e (3.47)), e Segal e Whitt ((3.28), (3.38) e (3.51)), ao se aproximar a superposição de chegadas pelo método híbrido expresso em (3.28) e (3.48). A penúltima linha da tabela (Total) apresenta o número médio de *jobs* na rede, obtido através de cada aproximação. Estes resultados foram obtidos em poucos segundos, utilizando-se um microcomputador Pentium 100 Mhz e a linguagem de modelagem *GAMS* (*General Algebraic Modeling System*, versão 2.25) (Brooke *et al.*, 1992), com o *solver GAMS/BDMLP*.

Tabela 7 - Número médio de *jobs* nas estações e na rede: A primeira coluna de resultados refere-se à simulação, as três colunas seguintes referem-se a aproximações da superposição de chegadas pelo método assintótico, e as três últimas referem-se a aproximações da superposição de chegadas pelo método híbrido

Est. j	$E(L_j)^0$	$E(L_j)^1$	$E(L_j)^2$	$E(L_j)^3$	$E(L_j)^4$	$E(L_j)^5$	$E(L_j)^6$
1	1,96	1,63	1,63	1,63	1,63	2,29	1,63

2	3,23	3,61	3,34	2,80	3,83	4,26	3,28
3	2,18	3,08	2,22	2,05	3,09	2,49	2,09
4	2,27	2,72	2,28	2,07	2,77	2,66	2,44
5	2,10	2,84	2,25	1,85	2,86	2,51	2,13
6	2,38	3,44	2,61	2,08	3,48	3,00	2,54
7	1,46	1,81	1,46	1,23	1,84	1,69	1,58
8	1,58	2,14	1,78	1,42	2,14	2,01	1,46
9	7,74	11,01	9,14	7,04	11,04	9,66	7,33
10	1,45	1,82	1,60	1,31	1,85	1,80	1,43
11	2,03	2,36	2,24	1,90	2,43	2,45	2,14
12	2,17	2,82	2,24	1,88	2,86	2,62	2,27
13	3,19	4,50	3,73	2,78	4,54	4,07	3,14
Total:	33,71	43,80	36,53	30,06	44,37	41,52	33,48
Desvio:	0%	29,92%	8,36%	-10,83%	31,62%	23,14%	-0,71%

(^{0}) simulação

(^{1,4}) aproximações de Whitt (1983a) utilizando respectivamente (3.27), (3.38) e (3.32), e (3.28), (3.38) e (3.32).

(^{2,5}) aproximações de Bitran e Tirupati (1988) utilizando respectivamente (3.27), (3.38) e (3.47), e (3.48), (3.38) e (3.47).

(^{3,6}) aproximações de Segal e Whitt (1989) utilizando respectivamente (3.27), (3.38) e (3.51), e (3.28), (3.38) e (3.51).

Para se comparar os resultados das aproximações acima, várias replicações independentes foram simuladas, cada uma com um total de 200000 *jobs* gerados. Os primeiros 100000 *jobs* de cada replicação foram descartados na tentativa de se evitar os efeitos dos estados transitórios. Utilizou-se o programa de simulação *GPSS/H* e seu gerador de números pseudo-aleatórios, baseado no algoritmo de Lehmer (Schriber, 1991). Para garantir a independência entre as replicações, todos os intervalos de tempo entre chegadas e tempos de processamento de *jobs* nas estações foram sorteados de posições diferentes da seqüência de 2^{31}-1 números gerada pelo algoritmo de Lehmer no *GPSS/H*. Dado que, para este exemplo, a hipótese das estimativas pontuais serem normalmente distribuídas parece razoável, devido aos processos aditivos envolvidos (veja, p.e., Schriber (1991)), optou-se por gerar uma pequena amostra de 10 replicações. Para garantir a uniformidade dos números sorteados, as replicações rejeitadas no teste de *chi-quadrado* foram sendo descartadas, até que 10 replicações fossem aceitas. Ao todo foram necessárias 12 replicações (2 replicações foram rejeitadas), que consumiram pouco mais de uma hora de execução no microcomputador. As médias do número médio de *jobs* das 10 replicações aceitas estão apresentadas para cada estação na coluna $E(L_j)^0$.

Note que em média foram obtidos 33,71 *jobs* na rede, com um pequeno desvio padrão de 0,63 e intervalo de 95% de confiança [33,27; 34,16] construído a partir da distribuição *t*-Student. A última linha da tabela 7 (Desvio) apresenta os desvios das diversas aproximações em relação ao número médio de *jobs* na rede da simulação. A aproximação (3.27), (3.38) e (3.47) de Bitran e Tirupati (1988) e a aproximação (3.28), (3.38) e (3.51) de Segal e Whitt (1989) resultaram em desvios menores que 10% (note que ambas as aproximações levam em conta o efeito da interferência entre as classes, discutido no final da seção 3.2.3). Convém salientar que esta comparação aqui é meramente ilustrativa. Conforme mencionado na seção anterior, uma comparação efetiva entre as aproximações da tabela 7 está além dos objetivos desta tese e é um tópico para pesquisa futura.

A tabela 8 apresenta os *leadtimes* médios em (3.21) obtidos para cada classe de produtos com a aproximação (3.27), (3.38) e (3.47) (esta aproximação é utilizada no capítulo 5 para gerar as

curvas de *trade-off*). A coluna $E(Wq_k)$ corresponde ao atraso médio da classe k, dado por $\sum_{l=1}^{n_k} E(Wq_{n_{kl}})$. Note na última coluna da tabela que, exceto para a classe 9, os produtos esperam na fila, em média, mais de 2/3 dos seus *leadtimes* de produção $E(T_k)$.

Tabela 8 - Resultados obtidos para cada classe com a aproximação (3.27), (3.38) e (3.47) de Bitran e Tirupati (1988)

Classe k	$E(T_k)$	$E(Wq_k)$	$E(Wq_k)/E(T_k)$
1	27,48	20,53	74,71%
2	34,30	24,53	71,52%
3	31,02	23,31	75,16%
4	31,14	22,41	71,98%
5	29,75	23,09	77,60%
6	34,44	25,40	73,77%
7	23,86	16,15	67,70%
8	49,61	35,23	71,01%
9	48,49	31,86	65,71%
10	55,25	39,31	71,14%

Finalmente, este exemplo foi analisado assumindo-se que todas as distribuições das tabelas 5 e 6 fossem exponenciais, isto é, como uma rede de Jackson com múltiplas classes e roteiros determinísticos. Para isso, fixou-se $ca_k^{'} = 1$, $k = 1, ..., 10$, e $cs_j = 1$, $j = 1, ..., 13$, na aproximação (3.27), (3.38) e (3.47) (lembre-se que ela é exata para redes de Jackson). O número médio de *jobs* obtido na rede foi de 63,55, correspondendo a um desvio de mais de 88% em relação à simulação! Conforme diversos autores já observaram, isto mostra que as soluções em forma de produto das redes de Jackson descritas na seção 3.1 não são boas aproximações para as redes de Jackson generalizadas.

4. Modelos de otimização

O capítulo 3 examinou modelos para avaliar medidas de desempenho de *OQN* representando sistemas discretos de manufatura, particularmente *job-shops*. O presente capítulo analisa modelos de otimização que utilizam os modelos do capítulo 3 para projetar ou reprojetar *job-shops*. Naturalmente, se o projeto envolve selecionar uma configuração dentre um pequeno número de alternativas, então pode-se aplicar os modelos do capítulo 3 para escolher a opção de melhor desempenho, caso contrário, necessita-se de modelos baseados em técnicas de otimização e programação matemática para encontrá-la. Bitran e Dasu (1992) classificaram os modelos de otimização de *OQN* em:
(i) *projeto ótimo*
(ii) *controle ótimo.*

Modelos de *projeto ótimo* determinam o projeto ótimo do sistema para uma dada regra operacional (p.e., a disciplina *FCFS*). Modelos de *controle ótimo* determinam a regra operacional ótima para uma dada configuração do sistema. Esta tese está particularmente interessada nos modelos de projeto ótimo. Para um exame dos modelos de controle ótimo baseados no movimento Browniano, veja Harrison e Nguyen (1993).

Os três problemas *SP1.1*, *SP2.1* e *SP3.1* apresentados no capítulo 1 são exemplos de problemas de projeto ótimo. Como foi discutido, diversas medidas de desempenho podem ser utilizadas, tais como *WIP*, *leadtimes*, e taxa média de produção. A seguir, os problemas *SP1.1* e *SP2.1* são formulados escolhendo o *WIP* como a medida de desempenho. Uma vez que o *WIP* e o *leadtime* são linearmente relacionados através da *lei de Little* (3.13c), os algoritmos descritos neste capítulo também se aplicam para o *leadtime*. Estudos similares utilizando a taxa de produção como a medida de desempenho podem ser encontrados em Bitran e Sarkar (1994a).

Por simplicidade, adota-se a notação $\overline{L}_j$ e $\overline{W}_j$, ao invés de $E(L_j)$ e $E(W_j)$ usada no capítulo anterior, para denotar respectivamente o número médio de *jobs* e o tempo médio de espera em fila e serviço na estação *j*. Se a rede de filas contiver múltiplas classes, estas medidas referem-se à classe agregada, conforme procedimentos de agregação apresentados no capítulo 3. Sejam ainda:

μ_j taxa média de serviço de cada máquina na estação *j*,
m_j número de máquinas idênticas na estação *j*,
$F_j(\mu_j, m_j)$ custo (ou investimento) de alocar a capacidade (μ_j, m_j) na estação *j*,
F_T orçamento disponível para a capacidade da rede,
$\overline{L}_j(\mu, \mathbf{m})$ o número médio de *jobs* na estação *j*, como uma função da capacidade $\boldsymbol{\mu} = (\mu_1, \mu_2, ..., \mu_n)$ e $\mathbf{m} = (m_1, m_2, ..., m_n)$ da rede,
v_j valor monetário médio de um *job* na estação *j*, independente de sua classe,
L_T limitante superior para o *WIP* da rede.

Lembre-se que o *WIP* $L(\boldsymbol{\mu},\mathbf{m})$ é um valor monetário médio do número médio de *jobs* na rede, definido como:

$$L(\mu,\mathbf{m}) = \sum_{j=1}^{n} v_j \bar{L}_j(\mu,\mathbf{m}) \qquad (4.1)$$

Cada valor monetário v_j, associado a um *job* na estação *j*, pode ser estimado usando-se experiência prática, ou como uma média ponderada proporcional à taxa média de chegada e ao tempo médio de espera de cada classe (o tempo médio de espera pode ser computado aproximadamente através de um procedimento proposto em Albin, 1986). Obviamente, se $v_j = 1$ para todo *j*, então o *WIP* corresponde ao número médio de *jobs* na rede.

O problema de *WIP* desejado *SP1.1* é o problema de determinar a capacidade (μ_j, m_j) para cada estação *j* de maneira a minimizar o custo total e satisfazer a restrição do nível desejado de *WIP* na rede, L_T. *SP1.1* é formulado como:

$$(SP1.1) \qquad \begin{array}{ll} \min & F(\mu,\mathbf{m}) = \sum_{j=1}^{n} F_j(\mu_j, m_j) \\ s.a. & L(\mu,\mathbf{m}) \le L_T \\ & (\mu_j, m_j) \in P_j,\ j = 1,\ldots,n \end{array}$$

onde P_j é um dado domínio das variáveis. Similarmente, o problema de *WIP* ótimo *SP2.1* é o problema de determinar a capacidade (μ_j, m_j) para cada estação *j* de maneira a minimizar o *WIP* da rede e satisfazer a restrição do orçamento disponível, F_T. *SP2.1* é formulado como:

$$(SP2.1) \qquad \begin{array}{ll} \min & L(\mu,\mathbf{m}) = \sum_{j=1}^{n} v_j \bar{L}_j(\mu,\mathbf{m}) \\ s.a. & \sum_{j=1}^{n} F_j(\mu_j, m_j) = F_T \\ & (\mu_j, m_j) \in P_j,\ j = 1,\ldots,n \end{array}$$

Diversos autores apresentaram métodos de solução para os dois problemas acima. A seguir, alguns destes procedimentos são revistos. Com a finalidade de apresentá-los de uma forma mais estruturada, adotou-se a notação sugerida em Bitran e Dasu (1992) que denota cada instância do problema por:

$$\alpha/\beta/\chi/\delta$$

onde $\alpha \in \{SP1.1, SP2.1, SP3.1\}$, $\beta \in \{J, G\}$, $\chi \in \{S, M\}$ e $\delta \in \{R, N\}$. O símbolo α indica o tipo de problema, β indica se o problema é aplicado a uma rede de Jackson (J) ou a uma rede genérica (G), χ indica se as estações têm apenas uma única máquina (S) ou múltiplas máquinas (M), e δ indica se a variável de decisão do modelo é taxa média de serviço (R) ou o número de máquinas (N) em cada estação. Por exemplo, a notação *SP2.1*/J/M/N indica um problema de *WIP* ótimo (*SP2.1*) aplicado a uma rede de Jackson (J), com múltiplas máquinas nas estações (M) que correspondem às variáveis de decisão (N).

Ambos os problemas *SP1.1* e *SP2.1* são considerados nas seções 4.1 e 4.2. A seção 4.1 revê modelos e métodos de solução para as *redes de Jackson* e a seção 4.2, para as *redes de Jackson generalizadas*. Conforme mencionado anteriormente, esta tese preocupa-se com modelos de otimização apenas em *OQN*. Exemplos de modelos de otimização em *CQN* podem ser encontrados, por exemplo, em Shanthikumar e Yao (1987, 1988), Dallery e Stecke (1990),

Vinod e Solberg (1991), Schweitzer e Seidmann (1991), Calabrese (1992), Askin e Krisht (1994), Stecke e Raman (1994) e Kouvelis e Lee (1995). Convém salientar que, ao contrário dos modelos de avaliação de desempenho, poucos autores apresentaram exames cuidadosos dos modelos de otimização em *OQN* e suas aplicações em *job-shops* e, portanto, a revisão a seguir pretende ser uma contribuição útil para a literatura.

4.1 Redes de Jackson (Modelos ./J/./.)

Conforme visto no capítulo 3, nas redes de Jackson pode-se decompor cada estação *j* como se fosse um sistema de fila $M/M/m_j$ "estocasticamente independente". Desta maneira, $\bar{L}_j$ em *SP1.1* e *SP2.1* torna-se uma função apenas de μ_j e m_j, ao invés de uma função de $\boldsymbol{\mu}$ e **m**.

4.1.1 Modelos ./J/./R

Kleinrock (1964, 1976) estudou o problema de minimizar o número médio de *jobs* em redes de Jackson com uma máquina em cada estação (i.e. $m_j = 1$). As variáveis de decisão são as taxas de serviço μ_j, $j = 1, ..., n$. Kleinrock assumiu que o custo F_j seja proporcional a μ_j (i.e., $F_j(\mu_j) = f_j \mu_j$, onde f_j é o custo unitário da capacidade na estação *j*). Note em (3.12) que para $m_j = 1$, obtém-se $\bar{L}_j(\mu_j) = \lambda_j / (\mu_j - \lambda_j)$, onde λ_j é computado conforme (3.4). Kleinrock formulou *SP2.1*/J/S/R como (compare com *SP2.1* da seção anterior):

$$(SP2.1/\text{J/S/R}) \qquad \begin{aligned} \min \quad & \sum_{j=1}^{n} v_j \bar{L}_j(\mu_j) \\ s.a. \quad & \sum_{j=1}^{n} f_j \mu_j = F_T \\ & \mu_j \geq 0, j = 1,\ldots,n \end{aligned}$$

Introduzindo o multiplicador de Lagrange *y* associado à restrição de igualdade em *SP2.1*/J/S/R, a função Lagrangeana resulta em:

$$\sum_{j=1}^{n} \bar{L}_j(\mu_j) + y\left(\sum_{j=1}^{n} f_j \mu_j - F_T\right)$$

Ao derivar esta função em relação à cada μ_j e *y*, a solução ótima de *SP2.1*/J/S/R é obtida em forma fechada, dada por:

$$\mu_j^* = \lambda_j + \frac{\sqrt{f_j \lambda_j}}{\sum_{i=1}^{n} \sqrt{f_i \lambda_i}} \frac{F_T - \sum_{i=1}^{n} f_i \lambda_i}{f_j} \qquad (4.2)$$

Note que se o custo unitário de capacidade for igual em todas as estações, diga-se $f_j = f$, então (4.2) se reduz a:

$$\mu_j^* = \lambda_j + \left(\frac{F_T - f\sum_{i=1}^{n} \lambda_i}{f\sum_{i=1}^{n} \sqrt{\lambda_i}}\right)\sqrt{\lambda_j} \qquad (4.2a)$$

e portanto, a capacidade ótima na estação *j*, μ_j^*, é composta de uma parcela para compensar a taxa média de chegada λ_j, mais uma parcela proporcional à raiz quadrada da taxa média de chegada, $\sqrt{\lambda_j}$.

Cinco condições são satisfeitas no modelo *SP2.1*/J/S/R (Bitran e Dasu, 1992):
(i) $\overline{L}_j(\mu_j)$ é uma função convexa de μ_j (o número médio de *jobs* na estação *j* é uma função convexa da capacidade da estação *j*),
(ii) $\overline{L}_j(\mu_j)$ não depende de μ_i, $i \neq j$, $i = 1, ..., n$ (adições de capacidade em outras estações não têm efeito no número médio de *jobs* da estação *j*),
(iii) μ_j é contínuo (as variáveis de decisão são variáveis contínuas),
(iv) $F_j(\mu_j)$ é uma função convexa de μ_j (o custo de capacidade da estação *j* é uma função convexa da capacidade da estação *j*),
(v) $\overline{L}_j(\mu_j)$ (ou o tempo médio de permanência $\overline{W}_j(\mu_j)$) pode ser expresso em forma fechada.

Estas condições são úteis para analisar os demais modelos deste capítulo. As condições (i)-(iv) reduzem o modelo *SP2.1*/J/S/R num programa convexo, que pode ser resolvido otimamente via métodos exatos de ótimo local (p.e., Bazaraa *et al.*, 1993). A condição (v) permite que a solução do problema seja em forma fechada.

Pode-se formular *SP2.1*/J/M/R exatamente como *SP2.1*/J/S/R, onde $\overline{L}_j(\mu_j)$ é redefinido para filas $M/M/m_j$ (compare com (3.14)). Harel e Zipkin (1987) mostraram que o tempo médio de permanência $\overline{W}_j(\mu_j)$ (e o número médio de *jobs* $\overline{L}_j(\mu_j)$) em um sistema $M/M/m_j$ também é uma função convexa em μ_j. Assim, as condições (i)-(iv) também são satisfeitas e *SP2.1*/J/M/R também pode ser reduzido a um programa convexo.

Os modelos *SP1.1*/J/S/R e *SP1.1*/J/M/R podem ser definidos e analisados de maneira similar. Em particular, Bretthauer (1996) analisou *SP1.1*/J/S/R com as variáveis μ_j, $j = 1, ..., n$, pertencendo a um conjunto finito de alternativas discretas, ao invés de serem definidas como variáveis contínuas, e com $F_j(\mu_j)$ definida como uma função côncava em μ_j (as condições (iii) e (iv) não são satisfeitas):

(*SP1.1*/J/S/R)

$$\begin{aligned} \textit{min} \quad & F(\mu) = \sum_{j=1}^{n} F_j(\mu_j) \\ & \sum_{j=1}^{n} v_j \overline{L}_j(\mu_j) \leq L_T \\ \textit{s.a.} \quad & \mu_j^L \leq \mu_j \leq \mu_j^U, \text{ com } \mu_j \text{ selecionado} \\ & \text{de um conjunto discreto de alternativas.} \end{aligned}$$

(μ^L_j e μ^U_j são limitantes de μ_j). Apesar de dificultar a solução do modelo, o uso de funções côncavas de custo de capacidade pode ser útil para tratar economia de escala e custos fixos. Para resolver *SP1.1*/J/S/R, Bretthauer (1996) apresentou uma algoritmo *branch-and-bound* com garantia de otimalidade, baseado na divisão da região de factibilidade em subconjuntos sucessivamente menores, usando subretângulos *n*-dimensionais:

$$\left\{\mu \in R \mid l_j \leq \mu_j \leq u_j, j = 1,\ldots,n\right\}, \quad \text{com } \mu_j^L \leq l_j \leq u_j \leq \mu_j^U$$

do retângulo original $R = \left\{ \mu \in \Re^n \mid \mu_j^L \le \mu_j \le \mu_j^U , j = 1,\ldots,n \right\}$. Procedimentos adicionais também foram apresentados para permitir que estes subproblemas fossem resolvidos mais eficientemente. Para maiores detalhes do método, o leitor é referido a Bretthauer (1996).

4.1.2 Modelos ./J/./N

A seguir discute-se os modelos *SP1.1*/J/M/N e *SP2.1*/J/M/N. Note que as variáveis de decisão são inteiras, correspondendo ao número de máquinas em cada estação. Boxma *et al.* (1990) apresentaram um algoritmo heurístico e outro exato para resolvê-los, respectivamente. A rede de manufatura é representada por uma *OQN* de Jackson com múltiplos servidores, múltiplas classes, e diferentes roteiros determinísticos para cada classe (veja seções 3.1.2 e 3.2.3).

Considere novamente os dados de entrada $\{m_j , j = 1, ..., n, \lambda_k^{'} , n_{kl}, \mu_{kl}, k = 1, ..., r; l = 1, ..., n_k\}$ descritos na seção 3.1.1. Ao agregar todas as classes em uma única classe, obtém-se cada estação j descrita pelos três parâmetros $\{m_j, \lambda_j, \mu_j\}$, conforme expressões (3.19a) e (3.20). Conforme mostrado em (3.9) na seção 3.1.1, a distribuição de equilíbrio do número de *jobs* na rede pode ser expressa na forma de produto, e cada estação j se comporta como um sistema $M/M/m_j$. O número médio de *jobs* na estação j, $\bar{L}_j$, é função de m_j, λ_j e μ_j, conforme (3.12).

Boxma *et al.* (1990) consideraram m_j, $j = 1, ..., n$, como variável inteira nos modelos *SP1.1*/J/M/N e *SP2.1*/J/M/N, e observaram que $\bar{L}_j(m_j)$ em (3.12) é uma função convexa e decrescente em m_j, independente de m_i, $i \neq j$, $i = 1, ..., n$ (as condições (i) e (ii) são satisfeitas). Eles escolheram o *WIP* como medida de desempenho da rede. Note que esta análise pode ser facilmente estendida para o *leadtime*, dado que o *WIP* e o *leadtime* são linearmente relacionados. O *WIP* da rede, definido em (4.1), pode ser rescrito por:

$$L(\mathbf{m}) = \sum_{j=1}^{n} v_j \bar{L}_j(m_j) \tag{4.3}$$

A escolha de m_j em cada estação deve satisfazer a condição $\rho_j < 1$ para evitar a instabilidade do sistema. Seja $\lfloor z \rfloor$ denotando o maior inteiro menor ou igual a z. Usando esta condição, segue que m_j deve ser um número inteiro maior ou igual ao limitante inferior m^0_j, definido por:

$$m_j^0 = \left\lfloor \frac{\lambda_j}{\mu_j} \right\rfloor + 1 \tag{4.4}$$

Modelo *SP1.1*/J/M/N

No modelo *SP1.1*/J/M/N deseja-se encontrar a solução de custo mínimo satisfazendo um nível de *WIP* menor ou igual a um limite especificado L_T, com $L_T < L(\mathbf{m}^0)$. Seja $F_j(m_j)$ o custo de alocar m_j máquinas na estação j, definido como uma função convexa não-decrescente de m_j (a condição (iv) é satisfeita). Utilizando (4.3) e (4.4), obtém-se o problema de *WIP* desejado (também chamado *problema de alocação de servidores*):

$$
\begin{array}{lll}
 & min & F(\mathbf{m}) = \sum_{j=1}^{n} F_j(m_j) \\
(SP1.1/\text{J/M/N}) & & L(\mathbf{m}) \le L_T \\
 & s.a. & m_j \ge m_j^0, \text{ inteiro, } j = 1,\ldots,n
\end{array}
$$

Note que, dado que *SP1.1*/J/M/N é um programa convexo com variáveis inteiras, o uso de análise marginal não leva necessariamente à otimalidade (a condição (iii) não é satisfeita). Seja $PI_j(m_j)$ um *índice de prioridade* definido como o quociente entre o aumento de custo e a redução de *WIP* na estação *j*, dado por:

$$PI_j(m_j) = \frac{\Delta F_j(m_j + 1)}{-v_j \Delta \bar{L}_j(m_j + 1)} \tag{4.5}$$

onde:
$$\Delta F_j(m_j + 1) = F_j(m_j + 1) - F_j(m_j) \ge 0$$
$$\Delta \bar{L}_j(m_j + 1) = \bar{L}_j(m_j + 1) - \bar{L}_j(m_j) < 0$$

PI_j é resultado da análise marginal de F_j e $\bar{L}_j$. Boxma *et al.* (1990) apresentaram um simples algoritmo heurístico (algoritmo 1) baseado no *método guloso* para resolver o problema *SP1.1*/J/M/N (veja também a abordagem em Sundarraj *et al.* (1994) para um problema similar). O algoritmo começa com a menor alocação de máquinas possível (4.4) para cada estação. Em cada iteração, ele adiciona uma máquina na estação com o mínimo índice de prioridade (4.5). O algoritmo termina assim que a adição de uma máquina resultar numa alocação factível.

Algoritmo 1

Passo 1: Comece com a alocação $m_j = m_j^0$, j = 1, ..., n. Esta solução é infactível ($L(\mathbf{m}^0) > L_T$) e seu custo $F(\mathbf{m}^0)$ é menor do que o custo mínimo de *SP1.1*/J/M/N.

Passo 2: Em cada iteração, atualize o custo $F(\mathbf{m})$, o *WIP* $L(\mathbf{m})$ (usando (4.3) e (3.12)) e $PI_j(m_j)$ (usando (4.5)). Adicione uma máquina na estação j^* que resultar no menor quociente PI_{j*} (estratégia gulosa), dado por:

$$PI_{j^*} = \min\left\{ PI_j(m_j), j = 1,\ldots,n \right\} \tag{4.6}$$

Passo 3: Pare assim que $L(\mathbf{m})$ atingir o limite L_T (solução factível).

Note que a estação j^* escolhida em (4.6) produz o menor aumento em $F(\mathbf{m})$ por unidade de redução em $L(\mathbf{m})$, indicado por PI_{j*}. Devido a convexidade de F_j e $\bar{L}_j$, tem-se que:

$$\frac{\Delta F_j(m_j + 1)}{-v_j \Delta \bar{L}_j(m_j + 1)} \ge \frac{\Delta F_j(m_j)}{-v_j \Delta \bar{L}_j(m_j)} \tag{4.7}$$

Um resultado interessante que decorre de (4.7) é que pode-se verificar a qualidade da solução heurística gerada pelo algoritmo 1, apenas comparando as soluções geradas nas duas últimas iterações. Seja p a última iteração, e $\mathbf{m}^1$, $\mathbf{m}^2$, ..., $\mathbf{m}^{p-1}$, $\mathbf{m}^p$ as soluções geradas em cada iteração.

Obviamente, $\mathbf{m}^{p-1}$ é infactível e $\mathbf{m}^p$ é factível. Denote por $\mathbf{m}^*$ a solução ótima de *SP1.1*/J/M/N. Boxma *et al.* (1990, teoremas 1 e 2) mostraram que:

$$F(\mathbf{m}^{p-1}) < F(\mathbf{m}^*) \leq F(\mathbf{m}^p)$$

e portanto, $F(\mathbf{m}^{p-1})$ e $F(\mathbf{m}^p)$ são limitantes para o valor da solução ótima. Experimentos computacionais utilizando duas redes de manufatura reais resultaram num erro relativo de 5% entre $F(\mathbf{m}^p)$ e $F(\mathbf{m}^{p-1})$. Estas experiências sugerem que o algoritmo 1 gera uma alocação suficientemente próxima da alocação ótima de *SP1.1*/J/M/N.

