

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE SISTEMAS

MODELAGEM DE UM SISTEMA KANBAN ATRAVÉS DE
PROGRAMAÇÃO ESTOCÁSTICA E MÉTODOS DE RESOLUÇÃO

LÚCIA FILOMENA DE ALMEIDA, GUIMARÃES†

Orientador

Prof. Dr. Vinicius Amaral Armentano†

Tese apresentada à Faculdade de Engenharia Elétrica e de
Computação da Universidade Estadual de Campinas -
UNICAMP, como parte dos requisitos exigidos para
obtenção do título de Doutor em Engenharia Elétrica.

- AGOSTO 1997 -

0713847

Este exemplar contém a redação final da tese defendida por <u>Lúcia F. A. Guimarães</u> e aprovada pela Comissão Julgada em <u>15 / 08 / 1997</u> <u>Vinicius A. Armentano</u> Orientador
--

G947m

UNICAMP

UNIDADE	73C
N.º CHAMADA:	
	Unicamp
	G947m
EX.	
	31854
	281/97
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	R\$ 11,00
DATA	18/10/97
N.º CPD	

CM-00102687-7

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

G947m Guimarães, Lúcia Filomena de Almeida
Modelagem de um sistema kanban através de
programação estocástica e métodos de resolução / Lúcia
Filomena de Almeida Guimarães.--Campinas, SP: [s.n.],
1997.

Orientador: Vinícius Amaral Armentano.
Tese (doutorado) - Universidade Estadual de Campinas,
Faculdade de Engenharia Elétrica e de Computação.

1. Programação estocástica. 2. Just-in-time. 3.
Heurística. I. Armentano, Vinícius Amaral. II.
Universidade Estadual de Campinas. Faculdade de
Engenharia Elétrica e de Computação. III. Título.

*Para fazer um grande sonho se tornar realidade é
preciso primeiro ter um grande sonho.*

- Hans Selye -

AGRADECIMENTOS

- a Vinícius A. Armentano pela dedicada orientação durante o desenvolvimento desta tese e estímulo à pesquisa;
- a Michel Gendrau pela orientação durante a minha estadia no Canadá e pela paciência nas nossas “falhas” de comunicação;
- a minha mãe pelo amor, carinho e confiança;
- a Rita, Guilherme e Livia pelo carinho, apoio e constante interesse;
- ao tio Abel, tia Deda e Sô pelo apoio nas horas mais difíceis;
- a Cíntia, Fran, Débora e Regina pela valiosa amizade;
- a Renata pela paciência e ajuda nas indecisões;
- ao Walcir e Luciana pela paciência e apoio computacional;
- aos professores e funcionários do Densis pelo apoio;
- a Capes pelo apoio financeiro;
- finalmente a todos que colaboraram indiretamente na realização deste trabalho.

RESUMO

O sistema kanban de produção é um sistema do tipo “puxar” para controle de chão de fábrica. Ele é motivado pelo conceito de produção *Just-in-Time* e tem como objetivo programar a produção, reduzindo ao mínimo os níveis de estoque. Neste trabalho considera-se um ambiente de produção com estrutura de montagem multi-estágio e restrições de capacidade dos itens componentes e do item final. Um modelo estocástico de otimização com recurso que incorpora as incertezas da demanda do item final e capacidade de produção é proposto. Demanda não atendida é permitida no estágio final do modelo. Um método do tipo *Branch and Cut* é desenvolvido para obter soluções ótimas. Soluções sub-ótimas são obtidas através de um método heurístico e de busca tabu. Testes computacionais são apresentados.

ABSTRACT

The kanban production system is a pull-type system for shop floor control. It is based on the concept of Just-in-Time production and has the purpose of scheduling production, reducing inventory levels to a minimum. In this work, a multi-stage assembly-structure production setting with capacity constraints for the component items and the final item is considered. A stochastic optimization model with recourse that takes into account uncertainties of demand of the final item and production capacity is proposed. Backlog is allowed in the final stage. A Branch and Cut method is developed in order to obtain optimal solutions. Sub-optimal solutions are obtained by a heuristic method and by tabu search. Computational tests are reported.

SUMÁRIO

INTRODUÇÃO.....	1
CAPÍTULO 1. A METODOLOGIA <i>JUST IN TIME</i> E O SISTEMA KANBAN DE PRODUÇÃO.....	4
1.1. Introdução.....	4
1.2. A metodologia <i>Just in Time</i>	4
1.3. O sistema kanban de produção.....	7
1.3.1. Sistema kanban de um cartão.....	8
1.3.2. Sistema kanban de dois cartões.....	10
1.4. Revisão Bibliográfica.....	12
CAPÍTULO 2. MODELOS DE PROGRAMAÇÃO ESTOCÁSTICA PARA O SISTEMA KANBAN DE PRODUÇÃO.....	15
2.1. Introdução.....	15
2.2. Modelos de Recurso (ou em dois estágios).....	16
2.3. Um modelo estocástico para o sistema kanban de produção.....	17
2.4. Extensões do Modelo.....	24

2.4.1. Modelo com containers parcialmente cheios na produção e no estoque.....	24
2.4.2. Modelo com vários itens finais e demanda externa para todos os estágios de produção.....	29
CAPÍTULO 3. UM ALGORITMO ÓTIMO PARA O MODELO ESTOCÁSTICO.....	32
3.1. Introdução.....	32
3.2. O método L-Shaped inteiro para programação estocástica inteira com recurso.....	32
3.3. O método L-Shaped inteiro aplicado ao modelo estocástico de um sistema kanban de produção.....	36
3.3.1. O algoritmo L-shaped inteiro para o modelo estocástico de um sistema kanban.....	40
3.4. Implementação e Análise.....	43
3.4.1. Geração dos problemas.....	43
3.4.2. Análise dos resultados.....	46
CAPÍTULO 4. O MÉTODO HEURÍSTICO E A BUSCA TABU.....	48
4.1. Introdução.....	48
4.2. O Método Heurístico.....	48
4.3. A Busca Tabu.....	51

4.3.1. Testes preliminares.....	52
4.4. Implementação e Análise.....	53
4.4.1. Um limitante inferior.....	53
4.4.2. Estruturas pequenas de produção.....	54
4.4.3. Novas estruturas de produção.....	57
4.4.4. Análise dos resultados.....	64
4.4.5. Um novo ponto de partida para o método heurístico.....	68
CAPÍTULO 5. UM MODELO DETERMINÍSTICO PARA O SISTEMA KANBAN DE PRODUÇÃO.....	73
5.1. Introdução.....	73
5.2. Um modelo determinístico.....	73
5.3. Implementação e análise dos resultados.....	77
CAPÍTULO 6. CONCLUSÕES.....	81
APÊNDICE A. DEMONSTRAÇÕES.....	84
APÊNDICE B. COMPLEXIDADE DA HEURÍSTICA.....	93
BIBLIOGRAFIA.....	103

INTRODUÇÃO

Uma empresa pode ser considerada como um sistema integrado por diversas partes relacionadas entre si, que trabalham em harmonia com a finalidade de alcançar uma série de objetivos. As empresas que tem suas atividades voltadas para a produção de bens podem ser divididas, basicamente, em três áreas que interagem entre si: marketing, finanças e produção.

Marketing e finanças ocupam-se das necessidades da área de produção que, por sua vez, é a responsável pela transformação da matéria prima em produtos. Assim, a produção necessita de um sistema de gerenciamento que coordene todas as atividades envolvidas no processo produtivo, desde a aquisição de matéria prima até a entrega do produto final.

O sistema de planejamento e controle da produção (PCP) tem este objetivo e divide o planejamento em três níveis distintos: estratégico, que consiste no estabelecimento dos objetivos globais da empresa a longo prazo; tático, responsável pelo planejamento do fluxo de materiais, gerenciamento da demanda e decisões sobre nível de estoque e o nível operacional que procura executar os planos em termos de programação detalhada da produção e compras.

O *Just in Time* pode ser descrito como uma metodologia de gerenciamento para otimizar a utilização dos recursos da empresa, buscando constantemente a eficiência, integração do sistema e simplificação do processo produtivo. A implementação desta metodologia causa um grande impacto no sistema PCP. No nível operacional, onde seus efeitos são mais visíveis, o *Just in Time* mostra-se extremamente adequado para o controle de chão de fábrica e compras de materiais. No nível tático reduz a complexidade do planejamento de necessidades de materiais através da redução do número de níveis da estrutura dos produtos e do número de componentes. Finalmente, no planejamento estratégico leva a mudanças no plano mestre e no planejamento agregado de produção para que haja uma utilização adequada de capacidade para um nível operacional suave de chão de fábrica.

O enfoque deste trabalho está nas atividades do *Just in Time* a nível de chão de fábrica, que possui como principal elemento o sistema kanban de produção. Este, é um sistema de informação cuja finalidade é programar e controlar harmoniosamente a produção. É do tipo “puxar”, onde os itens dos estágios predecessores são retirados à medida que o estágio sucessor os

consome em seu processo de produção. O objetivo básico do sistema kanban é minimizar o estoque em processo.

O sistema de produção considerado é um sistema multi-estágio com estrutura de montagem e um item final. Quando uma operação é realizada em um item este é enviado para o seu único estágio sucessor. A figura I.1 representa um exemplo deste sistema através de um grafo acíclico onde cada nó representa um estágio de produção. O item final é produzido no estágio zero. O arco (i, j) indica que os itens produzidos no estágio i são utilizados na produção do estágio j .

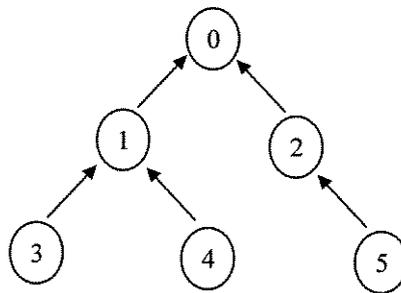


FIGURA I.1 - ESTRUTURA DE MONTAGEM

Neste trabalho é apresentado um modelo estocástico para um sistema kanban de produção. A programação estocástica com recurso é utilizada no intuito de tornar o planejamento de produção mais flexível pois possibilita a tomada de decisões perante incertezas. Assume-se que a demanda para o estágio final e a capacidade de produção para cada estágio são variáveis aleatórias discretas ao longo do horizonte de planejamento. Demanda não atendida é permitida no estágio final. O objetivo é determinar o número de kanbans em cada estágio que minimize o custo de estoque em todos os estágios e o custo de demanda não atendida no estágio final.

No capítulo 1 os principais elementos e objetivos da metodologia *Just in Time* são apresentados. O sistema kanban é definido e as diferenças entre os sistema kanban de um cartão e dois cartões são discutidas. Consta também deste capítulo a revisão bibliográfica sobre os principais artigos da área.

A programação estocástica inteira com recurso, bem como o modelo estocástico para o sistema kanban, são introduzidos no capítulo 2. Generalizações do modelo também são apresentadas.

O capítulo 3 trata do algoritmo ótimo para a resolução do modelo. Estratégias para a execução deste método são discutidas e resultados computacionais são analisados.

Um método heurístico é apresentado no capítulo 4, assim como uma implementação de busca tabu para a obtenção de melhores soluções. Estes métodos são testados e os resultados obtidos são discutidos.

No capítulo 5 um modelo determinístico é introduzido. Demanda não atendida não é permitida. O método heurístico e a implementação de busca tabu do capítulo 4 são modificados para este caso.

Finalmente, o capítulo 6 apresenta as conclusões finais bem como sugestões para trabalhos futuros.

CAPÍTULO 1

A METODOLOGIA *JUST-IN-TIME* E O SISTEMA KANBAN DE PRODUÇÃO

1.1 INTRODUÇÃO

Na metade da década de 70 os produtos japoneses começaram a chamar a atenção do mercado mundial pela sua qualidade e baixo preço. A indústria japonesa introduzia assim, novos conceitos e técnicas de produção.

Baixos estoques, rápido atendimento das necessidades do consumidor e alta qualidade são características visíveis destes processos de produção e são obtidos através da utilização da metodologia *Just in Time* ou simplesmente JIT.

Neste capítulo a metodologia JIT é definida. Apresenta-se o sistema kanban de produção e a revisão bibliográfica também é realizada.

1.2 A METODOLOGIA *JUST-IN-TIME*

O termo JIT descreve uma metodologia que engloba um conjunto de técnicas e conceitos que tem como finalidade a melhoria contínua do sistema de produção. Esta melhoria pode ser interpretada como uma otimização dos processos e procedimentos de produção através da redução sistemática dos desperdícios. Entende-se por desperdício todas as atividades realizadas na fábrica que não agregam valor à produção, tais como, desperdício de produzir itens defeituosos, desperdício de movimento e desperdício de estoque.

Esta metodologia também enfatiza o fluxo contínuo de produção, ou seja, os produtos fluem de forma suave e contínua através das diversas fases do processo produtivo. Seus principais objetivos são:

- **redução contínua dos níveis de estoque**

Tradicionalmente, os estoques são reconhecidos como um meio seguro de manter o processo de produção quando ocorrem problemas ou defeitos que poderiam ocasionar sua parada. O JIT entende que o excesso de estoque

reduz a necessidade de investigar e resolver os problemas que são encobertos pelo estoque. Em um programa JIT, o mínimo de materiais está em movimentação e qualquer problema no processo se evidenciará como uma perda de produção. Isto força problemas de produção e qualidade a virem à tona para serem solucionados. À medida que o problema é localizado e solucionado, a linha de produção é capaz de atender as necessidades do processo de manufatura sem excesso de estoques, aumentando dessa forma o número de vezes que o estoque é movimentado durante o ano. A rotatividade do estoque é uma medida de eficiência da empresa na administração de materiais.

- **redução ao mínimo dos tempos de preparação (*setup*) de máquinas**
A redução do tempo de preparação leva à redução do lote produzido e conseqüentemente do estoque. Além disso, contribui para a redução do *lead time* de produção que também é uma meta do JIT.
- **redução dos tamanhos de lotes**
Com o tamanho de lotes reduzidos obtém-se, como vantagem, a redução dos estoques intermediários e do produto final. Além disso, se houver problemas de qualidade em um lote e este tiver que ser descartado, o custo de perda é menor.
- **flexibilidade de manufatura**
O JIT aumenta a flexibilidade de resposta do sistema pela redução dos tempos envolvidos no processo. Esta redução permite que o sistema produtivo seja mais flexível em relação às variações da composição (*mix*) de produtos.

Segundo Smith (1989), a obtenção destes objetivos está associada à implementação de:

1. Controle de Qualidade Total

É a visão da qualidade na empresa como um todo, com objetivo de eliminar os problemas desta natureza. O JIT procura diminuir a importância da inspeção como função de garantia de qualidade, e aumentar a importância do controle de qualidade como função de prevenção. Nos sistemas tradicionais, existe um compromisso entre custo e qualidade, isto é, um produto de baixo custo é, em geral, incompatível com alta qualidade. Para o JIT, realizar tarefas corretamente assegura a qualidade do produto que por sua vez leva a uma redução dos custos associados a desperdícios e perdas.

2. Manufatura Celular

É uma forma de organizar a produção baseado na divisão dos produtos finais em famílias com processo de fabricação similar. Uma vez que as famílias são identificadas, há a formação de células de manufatura que tem como principais vantagens a redução do tempo gasto com a preparação de máquinas e movimentação de matérias.

3. Envolvimento dos Operários

Tradicionalmente, a gerência e engenharia são responsáveis pelo conhecimento que caracteriza a empresa. Estas pessoas possuem um alto grau de instrução e aos trabalhadores compete apenas seguir suas ordens. No JIT, os operários são estimulados a participar da resolução dos problemas de produção e garantir a manutenção de equipamentos. Os engenheiros e gerentes são o suporte para a resolução dos mesmos. Além disso, os operários são treinados para a operação de diferentes tipos de máquinas possibilitando a redução de custos operacionais e melhoria de qualidade.

4. Manutenção Preventiva

Esta tem como objetivo evitar paradas na fabricação. Durante os intervalos de produção os equipamentos são avaliados.

5. Carregamento Uniforme

Com o objetivo de manter o fluxo contínuo de produção, o plano mestre de produção fixa a produção diária do mesmo número de produtos finais, dentro de um determinado período de tempo (um mês, por exemplo), permitindo assim a regularidade e estabilidade na utilização de capacidade do equipamento (recursos produtivos).

6. Fluxo Balanceado de Trabalho

Em empresas que produzem produtos finais diferentes, é importante distribuir uniformemente a carga nos centros de trabalho para evitar sobrecarga em uns e ociosidade em outros. Por este motivo, adota-se um programa de produção, onde diariamente são produzidas a mesma proporção de cada produto durante o horizonte de planejamento determinado pelo carregamento uniforme.

7. JIT no Fornecedor

É uma extensão dos conceitos desta metodologia para “fora da fábrica”, tendo como objetivo obter um freqüente número de pequenas remessas de componentes / itens / matéria-prima com alta qualidade e apenas quando a fábrica precisar.

8. Sistema Kanban de Produção

O JIT possui um sistema de controle de chão de fábrica baseado na idéia que uma operação em um estágio dispara o início de uma operação em um estágio predecessor, ou seja, o material de produção é “puxado” pelo sistema à medida que se torna necessário. O mais famoso sistema de puxar é o sistema kanban de produção, criado pela Toyota, e que será discutido na próxima seção.

Em contraste com o sistema de puxar, empresas ocidentais empregam sistemas de produção do tipo “empurrar”. Nestes sistemas uma previsão de demanda, incluindo *lead time*, é determinada para cada estágio do processo produtivo. O processo de empurrar é controlado através do nível de estoque em cada estágio. Em geral, o nível de estoque é no mínimo igual ao estoque de segurança para a proteção contra erros de previsão. Sistemas MRP (Volmann *et al.* (1988), são considerados do tipo empurrar sendo seu objetivo determinar em cada período as necessidades em termos de componentes/matéria prima a fim de que o sistema de produção possa cumprir seu programa sem atraso.

A metodologia JIT é discutida em detalhes por Corrêa e Gianesi(1993) e Schmenner(1990)

1.3 O SISTEMA KANBAN DE PRODUÇÃO

Um dos elementos mais importantes do JIT é o sistema kanban de produção. Este é considerado um sistema de puxar, que se caracteriza pela retirada dos itens dos estágios predecessores à medida que o estágio sucessor os consome em seu processo de produção. O seu objetivo básico é minimizar o estoque em processo (Monden, 1983 e Schonberger, 1982).

A palavra kanban em japonês significa cartão. No sistema kanban, os cartões são utilizados para autorizar a produção ou transporte de uma certa quantidade de itens. O kanban contém informações sobre o tipo e quantidade de itens, a operação (produção ou transporte) que está autorizando e o local onde o item é armazenado.

Em um ambiente JIT podem existir, basicamente, dois tipos de sistema kanban: de um cartão ou de dois cartões.

1.3.1 SISTEMA KANBAN DE UM CARTÃO

Este sistema se caracteriza por possuir apenas um local de estoque entre duas estações de trabalho. Os passos envolvidos em sua operação são representados por números na Figura 1.1 e descritos abaixo

1. Um container de itens é selecionado para ser usado pela próxima estação de trabalho. O kanban ligado a este container é retirado e deixado na caixa de kanbans;
2. Quando o número de kanbans atinge um determinado nível, ou no final de cada período de tempo, estes são retirados da caixa de kanbans e afixados no quadro de kanbans;
3. Durante um período de tempo, cada vez que a estação de trabalho está apta a produzir, o quadro de kanbans é verificado. Cada kanban autoriza a produção de um novo container de itens;
4. Quando um container é preenchido na produção, ele é armazenado em um estoque intermediário, com um kanban ligado a ele, até que seja necessário na próxima estação de trabalho.

Note que o número de kanbans circulando entre a caixa de kanbans e o quadro de kanbans é invariante no tempo e o nível de estoque é limitado por este número de kanbans. Portanto, controlando o número de kanbans em circulação pode-se minimizar o custo de estoque.