Mais tarde, Frenk *et al.* (1994) apresentaram dois algoritmos para resolver *SP1.1*/J/M/N, que podem ser vistos como extensões do algoritmo 1. Eles provaram que estes algoritmos têm desempenhos de pior caso iguais a 2 e 3/2, respectivamente. Resultados computacionais foram apresentados indicando que o erro relativo médio dos algoritmos foi substancialmente menor do que o erro relativo médio do algoritmo 1. Enquanto o algoritmo 1 tem complexidade de $O(Mn)$, a complexidade dos algoritmos melhorados é $O(Mn^2)$ e $O(Mn^3)$, respectivamente, onde M é o máximo número de máquinas entre as *alocações não-dominadas* (veja definição 1 em Frenk *et al.*, 1994).

Modelo *SP2.1*/J/M/N

No modelo *SP2.1*/J/M/N deseja-se alocar (ou realocar) máquinas de maneira a otimizar uma medida de desempenho, por exemplo, minimizar o *WIP* na rede. Admite-se que um total de M máquinas homogêneas deve ser alocado às estações, onde $M > \sum_{j=1}^{n} m_j^0$. Esta situação ocorre por exemplo no projeto de *FMS*, onde se pode ter máquinas iguais desempenhando operações diferentes ao se instalar ferramentas diferentes. Utilizando (4.3) e (4.4) obtém-se o problema de *WIP* ótimo (também chamado de *problema de realocação de servidores*):

$$(SP2.1/\text{J/M/N}) \qquad \begin{aligned} \min \quad & L(\mathbf{m}) = \sum_{j=1}^{n} v_j \bar{L}_j(m_j) \\ s.a. \quad & \sum_{j=1}^{n} m_j = M \\ & m_j \geq m_j^0, \text{inteiro}, \; j = 1, \ldots, n \end{aligned}$$

Novamente, tem-se um programa convexo com variáveis inteiras (as condições (i), (ii) e (iv) são satisfeitas mas a condição (iii) é violada), e o uso de análise marginal pode não produzir a solução ótima de *SP2.1*/J/M/N. Seja agora $PI_j(m_j)$ um índice de prioridade definido agora como a redução de *WIP* por unidade de máquina na estação j, dado por:

$$PI_j(m_j) = -v_j \Delta\bar{L}_j(m_j + 1) \qquad (4.8)$$

onde $\Delta\bar{L}_j(m_j + 1) = \bar{L}_j(m_j + 1) - \bar{L}_j(m_j) < 0$, conforme seção anterior.

Boxma *et al.* (1990) apresentaram um simples algoritmo (algoritmo 2), similar ao algoritmo 1, também baseado no método guloso para resolver *SP2.1*/J/M/N. O algoritmo começa com a menor alocação de máquinas possível (4.4) para cada estação. Em cada iteração ele adiciona uma máquina na estação com o máximo índice de prioridade (4.8). O algoritmo termina quando todas as M máquinas tiverem sido alocadas.

Algoritmo 2

Passo 1: Comece com a alocação $m_j = m_j^0$, $j = 1, ..., n$. Esta solução é infactível ($\sum_{j=1}^{n} m_j^0 < M$) e seu *WIP* $L(\mathbf{m}^0)$ é maior do que o *WIP* mínimo de *SP2.1*/J/M/N.

Passo 2: Em cada iteração, atualize o *WIP* $L(\mathbf{m})$ (usando (4.3) e (3.12)) e $PI_j(m_j)$ (usando (4.8)). Adicione uma máquina na estação j^* que resultar no maior PI_{j*} (estratégia gulosa), dado por:

$$PI_{j^*} = \max\left\{ PI_j(m_j), j = 1,\ldots,n \right\} \tag{4.9}$$

Passo 3: Pare assim que o número total de máquinas alocadas atingir o limite M (solução factível).

Note que a estação j^* escolhida em (4.9) produz a maior redução em $L(\mathbf{m})$ por unidade de máquina, indicada por PI_{j*}. Devido a convexidade de $\bar{L}_j$, tem-se que:

$$v_j \Delta\bar{L}_j(m_j + 1) \geq v_j \Delta\bar{L}(m_j) \tag{4.10}$$

Usando (4.10), Boxma *et al.* (1990, teorema 3) provaram que o algoritmo 2 é exato e termina com a solução ótima de *SP2.1*/J/M/N (apesar da condição (iii) não ser satisfeita). Além disto, esta solução é gerada num intervalo de tempo limitado por uma função polinomial no número de estações da rede, ou seja, em $O(Mn)$.

4.2 Redes de Jackson Generalizadas (Modelos ./G/./.)

Esta seção estuda os modelos *SP1.1*/G/S/R, *SP2.1*/G/S/R (Bitran e Tirupati, 1989a, Bitran e Sarkar, 1994a, e Wein, 1990a), *SP1.1*/G/M/R com variáveis discretas (Bitran e Tirupati, 1989b), e *SP1.1*/G/M/N e *SP2.1*/G/M/N (Van Vliet e Rinnooy Kan, 1991).

4.2.1 Modelos ./G/./R

Inicialmente, apresenta-se dois algoritmos introduzidos por Bitran e Tirupati (1989a) para os modelos *SP1.1*/G/S/R e *SP2.1*/G/S/R. Estes algoritmos podem ser facilmente estendidos para tratar os modelos *SP1.1*/G/M/R e *SP2.1*/G/M/R. A rede de manufatura é representada por uma *OQN* com múltiplas classes e roteiros determinísticos para cada classe (veja seção 3.2.3). Na seção anterior, medidas de desempenho como o *WIP* em (4.3) foram facilmente avaliadas devido aos resultados exatos das redes de Jackson. Na falta de métodos exatos para as redes de Jackson generalizadas, o método aproximado de decomposição discutido na seção 3.2.3 é utilizado para aproximar os parâmetros de variabilidade em cada estação.

Considere os dados de entrada $\{m_j, j = 1, ..., n, \lambda_k', ca_k', n_{kl}, \mu_{kl}, cs_{kl}, k = 1, ..., r; l = 1, ..., n_k\}$ descritos na seção 3.2.3. Por simplicidade, suponha que $m_j = 1$ para toda estação j (os algoritmos a seguir são facilmente estendidos se $m_j > 1$). Ao se agregar as classes conforme (3.18), (3.20), (3.43) e (3.44), obtém-se os parâmetros $\{\lambda_{0j}, ca_{0j}, \mu_j, cs_j\}$ para cada estação j. Em seguida, aplicando-se o passo 1 do método de decomposição, obtém-se o sistema de equações (3.19a), (3.27), (3.30) e (3.47), descritos a seguir simplesmente por:

$$\Phi(\lambda, \mathbf{ca}, \mu, \mathbf{cs}) = 0 \qquad (4.11)$$

onde os vetores $\boldsymbol{\lambda}$, **ca**, $\boldsymbol{\mu}$ e **cs** denotam $\{\lambda_j, ca_j, \mu_j, cs_j\}$ para todo j. Uma vez que $\bar{L}_j$ em (3.33) é uma função de λ_j, ca_j, μ_j e cs_j, e estes parâmetros estão relacionados conforme (4.11), segue que $\bar{L}_j$ é uma função de $\boldsymbol{\lambda}$, **ca**, $\boldsymbol{\mu}$ e **cs**. Bitran e Tirupati (1989a) consideraram cada capacidade μ_j, $j = 1, ..., n$, como variável de decisão contínua (portanto, a condição (iii) é satisfeita), assumindo que capacidade adicional possa ser incorporada na estação em pequenos incrementos, quando comparados com a capacidade total (lembre-se que foi suposto que cada estação tem uma única máquina). Para um dado $\boldsymbol{\lambda}$, (4.11) e (3.33) sugerem que mudanças na capacidade $\boldsymbol{\mu}$ resultem em mudanças em **ca** e **cs**. Assim, $\bar{L}_j$ é uma função de $\mu_1, \mu_2, ..., \mu_n$. Entretanto, esta relação funcional é complexa, dado que o sistema de equações em (4.11) é não-linear e não é fácil de ser analisado.

Bitran e Tirupati assumiram que:

(a) cada cs_j seja independente à pequenas mudanças na capacidade μ_j (i.e. a variância e o quadrado da média do tempo de processamento na estação j variam na mesma proporção e assim, cs_j se mantém aproximadamente constante), e

(b) $\bar{L}_j$ depende apenas de μ_j, ao invés de $\mu_1, \mu_2, ..., \mu_n$ (condição (ii) satisfeita).

A hipótese (b) parece se verificar a medida que o número de classes aumenta e a proporção da demanda devida por cada classe diminui (veja os resultados numéricos da seção 3.3 e em Bitran e Tirupati (1988), e a discussão em Whitt (1988)). A conseqüência das suposições (a) e (b) é que pode-se inicialmente resolver o sistema (4.11) para uma dada capacidade, e em seguida tratar os **ca** obtidos como parâmetros conhecidos em (3.33). Sob tais hipóteses, Bitran e Tirupati mostraram que $\bar{L}_j(\lambda_j, ca_j, \mu_j, cs_j)$ em (3.33) é uma função convexa em μ_j, denotada agora simplesmente por $\bar{L}_j(\mu_j)$ (condição (i) é satisfeita). Seja o vetor de variáveis. O *WIP* em (4.1) pode ser rescrito por:

$$L(\mu) = \sum_{j=1}^{n} v_j \bar{L}_j(\mu_j) \qquad (4.12)$$

Modelo *SP1.1*/G/S/R

Similarmente à seção 4.1.2, seja L_T uma meta para o nível de *WIP* da rede, tal que $L_T < L(\mu^0)$, onde $\boldsymbol{\mu}^0$ é a capacidade inicial na estação j, suficientemente pequena, satisfazendo:

$$\mu_j^0 > \lambda_j \qquad (4.13)$$

Seja também $F_j(\mu_j)$ o custo de alocar capacidade μ_j na estação j, definida como uma função convexa não-decrescente e diferenciável de μ_j (a condição (iv) é satisfeita). Utilizando (4.12), obtém-se o seguinte problema de programação convexa:

$$(SP1.1/\text{G/S/R}) \qquad \begin{array}{ll} \min & F(\mu) = \sum_{j=1}^{n} F_j(\mu_j) \\ s.a. & L(\mu) \le L_T \\ & \mu_j > \lambda_j,\ j = 1,\ldots,n \end{array}$$

Bitran e Tirupati (1989a) apresentaram um algoritmo heurístico (algoritmo 3) para resolver *SP1.1*/G/S/R e gerar curvas de *trade-off* entre $F(\mu)$ e $L(\mu)$. Seja $PI_j(\mu_j)$ um índice de prioridade, agora definido como o quociente do aumento marginal de custo pela redução marginal de *WIP* na estação *j*, dado por:

$$PI_j(\mu_j) = \frac{\partial F_j(\mu_j)/\partial \mu_j}{-v_j \partial L_j(\mu_j)/\partial \mu_j} \tag{4.14}$$

O algoritmo 3 é similar ao algoritmo 1 da seção 4.1.2. Seja Δ um incremento de capacidade em cada iteração, previamente especificado. Começa-se com a capacidade inicial μ^0 suficientemente pequena, satisfazendo (4.13). Em cada iteração, aumenta-se de Δ a capacidade da estação com o mínimo índice de prioridade em (4.14). O procedimento é repetido até que a meta L_T seja alcançada.

Algoritmo 3

Passo 1: Comece com a alocação $\mu_j = \mu_j^0$ (suficientemente pequena) e compute os valores ca_j e cs_j (usando (4.11)) para cada estação *j*, *j* = 1, ..., *n*. Esta solução é infactível ($L(\mu^0) > L_T$) e seu custo $F(\mu^0)$ é menor do que o custo mínimo de *SP1.1*/G/S/R.

Passo 2: Em cada iteração, atualize o custo $F(\mu)$, o *WIP* $L(\mu)$ (usando (4.12) e (3.33)) e o quociente $PI_j(\mu_j)$ (usando (4.14)). Adicione a capacidade Δ na estação j^* que resultar no menor PI_{j*} (estratégia gulosa), dado por:

$$PI_{j^*} = \min\left\{ PI_j(\mu_j),\ j = 1,\ldots,n \right\} \tag{4.15}$$

Passo 3: Pare assim que $L(\mu)$ atingir o limite L_T (solução factível).

A medida que são escolhidos valores menores para Δ, o algoritmo 3 gera curvas de *trade-off* mais precisas. Bitran e Tirupati (1989a, proposição 2) mostraram que no limite $\Delta \to 0$, o algoritmo 3 resolve otimamente *SP1.1*/G/S/R (lembre-se que foi assumido que todas as condições (i)-(iv) são satisfeitas), e o $PI_j(\mu_j)$ obtido na última iteração corresponde ao multiplicador dual associado a restrição de *WIP* da estação *j*.

Bitran e Tirupati (proposição 3) também apresentaram um limitante para o erro do valor da solução aproximada obtida pelo algoritmo 3. Suponha que o algoritmo 3 encontre uma solução factível após *p* iterações, denotada por μ^p, e seja μ^* a solução ótima de *SP1.1*/G/S/R. Tem-se que:

$$0 \le F(\mu^p) - F(\mu^*) \le \frac{L_T - L(\mu^p)}{PI_{j^*}^p} + \delta \tag{4.16}$$

onde: $\delta = \Delta \sum_{i=1}^{p} \left(1 - \frac{PI_{j^*}^{i}}{PI_{j^*}^{p}} \right)$

e $PI_{j^*}^{i}$ é o quociente obtido em (4.15) na *i*-ésima iteração, $i = 1, ..., p$. Experiências computacionais com Δ=0,1 aplicado em um exemplo real resultaram num erro relativo de 0,6% entre $F(\boldsymbol{\mu}^p)$ e $F(\boldsymbol{\mu}^*)$. Este erro é aceitável em muitas situações práticas. Estas experiências também indicaram que a suposição anterior de que **ca** e **cs** sejam independentes a mudanças de $\boldsymbol{\mu}$ é razoável (observe no algoritmo 3 que **ca** e **cs** permanecem constantes ao longo das iterações). Como ilustração, ao reduzir o *WIP* de um valor inicial de 70000 para 30000, a maior variação encontrada no valor de **ca** foi de 3%. Esta variação foi obtida atualizando-se os valores de **ca** e **cs** conforme (4.11) para a configuração final da rede.

Uma abordagem mais refinada, porém, demandando maior esforço computacional do que o algoritmo 3, foi apresentada em Bitran e Sarkar (1994b). Esta abordagem atualiza os *scv* ca_j, $j = 1, ..., n$, ao longo das iterações e desta maneira, reflete a dependência de cada $\bar{L}_j$ com relação a $\mu_1, \mu_2, ..., \mu_n$. A seguir, um algoritmo iterativo exato (algoritmo 3a) é apresentado, baseado neste refinamento, para resolver o caso particular do modelo *SP1.1*/G/S/R quando $L(\boldsymbol{\mu}) = L_T$ (ao invés de $L(\boldsymbol{\mu}) \leq L_T$), onde $L_T = L(\boldsymbol{\mu}^0)$ e $\boldsymbol{\mu}^0$ agora denota a capacidade inicial *existente* na rede. Desta maneira, o algoritmo 3a minimiza o custo de capacidade da rede necessário para manter o nível desejado L_T de *WIP* na rede.

Algoritmo 3a

Passo 0: Dados os parâmetros iniciais $\{\lambda_k', ca_k', n_{kl}, k = 1, ..., r; l = 1, ..., n_k, \mu_j^0, cs_j, j = 1, ..., n\}$, aplique o método de decomposição (seção 3.2.3) para obter os parâmetros $\{\lambda_j, ca_j^0, \mu_j^0, cs_j, j = 1, ..., n\}$, onde ca_j^0 e μ_j^0 denotam, respectivamente, o *scv* inicial do intervalo de tempo entre chegadas e a capacidade inicial na estação *j*. Defina $L_T = L(\boldsymbol{\mu}^0)$ e faça $p = 1$.

Passo 1: Em cada iteração *p*, utilize os *scv* ca_j^{p-1}, $j = 1, ..., n$, para resolver o seguinte programa convexo nas variáveis μ_j:

$$\min \quad F(\mu) = \sum_{j=1}^{n} F_j(\mu_j) \qquad (4.17a)$$

$$s.a. \quad L(\mu) = L_T \qquad (4.17b)$$

$$\mu_j > \lambda_j, \; j = 1, ..., n \qquad (4.17c)$$

Sejam μ_j^p, $j = 1, ..., n$, denotando a solução ótima do problema (417a)-(4.17b) usando μ_j^{p-1}.

Passo 2: Aplique o método de decomposição com os parâmetros $\{\lambda_k', ca_k', n_{kl}, k = 1, ..., r; l = 1, ..., n_k, \mu_j^p, cs_j, j = 1, ..., n\}$ para obter os parâmetros $\{\lambda_j, ca_j^p, \mu_j^p, cs_j, j = 1, ..., n\}$. Pare se ca_j^{p-1} e ca_j^p forem suficientemente próximos; caso contrário, faça $p = p + 1$ e volte para o passo 1.

Em cada iteração, o algoritmo 3a assume que o *scv* ca_j seja independente a mudanças na capacidade das estações (veja passo 1 do algoritmo). Dado que a função $\bar{L}_j(\mu_j)$ é convexa em μ_j (Bitran e Tirupati, 1989a), resulta que o problema (4.17a)-(4.17c) é convexo nas em variáveis μ_j, j = 1, 2, ..., n, podendo ser resolvido através das diversas técnicas de programação convexa (veja, p.e., Bazaraa *et al.*, 1993). Bitran e Sarkar (1994b) mostraram que o algoritmo converge para uma solução ótima sob condições usualmente encontradas na prática.

Note que se v_j = 1, j = 1, ..., n, então $L_T = \sum_{j=1}^{n} \bar{L}_j(\mu^0{}_j)$ conforme (4.12). Neste caso, o algoritmo 3a redistribui o número médio de produtos L_T na rede tal que os recursos necessários $F(\boldsymbol{\mu})$ sejam mínimos. Além disso, se $F_j(\mu_j) = \mu_j$, então o algoritmo determina a mínima capacidade para manter o número médio de produtos L_T na rede.

Modelo *SP2.1*/G/S/R

A seguir, analisa-se o problema de redistribuir capacidade existente nas estações de maneira a minimizar o *WIP* na rede. Esta redistribuição faz sentido em redes com capacidade homogênea e intercambiável, isto é, recursos que possam ser compartilhados por diferentes estações (e.g., mão-de-obra). Utilizando (4.12), tem-se:

$$(SP2.1/\text{G/S/R}) \qquad \begin{aligned} \min \quad & L(\mu) = \sum_{j=1}^{n} v_j \bar{L}_j(\mu_j) \\ s.a. \quad & \sum_{j=1}^{n} \mu_j = \sum_{j=1}^{n} \mu^0{}_j \\ & \mu_j > \lambda_j, \; j = 1, \ldots, n \end{aligned}$$

onde μ_j^0 é a capacidade inicial existente na rede, satisfazendo (4.13). Bitran e Tirupati (1989a) apresentaram um algoritmo heurístico (algoritmo 4), também baseado no método guloso, para resolver *SP2.1*/G/S/R (novamente, as condições (i)-(iv) são satisfeitas). Sejam definido conforme anteriormente e $PI_j(\mu_j)$ definido agora como a redução marginal de *WIP* na estação j, dado por:

$$PI_j(\mu_j) = -v_j \frac{\partial \bar{L}_j(\mu_j)}{\partial \mu_j} \tag{4.18}$$

Algoritmo 4

Passo 1: Comece com a alocação $\mu_j = \mu_j^0$ (solução factível) e compute os valores ca_j e cs_j (usando (4.11)) para cada estação j, j = 1, ..., n. Defina J_0 como o conjunto das estações disponíveis, J_1 como o conjunto das estações cuja capacidade for aumentada e J_2 como o conjunto das estações cuja capacidade for reduzida. Inicialmente $J_0 = \{1, 2, ..., n\}$, e J_1 e J_2 são vazios. Compute tal que:

$$PI_j(\mu_j)(\lambda_j + \varepsilon_j) = \max\left\{ PI_j(\mu_j), \; j \in J_0 \right\} \tag{4.19}$$

Passo 2: Em cada iteração, atualize o *WIP* $L(\boldsymbol{\mu})$ (usando (3.33) e (4.12)) e $PI_j(\mu_j)$ (usando (4.18)). Encontre a estação j_1 que resultar no menor PI_{j_1} dado por:

$$PI_{j_1} = \min\left\{PI_j(\mu_j), j \in J_0\right\} \quad (4.20)$$

e a estação j_2 que resultar no maior PI_{j_2} dado por:

$$PI_{j_2} = \max\left\{PI_j(\mu_j), j \in J_0\right\} \quad (4.21)$$

Passo 2a: Se $j_1 \in J_1$, então faça $J_0 \leftarrow J_0 - \{j_1\}$

Passo 2b: Se $j_2 \in J_2$, então faça $J_0 \leftarrow J_0 - \{j_2\}$

Passo 2c: Se $j_1 \notin J_1$ e $j_2 \notin J_2$, então defina $\Delta_1 = \min\left\{\Delta, \mu_{j_1} - \lambda_{j_1} - \varepsilon_{j_1}\right\}$ e faça $\mu_{j_1} \leftarrow \mu_{j_1} - \Delta_1$, $\mu_{j_2} \leftarrow \mu_{j_2} + \Delta_1$, $J_1 \leftarrow J_1 \cup \{j_2\}$, $J_2 \leftarrow J_2 \cup \{j_1\}$.

Passo 3: Pare se J_0 for nulo ou unitário, ou $PI_{j_1} = PI_{j_2}$.

Note que, em cada iteração, (4.20) e (4.21) correspondem às estações que produzem a maior e a menor redução marginal em $L(\boldsymbol{\mu})$, respectivamente. A expressão (4.19) junto com Δ_1 garante que a solução gerada pelo algoritmo 4 satisfaz $\mu_j \geq \lambda_j + \varepsilon_j$, j = 1, ..., n, e portanto, é factível. Bitran e Tirupati (1989a) mostraram que no limite $\Delta \rightarrow 0$, esta solução é ótima para *SP2.1*/G/S/R (lembre-se que foi assumido que todas as condições (i)-(iv) satisfeitas), e todos os $PI_j(\mu_j)$ produzidos na última iteração têm o mesmo valor. Similarmente à seção anterior, cada $PI_j(\mu_j)$ pode ser interpretado como o multiplicador dual associado às restrições de capacidade na estação j. Ele representa a taxa de redução do *WIP* em relação a adições marginais de capacidade nesta estação.

Bitran e Tirupati (proposição 4) também apresentaram um limitante para o erro do valor da solução aproximada gerada pelo algoritmo 4. Seja $\boldsymbol{\mu}^p$ a solução heurística gerada na última iteração p, e $\boldsymbol{\mu}^*$ a solução ótima de *SP2.1*/G/S/R. Então:

$$0 \leq F(\mu^p) - F(\mu^*) \leq n\Delta PI_{j_2}^p \quad (4.22)$$

onde $PI_{j_2}^p$ é o índice de prioridade obtido em (4.21) na última iteração p. Observe em (4.22) que a solução $\boldsymbol{\mu}^p$ é ótima no limite $\Delta \rightarrow 0$. Experiências computacionais com Δ = 0,02 aplicadas no mesmo exemplo testado com o algoritmo 3 resultaram num erro relativo menor que 2% entre $F(\boldsymbol{\mu}^p)$ e $F(\boldsymbol{\mu}^*)$, indicando que o algoritmo 4 é uma boa aproximação para *SP2.1*/G/S/R. Bitran e Tirupati (1989a) reportaram um resultado interessante deste exemplo: o *WIP* foi reduzido de um valor inicial de 70000 para um valor final perto de 40000 apenas redistribuindo a capacidade inicial da rede (note no entanto que eles supuseram que a capacidade de cada estação fosse completamente transferível para as outras estações). Com a finalidade de testar a hipótese da independência de **ca** em relação a alterações da capacidade, eles recomputaram **ca** conforme (4.11) para a configuração final da rede (lembre-se que o algoritmo 4 mantém **ca** e **cs** fixados durante as iterações). A maior variação de **ca** foi em torno de 3%.

A seguir, um algoritmo iterativo exato (algoritmo 4a) que atualiza **ca** ao longo das iterações é apresentado para resolver o seguinte problema de *WIP* ótimo:

$$(SP2.1/\text{G/S/R}) \qquad \begin{aligned} \min \quad & L(\mu) = \sum_{j=1}^{n} v_j \bar{L}_j(\mu_j) \\ s.a. \quad & \sum_{j=1}^{n} F_j(\mu_j) = F_T \\ & \mu_j > \lambda_j, j = 1,\ldots,n \end{aligned}$$

onde $F_T = \sum_{j=1}^{n} F_j(\mu_j^0)$ e μ_j^0 é a capacidade inicial existente na estação *j*, satisfazendo (4.13). O algoritmo 4a minimiza o *WIP* da rede mantendo o mesmo custo de capacidade na rede. As mesmas suposições com respeito à convexidade de $\bar{L}_j$ e F_j, e à convergência do algoritmo 3a são assumidas no algoritmo 4a.

Algoritmo 4a

Passo 0: Dados os parâmetros iniciais $\{\lambda_k^{'}, ca_k^{'}, n_{kl}, k = 1, ..., r; l = 1, ..., n_k, \mu_j^0, cs_j, j = 1, ..., n\}$, aplique o método de decomposição (seção 3.2.3) para produzir os parâmetros $\{\lambda_j, ca_j^0, \mu_j^0, cs_j, j = 1, ..., n\}$, onde ca_j^0 e μ_j^0 denotam, respectivamente, o *scv* inicial do intervalo de tempo entre chegadas e a capacidade inicial na estação *j*. Defina $F_T = \sum_{j=1}^{n} F_j(\mu_j^0)$ e faça $p = 1$.

Passo 1: Em cada iteração *p*, utilize os *scv* ca_j^{p-1}, $j = 1, ..., n$, para resolver o seguinte programa convexo nas variáveis μ_j:

$$\begin{aligned} \min \quad & L(\mu) = \sum_{j=1}^{n} v_j \bar{L}_j(\mu_j) && (4.23a) \\ s.a. \quad & \sum_{j=1}^{n} F_j(\mu_j) = F_T && (4.23b) \\ & \mu_j > \lambda_j, j = 1,\ldots,n && (4.23c) \end{aligned}$$

Sejam μ_j^p, $j = 1, ..., n$, denotando a solução ótima do problema (4.23a)-(4.23c) usando $ca^{p-1}{}_j$.

Passo 2: Aplique o método de decomposição com os parâmetros $\{\lambda_k^{'}, ca_k^{'}, n_{kl}, k = 1, ..., r; l = 1, ..., n_k, \mu_j^p, cs_j, j = 1, ..., n\}$ para obter os parâmetros $\{\lambda_j, ca_j^p, \mu_j^p, cs_j, j = 1, ..., n\}$. Pare se ca_j^{p-1} e ca_j^p forem suficientemente próximos; caso contrário, faça $p = p + 1$ e volte para o passo 1.

Note que se $v_j = 1$, $j = 1, ..., n$, então $L(\mu) = \sum_{j=1}^{n} \bar{L}_j(\mu_j^0)$ conforme (4.3). Neste caso, o algoritmo 4a redistribui os recursos disponíveis F_T tal que o número médio de produtos $L(\mu)$ na rede seja mínimo. Além disso, se $F_j(\mu_j) = \mu_j$, então o algoritmo redistribui capacidade ao invés de investimento em capacidade (recursos). Neste caso, se diz que o sistema está sendo *balanceando* (lembre-se que foi admitido que a capacidade seja homogênea e intercambiável).

Assim, os algoritmos 2, 4 e 4a (problemas de *WIP* ótimo) ajudam a balancear o sistema de manufatura, enquanto os algoritmos 1, 3 e 3a (problemas de *WIP* desejado) adicionam eficientemente recursos ao sistema. Pode-se gerar curvas de *trade-off* entre o capital de trabalho (*WIP*) e o capital de investimento, primeiramente, aplicando o algoritmo 4 para a configuração

original do sistema e então, utilizando a solução obtida $\boldsymbol{\mu}^p$ como capacidade inicial no algoritmo 3 (i.e., $\boldsymbol{\mu}^0 \leftarrow \boldsymbol{\mu}^p$, onde $\boldsymbol{\mu}^0$ é a capacidade inicial no passo 1 do algoritmo 3). Também pode-se utilizar os algoritmos 3a e 4a, conforme é explorado no próximo capítulo.