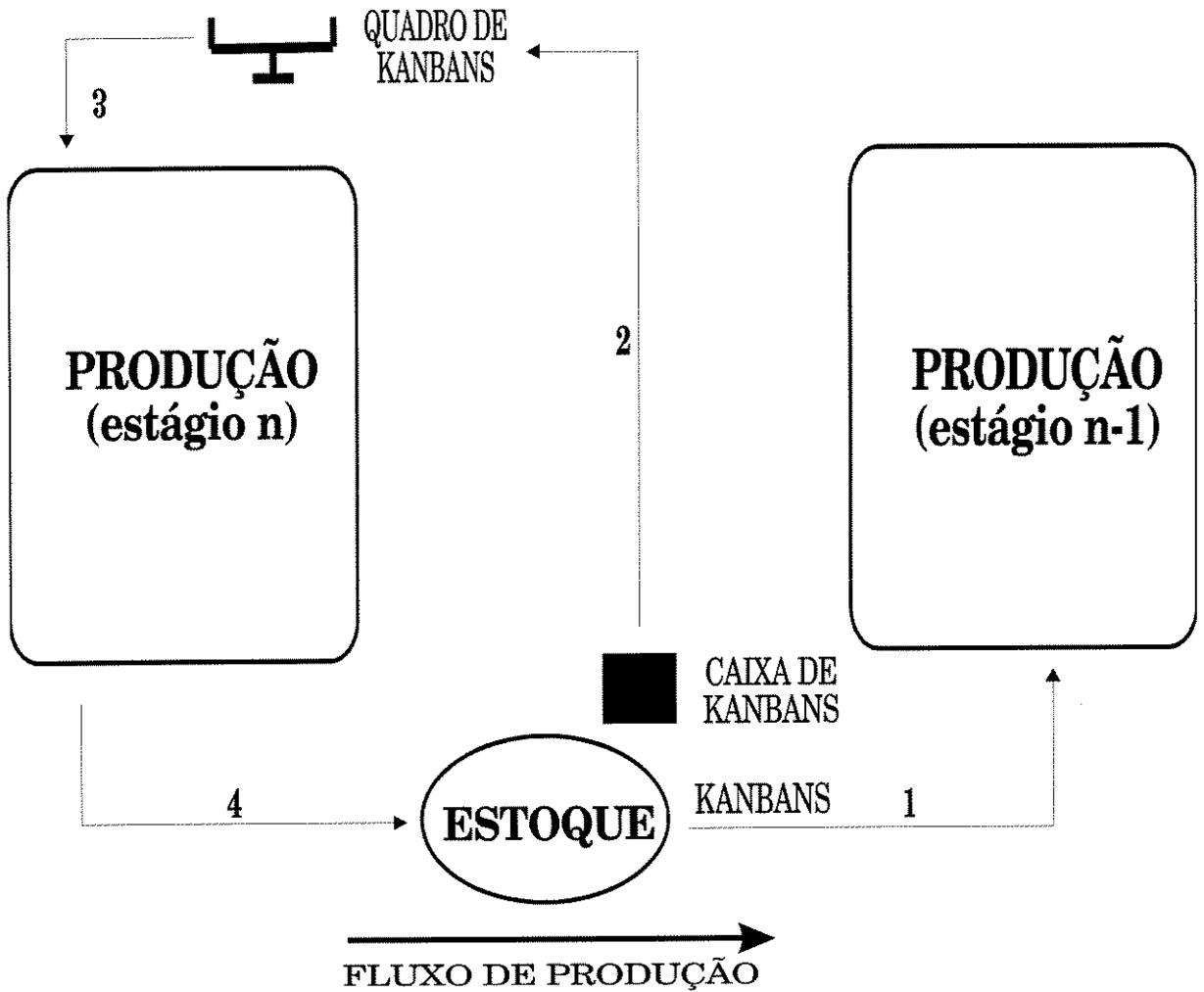


FIGURA 1.1 - SISTEMA KANBAN DE PRODUÇÃO COM UM CARTÃO

1.3.2 SISTEMA KANBAN DE DOIS CARTÕES

O ambiente de produção, neste sistema, tem como característica a existência de dois locais de estoque em cada centro de produção: um imediatamente antes da produção mantendo os itens do estágio predecessor (estoque de entrada), e o outro imediatamente após a produção contendo os itens produzidos (estoque de saída). O sistema opera com dois tipos de kanbans: requisição e produção. O kanban de requisição representa um pedido de itens ao estoque de saída de um estágio predecessor. É ele que determina o máximo de estoque intermediário. O kanban de produção indica o início de produção de um centro de trabalho. É o responsável pela programação e controle da produção.

Segundo Nahmias (1989) a operação deste sistema contém os passos, indicados nos locais apropriados na Figura 1.2:

1. quando o número de kanbans de requisição alcança um nível pré-determinado no quadro de kanbans de requisição, um trabalhador leva esses kanbans para o estoque de saída de seu estágio predecessor;
2. o trabalhador, no local de estoque de saída compara o número do item do kanban de requisição com o de produção;
3. se estes números são iguais, ele remove do container o kanban de produção, que é colocado no quadro de kanbans de produção no estágio anterior e associa ao container o kanban de requisição deixando-o no estoque de saída;
4. a produção, no estágio anterior é executada segundo uma prioridade estabelecida no quadro de kanbans de produção
5. um trabalhador transportará os containers do estoque de saída para o estoque de entrada do estágio posterior, onde ficarão até que sejam requisitados;
6. quando os itens começam a participar da produção no estágio posterior, o kanban de requisição é removido e colocado no quadro de kanbans de requisição

No sistema kanban de dois cartões o número de kanbans de produção é igual ao número de kanbans de requisição, pois não existe requisição se não há produção e vice-versa. O kanban de produção no sistema de 2 cartões desempenha o mesmo papel que o kanban no sistema de um cartão. Portanto, estes sistemas são semelhantes quando o objetivo é determinar o número de kanbans a ser injetado no sistema de produção. Yavuz e Sartir (1995) comentam que freqüentemente o sistema kanban de um cartão é usado como estágio de desenvolvimento para o sistema kanban de dois cartões.

Detalhes adicionais sobre os princípios deste sistema de produção podem ser encontrados em Kimura e Terada(1981), Monden (1983), Schoenberger (1982), Sugimori e Uchikawa (1977).

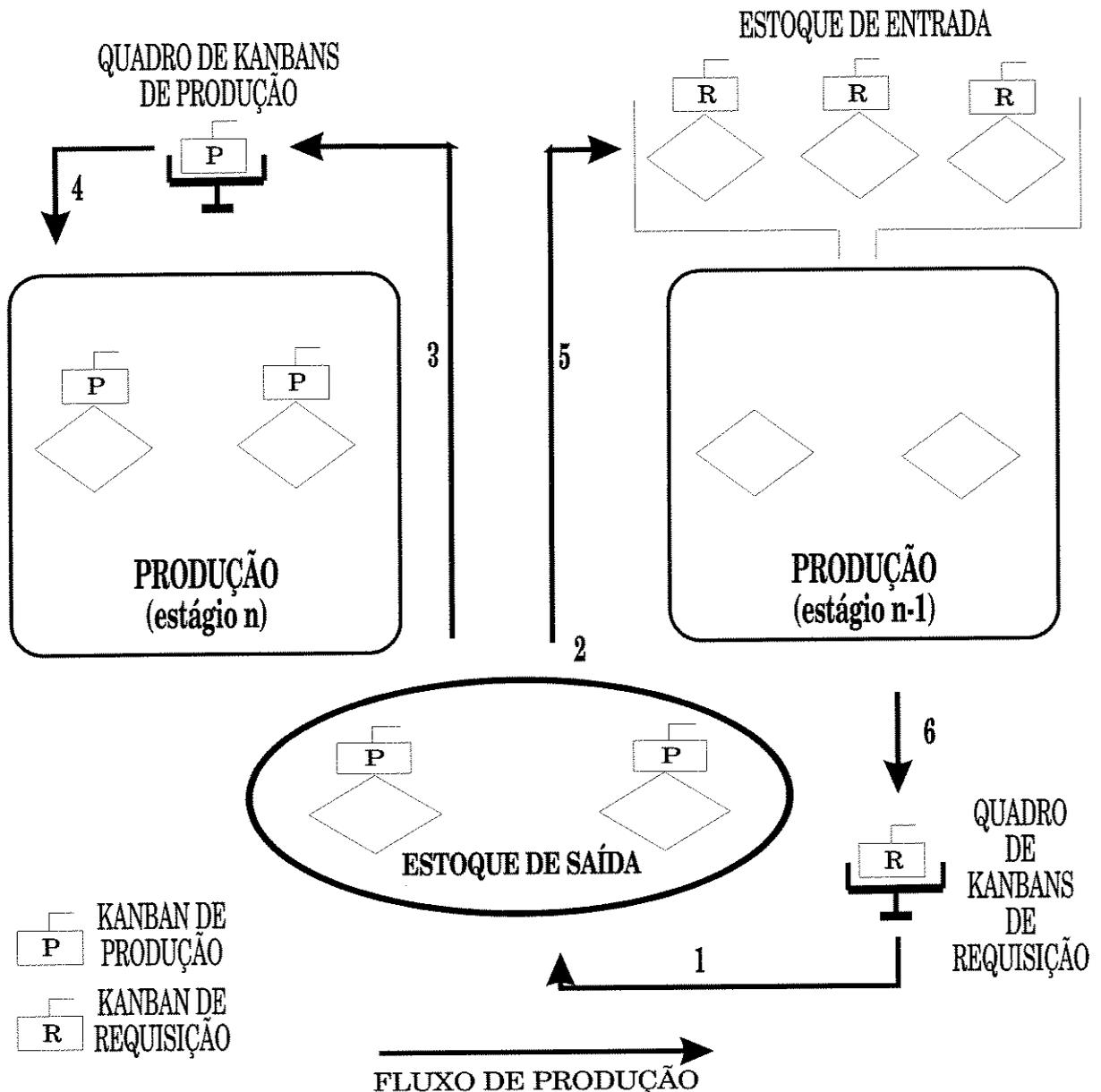


FIGURA 1.2 - SISTEMA KANBAN DE PRODUÇÃO COM DOIS CARTÕES

Este trabalho considera um sistema kanban de um cartão e tem como objetivo determinar o número de kanbans em cada estágio que minimize o custo de estoque em todos os estágios e o custo de demanda não atendida (*backlog*^{*}) no estágio final.

1.4 REVISÃO BIBLIOGRÁFICA

A literatura referente ao sistema kanban de produção e metodologia JIT é extensa devido ao seu grande sucesso no mercado mundial, o que atraiu a atenção de um grande número de pesquisadores.

Uma análise bibliográfica sobre o JIT, classificando os artigos através de quatro diferentes tópicos: definições/objetivos do JIT como metodologia, aspectos de implementação do JIT, modelos matemáticos para o JIT e outros é realizada por Goyal e Deshmukh (1992). Golhar e Stamm (1991) revisam 211 artigos classificando-os de acordo com as principais áreas do JIT incluindo implementação, kanban, compras e gerenciamento de recursos humanos. Além disso, identificam as variáveis críticas associadas com os princípios básicos do JIT. Os dois trabalhos apresentam sugestões de possíveis pesquisas na área.

A implementação gradativa do JIT através de um processo onde os elementos desta metodologia são inseridos compassadamente é descrito por Miltenburg e Wingaard (1991).

Estudos comparativos entre os sistemas de produção de puxar (JIT/kanban) e os de empurrar (MRP) são realizados por Flapper *et al.* (1991), Spearman e Zazanis (1992), Saker e Fitzsimmons (1989) e Deleersnyder *et al.* (1992).

O uso de simulação como ferramenta no estudo do JIT é apresentado por Chu e Shih (1992) que examinam de forma extensiva o uso desta técnica em sistemas de produção JIT. Gravel e Price (1991) expõem como a simulação visual iterativa pode convencer a gerência de uma pequena empresa sobre o benefício do método kanban. Em outro trabalho, Gravel *et al.* (1992), mostram através de simulação que o sistema kanban pode ser adaptado com sucesso em ambientes *job-shop*.

Kimura e Terada (1981) foram os primeiros a modelar matematicamente um sistema kanban de produção e seu artigo serviu como referência aos trabalhos subseqüentes.

Alguns autores, como por exemplo, Rees *et al.* (1989), Sarker e Harris (1988), Yavuz e Sartir (1995) estudam, através de simulação, o efeito no desempenho do sistema de produção de variações de fatores tais como, o tempo de processamento, tempo de preparação das máquinas, tamanho de lotes e número de kanbans.

* Neste texto a palavra *backlog* é traduzida por demanda não atendida

Através de um processo markoviano a tempo discreto, Deleersnyder *et al.* (1989) analisam os efeitos do número de kanbans, variação da produção e demanda, e de estoques de segurança no desempenho do sistema kanban.

Um método analítico para estimar a quantidade necessária de estoque de segurança considerando flutuação da demanda, quebra de máquinas e tempo de processamento é proposto por So e Pinault (1988). Através de simulações mostram que este método fornece bons resultados.

Com o objetivo de determinar o tamanho ótimo dos lotes a serem usados no sistema kanban para minimizar o estoque nos centros de trabalhos e os custos de preparação de máquinas, Philipoom *et al.* (1990) propõem um modelo matemático de programação inteira.

Alguns trabalhos foram desenvolvidos para determinar o número ótimo de kanbans. Considerando a demanda determinística, Bitran e Chang (1987) constroem a partir do trabalho de Kimura e Terada, um modelo geral de programação inteira não linear com o objetivo de minimizar uma combinação ponderada do número de kanbans em circulação no sistema. O modelo é elaborado para um ambiente com estrutura de montagem com um único produto final e o não atendimento da demanda do período não é permitido. Os autores apresentam uma transformação para um modelo de programação linear inteira.

A partir deste trabalho vários outros foram desenvolvidos. Li e Co (1991) utilizam programação dinâmica para resolver um modelo de programação inteira para um sistema de montagem com o objetivo de minimizar os custos associados a injeção de kanbans no sistema. Sugerem também um método eficiente para determinar o número de kanbans em cada estágio do sistema de produção.

Bard e Golany (1991) desenvolvem um modelo de programação inteira mista com vários itens finais. Consideram também restrição de capacidade, custo de preparação, demanda não atendida e demanda externa para todo os itens. Este modelo é implementado em um problema prático considerando 7 itens componentes, 3 itens finais e 5 períodos de tempo, que é resolvido através do GAMS/MINOS.

Um modelo matemático de programação inteira para determinar o número de kanbans que minimiza o *makespan*, encorajando o planejamento de operações para que as entregas sejam feitas o mais cedo possível, é apresentado por Price *et al.* (1995). Consideram um sistema de produção onde cada estágio é responsável pelo processamento de diferentes operações. Devido a complexidade da resolução do problema de programação inteira, resolvem somente o problema relaxado que leva a informações interessantes sobre o uso do método kanban em um sistema de montagem.

Funções recursivas que determinam o número ótimo de kanbans para sistemas de produção em série com custos associados a funções monótonas lineares não crescentes são elaboradas por Co e Jacobson (1994). Nestas funções, demanda não atendida é considerada.

Para um ambiente de montagem com vários itens finais e demanda externa para todos os estágios de produção, Mitwasi e Askin (1994) desenvolvem um modelo de programação inteira mista que tem como função objetivo minimizar os custos associados ao estoque. Apresentam também um método heurístico e um método ótimo. Testes computacionais são realizados e demonstram que a qualidade da heurística é boa.

No caso de demanda estocástica modelos markovianos são propostos. Wang e Wang (1990), desenvolvem um modelo markoviano contínuo com o objetivo de minimizar o estoque intermediário. Os autores consideram a variação do número de containers entre os estágios ao longo do horizonte de planejamento como um processo estocástico. Um método para ajustar o número de kanbans em um sistema de produção onde existe quebra de máquinas também é discutido e um exemplo numérico é apresentado.

Para minimizar o custo de estoque e demanda não atendida, Askin *et al.* (1993) apresentam um modelo markoviano contínuo onde os estágios são considerados independentes e ligados por taxas proporcionais de demanda. Custos de preparação de máquina também integram o modelo.

Tendo como objetivo minimizar a média de todos os custos por período, Ohno *et al.* (1995) apresentam um algoritmo para determinar o número de kanbans em um ambiente onde o tempo de processamento é determinístico e a demanda não atendida possui distribuição estacionária e contínua.

CAPÍTULO 2

MODELOS DE PROGRAMAÇÃO ESTOCÁSTICA PARA O SISTEMA KANBAN DE PRODUÇÃO

2.1 INTRODUÇÃO

As rápidas mudanças nos ambientes atuais enfatizam a necessidade de modelos e métodos capazes de lidar com as incertezas inerentes em sistemas relacionados com economia, produção, energia e transporte. Sistemas envolvendo interações entre o homem, natureza e tecnologia estão sujeitos a distúrbios que não existiram no passado.

Com o objetivo de lidar com tais situações é necessário desenvolver modelos que possam facilitar a tomada de decisões perante as incertezas. Dois tipos principais de mecanismos podem ser identificados para tratar esses problemas. Um deles é o ajuste adaptativo de curto prazo, que é executado após a realização de algumas observações sobre os parâmetros do sistema. O outro consiste de ações antecipativas de longo prazo tomadas a priori através de conhecimentos probabilísticos e que não dependem de observações futuras dos parâmetros do sistema. O principal desafio, neste sentido, é desenvolver modelos que combinem ambos mecanismos (antecipativo e adaptativo) na presença de um grande número de incertezas e que sejam computacionalmente tratáveis (Ermoliev, 1992).

Wets (1989) define o modelo de **programação estocástica** como uma extensão dos modelos de programação determinística linear e não linear onde os coeficientes não são conhecidos com certeza (possuem uma representação probabilística). Resumindo, este termo refere-se ao campo de modelagem, teoria e procedimentos de solução que tem como objetivo a otimização de problemas envolvendo quantidades aleatórias.

Um modelo estocástico para o sistema kanban de produção é apresentado neste capítulo. Também são discutidas generalizações para este modelo.

2.2 MODELOS DE RECURSO (OU EM DOIS ESTÁGIOS)

O problema de recurso (ou em dois estágios) pode ser interpretado como sendo aquele onde uma decisão tem que ser tomada antes que os eventos aleatórios tenham sido observados (primeiro estágio). Após a ocorrência dos eventos aleatórios uma ação corretiva é tomada (segundo estágio). Portanto, é uma tentativa de incorporar os mecanismos fundamentais de antecipação e adaptação no mesmo modelo matemático. A versão geral linear para os problemas de recurso é formulada como:

$$\begin{aligned} \text{Minimizar } z &= cx + E_{\xi} \{q(\omega)y\} & (2.1) \\ \text{sujeito a } Ax &= b \\ Wy &= h(\omega) - T(\omega) \\ x &\in X, y \in Y \end{aligned}$$

onde c é o vetor conhecido e definido em \mathfrak{R}^{n_1} , b é um vetor conhecido em \mathfrak{R}^{m_1} , A e W são matrizes conhecidas de dimensões $m_1 \times n_1$ e $m_2 \times n_2$, respectivamente. ξ é um vetor aleatório de dimensão N definido no espaço de probabilidade $(\Omega, \mathfrak{F}, P)$ onde Ω é o espaço amostral, \mathfrak{F} é a família de subconjuntos de Ω (campo de Borel) e P é uma medida de probabilidade em \mathfrak{F} . Para cada $\omega \in \Omega$, $T(\omega)$, $q(\omega)$ e $h(\omega)$ possuem a dimensão adequada. Agrupando os componentes estocásticos do problema, obtém-se um vetor $\xi(\omega) = (q(\omega), h(\omega), T_1(\omega), \dots, T_{m_2}(\omega))$ com $N = n_2 + m_2 + (m_2 \times n_1)$ componentes, onde $T_i(\omega)$ é a i -ésima linha de $T(\omega)$. E_{ξ} representa a esperança matemática em relação a ξ .

A formulação acima é geral no sentido de que os conjuntos X e Y podem definir restrições sobre integralidade de variáveis. Esta formulação pode ser interpretada da seguinte forma: uma decisão é tomada sobre o valor de x , como por exemplo $x = \bar{x}$. O evento aleatório $\xi = \hat{\xi}$ é observado e uma ação corretiva y é tomada tal que $y \in Y$, $Wy = h(\omega) - T(\omega)$ e qy é minimizado.

O problema acima é equivalente a:

$$\begin{aligned} \text{Minimizar } z &= cx + Q(x) \\ \text{sujeito a } Ax &= b \\ x &\in X \end{aligned} \tag{2.2}$$

onde

$$Q(x) = E_{\xi} Q(x, \xi(\omega)) \quad (2.3)$$

e

$$Q(x, \xi(\omega)) = \min_y \{q(\omega)y \mid Wy = h(\omega) - T(\omega), y \in Y\} . \quad (2.4)$$

Dantzig (1955) e Beale (1955, *apud in* Hansotia 1990) independentemente, apresentaram modelos que se tornaram conhecidos na literatura como programa linear de dois estágios sob incerteza. Wets (1966a, b, c) desenvolveu a teoria e métodos para este tipo de problema e denominou-o de programa linear estocástico com recurso. Uma introdução à programação estocástica com recurso é apresentada por Hansotia (1990)

Uma visão geral sobre programação estocástica é apresentada por Wets (1989). Teoria, técnicas de resolução e aplicações encontram-se no livro editado por Ermoliev e Wets(1988).

A programação estocástica inteira com recurso é apresentada como uma extensão da programação estocástica contínua por Loveuax e Vlerk (1993). Estes também analisam as propriedades da função recurso esperada. Exemplos da aplicação da programação estocástica binária em problemas de roteamento de veículos e de localização encontram-se em Laporte *et al.* (1989, 1992, 1994).