Wein (1990a) analisou o problema *SP2.1*/G/S/R numa *OQN GI/G/1* de classe única com todos os *jobs* percorrendo um roteiro probabilístico. Partindo do modelo Browniano proposto por Harrison e Williams (1987), que é baseado na aproximação de tráfego pesado de Reiman (1984), Wein obteve o número médio de *jobs* na estação *j* (em equilíbrio), dado por:

$$\bar{L}_j(\mu_j) = \frac{\sigma_j}{2(\mu_j - \lambda_{0j})} \tag{4.24}$$

onde: $\sigma_j = \lambda_{0j} ca_j + \lambda_j cs_j + \sum_{i=1}^{n} \lambda_i q_{ij} (cs_i q_{ij} + 1 - q_{ij})$.

Note que (4.24) não deriva dos métodos aproximados de decomposição discutidos no capítulo 3, ao contrário da expressão (3.33). Além disto, (4.24) é válida somente se uma certa condição, denominada *skew-symmetry*, for satisfeita (veja a expressão (4.4) em Wein, 1990a). Considere novamente a restrição de orçamento utilizada em Kleinrock (1964) e discutida na seção 4.1.1. Supondo-se que a condição de *skew-symmetry* seja satisfeita e utilizando-se (4.24), o modelo *SP2.1*/G/S/R pode ser formulado por:

(*SP2.1*/G/S/R') $$\begin{aligned} \min \quad & \sum_{j=1}^{n} \bar{L}_j(\mu_j) \\ s.a. \quad & \sum_{j=1}^{n} f_j \mu_j = F_T \\ & \mu_j > 0, j = 1, \ldots, n \end{aligned}$$

Após derivar a função Lagrangeana deste problema, Wein obteve a solução em forma fechada μ_j^*, j = 1, ..., n, dada por:

$$\mu_j^* = \lambda_j + \frac{\sqrt{f_j \sigma_j}}{\sum_{i=1}^{n} \sqrt{f_i \sigma_i}} \frac{F_T - \sum_{i=1}^{n} f_i \lambda_i}{f_j} \tag{4.25}$$

Wein observou que a condição de *skew-symmetry* é satisfeita para as redes de Jackson (i.e., sistemas *M/M/1* com $ca_j = 1$ e $cs_j = 1$, j = 1, ..., n), e que (4.25) se reduz a (4.2), que é a solução ótima de *SP2.1*/J/S/R'. Note que, se f_j for igual em todas as estações, então (4.25) primeiro aloca capacidade na estação *j* apenas para compensar λ_j, e depois aloca capacidade na estação *j* na proporção da raiz quadrada do parâmetro σ_j (similarmente a (4.2a)).

Wein apresentou experiências computacionais com um exemplo simples de uma *OQN* satisfazendo a condição de *skew-symmetry*. Estes resultados mostraram que (4.25) produz uma solução muito próxima da solução ótima gerada através de simulação. Embora (4.25) seja derivada sob condições de tráfego pesado ($\rho_j \geq 0{,}9$), ela também pode produzir boas aproximações para baixas intensidades de tráfego. Uma questão importante é investigar a qualidade da solução gerada por (4.25) em situações onde a condição de *skew-symmetry* não é satisfeita.

4.2.2 Modelos ./G/./R com variáveis discretas

Bitran e Tirupati (1989b) apresentaram um algoritmo para o modelo *SP1.1*/G/M/R com *alternativas discretas* para mudança de capacidade em cada estação. Os *jobs* pertencem a múltiplas classes e cada classe segue um roteiro determinístico, conforme a seção 3.2.3.

Similarmente ao que foi feito na seção 4.2.1, ao agregar os dados de entrada descritos na seção 3.2.3, obtém-se os parâmetros $\{m_j, \lambda_{0j}, ca_{0j}, \mu_j, cs_j\}$ para cada estação j. Em seguida, aplicando-se o passo 1 do método de decomposição, obtém-se o sistema de equações (3.19a), (3.27), (3.38) e (3.47), descrito a seguir simplesmente por:

$$\Phi(\mathbf{m}, \lambda, \mathbf{ca}, \mu, \mathbf{cs}) = 0 \qquad (4.26)$$

onde os vetores **m**, $\boldsymbol{\lambda}$, **ca**, $\boldsymbol{\mu}$ e **cs** denotam respectivamente os parâmetros $\{m_j, \lambda_j, ca_j, \mu_j, cs_j\}$ para todas as estações j, $j = 1, ..., n$. Similarmente a (3.33), o número médio de *jobs* $\overline{L}_j$ em (3.40) é função de m_j, λ_j, ca_j, μ_j e cs_j, que por sua vez estão relacionados em (4.26).

Ao invés de escolher **m** ou $\boldsymbol{\mu}$ como as variáveis de decisão de modelo, Bitran e Tirupati consideraram um número finito de alternativas para mudança de capacidade em cada estação. Seja n_j o número total de alternativas para a estação j. Para cada alternativa i, $i = 1, ..., n_j$, são dados:

m_{ji} número de máquinas idênticas da estação j na alternativa i,
μ_{ji} taxa média de serviço de cada máquina da estação j na alternativa i,
F_{ji} custo da capacidade (m_{ji}, μ_{ji}) na estação j (i.e., na alternativa i).

Defina y_{ji} como uma variável de decisão 0-1 (condição (iii) não é satisfeita) tal que:

$y_{ji} =$ 1, se a alternativa i é escolhida para a estação j, e
0, caso contrário.

onde $\sum_{i=1}^{n_j} y_{ji} = 1$.

Para cada estação j, a escolha de capacidade é representada pelo vetor $(y_{j1}, y_{j2}, ..., y_{j,n_j})$ em que todos os elementos são nulos, exceto um. Desta maneira, tem-se que $m_j = \sum_{i=1}^{n_j} m_{ji} y_{ji}$ e $\mu_j = \sum_{i=1}^{n_j} m_{ji} y_{ji}$ e assim, (4.26) e (3.40) dependem de y_{ji}. Seja $\mathbf{y} = \{y_{ji}, j = 1, ..., n; i = 1, ..., n_j\}$. Similarmente à seção 4.2.1, Bitran e Tirupati assumiram que **ca** e **cs** sejam independentes a mudanças de capacidade na rede (veja a discussão na seção 4.2.1). Como conseqüência, pode-se primeiro resolver o sistema (4.26) para um dado **y** (i.e., uma dada capacidade **m** e $\boldsymbol{\mu}$) e então, tratar os **ca** e **cs** obtidos como parâmetros fixos em (3.40). Além disto, ao se escolher a alternativa i na estação j (i.e., $y_{ji} = 1$ e $y_{jl} = 0$, $l \neq i$), $\overline{L}_j(m_j, \lambda_j, ca_j, \mu_j, cs_j)$ em (3.40) é referido simplesmente como $\overline{L}_{ji}$, onde $\overline{L}_{ji} = \overline{L}_j(m_{ji}, \lambda_j, ca_j, \mu_{ji}, cs_j)$. Note que, desta maneira, pode-se computar $\overline{L}_{ji}$ para toda alternativa i e toda estação j utilizando (3.40). Sem perda de generalidade, assume-se que se $\overline{L}_{ji} > \overline{L}_{jl}$, então $F_{ji} < F_{jl}$, $i \neq l$, $i, l = 1, ..., n_j$. Similarmente a (4.12), o *WIP* da rede pode ser rescrito por:

$$L(\mathbf{y}) = \sum_{j=1}^{n}\sum_{i=1}^{n_j} v_j \bar{L}_{ji} y_{ji} \qquad (4.27)$$

onde v_j é o valor médio de um *job* na estação *j*, conforme anteriormente. Utilizando (4.27) obtém-se o seguinte problema:

$$(SP1.1/\text{G/M/R}) \qquad \begin{aligned} \min \quad & F(\mathbf{y}) = \textstyle\sum_{j=1}^{n}\sum_{i=1}^{n_j} F_{ji} y_{ji} \\ & L(\mathbf{y}) \le L_T \\ s.a. \quad & \textstyle\sum_{i=1}^{n_j} y_{ji} = 1,\ j = 1,..,n \\ & y_{ji} \in \{0,1\},\ j = 1,\ldots,n; i = 1,..,n_j \end{aligned}$$

onde L_T é um dada meta para o *WIP* da rede. Note que $L(\mathbf{y})$ e $F(\mathbf{y})$ foram admitidas funções lineares de $\mathbf{y}$. *SP1.1*/G/M/R modela situações cujo alvo L_T é alcançado aumentando-se a capacidade por meio de máquinas adicionais, trabalhadores, ou aumentando-se a disponibilidade com horas extras e turnos adicionais de trabalho. Bitran e Tirupati (1989b) propuseram um algoritmo heurístico (algoritmo 5) para resolver o programa linear inteiro em *SP1.1*/G/M/R acima. Eles mostraram que:

(i) a solução ótima da relaxação LP de *SP1.1*/G/M/R tem *zero* ou *duas* diferentes variáveis y_{ji} com valores fracionários (proposição 3.1)

(ii) se esta solução ótima tiver *duas* variáveis com valores fracionários, então elas correspondem à mesma estação (corolário).

O algoritmo descrito abaixo produz a solução aproximada $\mathbf{y}^1$:

Algoritmo 5

Passo 1: Seja $\mathbf{y}^0$ a solução ótima da relaxação PL de *SP1.1*/G/M/R. Se $\mathbf{y}^0$ for uma solução factível de *SP1.1*/G/M/R, então $\mathbf{y}^1 = \mathbf{y}^0$ é uma solução ótima de *SP1.1*/G/M/R e portanto, pare; caso contrário, vá para o passo 2.

Passo 2: Seja *h* a estação cujas variáveis são valores fracionários para certos i_1 e i_2 ($0 \le y_{h,i_1} < 1$ e $0 \le y_{h,i_2} < 1$). Uma solução factível de *SP1.1*/G/M/R é dada por:

$$y^1_{ji} = y^0_{ji},\ j \ne h,\ j = 1,..,n; i = 1,..,n_j$$

$$y^1_{hi} = \begin{cases} 1, & \text{se } i = l \\ 0, & \text{se } i \ne l \end{cases}$$

onde *l* é tal que $\bar{L}_{hl} = \max\left\{\bar{L}_{hi} \mid \bar{L}_{hi} \le \bar{L}_{hi_1} y^0_{hi_1} + \bar{L}_{hi_2} y^0_{hi_2}, i = 1,..,n_h\right\}$.

Bitran e Tirupati também apresentaram um limitante para o erro do valor da solução aproximada $\mathbf{y}^1$ gerada pelo algoritmo 5. Sem perda de generalidade, assuma que $\bar{L}_{hi} > \bar{L}_{hi_2}$ e denote por $\mathbf{y}^*$ a solução ótima de *SP1.1*/G/M/R. Então,:

$$0 \le F(\mathbf{y}^1) - F(\mathbf{y}^*) \le F_{hi_1} - F_{hi_2} \le \max\left\{F_{ji},\ j = 1,..,n; i = 1,..,n_j\right\}$$

Experiências computacionais com um exemplo de uma rede real indicaram que o algoritmo 5 pode ser uma boa aproximação para *SP1.1*/G/M/R quando o número de classes é relativamente

grande. Neste exemplo, ao reduzir o *WIP* da rede de um valor inicial de 80000 para uma valor final abaixo de 30000, o erro relativo entre $F(\mathbf{y}^1)$ e $F(\mathbf{y}^*)$ foi menor do que 0,08%. A maior variação no valor de **ca** foi de 4.6%, correspondendo a uma variação de 0,5% no *WIP* (lembre-se que os valores de **ca** e **cs** foram mantidos constantes no algoritmo).

Um refinamento desta abordagem é atualizar os *scv* ca_j, $j = 1, ..., n$, ao longo das iterações, conforme os algoritmos 3a e 4a. A seguir, o algoritmo 3a é adaptado para *SP1.1*/G/S/R com alternativas discretas (o algoritmo 4a também pode ser adaptado para *SP2.1*/G/S/R de maneira similar). Os experimentos computacionais indicam que o algoritmo converge após poucas iterações (veja p.e. os resultados da seção 5.5 do próximo capítulo). Uma prova formal da convergência do algoritmo está além do escopo desta tese.

Algoritmo 5a (Algoritmo 3a para alternativas discretas)

Passo 0: Dados os parâmetros iniciais $\{\lambda_k', ca_k', n_{kl}, k = 1, ..., r; l = 1, ..., n_k, \mu_j^0, cs_j, j = 1, ..., n\}$, aplique o método de decomposição (seção 3.2.3) para obter os parâmetros $\{\lambda_j, ca_j^0, \mu_j^0, cs_j, j = 1, ..., n\}$, onde ca_j^0 e μ_j^0 denotam respectivamente o *scv* inicial do intervalo entre chegadas, e a capacidade inicial da estação j (p.e., $\mu_j^0 = \mu_{j1}, j = 1, ..., n$). Defina L_T e faça $p = 1$.

Passo 1: Em cada iteração p, utilize os *scv* ca_j^{p-1}, $j = 1, ..., n$, para computar F_{ji} para cada μ_{ji}, e resolva o seguinte programa linear inteiro nas variáveis y_{ji}:

$$\begin{aligned}
\min \quad & F(\mathbf{y}) = \sum_{j=1}^{n}\sum_{i=1}^{n_j} F_{ji} y_{ji} && (4.28a)\\
& L(\mathbf{y}) \le L_T && (4.28b)\\
s.a. \quad & \sum_{i=1}^{n_j} y_{ji} = 1, j = 1,..,n && (4.28c)\\
& y_{ji} \in \{0,1\},\ j = 1,...,n; i = 1,..,n_j && (4.28d)
\end{aligned}$$

Seja y_{ji}^p, $j = 1, ..., n$, $i = 1, ..., n_j$, denotando a solução ótima do problema (4.28a)-(4.28d) usando ca_j^{p-1}. Note que se $y_{ji}^p = 1$, então a capacidade μ_{ji} é alocada na estação j. Seja μ_p^j, $j = 1, ..., n$, a capacidade alocada na estação j.

Passo 2: Aplique o método de decomposição com os parâmetros $\{\lambda_k', ca_k', n_{kl}, k = 1, ..., r; l = 1, ..., n_k, \mu_p^j, cs_j, j = 1, ..., n\}$ para obter os parâmetros $\{\lambda_j, ca_j^p, \mu_p^j, cs_j, j = 1, ..., n\}$. Pare se ca_j^{p-1} e $ca^p{}_j$ forem suficientemente próximos ou se p atingir um certo limite de iterações; caso contrário, faça $p = p + 1$ e volte para o passo 1.

O problema (4.28a)-(4.28d) pode ser resolvido pelas técnicas de programação linear inteira conhecidas da literatura (veja p.e. Nemhauser e Wolsey, 1988), ou pelo algoritmo heurístico 5, que demanda pouco esforço computacional para encontrar soluções relativamente boas. Nos resultados computacionais apresentados no capítulo 5, esse problema foi resolvido por um procedimento exato do tipo *branch-and-bound*.

Um tópico para pesquisa futura é desenvolver abordagens para situações envolvendo um pequeno número de classes e misturas de roteiros determinísitcos e probabilísticos.

4.2.3 Modelos ./G/./N

Van Vliet e Rinnooy Kan (1991) apresentaram dois algoritmos para resolver os modelos *SP1.1*/G/M/N e *SP2.1*/G/M/N, baseados em análise marginal e no método guloso. Estes algoritmos estão estreitamente relacionados com os dois algoritmos apresentados por Boxma *et al.* (1990) para resolver os modelos *SP1.1*/J/M/N e *SP2.1*/J/M/N (descritos na seção 4.1.2). Novamente, os *jobs* pertencem a múltiplas classes e cada classe percorre um roteiro determinístico diferente. Ao contrário da seção 4.2.2, as variáveis de decisão correspondem ao número de máquinas em cada estação.

Considere novamente o sistema linear em (4.26), e a expressão (3.40) para o número médio de *jobs* numa fila $GI/G/m_j$ da estação j. Van Vliet e Rinnooy Kan consideraram cada capacidade m_j, $j = 1, ..., n$, como uma variável de decisão *inteira*. Dados λ e μ, (4.26) e (3.40) sugerem que mudanças na capacidade **m** resultam em mudanças nos **ca** e **cs** ($\bar{L}_j$ é função de $m_1, m_2, ..., m_n$). Note, entretanto, que esta relação funcional não é fácil de ser analisada.

Baseados nos resultados de Bitran e Tirupati (1989a) (veja seção 4.2.1), Van Vliet e Rinnooy Kan assumiram que **ca** e **cs** são independentes a mudanças na capacidade **m**. Portanto, $\bar{L}_j$ não é dependente de m_i, $i \neq j$ (condição (ii) satisfeita). Eles argumentaram que ao modificar **m**, a média e a variância do tempo de serviço variam na mesma proporção e assim, **cs** permanece aproximadamente constante. Além disto, a sensibilidade de **ca** com relação a mudanças em **m** parece ser pequena a medida que o número de classes aumenta e a proporção da carga devido a cada classe decresce (compare com as hipóteses (a) e (b) admitidas na seção 4.2.1). Portanto, uma vez que o conjunto de equações (4.26) tenha sido calculado, **ca** e **cs** podem ser vistos apenas como parâmetros em (3.40). Isto significa que $\bar{L}_j(m_j, \lambda_j, ca_j, \mu_j, cs_j)$ em (3.40) pode ser visto como uma função apenas de m_j, agora denotada por $\bar{L}_j(m_j)$. Dado que $E(Lq_j)$ no lado direito de (3.40) é uma função convexa em m_j (veja discussão na seção 4.1.2), e que **ca** e **cs** foram assumidos como parâmetros, tem-se que $\bar{L}_j(m_j)$ também é convexa em m_j (condição (i) satisfeita). Seguindo os mesmos passos da seção 4.1.2, o *WIP* da rede $L(\mathbf{m})$ é definido conforme (4.3) (onde $\bar{L}_j(m_j)$ é dado por (3.40), ao invés de (3.12)), e o número inicial de máquinas $\mathbf{m}^0$ deve satisfazer (4.4).

Modelo *SP1.1*/G/M/N

O modelo *SP1.1*/G/M/N é formulado exatamente como *SP1.1*/J/M/N descrito na seção 4.1.2, onde $F(\mathbf{m})$ é uma função convexa não-decrescente de **m** (condição (iv) satisfeita), e $L(\mathbf{m})$ é suposta convexa em **m**, conforme a discussão acima. *SP1.1*/G/M/N, também chamado problema de alocação de servidores, é um programa convexo com variáveis inteiras e portanto, o uso de análise marginal não leva necessariamente à otimalidade (condição (iii) não é satisfeita). O problema pode ser visto como um problema de alocação de máquinas a custo mínimo, tal que o *WIP* da rede resulte menor do que uma dada meta.

Van Vliet e Rinnooy Kan (1991) utilizaram o algoritmo 1 (seção 4.1.2) para resolver *SP1.1*/G/M/N. O algoritmo inicia com a menor alocação possível de máquinas $\mathbf{m}^0$ para todas as estações (alocação infactível). Em cada iteração, ele adiciona uma máquina na estação com o menor índice de prioridade (i.e., o quociente entre o aumento da função objetivo e a redução do *WIP* da rede). Note que este índice de prioridade é o resultado de análise marginal. Ele é obtido

ao substituir (3.40) em (4.5). O algoritmo termina assim que a adição de uma máquina numa estação tornar a alocação factível.

O limitante de erro fornecido por Boxma *et al.* (1990) (discutido na seção 4.1.2) também pode ser aqui aplicado. Por exemplo, se p é a última iteração do algoritmo 1 e $\mathbf{m}^*$ é a solução ótima de *SP1.1*/G/M/N, tem-se que: $F(\mathbf{m}^{p-1}) < F(\mathbf{m}^*) \leq F(\mathbf{m}^p)$. Van Vliet e Rinnooy Kan geraram curvas de *trade-off* entre o custo e o *WIP* da rede, similares àquelas discutidas por Bitran e Tirupati (1989a). A partir dos resultados computacionais de dois exemplos de *OQN*, eles encontraram um erro relativo de 7% na solução do primeiro exemplo e 5% na solução do segundo. Este erro relativo diminui a medida que a meta escolhida para o *WIP* decresce. Van Vliet e Rinnooy Kan também recalcularam **ca** para a configuração final da rede, para verificar a sensibilidade de **ca** à mudanças de **m**. Eles obtiveram um erro de **ca** abaixo de 6%, sugerindo que esta abordagem é uma boa aproximação para o problema *SP1.1*/G/M/N.

Uma abordagem mais precisa para resolver *SP1.1*/G/M/N do que utilizar o algoritmo 1, porém, envolvendo maior custo computacional, é atualizar os *scv* ca_j, $j = 1, ..., n$, ao longo das iterações e, desta maneira, refletir a dependência de cada $\bar{L}_j$ com relação a $m_1, m_2, ..., m_n$. O algoritmo 3a pode ser adaptado para isto, substituindo-se o programa convexo (4.17a)-(4.17c) do passo 1 pelo seguinte programa inteiro nas variáveis m_j, $j = 1, ..., n$:

$$(SP1.1/\text{G/M/N}) \qquad \begin{array}{ll} min & F(\mathbf{m}) = \sum_{j=1}^{n} F_j(m_j) \\ s.a. & L(\mathbf{m}) \leq L_T \\ & m_j \geq m_j^0, \text{ inteiro, } j = 1,\ldots,n \end{array}$$

onde $L_T = L(\mathbf{m}^0)$.

Modelo *SP2.1*/G/M/N

Similarmente ao modelo *SP1.1*/G/M/N, o modelo *SP2.1*/G/M/N é formulado exatamente como o modelo *SP2.1*/J/M/N descrito na seção 4.1.2, onde M é o número de máquinas disponíveis tais que $M > \sum_{j=1}^{n} m_j^0$. Novamente, tem-se a função objetivo e a restrição do problema definidos como funções convexas de **m**. *SP2.1*/G/M/N, também chamado de problema de realocação de servidores, é um programa convexo com variáveis inteiras e pode ser visto como um problema de redistribuir M máquinas ao longo da rede de maneira a minimizar seu *WIP*.

Van Vliet e Rinnooy Kan (1991) utilizaram o algoritmo 2 (seção 4.1.2) para resolver *SP2.1*/G/M/N. O algoritmo inicia com a menor alocação possível de máquinas $\mathbf{m}^0$. Em cada iteração, ele adiciona uma máquina na estação com o máximo índice de prioridade (i.e., a maior redução de *WIP* por máquina). Note que este índice de prioridade é obtido através de análise marginal ao substituir (3.40) em (4.8). O algoritmo termina quando todas as M máquinas tiverem sido alocadas.

Uma vez que admite-se que $L(\mathbf{m})$ seja uma função convexa em **m**, o algoritmo 2 termina após $O(Mn)$ passos com a realocação ótima de máquinas (veja seção 4.1.2 para maiores detalhes). Experiências computacionais com os dois exemplos anteriores indicaram que a sensibilidade de **ca** à mudanças de **m** é pequena. Assim, a solução ótima produzida pelo algoritmo 2 para o suposto problema convexo pode ser utilizada como uma boa aproximação para o problema original.

Um abordagem mais refinada, porém, computacionalmente mais custosa do que utilizar o algoritmo 2, é atualizar os *scv* ca_j, $j = 1, ..., n$, ao longo das iterações. O algoritmo 4a pode ser adatpado para isto, substituindo o programa convexo do passo 1 por um programa inteiro representando *SP2.1*/G/M/N.

No próximo capítulo, ao analisar a geração das curvas de *trade-off*, apresenta-se alguns experimentos computacionais com os algoritmos 3a, 4a e 5a.

5. Geração e análise de curvas de *trade-off*

Este capítulo explora a geração e análise de curvas de *trade-off* para o projeto e planejamento de *job-shops*. Inicialmente, mostra-se como aplicar os modelos e métodos dos capítulos 3 e 4 para gerar as curvas entre os recursos de capacidade e o *WIP* da rede (a metodologia é facilmente estendida para outras medidas de desempenho, como *leadtimes* de produtos). As seções 5.1 e 5.2 apresentam, respectivamente, como minimizar o *WIP* sem adicionar recursos no sistema, e como minimizar os recursos sem aumentar o *WIP*. Para resolver estes problemas, utiliza-se os algoritmos da seção 4.2 do capítulo 4. Conforme é visto adiante, as soluções destes problemas correspondem a pontos na curva de *trade-off*. A seção 5.3 discute como utilizar esses resultados para gerar os demais pontos da curva.

A seção 5.4 apresenta outras curvas de *trade-off* para analisar os efeitos de redução de incertezas na rede (i.e., alterações no parâmetros de variabilidade) (seção 5.4.1), e de mudanças na taxa média de produção (seção 5.4.2) e no *mix* de produtos (seção 5.4.3). Finalmente, as seções 5.5 e 5.6 discutem como estender esta análise para o caso em que se está limitado a um conjunto de alternativas discretas para mudanças de capacidade nas estações, e o caso em que não se pode aproximar cada estação como uma única máquina.

Por comodidade, apresenta-se abaixo as notações utilizadas na seção 3.2.3 do capítulo 3:

- n número de estações internas na rede,
- r número de classes na rede.

Para cada estação j = 1, 2, ..., n:

- m_j número de máquinas na estação j
- μ_j taxa média de serviço para cada máquina na estação j ($\mu_j = 1/E(s_j)$)
- cs_j *scv* ou parâmetro de variabilidade do tempo de serviço na estação j ($cs_j = V(s_j)/E(s_j)^2$).

Para cada classe k = 1, 2, ..., r:

- n_k número de operações no roteiro da classe k.
- $\lambda_k^{'}$ taxa média de chegada externa da classe k ($\lambda_k^{'} = 1/E(a_k^{'})$)
- $ca_k^{'}$ *scv* ou parâmetro de variabilidade do intervalo de tempo entre chegadas externas da classe k ($ca_k^{'} = V(a_k^{'})/E(a_k^{'})^2$).

Para cada classe k = 1, 2, ..., r, e para cada operação l = 1, 2, ..., n_k do roteiro da classe k:

- n_{kl} estação visitada para a operação l do roteiro da classe k.

Para ilustrar a apresentação dos tópicos deste capítulo, utiliza-se o exemplo da rede *job-shop* analisado na seção 3.3 do capítulo 3, com r = 10 classes de *jobs* ou produtos percorrendo roteiros determinísticos, e n = 13 estações sem limitações de tamanho da fila de produtos. Várias curvas de *trade-off* foram geradas para analisar esta rede, como pode ser verificado nas

próximas seções. Conforme já mencionado, este exemplo deriva de uma situação real de uma fábrica de semicondutores, analisado em Bitran e Tirupati (1989b).

As tabelas 9 e 10 apresentam os dados de entrada deste exemplo. Note nestas tabelas que, para que a rede ficasse igual a analisada por Bitran e Tirupati (1989b), foram alterados os parâmetros do processo de chegada (λ'_k, ca'_k) de cada classe k, e do processo de serviço (μ_j, cs_j) de cada estação j, utilizados no capítulo 3 (compare as tabelas 9 e 10 com as tabelas 5 e 6). A taxa média de produção da rede, definida como $\sum_{k=1}^{r} \lambda'_k$, é igual a 10 produtos por unidade de tempo (veja tabela 9). Por simplicidade, considera-se inicialmente cada estação j como uma única máquina, com taxa de processamento μ_j (a seção 5.6 discute como considerar cada estação como um conjunto de máquinas).

Tabela 9 - Dados de entrada para as classes de produtos da rede *job-shop*

Classe k	λ'_k	ca'_k	n_{kl}	n_k
1	1	0,500	1, 2, 4, 2, 9, 10, 11	7
2	1	0,500	1, 2, 5, 2, 8, 9, 10, 11	8
3	1	0,333	1, 2, 6, 4, 2, 9, 12, 11	8
4	1	0,333	1, 2, 7, 4, 2, 9, 10, 11	8
5	1	0,333	1, 2, 4, 12, 2, 9, 2, 13	8
6	1	0,333	1, 2, 5, 12, 2, 9, 7, 13	8
7	1	0,250	1, 2, 6, 12, 2, 8, 2, 13	8
8	1	1,000	1, 2, 3, 7, 4, 12, 2, 8, 6, 9, 2, 13	12
9	1	1,000	1, 2, 3, 5, 4, 6, 12, 2, 8, 2, 10, 6, 13	13
10	1	0,333	1, 2, 3, 6, 2, 4, 12, 7, 2, 9, 11, 5, 13	13
total	10			93

Cada valor v_j da tabela 10 corresponde ao valor unitário (monetário) para um *job* arbitrário na estação j. Conforme mencionado no capítulo 4, este valor é estimado usando experiência prática, ou como uma média ponderada proporcional a taxa de chegada e ao tempo de espera esperados para cada classe (esse tempo médio de espera pode ser aproximadamente computado por um procedimento descrito em Albin, 1986). Obviamente, se $v_j = 1$, então o *WIP* na estação j corresponde ao número médio de unidades de produto $\bar{L}_j$ na estação j. A tabela 10 ainda apresenta os parâmetros α_j e β_j, definidos adiante.

Tabela 10 - Dados de entrada para as estações da rede *job-shop*

Estação j	μ_j	cs_j	v_j	α_j	β_j
1	13,004	0,500	100	5,68	-51,69
2	27,778	0,250	1612	2,59	-50,40
3	3,160	0,333	733	74,77	-165,40
4	10,000	0,500	1052	6,93	-48,53
5	5,631	0,333	912	12,62	-49,73
6	9,225	0,250	1683	7,51	-48,54
7	5,999	1,000	1662	11,11	-46,67
8	4,500	0,333	1812	27,66	-87,11
9	10,000	0,333	1730	7,47	-52,27
10	5,711	0,333	1600	15,34	-61,30
11	5,441	0,333	1882	27,03	-102,94

12	7,440	0,500	1486	13,01	-67,74
13	7,502	0,500	3250	14,22	-74,67
total	115,391				

A tabela 11 apresenta os parâmetros λ_j e ca_j computados para a rede das tabelas 9 e 10, por meio do método de decomposição discutido na seção 3.2.3 (expressões (3.19a), (3.27) (3.30) e (3.47)). Lembre-se que o método admite que os processos de chegada e partida em cada estação sejam processos de renovação, e que o sistema atinja equilíbrio. As colunas ρ_j, $\overline{L}_j$ e $v_j \overline{L}_j$ da tabela 11 correspondem à utilização média (3.5), número médio de produtos (3.33) e *WIP*. Somando-se o *WIP* de todas as estações, obtém-se o *WIP* da rede $L(\mu)$ conforme (4.12), com valor 71089.

Tabela 11 - Parâmetros e medidas de desempenho para a rede *job-shop* das tabelas 9 e 10

Estação j	λ_j	ca_j	ρ_j	$\overline{L}_j$	$v_j \overline{L}_j$	F_j
1	10	0,492	0,77	1,97	197,38	288,32
2	25	0,601	0,90	4,30	6929,06	598,46
3	3	0,760	0,959	10,69	7838,52	223,82
4	7	0,608	0,70	1,57	1650,82	207,70
5	4	0,613	0,71	1,50	1367,93	120,09
6	6	0,583	0,65	1,12	1881,39	191,33
7	4	0,619	0,67	1,71	2850,72	119,84
8	4	0,665	0,89	4,40	7979,09	168,19
9	8	0,642	0,80	2,33	4025,62	224,30
10	4	0,662	0,70	1,49	2382,31	150,24
11	5	0,684	0,92	6,19	11656,35	240,05
12	7	0,614	0,94	9,23	13709,10	216,22
13	6	0,677	0,80	2,65	8620,97	240,11
total				49,16	71089,25	2988,69

A última coluna F_j da tabela 11 refere-se aos recursos da estação j. Por conveniência, estes recursos são medidos por uma função de custo de capacidade (ou investimento em capacidade). Este custo é uma função da capacidade de cada estação j, μ_j. Um exemplo de tal função é:

$$F_j(\mu_j) = \alpha_j \mu_j^2 + \beta_j \mu_j + \chi_j \qquad (5.1)$$

onde α_j, β_j e χ_j são coeficientes conhecidos (o uso de funções quadráticas para custo de capacidade é usual na literatura; veja p.e. Hax e Candea, 1984). Admite-se que é possível adicionar capacidade na estação j em quantidades suficientemente pequenas para considerar μ_j como uma variável contínua (a seção 5.5 analisa o caso mais geral onde mudanças de capacidade são limitadas por um conjunto discreto). Com os valores de α_j e β_j da tabela 10 (por simplicidade, considerou-se que $\chi_j = 0$), obtém-se o custo de capacidade de cada estação (coluna F_j da tabela 11) e da rede, definido por: $F(\mu) = \sum_{j=1}^{n} F_j(\mu_j)$. Note que o valor do *WIP*, 71089, e o valor dos recursos, 2989, definem o ponto O da figura 19.

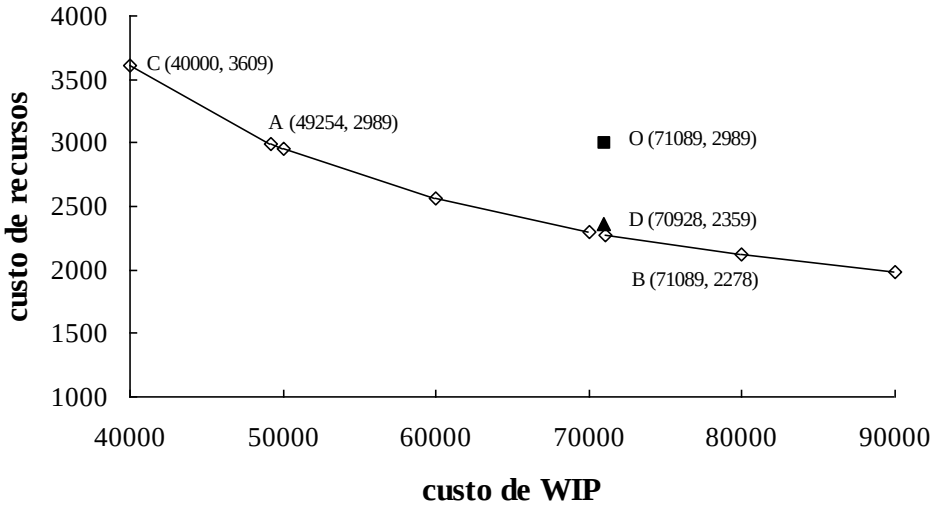


Figura 19 - Pontos O, A, B, C, D e fronteira eficiente

Inicialmente foi assumido que a capacidade da rede seja homogênea e intercambiável entre as estações. Um exemplo é a mão-de-obra treinada, que pode ser transferida de uma estação para outra. Os métodos discutidos adiante também podem ser aplicados quando a capacidade da estação não é transferível para todas as outras estações. No final da seção 5.3 discute-se esse caso mais geral.

5.1 Redistribuição eficiente de recursos

Admita que o sistema esteja no ponto O (71089, 2989). Considerando os dados das tabelas 9 e 10, pode-se formular a seguinte questão: É possível reduzir o *WIP* da rede para abaixo de 71089, sem adicionar recursos na rede? Em outras palavras, é possível redistribuir os recursos de 2989 (intercambiando capacidade entre as estações), tal que o *WIP* da rede seja reduzido? E se esta redução for possível, qual a redistribuição que leva ao mínimo *WIP* da rede?

O *problema de redistribuição ótima de recursos* pode ser resolvido pelo algoritmo 4a do capítulo 4. Lembre-se que este algoritmo admite que a medida que a capacidade μ_j varia, o valor esperado $E(s_j)^2$ e a variância $V(s_j)$ variam na mesma proporção e, portanto, cs_j permanece aproximadamente constante. O algoritmo também assume que, durante cada iteração, ca_j seja independente de mudanças de capacidade nas estações. Esta hipótese é razoável para o exemplo em questão, com número de classes relativamente grande e proporção de demanda de cada classe relativamente pequena, conforme discussão na seção 4.2.1.

Ao aplicar o algoritmo sobre a rede das tabelas 9 e 10, obtém-se o ponto A (49254, 2989), ilustrado na figura 19. O algoritmo converge após duas iterações para uma precisão de 0,001 nos valores de ca_j. A tabela 12 apresenta os valores finais de ca_j, μ_j, ρ_j, $\overline{L}_j$, $v_j \overline{L}_j$ e F_j para cada estação. Note que os valores ca_j da tabela 11 (ponto O) e 5.4 (ponto A) são quase os mesmos, apesar das mudanças de capacidade nas estações. Isto reforça a validade da hipótese de se considerar, em cada iteração do algoritmo, ca_j independente de mudanças de capacidade nas estações.

Tabela 12 - Parâmetros e medidas de desempenho relativos ao ponto A

Estação j	ca_j	μ_j	ρ_j	$\overline{L}_j$	$v_j \overline{L}_j$	F_j
1	0,492	10,604	0,94	8,61	860,86	90,54
2	0,602	28,041	0,89	3,97	6392,55	623,27
3	0,761	3,421	0,88	4,28	3136,30	309,22
4	0,610	8,712	0,80	2,59	2721,36	103,17
5	0,621	5,081	0,79	2,14	1952,44	73,10
6	0,589	7,818	0,77	1,79	3007,80	79,57
7	0,624	5,828	0,69	1,87	3115,22	105,36
8	0,665	4,999	0,80	2,37	4292,98	255,74
9	0,643	9,918	0,81	2,41	4178,75	216,38
10	0,666	5,296	0,75	1,89	3026,34	105,64
11	0,682	6,050	0,83	2,79	5260,48	366,62
12	0,611	8,403	0,83	3,10	4607,64	349,40
13	0,678	7,987	0,75	2,06	6701,74	310,68
Total		112,158		39,88	49254,48	2988,69

O ponto A indica que pode-se melhorar substancialmente o desempenho da rede (i.e., reduzir o *WIP* da rede de 71089 para 49254), sem alterar os recursos (2989). Esta redução é obtida redistribuindo apropriadamente os recursos ao longo das estações (compare as tabelas 11 e 12). Ela não implica numa modificação de processo ou tecnologia. A taxa média de produção da rede também é mantida, igual a 10 unidades de produto por unidade de tempo (veja tabela 9). A figura 20 compara, para cada estação, os recursos antes (ponto O) e depois (ponto A) da redistribuição.

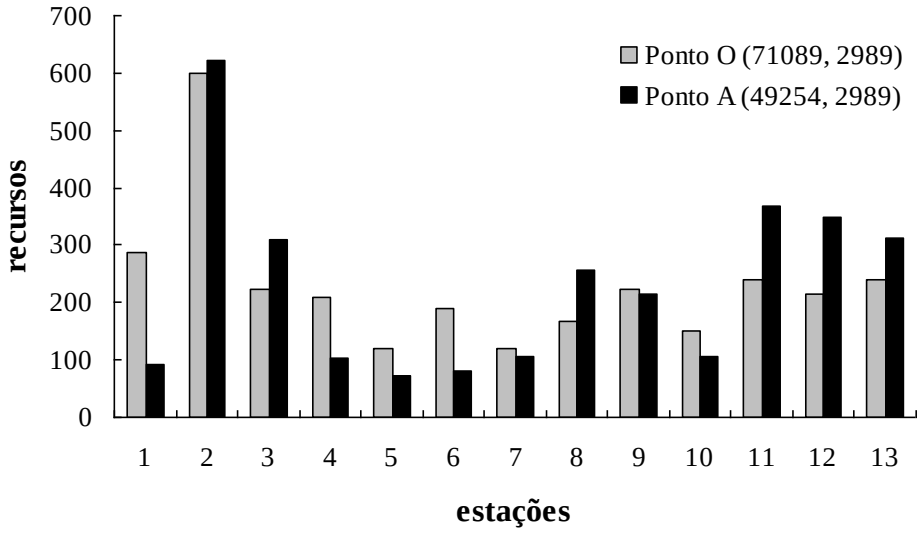


Figura 20 - Recursos em cada estação para os pontos O e A

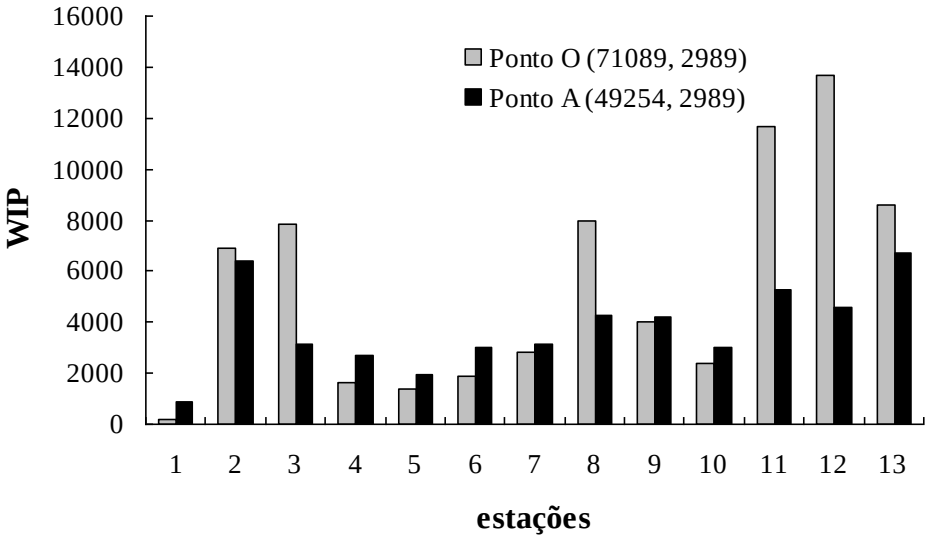


Figura 21 - *WIP* em cada estação para os pontos O e A

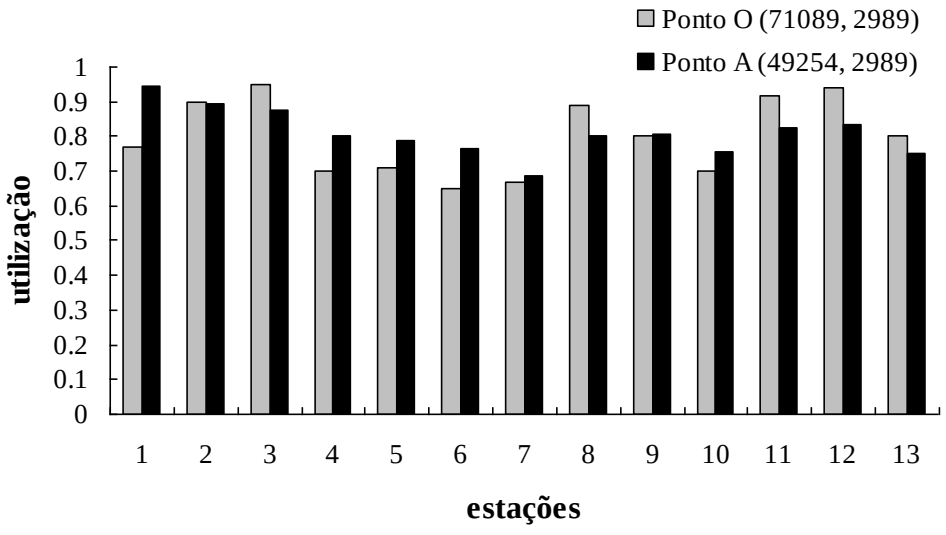


Figura 22 - Utilização em cada estação para os pontos O e A

A medida que se move o sistema do ponto O para o ponto A, deve-se "vender" capacidade das estações 1, 4, 5, 6, 7, 9 e 10, para poder "comprar" capacidade para as estações 2, 3, 8, 11, 12 e 13. Embora a capacidade da rede no ponto O, 115,4, seja diferente da capacidade da rede no ponto A, 112,2 (compare as tabelas 10 e 12), seus custos são exatamente o mesmo: 2989. A figura 21 mostra o impacto da mudança de capacidade no *WIP* de cada estação. Note que o *WIP* aumenta um pouco nas estações 1, 4, 5, 6, 7, 9 e 10, mas diminui substancialmente nas estações 3, 8, 11 e 12.

A figura 22 compara a utilização média das estações para os pontos O e A. As estações 3, 8, 11 e 12, com alta utilização no ponto O, tiveram suas utilizações reduzidas no ponto A. Por outro lado, a estação 1 teve sua utilização substancialmente aumentada no ponto A (de 0,77 para 0,94); entretanto, o efeito do *WIP* da rede não foi tão grande (de 197 para 861), uma vez que v_1 é pequeno comparado com o das outras estações (veja tabela 10 e figura 21).

5.2 Redistribuição eficiente de *WIP*

Admita que o sistema esteja novamente no ponto O (71089, 2989). Considerando os dados das tabelas 9 e 10, pode-se formular a seguinte (segunda) questão: É possível reduzir os recursos da rede para abaixo de 2989, sem mudar o *WIP* da rede de 71089? Em outras palavras, é possível redistribuir o *WIP* de 71089 (intercambiando capacidade entre as estações), tal que os recursos necessários na rede sejam reduzidos? E se esta redução for possível, qual a redistribuição que leva ao mínimo valor de recursos na rede?

O *problema de redistribuição ótima de WIP* pode ser resolvido pelo algoritmo 3a do capítulo 4. Similarmente ao algoritmo 4a, o algoritmo 3a admite que a medida que a capacidade μ_j varia, o valor esperado $E(s_j)^2$ e a variância $V(s_j)$ variam na mesma proporção e, portanto, cs_j permanece aproximadamente constante. O algoritmo também assume que, durante cada iteração, ca_j seja independente de mudanças de capacidade nas estações, o que é razoável para o exemplo em questão (veja discussão na seção 4.2.1).

Aplicando-o sobre a rede das tabelas 9 e 10, obtém-se o ponto B (71089, 2278) ilustrado na figura 19, com parâmetros e medidas de desempenho conforme a tabela 13. O algoritmo converge após duas iterações para uma precisão de 0,001 nos valores de ca_j. Note que, similarmente ao ponto A, os valores de ca_j no ponto B (tabela 12) são muito próximos dos valores do ponto O (tabela 10), apesar das mudanças de capacidade nas estações, o que reforça a validade da hipótese admitida no parágrafo anterior.

Tabela 13 - Parâmetros e medidas de desempenho relativos ao ponto B

Estação j	ca_j	μ_j	ρ_j	$\bar{L}_j$	$v_j \bar{L}_j$	F_j
1	0,492	10,390	0,96	13,11	1311,43	76,11
2	0,598	26,978	0,93	5,84	9415,94	525,35
3	0,760	3,275	0,92	6,35	4658,41	260,35
4	0,607	8,143	0,86	3,73	3923,52	64,33
5	0,616	4,720	0,85	3,04	2772,02	46,46
6	0,581	7,215	0,83	2,49	4189,14	40,72
7	0,617	5,255	0,76	2,69	4463,87	61,54
8	0,657	4,660	0,86	3,40	6163,57	194,74
9	0,638	9,270	0,86	3,46	5994,50	157,40

10	0,657	4,868	0,82	2,66	4262,77	65,11
11	0,672	5,690	0,88	4,05	7616,16	289,39
12	0,604	7,923	0,88	4,54	6741,80	279,96
13	0,668	7,330	0,82	2,95	9576,10	216,65
total		105,717		58,31	71089,25	2278,11

O ponto B indica que é possível reduzir os recursos da rede de 2989 para 2278, sem alterar o *WIP* da rede de 71089 (compare as tabelas 11 e 13). Similarmente à redistribuição eficiente de recursos, a redistribuição eficiente de *WIP* não implica numa mudança de processo, tecnologia, ou na taxa média de produção. A figura 23 apresenta, para cada estação, o *WIP* obtido antes (ponto O) e depois (ponto B) da redistribuição.

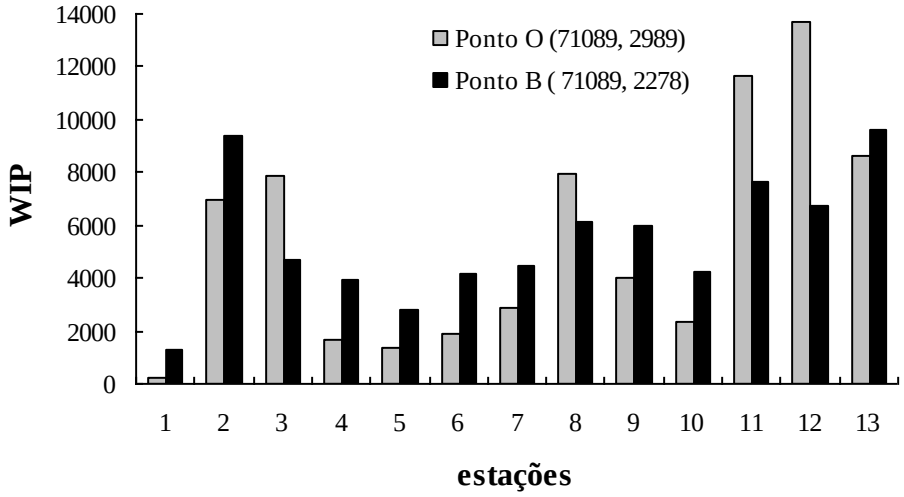


Figura 23 - *WIP* em cada estação para os pontos O e B

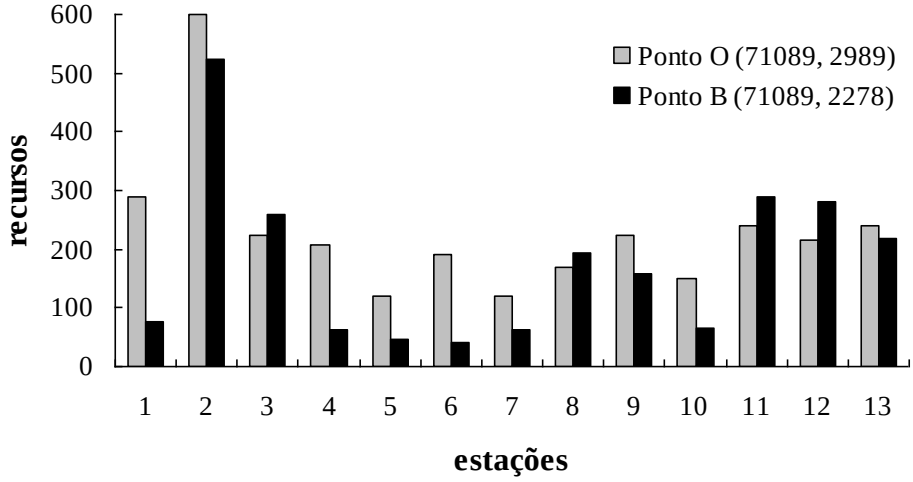


Figura 24 - Recursos em cada estação para os pontos O e B

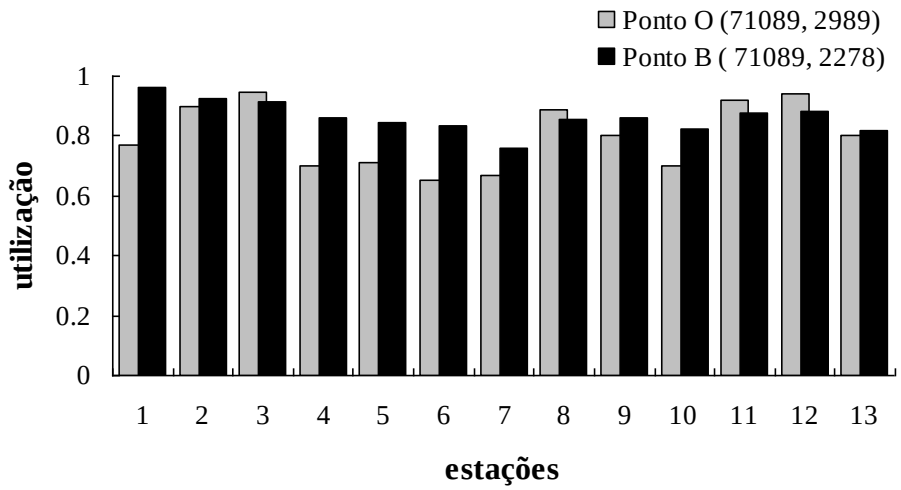


Figura 25 - Utilização em cada estação para os pontos O e B

A medida que se move o sistema do ponto O para o ponto B, "transfere-se" *WIP* das estações 3, 8, 11 e 12 para as outras estações (compare os pontos O e B da figura 23). Esta transferência é obtida intercambiando apropriadamente capacidade entre as estações, tal que o *WIP* da rede de 71089 seja mantido. A figura 24 ilustra os recursos em cada estação após a transferência de *WIP*. Note que os recursos crescem um pouco nas estações 3, 8, 11 e 12, mas decrescem mais da metade dos seus valores iniciais nas estações 1, 4, 6 e 10. A utilização média obtida antes e depois da redistribuição de *WIP* está ilustrada na figura 25.

5.3 Fronteira eficiente

Os algoritmos 3a e 4a movem o sistema para um ponto sobre a curva de *trade-off* ilustrada na figura 19 (lembre-se que os pontos A e B pertencem à curva). O algoritmo 4a move o sistema do ponto O para o ponto A, redistribuindo eficientemente os recursos do ponto O, enquanto que o algoritmo 3a move o sistema do ponto O para o ponto B, redistribuindo eficientemente o *WIP* do ponto O. Os demais pontos da curva podem ser obtidos aplicando os algoritmos 3a e 4a para valores arbitrários de recursos e *WIP* da rede. Em particular, a curva da figura 19 foi traçada

aplicando o algoritmo 3a para os valores de *WIP*: 40000, 50000, ..., 90000, indicados na figura (veja no gráfico os pontos correspondentes que originaram a curva).

Alternativamente, também poderia se ter gerado uma curva próxima da curva da figura 19, mas com menos esforço computacional, usando os algoritmos 3 e 4 (lembre-se que esses algoritmos não são exatos). Conforme descrito no capítulo 4, primeiro, o algoritmo 4 poderia ser aplicado para encontrar um ponto da curva e então, esta solução seria utilizada pelo algoritmo 3 para gerar os demais pontos. Lembre-se que ambos os algoritmos empregam uma *heurística gulosa* simples e intuitiva. A heurística do algoritmo 3 é ilustrada no seguinte exemplo: Considere que se deseja adicionar 100 horas de trabalho (capacidade) nas estações da rede. Por simplicidade, admita que o custo de adicionar 1 hora em cada estação seja constante, diga-se $1 (e portanto, alocar 100 horas é equivalente a alocar $100 na rede). A questão é: Como distribuir esta capacidade extra nas estações de maneira a minimizar o *WIP* da rede?

Dado que é possível adicionar capacidade nas estações em pequenas quantidades, pode-se particionar estas 100 horas em incrementos suficientemente pequenos e adicioná-los, um após o outro, de acordo com uma regra gulosa. O próximo incremento será adicionado na estação que resultar na maior redução de *WIP* da rede, e assim por diante, até que todos os incrementos tenham sido adicionados na rede. Quanto menores forem os incrementos, mais precisa é a solução gerada por este procedimento.

A curva de *trade-off* da figura 19 define uma *fronteira eficiente*, isto é, o valor mínimo de recursos necessário para produzir cada *WIP* ou, equivalentemente, o mínimo de *WIP* produzido por cada valor de recursos. Tome por exemplo o ponto A (49254, 2989) e considere que, de acordo com a estratégia competitiva, o sistema deve operar com um nível de *WIP* menor do que 40000. Qual o mínimo valor de recursos necessário para reduzir o *WIP* de 49254 para 40000? A medida que se desloca sobre os pontos da curva à esquerda do ponto A, encontra-se o ponto C (40000, 3609). Portanto, o sistema necessita de um investimento adicional em capacidade de 620 (i.e., 3609-2989).