2.3 UM MODELO ESTOCÁSTICO PARA O SISTEMA KANBAN DE PRODUÇÃO

Nesta seção, apresenta-se um modelo de otimização estocástica com recurso para o sistema kanban de produção para o qual serão desenvolvidas técnicas de resolução nos capítulos 3 e 4. Considera-se um sistema de produção multi-estágio com estrutura de montagem onde cada estágio recebe itens de estágios predecessores. Quando uma operação é realizada o item resultante é enviado para seu único estágio sucessor.

O sistema de montagem pode ser representado por um grafo orientado acíclico onde um nó representa o estágio de produção e um arco (i,j) indica que itens produzidos no estágio i são usados na produção do estágio j . Seja $n \in \{0,1,\dots,N\}$, o índice dos estágios de produção tal que se $n_1 < n_2$, então n_1 sucede n_2 . Um item é denominado pelo seu estágio de produção. O estágio final é o estágio 0, onde o item final é produzido. O estágio imediatamente sucessor ao estágio n é representado por $s(n)$ e $P(n)$ é o conjunto de estágios predecessores imediatos de n . Seja $t \in \{1,2,\dots,T\}$ os períodos de tempo do horizonte de planejamento. Um exemplo de sistema de montagem é mostrado na Figura 2.1.

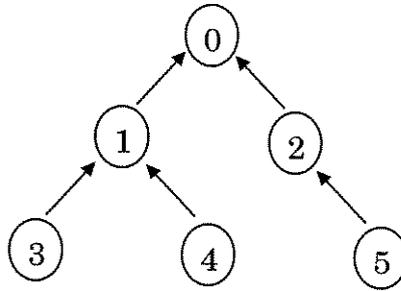


FIGURA 2.1 - ESTRUTURA DE MONTAGEM

Para este modelo, assume-se que a demanda para o estágio final e a capacidade de produção para cada estágio são variáveis aleatórias discretas ao longo do horizonte de planejamento. Demanda não atendida é permitida no estágio final. O objetivo é determinar o número de kanbans em cada estágio que minimize o custo de estoque em todos os estágios e o custo de demanda não atendida no estágio final.

As seguintes hipóteses são feitas sobre a forma de operação do sistema kanban de produção:

- no final de cada período de tempo os kanbans na caixa de kanbans são enviados para o quadro de kanbans. Neste momento, todos os kanbans do processo situam-se no estoque ou no quadro de kanbans;
- um kanban pode ser retirado do estoque em qualquer momento durante o período de tempo t .

Para a definição do modelo considere os seguintes parâmetros e variáveis

* Parâmetros

- $E_{n,s(n)}$ = número de containers cheios do item n necessários para produzir um container do item $s(n)$;
- τ_n = fração do período de tempo t necessária para produzir um container cheio do item n ;
- $\Phi_{n,0}$ = número de kanbans no estoque no início do horizonte de planejamento;
- Γ_0 = demanda não atendida no início do horizonte de planejamento;
- K_n = número máximo de kanbans no estágio n .

*** Variáveis Aleatórias**

Λ_t = demanda do item final no período $t \in \{1, 2, \dots, T\}$;

$\beta_{n,t}$ = capacidade de produção do estágio n no período de tempo t
($n = 0, 1, \dots, N$; $t = 1, 2, \dots, T$).

*** Variáveis de Decisão**

$\sigma_{n,0}$ = número de kanbans do estágio n a ser injetado pela gerência no início do horizonte de planejamento em cada estágio $n \in \{0, 1, \dots, N\}$.

Seja $\sigma_0 = (\sigma_{0,0}, \sigma_{1,0}, \dots, \sigma_{N,0})$ o vetor de variáveis de decisão.

*** Variáveis de Recurso**

Variáveis de recurso são aquelas que dependem das incertezas do modelo e das variáveis de decisão.

$\sigma_{n,t}$ = número de kanbans do estágio n no quadro de kanbans no final do período de tempo t ($n = 0, 1, \dots, N$; $t = 1, 2, \dots, T$);

$\Phi_{n,t}$ = número de containers no estoque, que estão disponíveis no estágio n , com um kanban ligado a cada um deles no final do período de tempo t (cada container possui um kanban ligado a ele)
($n = 0, 1, \dots, N$; $t = 1, 2, \dots, T$);

$\chi_{n,t}$ = número de containers cheios produzidos no estágio n no período de tempo t (cada container tem um kanban ligado a ele)
($n = 0, 1, \dots, N$; $t = 1, 2, \dots, T$);

Γ_t = demanda não atendida, em termo de containers do item final (estágio 0), no final do período de tempo t ($t = 1, 2, \dots, T$).

A Figura 2.2 mostra as variáveis de recurso para estágios intermediários

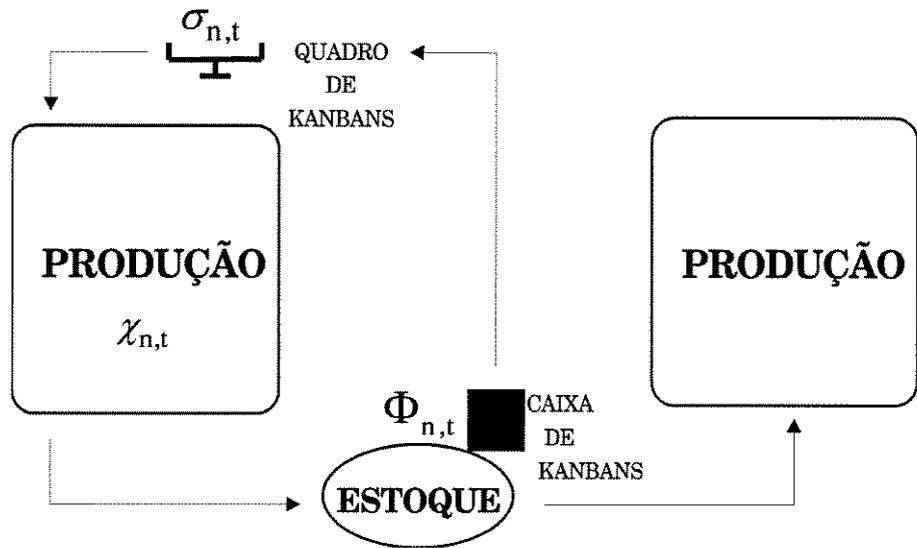


FIGURA 2.2 -. VARIÁVEIS DE RECURSO

* Descrição Matemática do Modelo

A seguir, é apresentado o modelo que descreve o sistema kanban de produção.

A evolução ao longo do tempo do número de containers de itens componentes no estoque é dado por:

$$\Phi_{n,t} = \Phi_{n,t-1} + \chi_{n,t} - E_{n,s(n)} \chi_{s(n),t} \quad n = 1, \dots, N; t = 1, \dots, T. \quad (2.5)$$

O balanço no quadro de kanbans no final do período t é descrito pela seguinte equação:

$$\sigma_{n,t} = \sigma_{n,t-1} - \chi_{n,t} + E_{n,s(n)} \chi_{s(n),t} \quad n=1, \dots, N; t=1, \dots, T. \quad (2.6)$$

O estoque do item final é descrito pela seguinte equação:

$$\Phi_{0,t} - \Gamma_t = \Phi_{0,t-1} - \Gamma_{t-1} + \chi_{0,t} - \Lambda_t \quad t = 1, \dots, T. \quad (2.7)$$

Como $\Phi_{0,t} \Gamma_t = 0$ conclui-se que:

$$\Phi_{0,t} = [\Phi_{0,t-1} + \chi_{0,t} - \Gamma_{t-1} - \Lambda_t]^+ \quad t = 1, \dots, T, \quad (2.8)$$

onde $[x]^+ = \max\{0, x\}$, e

$$\Gamma_t = -[\Phi_{0,t-1} + \chi_{0,t} - \Gamma_{t-1} - \Lambda_t]^- \quad t = 1, \dots, T, \quad (2.9)$$

onde $[x]^- = \min\{0, x\}$.

A equação do número de kanbans no quadro de kanbans para o item final pode ser escrita como:

$$\sigma_{0,t} = \sigma_{0,t-1} - \chi_{0,t} + \min\{\Phi_{0,t-1} + \chi_{0,t}, \Lambda_t + \Gamma_{t-1}\} \quad t = 1, \dots, T. \quad (2.10)$$

O último termo da equação (2.10) pode ser explicado da seguinte maneira. Se $\min\{\Phi_{0,t-1} + \chi_{0,t}, \Lambda_t + \Gamma_{t-1}\} = \Phi_{0,t-1} + \chi_{0,t}$, de (2.8) obtém-se que $\Phi_{0,t} = 0$, o que significa que o estoque no início do período de tempo t , mais o que foi produzido nele é utilizado para satisfazer a sua demanda e aquela que possivelmente não foi atendida no período anterior ($t-1$). Note que neste caso $\Gamma_t \geq 0$. Se $\min\{\Phi_{0,t-1} + \chi_{0,t}, \Lambda_t + \Gamma_{t-1}\} = \Lambda_t + \Gamma_{t-1}$, isto é $\Phi_{0,t} \geq 0$, o que implica que a demanda deste período de tempo e aquela não atendida no período de tempo anterior são satisfeitas.

O número de containers produzidos no estágio n no período de tempo t , $\chi_{n,t}$, é o mínimo entre o número de kanbans na caixa de kanbans no início do período t , a capacidade de produção, a disponibilidade de itens no estoque e a produção dos estágios predecessores imediatos.

$$\chi_{n,t} = \min \left\{ \begin{array}{l} \sigma_{n,t-1}, \\ \beta_{n,t} \\ \left\lfloor \left(\Phi_{q,t-1} + \chi_{q,t} \theta_n \right) / E_{q,n} \right\rfloor \quad q \in P(n) \end{array} \right\} \quad n = 0, \dots, N; t = 1, \dots, T \quad (2.11)$$

onde $\lfloor x \rfloor$ é o maior inteiro que é menor ou igual a x .

Nesta equação há a introdução do parâmetro θ_n que é definido como sendo a fração do período de tempo t na qual a produção do item n pode utilizar seus predecessores produzidos no mesmo período de tempo, $\theta_n = 1 - \tau_n$. Os itens predecessores produzidos no período de tempo t , são

considerados na produção do estágio n , no mesmo período, até o instante de início de produção do último container de n . A Figura 2.3 ilustra o parâmetro θ_n .