Capacidade heterogênea e não intercambiável

Por conveniência, até agora admitiu-se que a capacidade da rede seja homogênea e totalmente intercambiável entre as estações. Os algoritmos 3a e 4a também podem ser aplicados quando parte da capacidade μ_j da estação *j* não é intercambiável. Neste caso, basta impor um limitante inferior para a variável μ_j no passo 1 desses algoritmos. Para fazer isso, apenas adiciona-se aos problemas (4.17a)-(4.17c) e (4.23a)-(4.23c) a restrição $\mu_j \geq \mu_j^1$, j = 1, ..., n, onde μ_j^1 corresponde à capacidade não intercambiável da estação *j*. Outros tipos de restrições também podem ser incluídas para refletir as restrições sobre as transferências de recursos entre estações.

O caso mais geral onde a capacidade não precisa ser homogênea nem intercambiável envolve considerações adicionais. Por exemplo, se é possível apenas adicionar capacidade nas estações, deve-se incluir nos problemas (4.17a)-(4.17c) e (4.23a)-(4.23c) do passo 1 a restrição $\mu_j \geq \mu_j^0$, j = 1, ..., n, onde μ_j^0 é a capacidade inicial na estação *j*. Neste caso os algoritmos 3a e 4a podem ser aplicados para valores arbitrários de F_T e L_T em (4.17b) e (4.23b), tal que $F_T \geq \sum_{j=1}^{n} F_j(\mu_j^0)$ e $L_T \leq \sum_{j=1}^{n} v_j \bar{L}_j(\mu_j^0)$. Um exemplo ocorre no projeto de uma nova rede de manufatura, ou no reprojeto de uma rede existente, quando não se pode "vender" (i.e., remover) capacidade de uma estação para obter dinheiro para poder "comprar" (i.e., adicionar) capacidade para outra estação.

O caso onde também é possível vender capacidade das estações pode requerer modificações nos algoritmos, conforme discutido abaixo.

Quando a capacidade não é transferível entre as estações, pode-se ter que vender capacidade de uma estação para adquirir capacidade para outra. Em muitas aplicações práticas a venda de capacidade leva a uma perda financeira. Em tais casos a equação (4.23b) não se aplica e deve ser substituída pela restrição (5.2b) abaixo, onde c_j é um número positivo menor ou igual a 1, refletindo a perda financeira na transação. O algoritmo 4a pode ser adaptado para resolver esta situação substituindo o problema (4.23a)-(4.23c) do passo 1 por (similarmente para o algoritmo 3a):

$$\begin{aligned} \textit{min} \quad & L(\mu) = \sum_{j=1}^{n} v_j \bar{L}_j(\mu_j) && (5.2a)\\ \textit{s.a.} \quad & \sum_{j=1}^{n} c_j \max\left\{0, F_j(\mu_j^0) - F_j(\mu_j)\right\} = \sum_{j=1}^{n} \max\left\{0, F_j(\mu_j) - F_j(\mu_j^0)\right\} && (5.2b)\\ & \mu_j > \lambda_j,\ j = 1,\ldots,n && (5.2c) \end{aligned}$$

onde $\bar{L}_j(\mu_j)$ e $F_j(\mu_j)$ são definidos por (3.33) e (5.1) (note que F_j deve ser uma função não-decrescente de μ_j). Ao definir $y_j = \max\left\{0, F_j(\mu_j^0) - F_j(\mu_j)\right\}$ e $z_j = \max\left\{0, F_j(\mu_j) - F_j(\mu_j^0)\right\}$, e substituir as funções max por restrições disjuntivas (Nemhauser e Wolsey, 1988), pode-se rescrever o problema (5.2a)-(5.2c) por:

$$\begin{aligned} \textit{min} \quad & L(\mu) = \sum_{j=1}^{n} v_j \bar{L}_j(\mu_j)\\ & \sum_{j=1}^{n} c_j y_j = \sum_{j=1}^{n} z_j\\ & y_j \geq F_j(\mu_j^0) - F_j(\mu_j),\ \ j = 1,..,n\\ & y_j \leq B(1 - p_j),\ \ j = 1,..,n\\ & y_j \leq F_j(\mu_j^0) - F_j(\mu_j) + B(1 - q_j),\ \ j = 1,..,n\\ \textit{s.a.} \quad & z_j \geq F_j(\mu_j) - F_j(\mu_j^0),\ \ j = 1,..,n\\ & z_j \leq B(1 - q_j),\ \ j = 1,..,n\\ & z_j \leq F_j(\mu_j) - F_j(\mu_j^0) + B(1 - p_j),\ \ j = 1,..,n\\ & p_j + q_j = 1,\ \ j = 1,..,n\\ & \mu_j > \lambda_j, y_j \geq 0, z_j \geq 0,\ p_j, q_j \in \{0,1\},\ j = 1,\ldots,n \end{aligned}$$

onde B é um número positivo suficientemente grande, e p_j e q_j são variáveis 0-1 de controle. Se c_j =1, j = 1, ..., n, então a restrição (5.2b) é equivalente à restrição (4.23b), e o algoritmo 4a pode ser aplicado sem qualquer modificação. Os conceitos dos pontos A, B e a fronteira eficiente permanecem válidos. Para mostrar que as restrições (4.23b) e (5.2b) são equivalentes se c_j = 1, inicialmente defina N = {1, 2, ..., n}. Note que esse conjunto pode ser particionado em 3 subconjuntos disjuntos:

$$N_1 = \left\{ j \in N | F_j(\mu^0{}_j) - F_j(\mu_j) > 0 \right\}$$
$$N_2 = \left\{ j \in N | F_j(\mu^0{}_j) - F_j(\mu_j) < 0 \right\}$$
$$N_3 = \left\{ j \in N | F_j(\mu^0{}_j) - F_j(\mu_j) = 0 \right\}$$

onde N_1, N_2 e N_3 são os conjuntos de estações que tiveram seus recursos reduzidos, aumentados, e conservados, respectivamente, depois da redistribuição. A restrição (5.2b) pode ser rescrita por:

$$\sum_{j\in N_1} c_j (F_j(\mu_j^0) - F_j(\mu_j)) = \sum_{j\in N_2} (F_j(\mu_j) - F_j(\mu_j^0)) \qquad (5.3)$$

onde o lado esquerdo de (5.3) corresponde ao retorno total de vender recursos das estações em N_1 e o lado direito de (5.3), ao custo total de adquirir recursos para as estações em N_2. Dado que todas as estações em N_3 tiveram seus recursos conservados, então $\sum_{j\in N_3} F_j(\mu_j) = \sum_{j\in N_3} F_j(\mu_j^0)$. Pode-se rescrever (5.3) como:

$$\sum_{j\in N_1} c_j F_j(\mu_j) + \sum_{j\in N_2} F_j(\mu_j) + \sum_{j\in N_3} F_j(\mu_j) = \sum_{j\in N_1} c_j F_j(\mu_j^0) + \sum_{j\in N_2} F_j(\mu_j^0) + \sum_{j\in N_3} F_j(\mu_j^0)$$

que se reduz a (4.23b) para o caso particular onde $c_j = 1$ para todo $j \in N$.

Na próxima seção assume-se que a redistribuição eficiente de recursos (ou *WIP*) já foi realizada, e que o estado do sistema encontra-se sobre um certo ponto da curva da figura 19, diga-se o ponto A (49254, 2989). Na discussão anterior mostrou-se que, partindo de um ponto sobre a curva, é possível reduzir o *WIP* adicionando capacidade na rede (pontos da curva à esquerda deste ponto). A seguir, discute-se outras alternativas para reduzir o *WIP*, tais como redução de incertezas. As próximas curvas de *trade-off* a serem apresentadas foram geradas pelos algoritmos 3a e 4a.

5.4. Mudanças nos parâmetros de variabilidade, taxa média de produção, e *mix* de produtos

A curva de *trade-off* da figura 19 foi gerada com os dados das tabelas 9 e 10, onde os parâmetros de variabilidade (i.e., os ca'_k para todas as classes de produtos e os cs_j para todas as estações), a taxa média de produção e o *mix* de produtos permaneceram fixos. Variou-se a capacidade μ_j de cada estação e consequentemente, variou-se os recursos, o *WIP* e a utilização média de cada estação. Nesta seção analisa-se a sensibilidade da curva de *trade-off* a alterações dos parâmetros de variabilidade, taxa média de produção, e *mix* de produtos da rede.

5.4.1 Mudanças nos parâmetros de variabilidade

Existem muitos fatores endógenos e exógenos que contribuem para incerteza nos sistemas de manufatura. Exemplos de fatores endógenos são operadores mal treinados, quebras de máquinas, falhas de manutenção, faltas de energia, etc. Estas fontes de incerteza podem ser controladas, por exemplo, investindo no treinamento de mão-de-obra e na melhoria de processo. Em geral tem-se mais controle sobre fatores endógenos do que sobre fatores exógenos. Mas também é freqüentemente possível gerenciar a incerteza dos fatores exógenos, por exemplo, trabalhando mais perto dos fornecedores ou reduzindo o tempo total de ciclo dos produtos (incluindo projeto e produção).

Para analisar o efeito da redução de incertezas numa rede de manufatura, considere inicialmente o simples exemplo de um sistema de único estágio com fila *M/M/1* e apenas uma classe de produtos. A figura 26 ilustra, sob condição de equilíbrio, a variação do *leadtime* médio $\overline{W} = 1/[\mu(1-\rho)]$ em função da utilização média $\rho = \lambda / \mu$ (por simplicidade, considera-se μ = 1 produto por hora). Note que o *leadtime* salta de 2 para 20 horas, ao se aumentar a utilização média ρ (ou, equivalentemente, a taxa média de produção λ) de 0,5 para 0,95. Um aspecto que sempre deve ser considerado no planejamento de um sistema é sua sensibilidade à pequenas perturbações. Perturbações podem ocorrer devido à imprevistos como falhas de equipamentos, faltas inesperadas de energia, atrasos de fornecedores, etc. Por exemplo, o que acontece com o sistema *M/M/1* se as utilizações médias 0,5 e 0,95 forem perturbadas de, diga-se, δ = 0,01? Conforme a figura 26, a variação do *leadtime* no primeiro caso é de apenas 0,041 horas, mas, no segundo caso, sofre um aumento médio de 5 horas! Note que, sob altos níveis de utilização, uma pequena perturbação devido a imprevistos pode disparar uma grande crise no sistema.

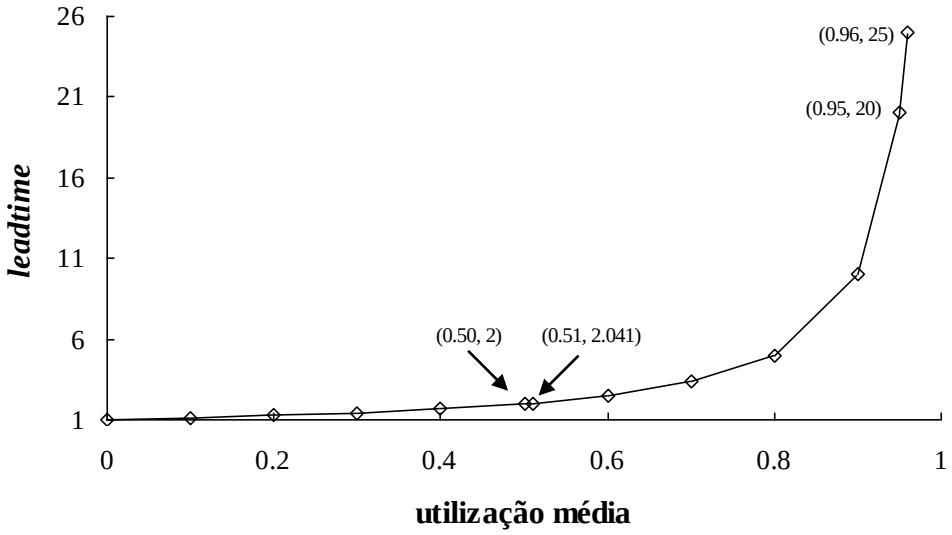


Figura 26 - Relação entre o *leadtime* médio e a utilização média num sistema *M/M/1* com μ = **1**

Entretanto, vários sistemas de manufatura têm que operar com altos níveis de utilização devido aos altos custos de aquisição de capacidade. Apenas para ilustrar, o *break-even-point* de algumas fábricas de semicondutores corresponde à utilizações médias acima de 0,7. Sob estas utilizações, é possível reduzir o *leadtime* dos produtos sem adicionar capacidade no sistema, reduzindo a variabilidade do sistema. A figura 27 apresenta a curva da figura 26 (curva 1, com os *scv ca* = *cs* = 1), ao lado das curvas correspondentes aos *scv ca* = *cs* = 0,5 e *ca* = *cs* = 0,1, obtidas por meio de (3.34). Note que, para a mesma utilização média 0,95, obtém-se *leadtimes* muito diferentes em função dos parâmetros de variabilidade *ca* e *cs* (*leadtimes* de 20, 14,42 e 2,65 horas, respectivamente). Para uma ilustração deste efeito num estudo de caso de uma estação de fotolitografia, veja Lynes e Miltenburg (1994). No limite quando *ca* e *cs* tendem a zero, a curva coincide com o eixo horizontal, correspondendo a um sistema puramente determinístico.

Figura 27 - Impacto de redução de incertezas num sistema de único estágio com $\mu = 1$: Curva 1 ($ca = cs = 1$), curva 2 ($ca = cs = 0{,}5$) e curva 3 ($ca = cs = 0{,}1$)

Estas observações podem ser estendidas para os sistemas de redes de filas. Ao reduzir os *scv* ca_k', $k = 1, ..., r$, e cs_j, $j = 1, ..., n$, espera-se o mesmo efeito de achatamento das curvas de *WIP* em função dos recursos da rede. Uma questão imediata é: Sob que condições a redução de incertezas produz melhores desempenhos (i.e., menores níveis de *WIP*) do que simplesmente investir em expansão de capacidade?

A figura 28 apresenta a curva de *trade-off* da figura 19 (curva 1) ao lado de três outras curvas geradas com valores menores de ca_k' e cs_j. Na primeira (curva 2), reduziu-se pela metade todos os valores de ca_k' da tabela 9, na segunda (curva 3), reduziu-se pela metade todos os valores de cs_j da tabela 10, e na terceira (curva 3), reduziu-se pela metade todos os valores de ca_k' e cs_j.

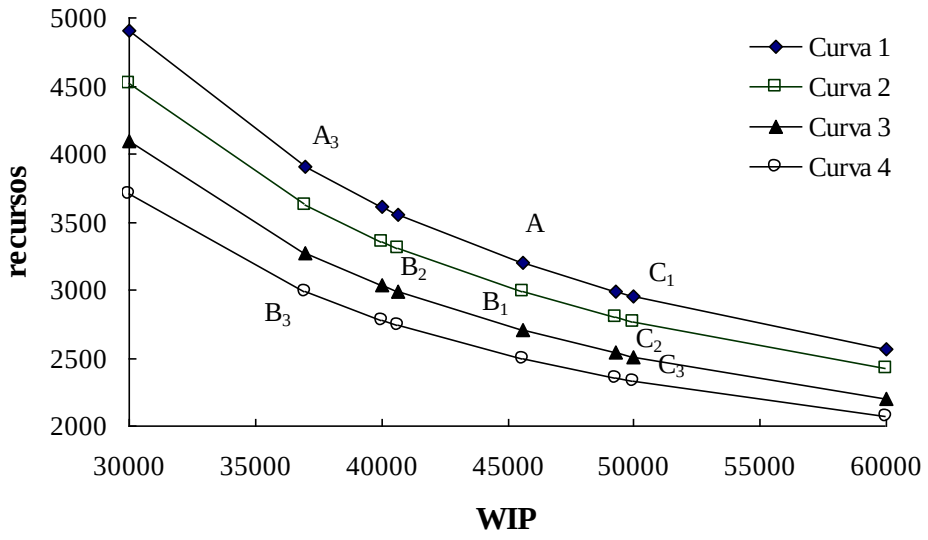


Figura 28 - Mudanças nos parâmetros de variabilidade: Curva 1 (ca_k', cs_j), curva 2 ($ca_k'/2$, cs_j), curva 3 (ca_k', $cs_j/2$) e curva 4 ($ca_k'/2$, $cs_j/2$)

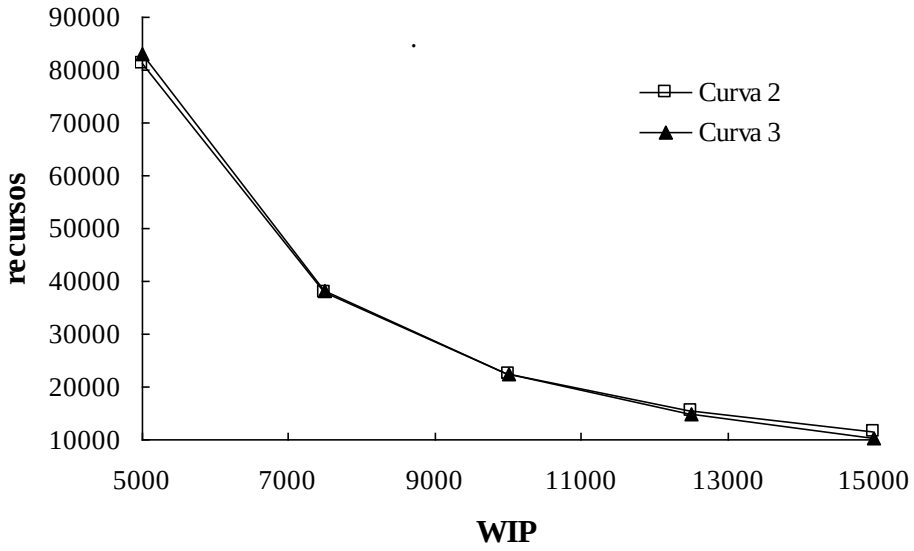


Figura 29 - Mudanças nos parâmetros de variabilidade para pequenos valores de *WIP*: Curva 2 ($ca_k'/2$, cs_j) e curva 3 (ca_k', $cs_j/2$)

Partindo do ponto A (49254, 2989), os pontos B_1 (45584, 2989), B_2 (40645, 2989) e B_3 (36948, 2989) podem ser obtidos pelo algoritmo 4a, e os pontos C_1 (49254, 2803), C_2 (49254, 2537) e C_3 (49254, 2351) pelo algoritmo 3a. Considere que o sistema esteja originalmente no ponto A e tome, por exemplo, a curva 4. Defina *V* como o investimento necessário para reduzir pela metade todos os parâmetros de variabilidade. Ao investir *V*, o estado do sistema é movido do ponto A para o ponto B_3 e assim, o *WIP* é reduzido para 36948. Este nível de *WIP* também poderia ser obtido investindo 918 (i.e., 3907-2989) em capacidade adicional na rede, ao invés de redução de variabilidade, para atingir o ponto A3 (36948, 3907). O valor 918 torna-se um limitante superior para o investimento *V*. Note que com as curvas da figura 28 em mãos, pode-se agora medir o *trade-off* entre investir em capacidade *versus* investir em redução de variabilidade.

A curva 4 é mais achatada do que a curva 3, que é mais achatada do que a curva 2, que por sua vez é mais achatada do que a curva 1. Para altos níveis de utilização, o efeito de reduzir cs_j é mais sensível do que o de reduzir ca_k' (compare as curvas 2 e 3). Isto pode ser explicado em parte pela equação (3.30) do *scv* dos tempos entre partidas da estação *j*, cd_j. Note na equação que, para altas utilizações na estação *j*, a contribuição de cs_j em cd_j é maior do que a de ca_j.

Entretanto, espera-se o inverso para baixas utilizações. Para ilustrar esse efeito, a figura 29 apresenta as curvas 2 e 3 para pequenos valores de *WIP* (menores do que 10000) e portanto, baixas utilizações nas estações. Note que para valores de *WIP* menores do que o ponto de cruzamento das curvas 2 e 3, o efeito de reduzir $ca_k^{'}$ torna-se mais sensível do que o de reduzir cs_j.

Substituição de tecnologia

Conforme foi visto, ao utilizar as curvas de *trade-off* da figura 28, pode-se avaliar o *trade-off* entre adicionar capacidade e investir em redução de incertezas, sem alterar a tecnologia, a taxa média de produção, ou o *mix* de produtos da rede. Suponha agora que se dispõe de uma tecnologia alternativa que permite produzir o mesmo *mix* de produtos, na mesma taxa média de produção. A figura 30 ilustra uma curva hipotética (curva 5), junto com as curvas da tecnologia atual (curva 1) e da tecnologia atual com redução de incertezas (curva 4). Estas duas últimas correspondem respectivamente às curvas 1 e 4 da figura 28. Note agora que tem-se uma nova análise do *trade-off* entre comprar esta tecnologia substituta *versus* investir em redução de incertezas no sistema atual.

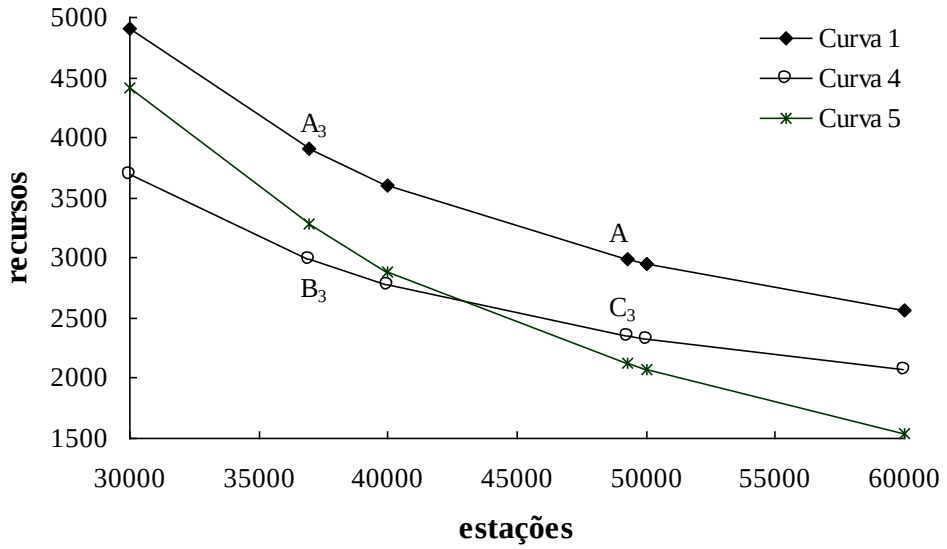


Figura 30 - *Trade-off* entre redução de incertezas e mudança de tecnologia: Curva 1 ($ca_k^{'}$, cs_j), curva 4 ($ca_k^{'}/2$, $cs_j/2$) e curva 5 (nova tecnologia)

5.4.2 Mudanças de taxa média de produção (*throughput*)

As curvas de *trade-off* também ajudam na análise de mudanças de taxa média de produção da rede. A figura 31 apresenta a curva 1 da figura 19, junto com outras duas curvas geradas ao variar a taxa média de produção original da rede, igual a 10 produtos por unidade de tempo (tabela 9). Na primeira (curva 2), reduziu-se em 10% as taxas médias de chegadas externas de todas as classes de produtos na rede (e portanto, a taxa média de produção da rede torna-se 9 produtos por unidade de tempo), e na segunda (curva 3), elas foram aumentadas em 10% (11 produtos por unidade de tempo). Note na figura que a curva 2 é mais achatada do que a curva 1, enquanto a curva 1 é mais achatada do que a curva 3. A variação na taxa média de produção aparentemente *translada* a curva e quanto menor for a taxa média de produção, mais achatada é a curva.

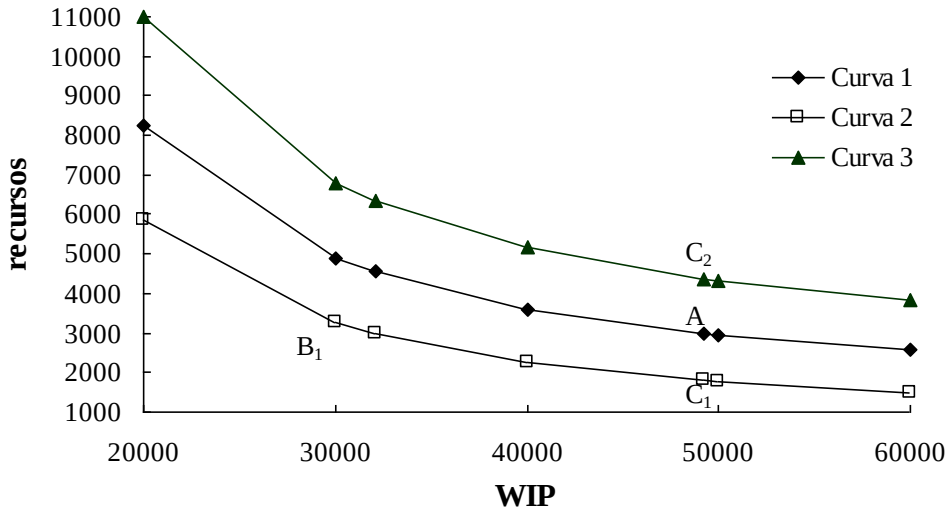


Figura 31 - Mudanças na taxa média de produção: Curva 1 (10 produtos/hora), curva 2 (9 produtos/hora) e curva 3 (11 produtos/hora)

Partindo do ponto A (49254, 2989), obtém-se os pontos B_1 (32079, 2989) e B_2 (98107, 2989) (este último não aparece na figura 31) aplicando o algoritmo 4a, e os pontos C_1 (49254, 1798) e C_2 (49254, 4378) aplicando o algoritmo 3a. Considere novamente que o sistema esteja no ponto A e tome, por exemplo, a curva 3. Note que é improvável que o sistema sobreviva a um crescimento de 10% da taxa média de produção, sem recursos adicionais (ponto B_2). Entretanto, mesmo um aumento de 50% sobre os recursos atuais não é suficiente para manter o mesmo nível de *WIP* do ponto A (ponto C_2).

5.4.3 Mudanças do *mix* de produtos

Os efeitos de variações no *mix* de produtos, tais como eliminação de produtos antigos, modificação da proporção entre os produtos, inclusão de novos produtos, também podem ser analisados com as curvas de *trade-off*. A figura 32 apresenta a curva 1 da figura 19, junto com outras três curvas geradas modificando o *mix* de produtos. Na primeira (curva 2), eliminou-se a classe de produtos 1 (i.e., $\lambda_1 = 0$), na segunda (curva 3), duplicou-se a taxa média de chegada da classe de produtos 1 (i.e., $\lambda_1 = 2$), e na terceira (curva 4), introduziu-se uma nova classe de

produtos (classe 11) com mesma taxa média de chegada da classe 1 (i.e., $\lambda_{11} = 1$), mas com um roteiro muito diferente. A tabela 14 apresenta os dados de entrada para a classe de produtos 11.

Tabela 14 - Dados de entrada para a classe de produtos 11

Classe k	λ_k'	ca_k'	n_{kl}	n_k
11	1	0,500	13, 1, 11, 3, 9, 5, 7	7

Note que a curva 2 corresponde a uma taxa média de produção de 9 produtos por unidade de tempo, enquanto as curvas 3 e 4, a uma taxa de 11 produtos por unidade de tempo. A curva 2 é mais achatada do que a curva 1, que por sua vez é mais achatada do que as curvas 3 e 4. Este resultado é consistente com a discussão da mudança de taxa média de produção da seção 5.4.2. Entretanto, a curva 3 é mais achatada do que a curva 4, apesar de ter a mesma taxa média de produção (lembre-se que, na curva 3, a taxa de chegada da classe 1 foi duplicada e, na curva 4, a classe 11 foi introduzida com a mesma taxa de chegada, parâmetros de variabilidade e número de operações da classe 1). Isto mostra que, neste exemplo, o roteiro da classe 11 produz maiores níveis de *WIP* do que o roteiro da classe 1.

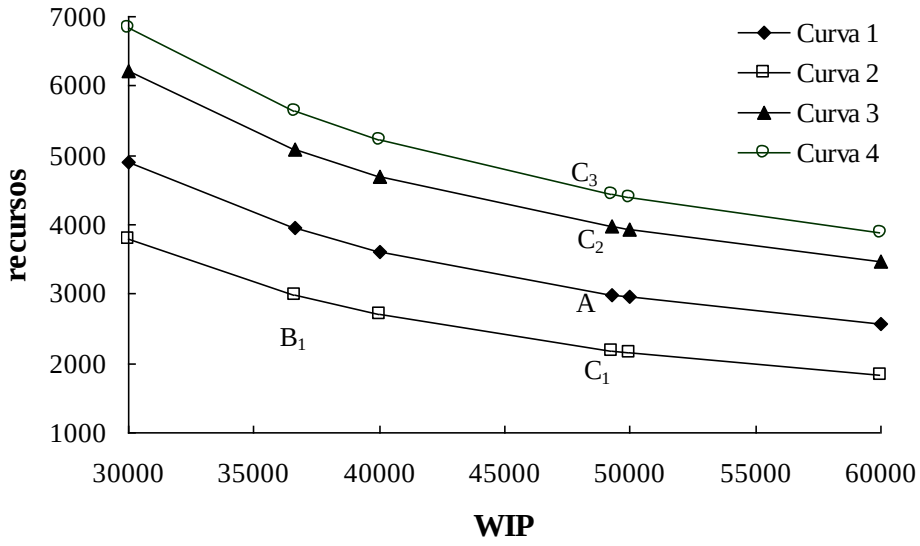


Figura 32 - Mudanças no *mix* de produtos: Curva 1 (curva original), curva 2 (classe 1 eliminada), curva 3 (classe 1 duplicada) e curva 4 (classe 11 incluída)

Partindo do ponto A (49254, 2989), obtém-se os pontos B_1 (36607, 2989), B_2 (77768, 2989) e B_3 (103211, 2989) (os pontos B_2 e B_3 não aparecem na figura 32) usando o algoritmo 4a, e os pontos C_1 (49254, 2184), C_2 (49254, 3973) e C_3 (49254, 4435) usando o algoritmo 3a. Note que ao se eliminar a classe de produtos 1, a utilização média da rede original é reduzida, e o efeito sobre o *WIP* da rede corresponde à distância horizontal entre os pontos A e B_1. Por outro lado, ao se duplicar a classe de produtos 1, ou introduzir a classe de produtos 11, a utilização média da capacidade é aumentada, produzindo um crescimento substancial do *WIP* da rede (pontos B_2 e B_3). Resultados similares foram encontrados na seção 5.4.2 ao aumentar em 10% a taxa média de produção da rede (compare as figuras 31 e 32).

Convém salientar que, além de redução de incertezas e mudanças na taxa média de produção e no *mix* de produtos, outras análises baseadas em curvas de *trade-off* também poderiam ter sido

feitas, apenas variando-se os parâmetros { λ_k', ca_k', n_{kl}, k = 1, ..., r, l = 1, ..., n_k; m_j, μ_j, cs_j, j = 1, ..., n} dos algoritmos 3a e 4a. Por exemplo, poderia-se considerar os efeitos de inclusões de novas estações na rede, alterações nos roteiros de fabricação de classes de produtos, ou mudanças nos padrões de demanda destas classes. Para isto, bastaria modificar adequadamente os parâmetros acima e aplicar os algoritmos para gerar as curvas correspondentes. Note que o potencial da análise de curvas de *trade-off* pode ser grande.

5.5 Alternativas discretas para mudanças de capacidade

Até agora admitiu-se que a capacidade pode ser adicionada ou removida de cada estação j em quantidades suficientemente pequenas, para poder considerar a capacidade total na estação μ_j como uma variável contínua. Isto nem sempre é válido. Esta seção analisa brevemente o caso mais geral, onde mudanças de capacidade em cada estação estão limitadas a um conjunto finito de alternativas discretas. Para isso, aplica-se o algoritmo 5a descrito no capítulo 4.

Considere que, ao invés de escolher qualquer valor de capacidade μ_j, deve-se escolher um elemento de um conjunto finito de n_j alternativas discretas em cada estação j. Este conjunto é descrito pelo vetor $(\mu_{j1}, \mu_{j2}, ..., \mu_{jn_j})$, onde μ_{ji} denota a capacidade na estação j sob a alternativa i, e satisfaz $\mu_{ji} > \lambda_j$ para todo i. A tabela 15 apresenta um conjunto com 5 possíveis alternativas de capacidade para cada estação da rede *job-shop*. Note que a primeira alternativa corresponde ao ponto O da figura 19 (compare com tabela 10).

Tabela 15 - Cinco alternativas discretas para mudanças de capacidade em cada estação

	Alternativas				
Estação j	1	2	3	4	5
1	13,004	10,5	11,0	14,0	15,0
2	27,778	26,0	27,0	28,0	30,0
3	3,160	3,5	3,5	4,0	4,5
4	10,000	7,5	8,0	9,0	11,0
5	5,631	4,5	4,7	5,0	6,0
6	9,225	6,5	7,0	9,0	12,0
7	5,999	5,0	5,5	6,0	6,5
8	4,500	4,5	5,0	5,5	6,0
9	10,000	8,5	9,0	10,0	11,0
10	5,711	4,5	4,7	5,0	6,0
11	5,441	5,3	5,6	5,9	6,0
12	7,440	7,5	8,0	8,5	9,0
13	7,502	6,5	7,0	7,5	8,0
total	115,391				

Cada alternativa i, i = 1, ..., 5, requer o nível de recursos $F_{ji}(\mu_{ji})$ na estação j, definido similarmente à expressão (5.1) como:

$$F_{ji}(\mu_{ji}) = \alpha_j \mu_{ji}^2 + \beta_j \mu_{ji} + \chi_j$$

Note que, para cada alternativa, pode-se calcular as necessidades de recursos na estação. Após escolher a alternativa de capacidade para cada estação j, pode-se aplicar o método de

decomposição (seção 3.2.3) para obter os 4 parâmetros $\{\lambda_j, ca_j, \mu_j, cs_j\}$. Suponha que se escolha a alternativa i na estação j (i.e., $\mu_j = \mu_{ji}$) e assim, obtém-se $\{\lambda_j, ca_j, \mu_{ji}, cs_j\}$. Usando estes parâmetros, pode-se calcular o *WIP* da estação j sob alternativa i como $v_j \ \bar{L}_{ji}(\lambda_j, ca_j, \mu_{ji}, cs_j)$, onde, conforme anteriormente, v_j é um dado valor monetário para um produto arbitrário na estação j, e $\bar{L}_{ji}(\lambda_j, ca_j, \mu_{ji}, cs_j)$ é o número médio de produtos (em fila e em processamento) na estação j sob alternativa i. $\bar{L}_{ji}$ pode ser calculado da mesma maneira que $\bar{L}_j$ na expressão (3.33). Note que, uma vez escolhida uma alternativa para cada estação, pode-se obter o *WIP* de cada estação e da rede. Para determinar a melhor escolha de capacidade nas estações, aplica-se o algoritmo 5a (ou 5) do capítulo 4

Usando o algoritmo 5a na rede *job-shop* das seções anteriores (com as 5 alternativas disponíveis da tabela 15), obtém-se o ponto D (70927, 2359) indicado na figura 19 e apresentado na tabela 16. O algoritmo converge após três iterações para uma precisão de 0,001 nos valores de ca_j e um máximo desvio relativo de 0,2% do valor da solução ótima. Note que o algoritmo seleciona diferentes alternativas para as estações (veja a segunda coluna da tabela 16). As colunas ca_{ji}, μ_{ji}, ρ_{ji}, e assim por diante, indicam os parâmetros e as medidas de desempenho na estação j sob alternativa i, relativos ao ponto D.

Tabela 16 - Parâmetros e medidas de desempenho do ponto D

Estação j	Alt. i	ca_{ji}	μ_{ji}	ρ_{ji}	$\bar{L}_{ji}$	$v_j \bar{L}_{ji}$	F_{ji}
1	2	0,492	10,5	0,95	10,31	1031,50	83,47
2	3	0,598	27,0	0,93	5,78	9321,13	527,31
3	2	0,760	3,5	0,86	3,65	2676,85	337,03
4	3	0,610	8,0	0,87	4,23	4448,63	55,28
5	4	0,619	5,0	0,80	2,28	2084,09	66,85
6	3	0,584	7,0	0,86	2,95	4971,32	28,21
7	3	0,622	5,5	0,73	2,27	3765,36	79,39
8	1	0,660	4,5	0,89	4,39	7947,18	168,12
9	3	0,637	9,0	0,89	4,30	7438,94	134,64
10	4	0,654	5,0	0,80	2,35	3756,94	77,00
11	3	0,671	5,6	0,89	4,60	8651,75	271,20
12	3	0,606	8,0	0,87	4,22	6265,35	290,72
13	4	0,666	7,5	0,80	2,64	8568,90	239,85
total			106,1		53,97	70927,95	2359,08

Os valores de ca_j no ponto D, assim como no ponto B, são muito próximos dos valores de ca_j no ponto O (compare as tabelas 11, 13 e 16). Note também que o *WIP* no ponto D pode ser menor ou igual a L_T (71089), ao invés de exatamente igual a L_T, como no ponto B (compare a desigualdade (4.28b) do algoritmo 5a com a igualdade (4.17b) do algoritmo 3a). Dado que $\mu_{ji} > \lambda_j$ para todo j e i, segue que a necessidade de recursos no ponto B, igual a 2278 (figura 19), torna-se um limitante inferior para a necessidade de recurso no ponto D, igual a 2359 (lembre-se que o ponto B é obtido do problema (4.17a)-(4.17c) onde as capacidades (variáveis de decisão) μ_j, j = 1, ..., n, podem ser contínuas).

Similarmente ao algoritmo 3a, o algoritmo 5a também pode ser aplicado para traçar a fronteira eficiente do problema com alternativas discretas para mudanças de capacidade. Naturalmente, essa fronteira agora não é mais definida com uma curva contínua, mas como um conjunto de pontos discretos. Experiências computacionais com o algoritmo 5a indicam que ele converge sob

tolerâncias razoáveis na precisão dos valores finais de ca_j (estas tolerâncias são fixadas para prevenir contra o efeito de ciclagem próximo da solução ótima). A prova da convergência do algoritmo 5a é um tópico para pesquisa futura.

Os três algoritmos 3a, 4a e 5a utilizados neste capítulo foram codificados na linguagem de modelagem *GAMS* (*General Algebraic Modeling System*, versão 2.25) (Brooke *et al.*, 1992). Para resolver em cada iteração os programas lineares e programas convexos dos algoritmos 3a e 4a, utilizou-se o *solver GAMS/MINOS*, e para resolver os programas lineares e programas inteiros do algoritmo 5a, utilizou-se o *solver GAMS/OSL*. As soluções apresentadas nas tabelas 11, 12 e 14, assim como as curvas de *trade-off*, foram obtidas em poucos minutos (incluindo a geração de relatórios detalhados) usando um microcomputador Pentium 100 Mhz. O desempenho computacional pode ser melhorado com a implementação de rotinas matemáticas que explorem as características particulares dos programas convexos e programas inteiros envolvidos.

5.6 Múltiplas máquinas

Nas seções 5.1-5.5, definiu-se capacidade de cada estação como a taxa média de processamento μ_j. Considerou-se cada estação j como uma "única máquina", ou um conjunto de máquinas, operadores, ferramentas, etc., que pode ser aproximado por uma única máquina com taxa média de processamento μ_j. Esta aproximação nem sempre é razoável. Pode-se ter situações onde a capacidade de cada estação tem que ser descrita como um conjunto de máquinas, cada máquina com uma dada taxa média de processamento.

No caso de todas as máquinas serem idênticas em cada estação (i.e., com mesma taxa média de processamento), algoritmos muito similares aos algoritmos 3a e 4a podem ser aplicados, conforme discussão na seção 4.2.3. Tais algoritmos consideram a variável de decisão em cada estação como o número de máquinas (m_j), ao invés da taxa média de processamento (μ_j). Além disso, medidas de desempenho, tais como o *WIP* definido na expressão (4.1), têm de ser redefinido de acordo com as fórmulas para múltiplas máquinas da teoria de filas (veja seção 3.2.2)

O caso mais geral em que se pode ter máquinas distintas na mesma estação envolve dificuldades adicionais, e é um tópico para pesquisa futura.

6. Conclusões e perspectivas para pesquisa futura

6.1 Conclusões

Esta tese explorou e evidenciou o potencial da análise de curvas de *trade-off* para o projeto e planejamento de sistemas discretos de manufatura, com ênfase em sistemas *job-shops*. Procurou-se mostrar como estas curvas podem desempenhar um papel importante para a tomada de decisões com respeito à quantidade e tipo de capacidade necessária para gerir o sistema eficientemente, para avaliar o impacto de incertezas na chegada e processamento de *jobs*, assim como as conseqüências de mudanças nas taxas médias de produção e no *mix* de produtos. A metodologia foi ilustrada com um exemplo derivado de uma aplicação real numa fábrica de semicondutores.

Para gerar as curvas de *trade-off*, mostrou-se como os sistemas podem ser representados por meio de redes de filas abertas e revisou-se diversos algoritmos para resolver modelos de otimização em *OQN* para duas categorias de problemas: na primeira minimiza-se o investimento em capital para satisfazer uma medida de desempenho do sistema (*WIP* ou *leadtime*), e na segunda busca-se otimizar a medida de desempenho sujeito a restrições de recursos. A questão central é a seleção entre as várias configurações para a rede, ou seja, como o investimento deve ser distribuído para alocar capacidade nas várias estações da rede. Uma contribuição desta tese foi apresentar o estado da arte destes modelos de otimização aplicados a sistemas *job-shops*.

Para estimar as medidas de desempenho utilizadas nos modelos de otimização, métodos de decomposição exatos e aproximados para avaliação de desempenho em *OQN* foram revisados. Os métodos aproximados, baseados em fórmulas matemáticas ao invés de simulação, produzem resultados precisos com pouco esforço computacional, para redes de manufatura relativamente grandes e complexas. Apesar da análise admitir que o sistema esteja em equilíbrio e basear-se apenas nos dois primeiros momentos das distribuições de chegada e processamento de *jobs* (tipicamente a média e o *scv*), as aproximações são relativamente robustas e permitem que esses métodos sejam usados para estimar medidas médias de desempenho com uma tolerância dentro de 10% dos valores reais (no caso de *OQN* genéricas).

6.2 Perspectivas para pesquisa futura

Vários tópicos para pesquisa futura foram levantados ao longo dos capítulos 1-5, tais como: (i) estender as abordagens aqui apresentadas para tratar *OQN* genéricas com máquinas distintas na mesma estação, ou com um pequeno número de classes e misturas de roteiros determinísticos e probabilísticos, (ii) realizar uma comparação computacional efetiva entre as diversas aproximações apresentadas na seção 3.2 para avaliar medidas de desempenho em *OQN*

genéricas (veja a tabela 7), e (iii) provar a convergência do algoritmo 5a da seção 4.2.2. Outros tópicos são discutidos a seguir.

6.2.1 Estudo de caso numa rede *job-shop* no Brasil

A principal perspectiva para pesquisa futura desta tese é a aplicação da metodologia de curvas de *trade-off* num estudo de caso de uma rede *job-shop* no Brasil. Exemplos de questões a serem respondidas são: Até que ponto as hipóteses admitidas nos modelos são razoáveis num ambiente de manufatura brasileiro? Quais as dificuldades encontradas no processo de coleta de dados? Como é o comportamento das curvas de *trade-off* obtidas no estudo de caso, em relação às curvas esboçadas no capítulo 5? Quais os benefícios da aplicação desta metodologia para a rede *job-shop* brasileira a ser estudada?

Hipótese de equilíbrio

Em particular, uma hipótese importante de ser validada neste estudo de caso é a de que o sistema atinge o estado de equilíbrio (*steady state*). Em todos os modelos discutidos nesta tese foi suposto que o sistema passasse por diversos *estados transitórios* até finalmente alcançar o estado de equilíbrio. Entretanto, este estado pode não existir, ou, se existir, não ser atingível pelo sistema durante sua vida útil. Alguns autores têm criticado a análise de equilíbrio em sistemas de manufatura discretos. Mudanças ocorrem freqüentemente no ambiente devido ao lançamento de novos produtos, obsolescência prematura de produtos existentes, alterações na capacidade das estações, atualizações tecnológicas do processo, entre outros motivos. Uma questão ainda aberta é se os sistemas *job-shops* modernos mantêm suas características durante um período de tempo suficientemente longo para atingir o estado de equilíbrio. Futuras pesquisas explorando este tópico seriam úteis para caracterizar instâncias em que a hipótese de equilíbrio pode ser assumida.

6.2.2 Aproximações de tráfego leve

Outra oportunidade para pesquisa futura são as *aproximações de tráfego leve* para analisar *OQN* genéricas com múltiplas classes e roteiros determinísticos. Por exemplo, na manufatura de placas de circuito impresso, o número de classes pode ser muito grande (entre 100 e 1000), com cada classe seguindo um roteiro diferente (Segal e Whitt, 1989). Sob certas condições, as aproximações de tráfego leve podem simplificar efetivamente a avaliação das medidas de desempenho de redes *job-shops* grandes e complexas e, consequentemente, facilitar a geração das curvas de *trade-off.*

Conforme discutido na seção 3.2.3 (interferência entre as classes), Bitran e Tirupati (1988) sugeriram que se o número de classes de *jobs* processados numa estação for suficientemente grande, pode-se ignorar a interação entre as estações e analisar cada estação independentemente. A média e a variância do fluxo de cada classe são preservadas ao longo da rede e, assim, pode-se assumir em cada estação que a média e a variância são as mesmas do processo de chegada externa dessa classe. Em outras palavras, à medida que o número de classes aumenta, espera-se que $q_{kl} \rightarrow 0$ e $cd_{kl} \rightarrow ca_{kl}$ em (3.47) para todo par (k,l) e portanto, $cd_{jk} \rightarrow ca_k'$. A rede pode ser decomposta num conjunto de estações independentes, cada uma analisada como um sistema de fila individual (similarmente às redes de Jackson discutidas na seção 3.1). Note que, desta maneira, virtualmente não haveriam limites para o tamanho da *OQN* que poderia ser analisada. Entretanto, redes de manufatura na prática têm usualmente uma ou

mais estações operando sob a condição de intensidade de tráfego pesado (altos níveis de utilização).

Baseado neste argumento, Whitt (1988) observou que se a taxa de chegada de uma classe numa estação for uma pequena proporção da taxa total de chegada (i.e., q_{kl} é pequeno para a classe k durante sua visita l na estação n_{kl}), então o processo de partida para essa classe a partir dessa estação deve ser muito próximo do processo de chegada dessa classe nessa estação (i.e., $cd_{kl} \rightarrow ca_{kl}$). Note que isto pode ser visto como uma aproximação de tráfego leve, onde apenas a classe de interesse tem que estar em tráfego leve. Ou seja, a estação não precisa também ter baixa utilização, mas a contribuição da taxa dessa classe na taxa total da estação deve ser pequena. Esta observação pode ser estendida para a rede. Se esta condição de tráfego leve ocorre em todas as estações do roteiro da classe de interesse, então os processos de chegada e de partida dessa classe em todas as estações do roteiro dessa classe devem ser muito próximos do processo de chegada externo da classe (i.e., $ca_{kl} \rightarrow ca_k'$ e portanto, $cd_{kl} \rightarrow ca_k'$). Novamente, note que apenas a classe de interesse precisa estar em tráfego leve nas estações. Este princípio torna-se mais significativo com o aumento do tamanho e da complexidade dos sistemas.

Whitt (1988) ainda observou que existe outra condição que também precisa ser satisfeita para a aproximação de tráfego leve. A carga ofertada da classe de interesse também precisa ser pequena, isto é, o tempo médio de serviço não pode crescer indefinidamente ao mesmo tempo. Em outras palavras, esta condição de tráfego leve supõe que a escala de tempo das chegadas e partidas da classe de interesse é muito maior do que a escala de tempo da agregação de todas as outras classes (classe agregada) na estação. Por exemplo, os intervalos de tempo entre chegadas de *jobs* da classe de interesse são de dias ou meses (e os tempos de processamento são de horas ou minutos), enquanto os intervalos de tempo entre chegadas de *jobs* da classe agregada são de horas ou minutos. Assim, os tempos de espera e serviço de *jobs* da classe de interesse são desprezíveis, se comparados com seus intervalos de tempo entre chegadas na estação.

O uso de aproximações de tráfego leve, portanto, poderia permitir enormes simplificações na análise de classes sob tráfego leve em *OQN* genéricas. Entretanto, resta uma importante questão pragmática: *Sob que condições estas aproximações poderiam ser realmente aplicadas em job-shops?* Por exemplo, quais valores práticos de intensidade de tráfego e proporção de uma classe na estação satisfariam a condição de tráfego leve para essa classe nessa estação? E como combinar aproximações de tráfego leve em estações bem ocupadas?

6.2.3 Problemas de partição da instalação (classe *SP3*)

Vários autores têm observado que os sistemas de manufatura modernos estão se tornando muito complexos para serem geridos, devido principalmente ao grande número de classes de produtos competindo pelos mesmos recursos, à incerteza na demanda dos produtos, e à redução dos *leadtimes*. Além de desenvolver métodos mais eficazes para analisar sistemas cada vez mais complexos, também podemos nos esforçar em reduzir a complexidade no ambiente de manufatura. Grande parte do sucesso de *just-in-time* e outros métodos relacionados é devido à simplificação. Exemplos de alternativas para reduzir a complexidade incluem a partição de linhas de produção existentes, a duplicação de recursos, e o reprojeto de produtos e processos de manufatura. Note que os problemas da classe *SP3* (p.e., o problema *SP3.1* no capítulo 1), podem ser vistos neste contexto.

Algumas tentativas baseadas em modelos de *OQN* têm sido realizadas para analisar os *trade-offs* entre a partição de linhas de produção e a duplicação de máquinas (Bitran e Sarkar, 1994c;

veja também Tang e Yoo, 1991, para um estudo relacionado de partição de clientes e alocação de servidores em um sistema de serviços com fila única). A idéia é relacionar os conceitos de *complexidade* e *previsibilidade* de um sistema, sugerindo que sistemas mais complexos tendem a ser menos previsíveis. Por exemplo, um gerente de produção deve ser capaz de prever o *leadtime* de produção de um produto com razoável precisão. A medida que se aumenta o número de classes de produtos no sistema, a complexidade tende a crescer e a previsibilidade tende a diminuir, devido à interferência entre as classes nas estações. Também é possível reduzir a complexidade por meio da *partição da instalação*. A seguir, sugere-se possíveis medidas de desempenho que procuram capturar estas idéias, e podem ser analisadas em curvas de *trade-off*. Considera-se:
(i) medidas de complexidade do ponto de vista da gerência de produto,
(ii) medidas de complexidade do ponto de vista da gerência de estação.

Medidas de complexidade do ponto de vista da gerência de produto

Gerentes devem ser capazes de prever o *leadtime* de produção de um produto o mais precisamente possível. Em outras palavras, a variância do *leadtime* deve ser pequena. Pode-se reduzir a variância, por exemplo, adicionando máquinas nas estações. Seja T_k o *leadtime* de um produto da classe *k*, w_k um peso associado ao produto da classe *k*, e *U* um limitante superior para a variância ponderada do *leadtime* de todas as classes na rede. Assim, pode-se formular a seguinte restrição de complexidade:

$$\sum_{k=1}^{r} w_k V(T_k) \le U \qquad (6.1)$$

Note que quanto menor for o valor fixado de *U*, maior é a previsibilidade no sistema. Cada variância $V(T_k)$ em (3.45) é definida como a soma das variâncias dos tempos de espera nas filas e dos tempos de serviço de todas as estações no roteiro da classe *k*. Por simplicidade, assuma que os tempos de serviço sejam determinísticos em todas as estações e portanto, suas variâncias são nulas. Obtém-se então (Bitran e Sarkar, 1994c):

$$V(T_k) = \sum_{l=1}^{n_k} \sum_{j=1}^{n} V(Wq_j) 1\{j = n_{kl}\} \qquad (6.2)$$

onde $V(Wq_j)$ é a variância do tempo de espera na estação *j* aproximada por (3.42), ou pela aproximação de tráfego pesado:

$$V(Wq_j) = (E(Wq_j)_{M/M/m_j})^2 \frac{(ca_j + cs_j)}{4} \qquad (6.3)$$

onde $E(Wq_j)_{M/M/m_j}$ é o tempo de espera médio num sistema de fila $M/M/m_j$ (a aproximação (6.3) pode ser mais adequada do que (3.42) em situações onde a utilização média ρ_j é alta). Uma vez que $E(Wq_j)_{M/M/m_j}$ é uma função convexa decrescente em m_j e assumindo-se que ca_j e cs_j sejam independentes à mudanças de capacidade na rede (conforme seção 4.2 do capítulo 4), segue de (6.3) que $V(Wq_j)$ decresce ao se aumentar m_j. Portanto, ao adicionar máquinas nas estações do roteiro da classe *k*, reduz-se a variância $V(T_k)$ em (6.2) e assim, a complexidade do sistema do lado esquerdo de (6.1). Desta maneira, pode-se analisar os *trade-offs* entre complexidade (ou previsibilidade) e capacidade do sistema.

As expressões (6.1)-(6.3) também sugerem que, para uma mesma capacidade da planta (unidade de produção), poderia se reduzir a complexidade do sistema particionando-se apropriadamente a planta em sub-plantas (unidades menores), com um *mix* de produtos mais homogêneo. Desta maneira, seriam obtidos parâmetros de variabilidade menores nas estações de cada sub-planta, tais que a variância total de cada classe em (6.2) fosse reduzida e, por conseguinte, a complexidade do sistema em (6.1).

Medidas de complexidade do ponto de vista da gerência de estação

Bitran e Sarkar (1994c) observaram que à medida que o número de máquinas cresce em uma estação, espera-se obter maior *flexibilidade* (em termos de programação e manutenção) para operar essa estação. Para determinar o número de máquinas em cada estação, deve-se considerar as incertezas dos intervalos de tempo entre chegadas e dos tempos de processamento dos produtos que visitem a estação.

Whitt (1992) discutiu algumas heurísticas que podem ser úteis para descrever restrições de complexidade com respeito ao *nível de serviço* de cada estação. O nível de serviço é uma medida que, quando fixada, mantém uma certa *medida de congestão* aproximadamente constante na estação (logo abaixo é discutida a relação entre o nível de serviço e a flexibilidade da estação). Por exemplo, seja φ_j o nível de serviço na estação *j*, dado por:

$$\varphi_j = (1-\rho_j)\sqrt{m_j} \tag{6.4}$$

A expressão (6.4) sugere uma *economia de escala*, isto é, para um certo nível de serviço, a utilização média cresce ao se aumentar o número de máquinas e a taxa média de chegada λ_j da estação *j* (lembre-se que $\rho_j = \lambda_j / (m_j \mu_j)$). Note, entretanto, que a taxa de crescimento do número de máquinas é menor do que a taxa de crescimento da taxa média de chegada. Whitt (1992) mostrou que ao manter φ_j constante em (6.4), a medida de congestão $P(Wq_j > 0)$ também se mantém aproximadamente constante (i.e., a probabilidade de um tempo de espera positivo). Este resultado é suportado por teoremas do limite de tráfego pesado, e foi observado em experimentos computacionais. Em particular, Whitt (1992) mostrou que para uma fila $GI/G/m_j$, tem-se a seguinte aproximação:

$$\varphi_j \approx \frac{(ca_j + cs_j)}{2\sqrt{m_j}\,E(Wq_j | Wq_j > 0)} \tag{6.5}$$

onde $E(Wq_j | Wq_j > 0)$ também é uma medida de congestão. Ela corresponde ao tempo médio de espera na fila da estação *j*, dado que o tempo de espera é maior do que zero. Combinando (6.4) e (6.5), obtém-se um outro exemplo de nível de serviço para a estação *j* definido como:

$$\eta_j = \frac{(1-\rho_j)m_j}{(ca_j + cs_j)} \approx \frac{1}{2E(Wq_j | Wq_j > 0)} \tag{6.6}$$

A equação (6.6) implica que para um dado nível de serviço η_j, a medida de congestão $E(Wq_j | Wq_j > 0)$ é mantida aproximadamente constante. Assumindo-se que ca_j e cs_j sejam independentes a mudanças de capacidade na rede (conforme seção 4.2), segue de (6.6) que ao se adicionar máquinas e aumentar a taxa média de chegada na estação *j*, a utilização média cresce para o mesmo nível de serviço. Seja a seguinte restrição para cada estação *j* da rede:

$$\frac{(1-\rho_j)m_j}{(ca_j + cs_j)} \geq G_j \qquad (6.7)$$

onde G_j é um limitante inferior para o nível de serviço na estação *j* (note que G_j também é um limitante superior para a medida de congestão $E(Wq_j | Wq_j > 0)$). O parâmetro G_j pode ser visto como a mínima flexibilidade desejada na estação *j*. Para maiores valores de ca_j e cs_j, o número de máquinas na estação *j* deve ser aumentado, de maneira a satisfazer a flexibilidade desejada. Desta maneira pode-se analisar, por exemplo, os *trade-offs* entre flexibilidade e capacidade no sistema.