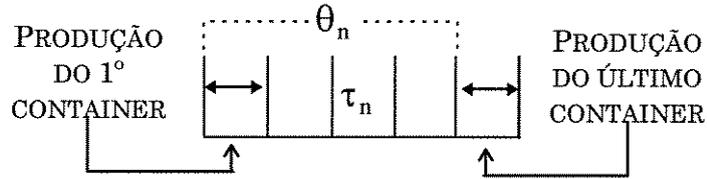


FIGURA 2.3- FRAÇÃO DE PERÍODO DE TEMPO θ_n

Proposição 2.1: Se as variáveis $\Phi_{n,t}, \sigma_{n,t}, \Gamma_t$ e $\chi_{n,t}$ satisfazem as equações (2.5, 2.6, 2.8, 2.9, 2.10 e 2.11) então:

- i) O número total de kanbans em cada estágio é invariante no tempo, ou seja, $\Phi_{n,0} + \sigma_{n,0} = \Phi_{n,t} + \sigma_{n,t}$ $n = 0, \dots, N; t = 1, \dots, T$
- ii) $\Phi_{n,t}, \sigma_{n,t}, \Gamma_t$ e $\chi_{n,t}$ são inteiros e não negativos
- iii) Uma vez que $\sigma_{n,0}$, $n = 0, \dots, N$ assume valores específicos, então as variáveis $\Phi_{n,t}, \sigma_{n,t}, \Gamma_t, \chi_{n,t}$ são unicamente determinadas.

Demonstração: Ver Apêndice A.

Pela propriedade i) da Proposição 2.1 tem-se que $\sigma_{n,t} = \sigma_{n,0} + \Phi_{n,0} - \Phi_{n,t}$. Portanto se as variáveis de decisão $\sigma_{n,0}$ e as variáveis de recurso $\Phi_{n,t}$ são determinadas, então $\sigma_{n,t}$ é conhecido e assim é possível omitir as equações (2.6) e (2.10).

Considere Λ_t e $\beta_{n,t}$, $t=1, \dots, T$ variáveis aleatórias discretas e independentes definidas no espaço de probabilidade (Ω, P) onde Ω é um espaço amostral finito e P uma medida de probabilidade definida em todos os conjuntos de Ω . Assume-se, por simplicidade, que as variáveis são equiprováveis.

Seja $E_{\Lambda, \beta}$, a esperança matemática com relação a Λ_t e $\beta_{n,t}$. Os custos de estoque e demanda não atendida por container no período t são representadas por h_n e p . Considere um cenário, uma combinação dos valores assumidos pelas variáveis Λ_t e $\beta_{n,t}$ e \mathbf{C} o conjunto de todos os possíveis cenários.

O modelo de programação estocástica pode ser formulado como:

$$\text{Minimizar } z = Q(\sigma_0) = E_{\Lambda, \beta} \left\{ \left[Q(\sigma_0, \Lambda, \beta) \right] \right\}$$

onde

$$Q(\sigma_0, \Lambda, \beta) = \min \left\{ \sum_{t=1}^T \left(\sum_{n=0}^N h_n \Phi_{n,t} \right) + p \Gamma_t \right\}$$

sujeito a $\sigma_0 \geq 0$ e inteiro

$$\Phi_{n,t} = \Phi_{n,t-1} + \chi_{n,t} - E_{n,s(n)} \chi_{s(n),t} \quad n = 1, \dots, N; t = 1, \dots, T$$

$$\Phi_{0,t} - \Gamma_t = \Phi_{0,t-1} + \chi_{0,t} - \Gamma_{t-1} - \Lambda_t \quad t = 1, \dots, T$$

$$\chi_{n,t} = \min \left\{ \begin{array}{l} \sigma_{n,0} + \Phi_{n,0} - \Phi_{n,t-1}, \\ \beta_{n,t}, \\ \left[(\Phi_{q,t-1} + \chi_{q,t} \theta_n) / E_{q,n} \right] q \in P(n) \end{array} \right. \quad \begin{array}{l} n = 0, \dots, N; \\ t = 1, \dots, T; \end{array}$$

Este é o modelo que será considerado para a execução dos métodos de resolução.

2.4 EXTENSÕES DO MODELO

Esta seção apresenta generalizações do modelo anterior .

2.4.1 MODELO COM CONTAINERS PARCIALMENTE CHEIOS NA PRODUÇÃO E NO ESTOQUE

O modelo considera a existência de um container parcialmente cheio na produção (**com** um kanban ligado a ele) e no estoque (**sem** um kanban ligado a ele) no final do período de tempo t . A demanda do item final no período de tempo t e a demanda não atendida no final do período t são consideradas em termos de containers cheios. Para descrever o modelo considere os novos parâmetros e variáveis.

* Parâmetros

$e_{n,s(n)}$ = número de unidades do item n necessárias para a produção de uma unidade do item $s(n)$;

α_n = capacidade do container, em termos de itens n .

* Variáveis de Recurso

$\varepsilon_{n,t}$ = número de itens n restantes em um container parcialmente cheio na **produção** no final do período t (este container possui um kanban ligado a ele)

$$\begin{cases} \varepsilon_{n,t} \in \{0,1,\dots,\alpha_n - 1\} & n = 0,\dots,N; t = 1,\dots,T \\ \varepsilon_{n,0} = 0 \quad \forall n \in N \end{cases};$$

$\Psi_{n,t}$ = número de unidades do item n , no final do período de tempo t , remanescentes em um container parcialmente cheio **no estoque**, no qual o kanban foi retirado

$$\begin{cases} \Psi_{n,t} \in \{0,1,\dots,\alpha_n - 1\} & n = 1,\dots,N; t = 1,\dots,T \\ \Psi_{n,0} = 0 & n = 1,\dots,N \\ \Psi_{0,t} = 0 & t = 1,\dots,T. \end{cases}$$

Note que $\Psi_{0,t} = 0$, $t = 1, \dots, T$ pois a demanda do item final é definida em termos de containers cheios.

Para este modelo definem-se as seguintes variáveis de recurso.

$\sigma_{n,t}$ = número de kanbans do estágio n no quadro de kanbans e na produção no final do período de tempo t ($n = 0, 1, \dots, N$; $t = 1, 2, \dots, T$);

$\chi_{n,t}$ = número de itens produzidos no estágio n no período de tempo t ($n = 0, 1, \dots, N$; $t = 1, 2, \dots, T$);

A Figura 2.4 mostra as novas variáveis de recurso para estágios intermediários

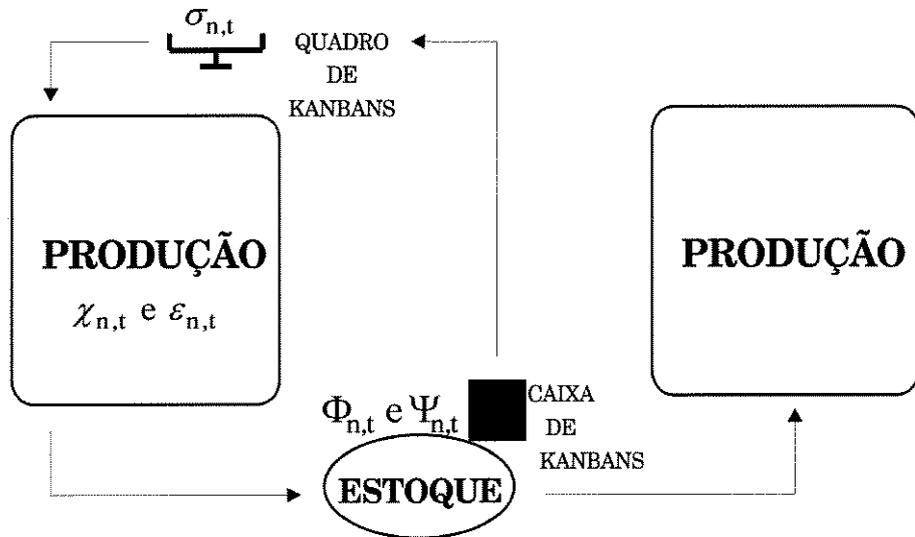


FIGURA 2.4. NOVAS VARIÁVEIS DE RECURSO

*** Descrição Matemática do Modelo**

Apresenta-se a seguir, o modelo que descreve este sistema kanban de produção.

$$\epsilon_{n,t} = \epsilon_{n,t-1} + \chi_{n,t} - \alpha_n \left\lfloor \frac{\epsilon_{n,t-1} + \chi_{n,t}}{\alpha_n} \right\rfloor \quad n = 0, \dots, N; t = 1, \dots, T \quad . \quad (2.12)$$

A equação (2.12) determina o número de itens n na produção, em um container parcialmente cheio no final do período t .

$$\Psi_{n,t} = \alpha_n \left[\left(e_{n,s(n)} \chi_{s(n),t} - \Psi_{n,t-1} \right) / \alpha_n \right] - \left(e_{n,s(n)} \chi_{s(n),t} - \Psi_{n,t-1} \right) \quad n=1,\dots,N; t=1,\dots,T \quad (2.13)$$

onde $\lceil x \rceil$ é o menor inteiro que é maior ou igual a x .

A equação (2.13) define o número de itens n , deixados em um container parcialmente cheio no estoque, no final do período de tempo t .