A expressão (6.7) sugere que, em certos casos, pode-se satisfazer a mínima flexibilidade desejada nas estações sem alterar o número total de máquinas na rede. Ao particionar apropriadamente a planta em sub-plantas com um *mix* de produtos mais homogêneo, pode-se obter *scv* menores, diga-se ca_j^i e cs_j^i para cada estação *j* de cada sub-planta *i*, tal que a lado esquerdo de (6.7) cresça para todo *j* e *i*.

Projeto de fábrica focalizada

O problema do *projeto de fábrica focalizada* envolve a alocação de produtos em linhas de produção, e a alocação de capacidade nas estações de cada linha. Este problema pode ser visto como um exemplo da classe *SP3* (veja, p.e., o problema *SP3.1* na seção 1), e é diferente dos problemas discutidos nos capítulos 4 e 5, onde apenas a alocação de capacidade estava envolvida.

Um tópico de pesquisa interessante é desenvolver modelos de otimização para analisar o *trade-off* entre a partição de linhas de produtos e a duplicação de máquinas num projeto de fábrica focalizada. Poderia se incorporar nestes modelos restrições de complexidade dos pontos de vista do produto e da estação, tais como (6.1) e (6.7) discutidas acima. Estas restrições podem ajudar a representar a previsibilidade desejada nos *leadtimes* de produtos e a flexibilidade desejada nas estações do sistema.

Uma pesquisa inicial baseada nessas idéias (Bitran e Sarkar, 1994c) revela um resultado inesperado:

> *"Ao contrário do senso comum, o número de máquinas requerido pode ser menor quando as linhas de produção são particionadas (comparado a quando elas são colocadas em uma única planta)."*

Esse resultado sugere que pode-se reduzir a complexidade da rede (lado esquerdo de (6.1)), ou aumentar a flexibilidade das estações (lado esquerdo de (6.7)), apenas particionando a planta em linhas de produto. Uma pesquisa adicional poderia investigar, por meio de análise de curvas de *trade-off*, a estabilidade das partições ótimas. Por exemplo, a sensibilidade da solução a mudanças na taxa média de chegada de produtos, ou a mudanças no nível de serviço desejado nas estações. Também poderia se considerar situações particulares onde classes de produtos privilegiados deveriam ter baixos *leadtimes*, ou percorrer roteiros através de estações com altos níveis de serviço.

Anexo 1 - Parâmetro de variabilidade do processo de partida num sistema de fila *GI/G/1*

Este anexo mostra que o *scv* do processo de partida de um sistema de fila *GI/G/1* com disciplina *FCFS* é dado, conforme (3.29), por:

$$cd = ca + 2\rho^2 cs - 2\rho(1-\rho)\mu E(Wq) \qquad \text{(A1.1)}$$

onde *ca* e *cs* são os *scv* dos processos de chegada e serviço, $\rho = \lambda/\mu$ é a utilização média, e $E(Wq)$ é o tempo médio de espera em fila. Considere a seguinte notação:

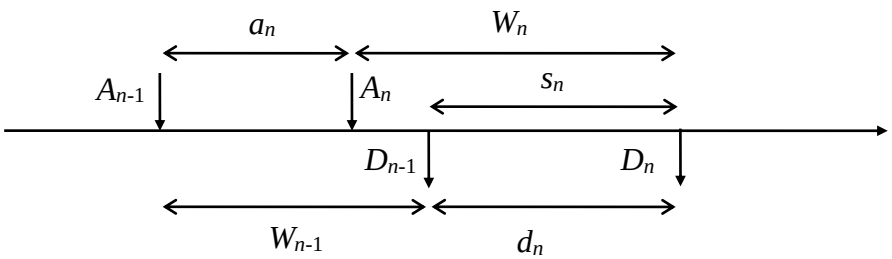
A_n instante em que o *n*-ésimo *job* chega no sistema
D_n instante em que o *n*-ésimo *job* parte do sistema
s_n tempo de serviço do *n*-ésimo *job* no sistema
W_n tempo de permanência (em serviço e em fila) do *n*-ésimo *job* no sistema

Os intervalos de tempo entre as chegadas e as partidas do (*n*-1)-ésimo e o *n*-ésimo *job* no sistema são denotados respectivamente por $a_n = A_n - A_{n-1}$ e $d_n = D_n - D_{n-1}$. Também se denota por I_n o intervalo de tempo entre a partida do (*n*-1)-ésimo *job* e a chegada do *n*-ésimo *job* no sistema. Note que $I_n \geq 0$ é o tempo em que o servidor fica ocioso esperando a chegada do *n*-ésimo *job*, dado por:

$$I_n = \left[A_n - D_{n-1}\right]^+$$

onde $[z]^+ = z$ se $z > 0$, e $[z]^+ = 0$, caso contrário. A figura 33 abaixo ilustra todas essas medidas para o (*n*-1)-ésimo e o *n*-ésimo *job* no sistema, nos casos em que $A_n < D_{n-1}$ (i.e., $I_n = 0$) e $A_n > D_{n-1}$ ($I_n > 0$).

(a) $I_n = 0$



(b) $I_n > 0$

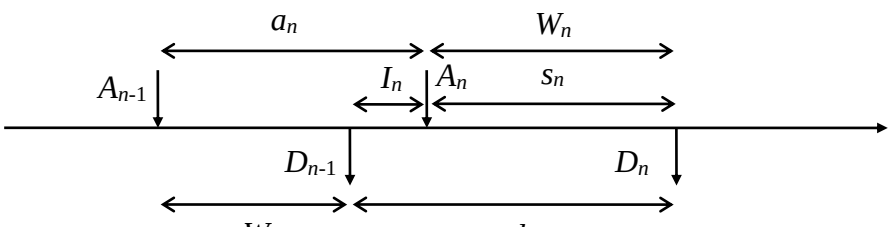


Figura 33 - Medidas do (n-1)-ésimo e do n-ésimo *job* no sistema *GI/G/1* quando: (a) $I_n = 0$ e (b) $I_n > 0$

Note na figura 33 que d_n também pode ser descrito como:

$$d_n = D_n - D_{n-1} = \left[A_n - D_{n-1}\right]^+ + s_n = I_n + s_n \quad \text{(A1.2)}$$

e que a expressão:

$$W_n - I_n = W_{n-1} - a_n + s_n \quad \text{(A1.3)}$$

sempre é válida. Aplicando o valor esperado em ambos os lados de (A1.2) e (A1.3), e no limite $n \to \infty$, obtém-se respectivamente:

$$E(d) = E(I) + E(s) \quad \text{(A1.4)}$$

$$E(I) = E(a) - E(s) \quad \text{(A1.5)}$$

(lembre-se que, no limite $n \to \infty$, tem-se que $E(W_n) = E(W_{n-1}) = E(W)$). Note em (A1.5) que para a ociosidade média $E(I)$ ser positiva, segue que $E(a) > E(s)$, ou equivalentemente, $\lambda < \mu$ (condição de equilíbrio do sistema). Combinando (A1.4) e (A1.5), obtém-se:

$$E(d) = E(a) \quad \text{(A1.6)}$$

o que era esperado, uma vez que em equilíbrio a taxa média de partida $1/E(d)$ deve ser igual a taxa média de chegada $\lambda = 1/E(a)$.

Como I_n e s_n são variáveis aleatórias independentes, aplicando a variância em ambos os lados de (A1.2) e no limite $n \to \infty$, tem-se que:

$$V(d) = V(I) + V(s)$$

Usando (A1.6) e lembrando que $V(I) = E(I^2) - E(I)^2$ e $V(s) = E(s^2) - E(s)^2$, o *scv* do processo de partida é dado por:

$$cd = \frac{V(d)}{E(d)^2} = \frac{V(I) + V(s)}{E(a)^2} = \frac{E(I^2) - E(I)^2 + E(s^2) - E(s)^2}{E(a)^2}$$

Como $E(I)^2 = (E(a) - E(s))^2 = E(a)^2 - 2E(a)E(s) + E(s)^2$ (veja (A1.5)) e $\rho = E(s) / E(a)$, obtém-se:

$$cd = \frac{E(I^2) - E(a)^2 + 2E(a)E(s) - 2E(s)^2 + E(s^2)}{E(a)^2}$$
$$= \frac{E(I^2)}{E(a)^2} - 1 + 2\rho - 2\rho^2 + \frac{V(s) + E(s)^2}{E(a)^2}$$

Dado que $-1 + 2\rho - \rho^2 = -(1-\rho)^2$ e observando que, ao multiplicar o numerador e o denominador do termo $(V(s) + E(s)^2) / E(a)^2$ por $E(s)^2$, obtém-se $\rho^2 cs + \rho^2$, chega-se em:

$$cd = \frac{E(I^2)}{E(a)^2} - (1-\rho)^2 + \rho^2 cs \qquad \text{(A1.7)}$$

A seguir mostra-se que ao relacionar $E(I^2)$ com $E(W)$, reduz-se (A1.7) em (A1.1). Rescrevendo (A1.3) como:

$$W_n - s_n - I_n = W_{n-1} - a_n$$

e elevando-se ao quadrado ambos os lados da expressão acima, obtém-se:

$$(W_n - s_n)^2 - 2(W_n - s_n)I_n + I_n^2 = W_{n-1}^2 - 2W_{n-1}a_n + a_n^2$$

ou,

$$W^2 - 2W_n s_n + s_n^2 - 2(W_n - s_n)I_n + I_n^2 = W_{n-1}^2 - 2W_{n-1}a_n + a_n^2 \qquad \text{(A1.8)}$$

Observe na figura 33 que se $W_n - s_n > 0$, então $I_n = 0$, e se $W_n - s_n = 0$, então $I_n > 0$. Logo, o termo $2(W_n - s_n)I_n$ em (A1.8) é sempre nulo e portanto, pode ser eliminado. Além disso, como W_n-s_n e s_n são independentes, então:

$$E((W_n - s_n)s_n) = E(W_n - s_n)E(s_n) = (E(W_n) - E(s_n))E(s_n) = E(W_n)E(s_n) - E(s_n)^2$$

Também pode-se escrever $E((W_n - s_n)s_n)$ como:

$$E((W_n - s_n)s_n) = E(W_n s_n - s^2) = E(W_n s_n) - E(s_n^2) = E(W_n s_n) - V(s_n) - E(s_n)^2$$

lembrando que $V(s_n) = E(s_n^2) - E(s_n)^2$. Igualando as duas expressões acima, o termo $E(W_n s_n)$ pode ser escrito como: $E(W_n s_n) = E(W_n)E(s_n) + V(s_n)$. Usando este resultado ao aplicar o valor esperado em ambos os lados de (A1.8), obtém-se

$$E(W_n^2) - 2E(W_n)E(s_n) - 2V(s_n) + E(s_n^2) + E(I_n^2) = E(W_{n-1}^2) - 2E(W_{n-1})E(a_n) + E(a_n^2)$$

que, no limite $n \to \infty$, se reduz a:

$$-2E(W)E(s) - 2V(s) + E(s^2) + E(I^2) = -2E(W)E(a) + E(a^2) \qquad \text{(A1.9)}$$

Note que a expressão (A1.9) relaciona $E(I^2)$ com $E(W)$. Finalmente, ao isolar $E(I^2)$ e substituir em (A1.7), obtém-se:

$$cd = \frac{E(a^2) + 2V(s) - E(s^2) - 2E(W)(E(a) - E(s))}{E(a)^2} - (1-\rho)^2 + \rho^2 cs$$

que, com V(a) = $E(a^2)$ - $E(a)^2$, $V(s) = E(s^2)$ - $E(s)^2$ e $\rho = E(s) / E(a)$, resulta em:

$$cd = ca + 2\rho^2 cs + 2\rho(1-\rho) - 2E(W)\lambda(1-\rho)$$

ou, ao substituir $E(W)$ por $E(Wq) + E(s)$, na expressão (A1.1).

Anexo 2 - Aproximações para medidas de desempenho num sistema de fila *GI/G/1*

Este anexo inicialmente apresenta algumas aproximações para medidas de desempenho num sistema de fila *GI/G/1*, tais como o número médio de *jobs* $E(L)$ e o tempo médio de espera $E(Wq)$. Estas aproximações são alternativas para as expressões (3.33) e (3.34) do capítulo 3. Também são discutidos detalhes das aproximações em (3.35a) e (3.35b) para $V(L)$ e $V(Wq)$ de cada estação. Finalmente, é mostrado que a expressão (3.30) é sempre uma subestimativa do valor real de *cd*.

Considere novamente a expressão (A1.9) do anexo 1, que relaciona $E(W)$ e $E(I^2)$. Isolando-se $E(W)$, obtém-se:

$$E(W) = \frac{E(a^2) + 2V(s) - E(s^2) - E(I^2)}{2(E(a) - E(s))}$$

e, lembrando que $V(s) = E(s^2) - E(s)^2$, resulta:

$$\begin{aligned} E(W) &= \frac{E(a^2) + E(s^2) - 2E(s)^2 - E(I^2)}{2(E(a) - E(s))} \\ &= \frac{E(a^2) + E(s^2) - 2E(a)E(s) - E(I^2)}{2(E(a) - E(s))} + E(s) \end{aligned} \qquad \text{(A2.1)}$$

Apesar de não se conhecer $E(I^2)$, pode-se definir limitantes para $E(I^2)$. Por exemplo, lembrando que para qualquer variável aleatória x, tem-se que $E(x^2) \geq E(x)^2$, segue que $E(I^2) \geq E(I)^2 = (E(a) - E(s))^2$, conforme (A1.5), e portanto:

$$\begin{aligned} E(W) &\leq \frac{E(a^2) + E(s^2) - 2E(a)E(s) - (E(a) - E(s))^2}{2(E(a) - E(s))} + E(s) \\ &= \frac{V(a) + V(s)}{2(E(s) - E(a))} + E(s) \\ &= \frac{(ca + \rho^2 cs)}{2\lambda(1 - \rho)} + E(s) \end{aligned} \qquad \text{(A2.2)}$$

onde $\rho = \lambda/\mu = E(s)/E(a)$. Também pode ser mostrado que $E(I^2) \geq E(I)^2 E(a^2) / E(a)^2$ (veja p.e. Buzacott e Shanthikumar, 1993). Substituindo este outro limitante em (A2.1), obtém-se:

$$E(W) \leq \frac{\rho(2-\rho)ca + \rho^2 cs}{2\lambda(1-\rho)} + E(s) \qquad \text{(A2.3)}$$

que é melhor do que (A2.2), uma vez que $\rho(2-\rho) < 1$. Para que os lados direitos de (A2.2) e (A2.3) sejam exatos para o caso particular de um sistema de fila *M/G/1*, isto é,

$$E(W)_{M/G/1} = \frac{\lambda E(s^2)}{2(1-\rho)} + E(s)$$

então deve-se multiplicá-los respectivamente pelos fatores $\rho^2(1+cs)/(1+\rho^2 cs)$ e $\rho(1+cs)/(2-\rho+\rho cs)$. Neste caso, obtém-se as duas aproximações para $E(W)$:

$$E(W) \approx \frac{\rho^2(1+cs)}{(1+\rho^2 cs)} \frac{(ca+\rho^2 cs)}{2\lambda(1-\rho)} + E(s) \qquad \text{(A2.4)}$$

$$E(W) \approx \frac{\rho(1+cs)}{(2-\rho+\rho cs)} \frac{\rho(2-\rho)ca+\rho^2 cs}{2\lambda(1-\rho)} + E(s) \qquad \text{(A2.5)}$$

Note que, usando (3.13a)-(3.13c), é fácil obter aproximações para $E(L)$, $E(Lq)$ e $E(Wq)$, a partir das aproximações em (A2.4) e (A2.5). Por exemplo, $E(L)$ é simplesmente $\lambda E(W)$, conforme (3.13c), e $E(Wq)$ corresponde ao lado direito de (A2.4) e (A2.5) sem o termo $E(s)$, conforme (3.13b). Diversos experimentos computacionais têm mostrado que (A2.4) e (A2.5) resultam em boas aproximações quando $ca \leq 2$. Por outro lado, para sistemas com muita variabilidade no processo de chegada (i.e., valores altos de ca), aproximações baseadas apenas nos dois primeiros momentos das distribuições, como é o caso de (A2.4) e (A2.5), não têm bom desempenho (veja p.e. a discussão em Buzacott e Shanthikumar, 1993, p.75).

Para ilustrar, a tabela 17 compara os resultados aproximados obtidos para $E(L)$ usando (A2.4), (A2.5) e (3.33) (fórmula de Kraemer e Lagenbach-Belz), em sistemas de filas $E_p/E_q/1$ (i.e., com distribuições de Erlang de ordem p e q para os processos de chegada e serviço) com ρ = 0,9. Note que para p = 1 (sistema *M/G/1*), as aproximações são exatas (veja última coluna da tabela). Para analisar a qualidade das aproximações para p < 1, a tabela inclui os resultados obtidos com simulação (conforme Buzacott e Shanthikumar, 1993, p.76).

Tabela 17 - Comparação das aproximações para $E(L)$ em (A2.4), (A2.5) e (3.33) com simulação para um sistema de fila $E_p/E_q/1$ com ρ = 0,9

		p	4	3	2	1
		Ca	0,25	0,333	0,5	1
q	*cs*	Método				
4	0,25	(A2.4)	2,81	3,16	3,86	5,96
		(A2.5)	2,81	3,16	3,86	5,96
		(3.33)	2,76	3,13	3,86	5,96
		Simulação	2,77	3,12	3,83	
3	0,333	(A2.4)	3,11	3,47	4,17	6,30
		(A2.5)	3,12	3,47	4,18	6,30
		(3.33)	3,10	3,47	4,20	6,30

		Simulação	3,10	3,45	4,16	6,30
2	0,5	(A2.4)	3,73	4,09	4,81	6,98
		(A2.5)	3,74	4,10	4,82	6,98
		(3.33)	3,77	4,14	4,88	6,98
		Simulação	3,75	4,11	4,82	
1	1	(A2.4)	5,64	6,02	6,76	9,00
		(A2.5)	5,66	6,03	6,77	9,00
		(3.33)	5,80	6,17	6,90	9,00
		Simulação	5,74	6,11	6,83	

As três aproximações ficam muito próximas dos resultados de simulação (o maior desvio em relação à simulação foi de apenas 1,74%). Para outros estudos de aproximações para $E(L)$ em sistemas *GI/G/1*, veja Buzacott e Shanthikumar (1993) e Whitt (1993).

A seguir mostra-se como obter $V(L)$ e $V(Wq)$ em (3.35a) e (3.35b). Seja c_{Wq} o *scv* de Wq. Note que:

$$c_{Wq} = \frac{V(Wq)}{E(Wq)^2} = \frac{E(Wq^2) - E(Wq)^2}{E(Wq)^2} = \frac{E(Wq^2)}{E(Wq)^2} - 1 \qquad \text{(A2.6)}$$

Pode-se rescrever $E(Wq)$ como:

$$E(Wq) = E(Wq|Wq > 0)P(Wq > 0) + E(Wq|Wq = 0)P(Wq = 0) = E(D)P(Wq > 0)$$

$$E(Wq^2) = E(Wq^2|Wq > 0)P(Wq > 0) + E(Wq^2|Wq = 0)P(Wq = 0) = E(D^2)P(Wq > 0)$$

onde $D = (Wq \mid Wq > 0)$ é o tempo de espera dado que o servidor está ocupado (i.e., o atraso). Substituindo estas expressões em (A2.6), obtém-se:

$$c_{Wq} = \frac{E(D^2)P(Wq > 0)}{E(D)^2 P(Wq > 0)^2} - 1 = \frac{V(D) + E(D)^2}{E(D)^2 P(Wq > 0)} - 1 = \frac{c_D + 1 - P(Wq > 0)}{P(Wq > 0)}$$

e portanto,

$$V(Wq) = E(Wq)^2 c_{Wq} = E(Wq)^2 \frac{c_D + 1 - P(Wq > 0)}{P(Wq > 0)} \qquad \text{(A2.7)}$$

que coincide com (3.35b). Para aproximar $P(Wq > 0)$ em (A2.7), Whitt (1983a) sugeriu utilizar a aproximação de Kraemer e Lagenbach-Belz, que resulta em:

$$P(Wq > 0) = \rho + (ca - 1)\rho(1 - \rho)h(\rho, ca, cs)$$

onde:

Note que esta fórmula é exata para sistemas *M/G/1*, produzindo corretamente $P(Wq > 0) = \rho$. Para aproximar o *scv* c_D em (A2.7), Whitt (1983a) utilizou o c_D exato de um sistema *M/G/1* com distribuição hiperexponencial de ordem 2 para o tempo de serviço s, se $cs \geq 1$, e distribuição Erlang de ordem k, se $cs = 1/k < 1$. A expressão é dada por:

$$c_D = 2\rho - 1 + \frac{4(1-\rho)E(s^3)}{3(cs+1)^2 E(s)^3} \qquad \text{(A2.8)}$$

Para evitar uma aproximação em função do terceiro momento $E(s^3)$ do tempo de serviço, a razão $E(s^3) / E(s)^3$ em (A2.8) foi substituída por:

$$\frac{E(s^3)}{E(s)^3} \approx \begin{cases} 3cs(1+cs), & \text{se } cs \geq 1 \\ (2cs+1)(cs+1), & \text{se } cs < 1 \end{cases}$$

que baseia-se nas distribuições hiperexponencial de ordem 2 e Erlang de ordem *k*. Note que em sistemas *M/M/1* (cs = 1), obtém-se corretamente o terceiro momento de uma exponencial $E(s^3) = 6E(s)^3$ que, ao ser substituído em (A2.8), resulta corretamente em $c_D = 1$.

Pode-se obter $V(L)$ a partir de $V(Wq)$. Inicialmente, note que, para um sistema *M/G/1* com disciplina *FCFS* e em equilíbrio, tem-se que:

$$E(L^2|W=w) = E(L|W=w)^2 + V(L|W=w) = (\lambda w)^2 + \lambda w$$

onde $W = Wq + s$ (lembre-se que a média e a variância de um processo de Poisson são iguais a sua taxa de chegada, λ, multiplicada pelo valor do tempo de permanência, w). Seja $f_W(w)$ a função densidade de probabilidade do tempo de permanência *W*. Logo,

$$\begin{aligned} E(L^2) &= \int_0^\infty E(L^2|W=w) f_W(w) dw = \int_0^\infty \left(\lambda^2 w^2 + \lambda w\right) f_W(w) dw = \lambda^2 E(W^2) + \lambda E(W) \\ &= \lambda^2 E(Wq^2) + 2\lambda^2 E(Wq)E(s) + \lambda^2 E(s^2) + \lambda E(Wq) + \lambda E(s) \\ &= \lambda^2 \left(V(Wq) + E(Wq)^2\right) + 2\lambda^2 E(Wq)E(s) + \lambda^2 \left(V(s) + E(s)^2\right) + \lambda E(Wq) + \rho \\ &= \lambda E(Wq) + \rho + \rho^2 cs + \lambda^2 V(Wq) + \lambda^2 E(Wq)^2 + 2\lambda^2 E(Wq)E(s) + \lambda^2 E(s)^2 \\ &= \lambda E(Wq) + \rho + \rho^2 cs + \lambda^2 V(Wq) + \left(\lambda(E(Wq) + E(s))\right)^2 \end{aligned}$$

e, portanto,

$$V(L) = E(L^2) - E(L)^2 = \lambda E(Wq) + \rho + \rho^2 cs + \lambda^2 V(Wq)$$

que coincide com (3.35a). Esta expressão, exata para um sistema *M/G/1*, pode ser utilizada como uma aproximação para um sistema *G/G/1*. Outras aproximações para $V(L)$ e $V(Wq)$ podem ser encontradas em Whitt (1983a), (1983b) e (1993)). Conforme reportado nestes artigos, as aproximações (3.33)-(3.35), utilizadas no *software QNA* da *AT&T Bell Laboratories*, têm produzido resultados precisos para sistemas *GI/G/1* não muito variáveis (ca < 5) e com níveis de utilização não muito baixas (ρ > 0,3).

Convém salientar que as aproximações para $E(Wq)$ em (3.34), (A2.4) e (A2.5), e a aproximação para $V(Wq)$ em (A2.7), reproduzem corretamente as equações de Pollaczek-Khinchine (Kleinrock, 1975, Askin e Standridge, 1993) para sistemas *M/G/1* (ca = 1):

$$E(Wq) = \frac{\rho(1+cs)}{2\mu(1-\rho)} = \frac{\lambda E(s^2)}{2(1-\rho)} \qquad \text{(A2.9)}$$

$$V(Wq) = \frac{\lambda E(s^3)}{3(1-\rho)} + \frac{\lambda^2 E(s^2)^2}{4(1-\rho)^2} \qquad \text{(A2.10)}$$

Uma observação final neste anexo é que, ao substituir $E(I^2)$ em (A1.7) pelo limitante inferior $E(I)^2 E(a^2) / E(a)^2$ utilizado em (A2.3), obtém-se:

$$\begin{aligned} cd &\geq \frac{(E(a) - E(s))^2 E(a^2) / E(a)^2}{E(a)^2} - (1-\rho)^2 + \rho^2 cs \\ &= \frac{(E(a)^2 - 2E(a)E(s) + E(s)^2)}{E(a)^2} \frac{E(a^2)}{E(a)^2} - (1-\rho)^2 + \rho^2 cs \\ &= (1-\rho)^2 \frac{(V(a) + E(a)^2)}{E(a)^2} - (1-\rho)^2 + \rho^2 cs \\ &= (1-\rho)^2 ca + \rho^2 cs \end{aligned}$$

que corresponde à aproximação (3.30). Isso mostra que (3.30) é sempre uma subestimativa para o valor real de *cd*.

Anexo 3 - Aproximações para medidas de desempenho num sistema de fila *GI/G/m*

Este anexo apresenta o refinamento proposto em Whitt (1993) das aproximações de *E*(*Wq*) e *V*(*Wq*) definidas em (3.41) e (3.42) para sistemas *GI/G/m* (a extensão para *E*(*L*) e *V*(*L*) é similar). Apesar de envolverem maiores esforços computacionais, as aproximações abaixo podem ser sensivelmente superiores àquelas apresentadas em (3.41) e (3.42), dependendo do sistema *GI/G/m*, conforme os extensivos resultados computacionais em Whitt (1993).