O balanço do estoque de containers cheios do item n no final do período t é dado por:

$$\Phi_{n,t} = \Phi_{n,t-1} + \left[(\chi_{n,t} + \varepsilon_{n,t}) / \alpha_n \right] - \left[\left(e_{n,s(n)} \chi_{s(n),t} - \Psi_{n,t-1} \right) / \alpha_n \right] \quad n = 0, \dots, N; t = 1, \dots, T \quad (2.14)$$

$$\sigma_{n,t} = \sigma_{n,t-1} + \left[\left(e_{n,s(n)} \chi_{s(n),t} - \Psi_{n,t-1} \right) / \alpha_n \right] - \left[(\chi_{n,t} + \varepsilon_{n,t}) / \alpha_n \right] \quad n = 0, \dots, N; t = 1, \dots, T \quad (2.15)$$

A equação (2.15) descreve o balanço no quadro de kanbans no final do período t .

$$\Phi_{0,t} - \Gamma_t = \Phi_{0,t-1} - \Gamma_{t-1} + \left[(\chi_{0,t} + \varepsilon_{0,t}) / \alpha_0 \right] - \Lambda_t \quad t = 1, \dots, T; \quad (2.16)$$

O balanço do estoque do item final (item zero) é dado por (2.16). Desta obtém-se que:

$$\Phi_{0,t} = \left[\Phi_{0,t-1} - \Gamma_{t-1} + \left[(\chi_{0,t} + \varepsilon_{0,t}) / \alpha_0 \right] - \Lambda_t \right]^+ \quad t = 1, \dots, T; \quad (2.17)$$

e

$$\Gamma_t = - \left[\Phi_{0,t-1} - \Gamma_{t-1} + \left[(\chi_{0,t} + \varepsilon_{0,t}) / \alpha_0 \right] - \Lambda_t \right]^- \quad t = 1, \dots, T. \quad (2.18)$$

O número de kanbans no quadro de kanbans para o item zero, no fim do período t é definido por:

$$\begin{aligned} \sigma_{0,t} = & \sigma_{0,t-1} - \left[(\chi_{0,t} + \varepsilon_{0,t-1}) / \alpha_0 \right] \\ & + \min \left\{ \Phi_{0,t-1} + \left[(\chi_{0,t} + \varepsilon_{0,t-1}) / \alpha_0 \right], \Lambda_t + \Gamma_{t-1} \right\} \quad t = 1, \dots, T. \end{aligned} \quad (2.19)$$

A produção de itens n , no período de tempo t , $\chi_{n,t}$, é o mínimo entre o número de itens cuja a disponibilidade esta associada com a caixa de kanbans no início do período; a capacidade de produção e a disponibilidade de itens no estoque e a parte da produção acessível, no mesmo período de tempo, dos seus predecessores.

$$\begin{aligned} \chi_{n,t} = \min \left\{ \alpha_n \sigma_{n,t-1} - \varepsilon_{n,t-1}, \right. \\ \beta_{n,t}, \\ \left. \left[(\alpha_q \Phi_{q,t-1} + (\chi_{q,t} + \varepsilon_{q,t-1}) \theta_n + \Psi_{q,t-1}) / e_{q,n} \right] q \in P(n) \right\} \end{aligned} \quad \begin{array}{l} n = 0, \dots, N; \\ t = 1, \dots, T \end{array} \quad (2.20)$$

Proposição 2.2: Se as variáveis $\Phi_{n,t}, \sigma_{n,t}, \Gamma_t$ e $\chi_{n,t}$ satisfazem as equações (2.13, 2.14, 2.15, 2.17, 2.18, 2.19, 2.20) então:

- i) O número total de kanbans em cada estágio é invariante no tempo, ou seja, $\Phi_{n,0} + \sigma_{n,0} = \Phi_{n,t} + \sigma_{n,t} \quad n = 0, \dots, N; t = 1, \dots, T$
- ii) $\Phi_{n,t}, \sigma_{n,t}, \Gamma_t, \varepsilon_{n,t}, \Psi_{n,t}$ e $\chi_{n,t}$ são inteiros e não negativos
- iii) Uma vez que $\sigma_{n,0}, n = 0, \dots, N$ assume valores específicos, então as variáveis $\Phi_{n,t}, \sigma_{n,t}, \Gamma_t, \chi_{n,t}$ são unicamente determinadas.
- iv) $\varepsilon_{n,t} \leq \alpha_n - 1 \quad n = 0, \dots, N; t = 1, \dots, T$
- v) $\Psi_{n,t} \leq \alpha_n - 1 \quad n = 1, \dots, N; t = 1, \dots, T$

Demonstração: Ver APÊNDICE A.

Da Proposição 2.2 i) segue-se que $\sigma_{n,t} = \sigma_{n,0} + \Phi_{n,0} - \Phi_{n,t}$ e portanto, é possível omitir as equações (2.15) e (2.19).

Então, o modelo de programação estocástica em dois estágios pode ser formulado como:

$$\text{Minimizar } z = Q(\sigma_0) = \mathbf{E}_{\Lambda, \beta} \left\{ \left[Q(\sigma_0, \Lambda, \beta) \right] \right\}$$

onde

$$Q(\sigma_0, \Lambda, \beta) = \min \left\{ \sum_{t=1}^T \left(\sum_{n=0}^N h_n \Phi_{n,t} \right) + p \Gamma_t \right\}$$

sujeito a $\sigma_0 \geq 0$ e inteiro, $n = 0, \dots, N$

$$\varepsilon_{n,t} = \varepsilon_{n,t-1} + \chi_{n,t} - \alpha_n \left[(\chi_{n,t} + \varepsilon_{n,t-1}) / \alpha_n \right] \quad n = 0, \dots, N; t = 1, \dots, T$$

$$\Psi_{n,t} = \alpha_n \left[(e_{n,s(n)} \chi_{s(n),t} - \Psi_{n,t-1}) / \alpha_n \right] - (e_{n,s(n)} \chi_{s(n),t} - \Psi_{n,t-1}) \quad n = 1, \dots, N; t = 1, \dots, T$$

$$\Phi_{n,t} = \Phi_{n,t-1} + \left[(\chi_{n,t} + \varepsilon_{n,t}) / \alpha_n \right] - \left[(e_{n,s(n)} \chi_{s(n),t} - \Psi_{n,t-1}) / \alpha_n \right] \quad n = 0, \dots, N; t = 1, \dots, T$$

$$\Phi_{0,t} - \Gamma_t = \Phi_{0,t-1} - \Gamma_{t-1} + \left[(\chi_{0,t} + \varepsilon_{0,t}) / \alpha_0 \right] - \Lambda_t \quad t = 1, \dots, T$$

$$\chi_{n,t} = \min \left\{ \begin{array}{l} \alpha_n \sigma_{n,t-1} - \varepsilon_{n,t-1}, \\ \beta_{n,t}, \\ \left[(\alpha_q \Phi_{q,t-1} + (\chi_{q,t} + \varepsilon_{q,t-1}) \theta_n + \Psi_{q,t-1}) / e_{q,n} \right]_{q \in P(n)} \end{array} \right\} \quad \begin{array}{l} n = 0, \dots, N; \\ t = 1, \dots, T \end{array}$$

2.4.2 MODELO COM VÁRIOS ITENS FINAIS E DEMANDA EXTERNA PARA TODOS OS ESTÁGIOS DE PRODUÇÃO

O modelo a seguir considera a existência de vários itens finais e para cada estágio de produção a possibilidade de demanda não atendida e de demanda externa.

Para a elaboração deste modelo, define-se:

*** Parâmetros**

- N = conjunto de todos os itens (estágios) da árvore de produção;
- F = conjunto dos itens finais ;
- $S(n)$ = conjunto dos estágios imediatamente sucessores ao estágio n ;
- $d_{n,t}$ = demanda externa do item n no período de tempo t ,
 $n = 1, \dots, N; t = 1, \dots, T$.

*** Variável Aleatória**

$\Lambda_{n,t}$ = demanda do item final n no período de tempo $t, n \in F, t \in \{1, \dots, T\}$;

*** Variável de Recurso**

$\Gamma_{n,t}$ = demanda não atendida, em termos de containers do item final n , no final do período de tempo $t, n \in F, t \in \{1, \dots, T\}$.

*** Descrição Matemática do Modelo**

As próximas equações continuam com o mesmo significado que as suas correspondentes no modelo da seção anterior.

$$\Phi_{n,t} = \Phi_{n,t-1} + \chi_{n,t} - \left(d_{n,t} + \sum_{j \in S(n)} E_{n,j} \chi_{j,t} \right) \quad n \in N / F; t = 1, \dots, T \quad (2.21)$$

$$\sigma_{n,t} = \sigma_{n,t-1} - \chi_{n,t} + \left(d_{n,t} + \sum_{j \in S(n)} E_{n,j} \chi_{j,t} \right) \quad n \in N / F; t = 1, \dots, T \quad (2.22)$$

$$\Phi_{n,t} - \Gamma_{n,t} = \Phi_{n,t-1} + \chi_{n,t} - \Lambda_{n,t} - \Gamma_{n,t-1} \quad n \in F, t = 1, \dots, T \quad (2.23)$$

$$\sigma_{n,t} = \sigma_{n,t-1} - \chi_{n,t} + \min\left\{\Phi_{n,t-1} + \chi_{n,t}, \Lambda_{n,t} + \Gamma_{n,t-1}\right\} \quad n \in F, t = 1, \dots, T \quad (2.24)$$

$$\chi_{n,t} = \min\left\{\begin{array}{l} \sigma_{n,t-1}, \\ \beta_{n,t} \\ \left\lfloor \left(\Phi_{q,t-1} + \chi_{q,t} \theta_n \right) / E_{q,n} \right\rfloor \quad q \in P(n) \end{array}\right\}_{\substack{n \in N \\ t = 1, \dots, T}} \quad (2.25)$$

Proposição 2.3: Se as variáveis $\Phi_{n,t}, \sigma_{n,t}, \Gamma_{n,t}$ e $\chi_{n,t}$ satisfazem as equações (2.21, 2.22, 2.23, 2.24, 2.25) então:

- i) O número total de kanbans em cada estágio é invariante no tempo, ou seja, $\Phi_{n,0} + \sigma_{n,0} = \Phi_{n,t} + \sigma_{n,t} \quad n = 0, \dots, N; t = 1, \dots, T$
- ii) $\Phi_{n,t}, \sigma_{n,t}, \Gamma_{n,t}$ e $\chi_{n,t}$ são inteiros e não negativos
- iii) Uma vez que $\sigma_{n,0}, n = 0, \dots, N$ assume valores específicos, então as variáveis $\Phi_{n,t}, \sigma_{n,t}, \Gamma_{n,t}, \chi_{n,t}$ são unicamente determinadas.

A demonstração desta proposição é semelhante à da Proposição 2.1 e por isso será omitida.

Aqui também é válido que $\sigma_{n,t} = \sigma_{n,0} + \Phi_{n,0} - \Phi_{n,t}$ podendo-se então, omitir as equações (2.22) e (2.24).