O valor esperado *E*(*Wq*) para um sistema *GI/G/m* é dado por:

$$E(Wq) = \phi(\rho, ca, cs, m)\frac{(ca+cs)}{2}E(Wq)_{M/M/m} \qquad \text{(A3.1)}$$

onde: $$\phi(\rho, ca, cs, m) = \begin{cases} \dfrac{4(ca-cs)}{4ca-3cs}\phi_1(m,\rho) + \dfrac{cs}{4ca-3cs}\psi\left(\dfrac{ca+cs}{2}, m, \rho\right), & \text{se } ca \geq cs \\ \dfrac{cs-ca}{2ca+2cs}\phi_3(m,\rho) + \dfrac{cs+3ca}{2ca+2cs}\psi\left(\dfrac{ca+cs}{2}, m, \rho\right), & \text{se } ca < cs \end{cases}$$

$$\psi(c, m, \rho) = \begin{cases} 1, & \text{se } c \geq 1 \\ \phi_4(m,\rho)^{2(1-c)} & \text{se } c < 1 \end{cases}$$

$$\phi_1(m,\rho) = 1 + \gamma(m,\rho)$$

$$\phi_2(m,\rho) = 1 - 4\gamma(m,\rho)$$

$$\phi_3(m,\rho) = \phi_2(m,\rho)\exp\left(\frac{-2(1-\rho)}{3\rho}\right)$$

$$\phi_4(m,\rho) = \min\left\{1, \frac{\phi_1(m,\rho)+\phi_3(m,\rho)}{2}\right\}$$

$$\gamma(m,\rho) = \min\left\{0.24, \frac{(1-\rho)(m-1)(\sqrt{4+5m}-2)}{16m\rho}\right\}$$

A variância *V*(*Wq*) para um sistema *GI/G/m* é aproximada usando a expressão (A2.7) do anexo 2, com c_D definido por (A2.8), *E*(*Wq*) agora definido por (A3.1), e *P*(*Wq* > 0) definido por:

$$P(Wq > 0) = \min\{1, \pi\}$$

onde: $$\pi = \begin{cases} \pi_1, & se \quad m \le 6 \text{ ou } \gamma \le 0{,}5 \text{ ou } ca \ge 1 \\ \pi_2, & se \quad m \ge 7 \text{ e } \gamma \ge 1 \text{ e } ca < 1 \\ \pi_3, & se \quad m \ge 7 \text{ e } ca < 1 \text{ e } 0{,}5 < \gamma < 1 \end{cases}$$

$$\pi_1 = \rho^2 \pi_4 + (1 - \rho^2)\pi_5$$

$$\pi_2 = ca\,\pi_1 + (1 - ca)\pi_6$$

$$\pi_3 = 2(1 - ca)(\gamma - 0{,}5)\pi_2 + \left(1 - \left[2(1 - ca)(\gamma - 0{,}5)\right]\right)\pi_1$$

$$\pi_4 = \min\left\{1, \frac{1 - \phi\left((1 + cs)(1 - \rho)\sqrt{m} / (ca + cs)\right)}{1 - \phi\left((1 - \rho)\sqrt{m}\right)} P(Wq_{M/M/m} > 0)\right\}$$

$$\pi_5 = \min\left\{1, \frac{1 - \phi\left(2(1 - \rho)\sqrt{m} / (1 + ca)\right)}{1 - \phi\left((1 - \rho)\sqrt{m}\right)} P(Wq_{M/M/m} > 0)\right\}$$

$$\pi_6 = 1 - \phi(\gamma)$$

$$z = \frac{ca + cs}{1 + cs}\ , \qquad \gamma = \frac{m - m\rho - 0{,}5}{\sqrt{m\rho z}}$$

Anexo 4 – Eliminação de arcos de realimentação imediata

Este anexo revisa o procedimento utilizado por Kuehn (1979) e Whitt (1983a) para reconfigurar a rede de maneira a eliminar arcos de realimentação imediata (lembre-se que se $q_{jj} > 0$, então ocorre uma realimentação imediata na estação j). A eliminação desses arcos melhora a qualidade das aproximações apresentadas na seção 3.2 do capítulo 3.

Considere que a estação j é tal que $q_{jj} > 0$. Logo, cada *job*, ao completar serviço na estação j, volta imediatamente para a estação j com probabilidade q_{jj}. Para eliminar a realimentação imediata, cada visita à estação j de um *job* vindo de outra estação, mais todas as possíveis revisitas imediatas subsequentes deste *job* à estação j, são analisadas como uma única visita. Para isso, o tempo médio de serviço desta visita única deve ser escolhido de maneira a compensar a eliminação das revisitas. Note que isso é equivalente a colocar o *job* no começo da fila da estação j, ao invés de colocá-lo no final da fila, toda vez que ele estiver revisitando a estação j, devido à uma realimentação imediata.

Não é difícil mostrar que, ao modificar desta maneira o tempo médio de serviço para eliminar o arco de realimentação imediata da estação j, o procedimento não altera a distribuição do número de *jobs* L_j na estação j, nem o tempo médio de permanência $E(W_j)$ na estação j, mas altera a distribuição do tempo de permanência W_j (isto pode ser mostrado, por exemplo, ao provar a lei de Little em (3.13c)).

O primeiro passo do procedimento é substituir q_{jj}, $E(s_j)$ e cs_j pelos novos parâmetros (figura 34):

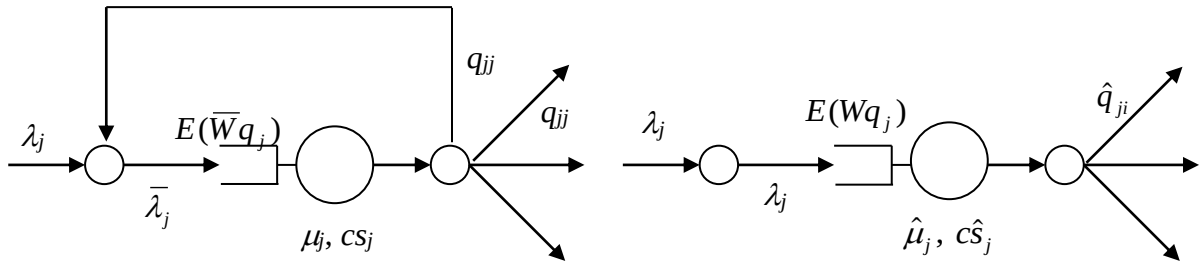
$$\hat{q}_{jj} = 0 \tag{A4.1}$$

$$\hat{q}_{ij} = \frac{q_{ij}}{1 - q_{jj}} \tag{A4.2}$$

$$E(\hat{s}_j) = \frac{E(s_j)}{1 - q_{jj}} \qquad \text{ou} \qquad \hat{\mu}_j = (1 - q_{jj})\mu_j \tag{A4.3}$$

$E(Wq_j)$

$$c\hat{s}_j = q_{jj} + (1 - q_{jj})cs_j \tag{A4.4}$$



(a) (b)

Figura 34 – Estação *j* antes (a) e depois (b) da eliminação do arco de realimentação imediata

As expressões (A4.1) e (A4.2) são óbvias. A seguir mostra-se como as expressões (A4.3) e (A4.4) foram obtidas. Seja $\hat{s}_j$ o tempo total de serviço de um *job* na estação *j*, incluindo os tempos de serviço das suas, diga-se, z_j revisitas imediatas à estação *j*. Note que $\hat{s}_j$ pode ser descrito como a soma de $z_j + 1$ variáveis aleatórias s_j *iid*:

$$\hat{s}_j = s_{j,1} + s_{j,2} + \ \ldots\ + s_{j,z_j} + s_{j,z_j+1}$$

onde $s_{j,1}, s_{j,2}, \ldots, s_{j,z_j}$ são os tempos de serviço das z_j revisitas e s_{j,z_j+1} é o tempo de serviço da primeira visita na estação *j*. Por conveniência, defina $z_j^{'} = z_j + 1$. Note que $z_j^{'}$ tem distribuição geométrica com probabilidade 1 - q_{jj}, logo,

$$E(z_j^{'}) = \frac{1}{1-q_{jj}} \qquad \text{e} \qquad V(z_j^{'}) = \frac{q_{jj}}{(1-q_{jj})^2}$$

A média da soma de z'_j variáveis aleatórias *iid* é dada por (p.e., Ross (1993)):

$$E(\hat{s}_j) = E(z_j^{'})E(s_j) = \frac{1}{1-q_{jj}} E(s_j)$$

que coincide com (A4.3), e a variância da soma de z'_j variáveis aleatórias *iid* é:

$$V(\hat{s}_j) = E(z_j^{'})V(s_j) + V(z_j^{'})E(s_j)^2 = \frac{1}{1-q_{jj}} V(s_j) + \frac{q_{jj}}{(1-q_{jj})^2} E(s_j)^2$$

Portanto, $c\hat{s}_j = V(\hat{s}_j) / E(\hat{s}_j)^2 = (1-q_{jj})cs_j + q_{jj}$, que coincide com (A4.4).

O segundo e último passo do procedimento é ajustar algumas medidas de desempenho obtidas depois de aplicar o método de decomposição da seção 3.2, com os parâmetros $\hat{q}_{ij}$, $E(\hat{s}_j)$ e $c\hat{s}_j$ em (A4.1)-(A4.4). Por exemplo:

$$\overline{\lambda}_j = \frac{\lambda_j}{1-q_{jj}}$$

$$E(\overline{W}q_j) = \frac{E(Wq_j)}{E(z^{'})} = (1-q_{jj})E(Wq_j)$$

onde λ_j e $E(Wq_j)$ são os valores obtidos pelo método de decomposição para a taxa média de chegada e o tempo médio de espera em fila na estação *j*, a partir dos parâmetros $\hat{q}_{ij}$, $E(\hat{s}_j)$ e $c\hat{s}_j$,

e $\overline{\lambda}_j$ e $E(\overline{W}q_j)$ denotam a taxa média de chegada *real* e o tempo médio de espera em fila *por visita* na estação *j* (lembre-se que $E(z'_j)$ corresponde ao número médio de visitas de um *job* na estação *j*) (veja figura 34). Para maiores detalhes deste procedimento, o leitor é referido a Kuehn (1979) e Whitt (1983a).

Referências bibliográficas

Albin, S. L. (1982). "Poisson approximations for superposition arrival processes in queues." *Management Science* 28(2), 126-137.

Albin, S. L. (1984). "Approximating a point process by a renewal process, II: Superposition arrival processes of queues." *Operations Research* 30(5), 1133-1162.

Albin, S. L. (1986). "Delays for customers from different arrival streams to a queue." *Management Science* 32(4), 329-340.

Askin, R. G. e Krisht, A. (1994). "Optimal operation of manufacturing systems with controlled work-in-process levels." *International Journal of Production Research* 32(7), 1637-1653.

Askin, R. G. e Standridge, C. R. (1993). *Modeling and analysis of manufacturing systems*, John Wiley & Sons, New York.

Baskett, F., Chandy, K. M., Muntz, R. R., Palacios, F. G. (1975). "Open, closed, and mixed networks of queues with different classes of customers." *Journal of the Association for Computing Machinery* 22(2), 248-260.

Bazaraa, M. S., Sherali, H. D. e Shetty, C. M. (1993). *Nonlinear programming: Theory and algorithms*, 2nd.ed., John Wiley & Sons, New York.

Bitran, G. R. e Dasu, S. (1992). "A review of open queueing network models of manufacturing systems." *Queueing Systems* 12, 95-134.

Bitran, G. R. e Morabito, R. (1995a). "Um exame dos modelos de redes de filas abertas aplicados a sistemas de manufatura discretos---Parte I." *Gestão & Produção* 2(2), 192-219.

Bitran, G. R. e Morabito, R. (1995b). "Um exame dos modelos de redes de filas abertas aplicados a sistemas de manufatura discretos---Parte II." *Gestão & Produção* 2(3), 297-320.

Bitran, G. R. e Morabito, R. (1995c). "Modelos de otimização de redes de filas abertas para projeto e planejamento de *job-shops*." *Pesquisa Operacional* 15(1), 1-22.

Bitran, G. R. e Morabito, R. (1995d). "Manufacturing systems design: Trade-off curve analysis." *Working Paper* WP#3805-95, Sloan School of Management, MIT, Cambridge, MA.

Bitran, G. R. e Morabito, R. (1996). "Open queueing networks: Optimization and performance evaluation models for discrete manufacturing systems." *Production and Operations Management* 5(2), 163-193.

Bitran, G. R. e Morabito, R. (1997). "An overview of trade-off curves in manufacturing system design." Aceito para publicação no *Production and Operations Management*.

Bitran, G. R. e Sarkar, D. (1994a). "Throughput analysis in manufacturing networks." *European Journal of Operational Research* 74, 448-465.

Bitran, G. R. e Sarkar, D. (1994b). "Targeting problems in manufacturing queueing networks - An iterative scheme and convergence." *European Journal of Operational Research* 76, 501-510.

Bitran, G. R. e Sarkar, D. (1994c). "Focused factory design: Complexity, capacity and inventory trade-offs." *Technical Memorandum*, AT&T Bell Lab., Holmdel, NJ.

Bitran, G. R. e Tirupati, D. (1988). "Multiproduct queueing networks with deterministic routing: Decomposition approach and the notion of interference." *Management Science* 34(1), 75-100.

Bitran, G. R. e Tirupati, D. (1989a). "Trade-off curves, targeting and balancing in manufacturing queueing networks." *Operations Research* 37(4), 547-564.

Bitran, G. R. e Tirupati, D. (1989b). "Capacity planning in manufacturing networks with discrete options." *Annals of Operations Research* 17, 119-136.

Bitran, G. R. e Tirupati, D. (1989c). "Approximations for product departures from a single server station with batch processing in multi-product queues." *Management Science* 35(7), 851-878.

Bitran, G. R. e Tirupati, D. (1991). "Approximations for network of queues with overtime." *Management Science* 37(3), 282-300.

Boxma, O. J., Rinnooy Kan, A. e Van Vliet, M. (1990). "Machine allocation problems in manufacturing networks." *European Journal of Operational Research* 45, 47-54.

Bramson, M. (1994). "Instability of FIFO queueing networks." *Annals of Applied Probability* 4(2), 414-431.

Brooke, A., Kendrick, D., Meeraus, A. (1992). *GAMS: A user's guide, Release 2.25*, The Scientific Press, San Francisco, CA.

Brown, E. (1988). "IBM combines rapid modeling technique and simulation to design PCB factory-of-the-future." *Industrial Engineering*, June, 23-90.

Buzacott, J. A. e Shanthikumar, J. G. (1992). "Design of manufacturing systems using queueing models." *Queueing Systems* 12, 135-214.

Buzacott, J. A. e Shanthikumar, J. G. (1993). *Stochastic models of manufacturing systems*, Prentice-Hall, Englewood Cliffs, NJ.

Buzacott, J. A. e Yao, D. D. (1986). "Flexible manufacturing systems: A review of analytical models." *Management Science* 32(7), 890-905.

Calabrese, J. M. (1992). "Optimal workload allocation in open networks of multiserver queues." *Management Science* 38(12), 1792-1802.

Cao, X. (1994). "Performance sensitivity analysis of open Markovian queueing networks". *European Journal of Operational Research* 76, 529-551.

Chase, R. B. e Aquilano, N. J. (1992). *Production and operations management - A life cycle approach*, Irwin, Homewood, MA.

Dai, J. G., Yeh, D. H. e Zhou, C. (1997). "The QNET method for re-entrant queueing networks with priority disciplines". *Operations Research* 45(4), 610-623.

Dallery, Y. e David, R. (1986). "Operational analysis of multiclass queueing networks." *Proceedings of the 25th IEEE Conference Decision and Control*, Athens, Greece, 1728-1732.

Dallery, Y. e Stecke, K. E. (1990). "On the optimal allocation of servers and workloads in closed queueing networks." *Operations Research* 38(4), 694-703.

Denning, P. J. e Buzen, J. P. (1978). "The operational analysis of queueing networks." *Association for Computing Machinery (ACM) Computing Surveys* 10(3), 225-261.

De Treville, S. (1992). "Time is money". *OR/MS Today*, Outubro.

Deuermeyer, B. L., Curry, L. e Feldman, R. M. (1993). "An automatic modeling approach to the strategic analysis of semiconductor fabrication facilities". *Production and Operations Management* 2(3), 195-220.

Disney, R. L. e Konig, D. (1985). "Queueing networks: A survey of their random processes." *SIAM Review* 27(3), 335-403.

Erlang, A. K. (1917). "Solution of some problems in the theory of probabilities of some significance in automatic telephone exchanges." *Post Office Electrical Engineer's Journal* 10, 189-197.

Fernandes, F. C. F. (1991). "Concepção de um sistema de controle da produção para a manufatura celular". *Tese de Doutorado*, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, SP.

Frenk, H., Labbe, M., Van Vliet, M., Zhang, S. (1994). "Improved algorithms for machine allocation in manufacturing systems." *Operations Research* 42(3), 523-530.

Gershwin, S. B. (1994). *Manufacturing systems engineering*, Prentice-Hall, Englewood Cliffs, NJ.

Harel, A. e Zipkin, P. (1987). "Strong convexity results for queueing systems." *Operations Research* 35(4), 405-418.

Harrison, J. e Nguyen, V. (1990). "The QNET method for two-moment analysis of open queueing networks." *Queueing Systems* 6, 1-32.

Harrison, J. e Nguyen, V. (1993). "Brownian models of multiclass queueing networks: Current status and open problems." *Queueing Systems* 13, 5-40.

Harrison, J. e Pich, M. T. (1996). "Two-moment analysis of open queueing networks with general workstation capabilities". *Operations Research* 44(6), 936-950.

Harrison, J. e Williams, R. (1987). "Brownian models of open queueing networks with homogeneous customer populations." *Stochastic* 22, 77-115.

Hax, A. C. e Candea, D. (1984). *Production and inventory management*, Prentice-Hall, New Jersey.

Ho, Y. C. (1987). "Performance evaluation and perturbation analysis of discrete event dynamic systems." *IEEE Transactions on Automatic Control* 32(7), 563-572.

Ho, Y. C. e Cao, X. (1983). "Perturbation analysis and optimization of queueing networks." *Journal of Optimization Theory and Applications* 40(4), 559-582.

Hsu, L. F., Tapiero, C. S. e Lin, C. (1993). "Network of queues modeling in flexible manufacturing systems: A survey." *Recherche Operationnelle/Operations Research* 27(2), 201-248.

Jackman, J. e Johnson, E. (1993). “The role of queueing network models in performance evaluation of manufacturing systems.” *Journal of the Operational Research Society* 44(8), 797-807.

Jackson, J. R. (1957). “Networks of waiting lines.” *Operations Research* 5(4), 518-521.

Jackson, J. R. (1963). “*Job* shop-like queueing systems.” *Management Science* 10(1), 131-142.

Karmarkar, U. S., Kekre, L., Freeman, S. (1985). “Lotsizing and leadtime performance in a manufacturing cell.” *Interfaces* 15(2), 1-9.

Kelly, F. P. (1975). “Networks of queues with customers of different types.” *Journal of Applied Probability* 12, 542-554.

Kelly, F. P. (1979). *Reversibility and Stochastic Processes*, John Wiley & Sons, New York.

Kleinrock, L. (1964). *Communication nets: stochastic message flow and delay*, Dover Publishing, New York.

Kleinrock, L. (1975). *Queueing systems, vol 1: Computer applications*, John Wiley & Sons, New York.

Kleinrock, L. (1976). *Queueing systems, vol 2: Computer applications*, John Wiley & Sons, New York.

Kobayashi, H. (1974). “Application of the diffusion approximation to queueing networks I: Equilibrium queue distributions.” *Journal of the Association for Computing Machinery* 21(2), 316-328.

Koenigsberg, E. (1982). “Twenty five years of cyclic queues and closed queue networks: A review.” *Journal of the Operational Research Society* 33(7), 605-619.

Kouvelis, P. e Lee, H. (1995). “An improved algorithm for optimizing a closed queueing network model of a flexible manufacturing system.” *IIE Transactions* 27, 1-8.

Kouvelis, P. e Tirupati, D. (1991). “Approximate performance modeling and decision making for manufacturing systems: A queueing network optimization framework.” *Journal of Intelligent Manufacturing* 2, 107-134.

Krajewski, L. J. e Ritzman, P. L. (1990). *Operations management: Strategy and analysis*, 2nd.ed., Addison-Wesley, Reading, MA.

Kuehn, P. J. (1979). “Approximate analysis of general networks by decomposition.” *IEEE Transactions Commun.* 27(1), 113-126.

Law, A. M. e Haider, S. W. (1989). “Selecting simulation software for manufacturing applications: Practical guidelines e software survey.” *Industrial Engineering*, May, 33-46.

Law, A. M. e McComas, M. G. (1989). “Pitfalls to avoid in the simulation of manufacturing systems.” *Industrial Engineering*, May, 28-69.

Lemoine, A. J. (1977). “Networks of queues - a survey of equilibrium analysis.” *Management Science* 24(4), 464-481.

Leung, Y. e Suri, R. (1990). “Performance evaluation of discrete manufacturing systems”. *IEEE Control Systems Magazine*, Junho, 77-86.

Lynes, K. e Miltenburg, J. (1994). “The application of an open queueing network to the analysis of cycle time, variability, throughput, inventory and cost in the batch production system of a microelectronics manufacturer”. *International Journal of Production Economics* 37, 189-203.

Mascolo, M., Frein, Y. e Dallery, Y. (1996). “Analytical method for performance evaluation of Kanban controlled production systems”. *Operations Research* 44(1), 50-64.

Nemhauser, G. L e Wolsey, L. A. (1988). *Integer and combinatorial optimization*, John Wiley & Sons, New York.

Papadopoulos. H. T. e Heavey, C. (1996). “Queueing theory in manufacturing systems analyisis and design: A classification of models for production and transfer lines.” *European Journal of Operational Research* 92, 1-27.

Pujolle, G. e Ai, W. (1986). “A solution for multiserver and multiclass open queueing networks.” *Information Systems and Operations Research* 24(3), 221-230.

Reiman, M. I. (1984). “Open queueing networks in heavy-traffic.” *Mathematics of Operational Research* 9, 441-458.

Reiman, M. I. (1990). “Asymptotically exact decomposition approximations for open queueing networks.” *Operations Research Letters* 9(6), 363-370.

Reiser, M. e Kobayashi, H. (1974). “Accuracy of diffusion approximations for some queueing systems.” *IBM Journal of Research and Development* 18, 110-124.

Ross, S. M. (1993). *Introduction to Probability Models*, 5th ed., Academic Press, Inc., San Diego, CA.

Schriber, T. J. (1991). *An introduction to simulation using GPSS/H*, John Wiley e Sons, New York.

Sevcik, K. C., Levy, A. I., Tripathi, S. K., Zahoran, J. L. (1977). “Improving approximations of aggregated queueing network systems” in *Computer Performance* (eds. K. Chandy and M. Reiser), North-Holland, 1-22.

Segal, M. e Whitt, W. (1989). “A queueing network analyzer for manufacturing” in *Teletraffic Science for New Cost-Effective Systems, Networks and Services*, ITC-12, M. Bonatti (ed.), Elsevier, North-Holland, Amsterdam, 1146-1152.

Schweitzer, P. e Seidmann, A. (1991). “Optimizing processing rates for flexible manufacturing systems.” *Managment Science* 37(4), 454-466.

Seidmann, A., Schweitzer, P. J., Shalev-Oren, S. (1987). “Computerized closed queueing network models of flexible manufacturing systems: A comparative evaluation.” *Large Scale Systems* 12, 91-107.

Shanthikumar, J. G. e Buzacott, J. A. (1981). “Open queueing network models of dynamic *job* shops.” *International Journal of Production Research* 19(3), 255-266.

Shanthikumar, J. G. e Buzacott, J. A. (1984). “The time spent in a dynamic *job* shop.” *European Journal of Operational Research* 17, 215-226.

Shanthikumar, J. G. e Yao, D. D. (1987). “Optimal server allocation in a system of multi-server stations.” *Management Science* 33(9), 1173-1180.

Shanthikumar, J. G. e Yao, D. D. (1988). “On server allocation in multiple center manufacturing systems.” *Operations Research* 36(2), 333-342.

Skinner, W. (1974). “The focused factory.” *Harvard Business Review*, May-June, 113-121.

Stecke, K. E. e Raman, N. (1994). “Production planning decisions in flexible manufacturing systems with random material flows”. *IIE Transactions* 26(5), 2-17.

Stecke, K. E. e Solberg, J. J. (1985). "The optimality of unbalancing both workloads and machine group size in closed queueing networks of multi-server queues." *Operations Research* 33, 882-910.

Sundarraj, R. P., Sundararaghavan, P. S., Fox, D. R. (1994). "Optimal server acquisition in open queueing networks." *Journal of the Operational Research Sociecty* 45(5), 549-558.

Suresh, S. e Whitt, W. (1990). "The heavy-traffic bottleneck phenomenon in open queueing networks." *Operations Research Letters* 9(6), 355-362.

Suri, R. (1989). "Perturbation analysis: The state of the art and research issues explained via the G/G/1 queue." *Proceedings IEEE* 77, 114-137.

Suri, R. e De Treville, S. (1991). "Full speed ahead: A look at rapid modeling technology in operations management". *OR/MS Today* 18, June, 34-42.

Suri, R. e De Treville, S. (1993). "Rapid modeling: The use of queueing models to support time-based competitive manufacturing", in *Operations Research in Production Planning and Control*, ed. G. Fandel, T. Gulledge e A. Jones, Springler-Verlag, 21-30.

Suri, R., Diehl, G. W., Dean, R. (1986). "Quick and easy manufacturing systems analysis using ManuPlan." *Proceedings Spring IIE Conference*, Dallas, TX, 195-205.

Suri, R., Sanders, J. L., Kamath, M. (1993). "Performance evaluation of production networks" in *Handbooks in OR/MS*, S. C. Graves (ed.), vol.4, Elsevier, North-Holland, Amsterdam.

Suri, R., Diehl, G. W., De Treville, S., Tomsicek, M. J. (1995). "From CAN-Q to MPX: Evolution of queueing software for manufacturing." *Interfaces* 25(5), 128-150.

Tang, C. S. e Yoo, S. (1991). "System planning and configuration problems for optimal system design." *European Journal of Operational Research* 54, 163-175.

Tetzlaff, U. (1996). "A queueing network model for flexible manufacturing systems with tool management". *IIE Transactions* 28, 309-317.

Tijms, H. C. (1986). *Stochastic modeling and analysis: A computational approach*, John Wiley & Sons, New York.

Van Vliet, M. e Rinnooy Kan, A. (1991). "Machine allocation algorithms for *job* shop manufacturing." *Journal of Intelligent Manufacturing* 2, 83-94.

Vinod, B. e Solberg, J. (1991). "Optimal design of flexible manufacturing systems." *International Journal of Production Research* 23(6), 1141-1151.

Walrand, J. (1990). "Queueing networks" in *Handbooks in OR/MS*, D. P. Heyman and M. J. Sobel (ed.), vol.2, Elsevier, North-Holland, Amsterdam.

Wein, L. M. (1990a). "Capacity allocation in generalized Jackson networks." *Operations Research Letters* 15, 215-242.

Wein, L. M. (1990b). "Scheduling networks of queues: Heavy traffic analysis of a two-station network with controllable inputs." *Operations Research* 38(6), 1065-1078.

Whitt, W. (1982). "Approximating a point process by a renewal process, I: Two basic methods." *Operations Research* 30(1), 125-147.

Whitt, W. (1983a). "The queueing network analyzer." *The Bell System Technical Journal* 62(9), 2779-2815.

Whitt, W. (1983b). "The queueing network analyzer." *The Bell System Technical Journal* 63(9), 1911-1979.

Whitt, W. (1984). “Open and closed models for networks of queues.” *AT&T Bell Laboratories Technical Journal* 63(9), 1911-1979.

Whitt, W. (1988). “A light-traffic approximation for single-class departures from multi-class queues.” *Management Science* 34(11), 1333-1346.

Whitt, W. (1992). “Understating the efficiency of multi-server service systems.” *Management Science* 38(5), 708-723.

Whitt, W. (1993). “Approximations for the *GI/G/m* queue.” *Production and Operations Management* 2(2), 114-161.

Whitt, W. (1994). “Towards better multi-class parametric-decomposition approximations for open queueing networks.” *Annals of Operations Research* 48, 221-248.

Whitt, W. (1995). “Variability functions for parametric decomposition approximations of queueing networks”. *Management Science* 41(10), 1704-1715.

Yao, D. D. e Buzacott, J. A. (1986). “The exponentialization approach to flexible manufacturing system models with general processing times.” *European Journal of Operational Research* 24, 410-416.

Índice