Assim, o modelo de programação estocástica pode ser formulado como:

$$\text{Minimizar } z = Q(\sigma_0) = E_{\Lambda, \beta} \left\{ \left[Q(\sigma_0, \Lambda, \beta) \right] \right\}$$

onde

$$Q[(\sigma_0, \Lambda, \beta)] = \min \left\{ \sum_{t=1}^T \left(\sum_{n \in N} h_n \Phi_{n,t} + \sum_{n \in F} p \Gamma_{n,t} \right) \right\}$$

sujeito a $\sigma_0 \geq 0$ e inteiro

$$\Phi_{n,t} = \Phi_{n,t-1} + \chi_{n,t} - \left(d_{n,t} + \sum_{j \in S(n)} E_{n,j} \chi_{j,t} \right) \quad n \in N / F; t = 1, \dots, T$$

$$\Phi_{n,t} - \Gamma_{n,t} = \Phi_{n,t-1} + \chi_{n,t} - \Lambda_{n,t} - \Gamma_{n,t-1} \quad n \in F; t = 1, \dots, T$$

$$\chi_{n,t} = \min \left\{ \begin{array}{l} \sigma_{n,t-1}, \\ \beta_{n,t}, \\ \left[(\Phi_{q,t-1} + \chi_{q,t} \theta_n) / E_{q,n} \right] q \in P(n) \end{array} \right\} \quad n \in N; t = 1, \dots, T.$$

CAPÍTULO 3

UM ALGORITMO ÓTIMO PARA O MODELO ESTOCÁSTICO

3.1 INTRODUÇÃO

O método L-shaped inteiro para programação estocástica inteira com recurso (Laporte *et al.* 1993) foi utilizado para a resolução do modelo estocástico para um sistema kanban de produção. É um método do tipo *branch-and-cut*, e é motivado por aquele proposto por Van Slyke e Wets (1969) para programação estocástica sem integralidade de variáveis que gera cortes de otimalidade e factibilidade de modo semelhante ao método de Benders (1962).

Neste capítulo o método L-shaped inteiro é apresentado e adaptado para o problema estocástico de um sistema kanban de produção. Cinco diferentes estratégias são propostas para a escolha da variável e para a escolha do problema candidato no algoritmo *branch-and-cut*. Uma análise comparativa das estratégias é realizada através de testes computacionais para problemas pequenos.

3.2 O MÉTODO L-SHAPED INTEIRO PARA PROGRAMAÇÃO ESTOCÁSTICA INTEIRA COM RECURSO

Seja o problema de programação estocástica inteira:

$$\begin{aligned} \text{Minimizar } z &= cx + Q(x) \\ \text{sujeito a } Ax &= b \\ x &\in B, \end{aligned} \tag{3.1}$$

onde B é o conjunto de vetores binários,

$$Q(x) = E_{\xi}(Q(x), \xi), \tag{3.2}$$

e

$$E_{\xi}(Q(x), \xi) = \min_y \{qy / Wy = h - Tx, y \geq 0\}. \quad (3.3)$$

Seja \bar{X} o politopo definido por :

$$\bar{X} = \{0 \leq x \leq 1 / \text{para todo } \xi \text{ discreto, existe } y \geq 0 \text{ tal que } Wy = h - Tx\}$$

O problema anterior é equivalente a:

$$\begin{aligned} \text{Minimizar } & z = cx + \lambda \\ \text{sujeiro a } & Ax = b \\ & Q(x) \leq \lambda \\ & x \in B \cap \bar{X}. \end{aligned}$$

Para uma dada iteração do método considere o problema candidato (PC):

$$\begin{aligned} \text{Minimizar } & z = cx + \lambda \\ \text{sujeito a } & Ax = b \end{aligned} \quad (3.4a)$$

$$(PC) \quad D_s x \geq d_s \quad s = 0,1,2,\dots,S \quad (3.4b) \quad (3.4)$$

$$K_r x + \lambda \geq k_r \quad r = 0,1,2,\dots,R \quad (3.4c)$$

$$x_i \in \{0,1\}, \quad \lambda \in \mathfrak{R}.$$

Seja o conjunto F formado por todas as variáveis x_i já fixadas nos seus valores binários. O problema candidato relaxado (PCR) é aquele onde as variáveis $x_i \notin F$ e $0 \leq x \leq 1$.

As restrições (3.4b) referem-se aos cortes de factibilidade e são usadas para fornecer uma representação apropriada de \bar{X} , que normalmente é desconsiderado no estágio inicial. Estas restrições são adicionadas à medida que são necessárias. Os cortes de otimalidade são representados por (3.4c). Um conjunto de cortes de otimalidade é válido, se para todo $x \in B \cap \bar{X}$, $(x, \lambda) \in \{(x, \lambda) | K_r x + \lambda \geq k_r, r = 1, \dots, R\}$ implica $\lambda \geq Q(x)$. Assim, estes cortes juntamente com a variável λ são usados para expressar uma aproximação apropriada de $Q(x)$.

A resolução através do método *branch-and-cut* possui uma estrutura em árvore, onde cada nó da árvore representa um problema candidato. Seja \bar{z} o valor da solução incumbente do problema, ou seja, a melhor solução obtida até o momento. Sejam r e s o número de cortes de otimalidade e factibilidade, respectivamente, inseridos no problema (PC) e v o número de iterações do algoritmo *branch-and-cut* que é apresentado a seguir.

- Passo 1:** Inicialização
 $s = r = v = 0$; $\bar{z} = \infty$;
 O único nó pendente corresponde ao problema candidato inicial.
- Passo 2:** Selecione um nó pendente na lista, se não existir, **pare**
- Passo 3:** Faça $v \leftarrow v + 1$;
 Resolva o (PCR);
 Se (PCR) não tem solução factível, sonde o nó atual e volte para o **passo 2**.
 Caso contrário, seja (x_v, λ_v) uma solução ótima.
 (Se $r=0$, faça λ igual a $-\infty$)
- Passo 4:** Verifique se existe a violação de alguma restrição.
 Se existir adicione um corte de factibilidade (3.4b), faça $s \leftarrow s + 1$ e retorne ao **passo 3**.
- Passo 5:** Se $cx_v + \lambda_v > \bar{z}$, sonde o problema candidato e retorne ao **passo 2**.
- Passo 6:** Verifique a integralidade das variáveis.
 Se esta é violada, crie dois novos nós e adicione na lista de nós pendentes e volte para o **passo 2**.
- Passo 7:** Calcule $Q(x_v)$ e $z_v = cx_v + Q(x_v)$
 Se $z_v < \bar{z}$ então atualize $\bar{z} \leftarrow z_v$.
- Passo 8:** Se $\lambda_v \geq Q(x_v)$, sonde o nó atual e volte para o **passo 2**;
 Caso contrário, crie um corte de otimalidade (3.4c), faça $r \leftarrow r + 1$ e retorne ao **passo 3**.

Note que as regras de sondagem são diferentes das usuais. Em particular, os nós não são necessariamente sondados quando as condições de integralidade são satisfeitas. A proposição a seguir decorre imediatamente das regras de sondagem e da definição do conjunto válido de cortes.

Proposição 3.1: Para qualquer problema onde existe um conjunto de cortes de factibilidade e um conjunto de cortes de otimalidade válidos, o método L-shaped inteiro obtém uma solução ótima (quando esta existe) em um número finito de passos.

Para a definição de cortes de otimalidade são necessárias as seguintes hipóteses:

Hipótese 3.1: Dado um vetor x de variáveis binárias de decisão do primeiro estágio, a função $Q(x)$ é calculável.

Hipótese 3.2: Existe um valor finito L que satisfaz $L \leq \min_x \{Q(x) / Ax = b, x \in B\}$.

Desde que as variáveis do primeiro estágio são binárias, existe somente um número finito de soluções factíveis para este estágio. Seja $m=1, \dots, M$, o índice destas soluções factíveis.

Proposição 3.2: Seja $x_i = 1, i \in S$ e $x_i = 0, i \notin S$ a m -ésima solução factível e λ_m o valor esperado do segundo estágio. Defina o corte de otimalidade como:

$$\lambda \geq (\lambda_m - L) \left(\sum_{i \in S} x_i - \sum_{i \notin S} x_i \right) - (\lambda_m - L)(|S| - 1) + L, \quad (3.5)$$

onde $|S|$ é a cardinalidade de S . Então, o conjunto de cortes de otimalidade (3.5) definido para todas as soluções factíveis do primeiro estágio é válido.

Demonstração: A expressão $\left(\sum_{i \in S} x_i - \sum_{i \notin S} x_i \right)$ é sempre menor ou igual a $|S|$ e assume este valor somente quando x é a m -ésima solução factível. Quando $\left(\sum_{i \in S} x_i - \sum_{i \notin S} x_i \right)$ é igual a $|S|$ então o lado direito assume o valor λ_m . Caso contrário, o lado direito assume um valor menor ou igual a L que é um limitante inferior da função $Q(x)$. Portanto, a desigualdade acima representa um corte válido de otimalidade

A aplicabilidade do método L-shaped Inteiro está diretamente relacionada com a satisfação das hipóteses (3.1) e (3.2). A validade da hipótese (3.1) depende obviamente, do problema. Geralmente, ela é assegurada quando o vetor aleatório tem um suporte discreto de pequena cardinalidade.

3.3 O MÉTODO L-SHAPED INTEIRO APLICADO AO MODELO ESTOCÁSTICO DE UM SISTEMA KANBAN DE PRODUÇÃO

A eficiência deste método está diretamente associada às definições apropriadas de um limitante inferior (\mathbf{L}) e à qualidade dos cortes de factibilidade e otimalidade. Além disso, a escolha adequada do problema candidato e da variável a ser dicotomizada também influenciam sua eficiência.

O método L-shaped inteiro assume que as variáveis do primeiro estágio são binárias. Assim, as variáveis inteiras do modelo estocástico para o sistema kanban de produção são transformadas em variáveis binárias.

$$\sigma_{n,0} = \sum_{i=1}^{\bar{K}_n} 2^{i-1} y_{n,i} \quad n = 0, \dots, N; y_{n,i} \in \{0,1\}, \quad (3.6)$$

onde $\bar{K}_n = K - \Phi_{n,0}$.

O modelo estocástico com recurso é reescrito como:

Minimizar $z = Q(\sigma_{n,0})$

$$\begin{aligned} \text{sujeito a} \quad & \sum_{i=1}^{\bar{K}_n} 2^{i-1} y_{n,i} \geq 0 & n = 0, \dots, N \\ & \sigma_{n,0} = \sum_{i=1}^{\bar{K}_n} 2^{i-1} y_{n,i} & n = 0, \dots, N \\ & y_{n,i} \in \{0,1\} & n = 0, \dots, N \end{aligned}$$

onde,

$$Q(\sigma_0) = \mathbf{E}_{\Lambda, \beta} \left\{ \left[Q(\sigma_0, \Lambda, \beta) \right] \right\},$$

e,

$$Q(\sigma_0, \Lambda, \beta) = \min \left\{ \sum_{t=1}^T \left(\sum_{n=0}^N h_n \Phi_{n,t} \right) + p \Gamma_t \right\}$$

$$\Phi_{n,t} = \Phi_{n,t-1} + \chi_{n,t} - \mathbf{E}_{n,s(n)} \chi_{s(n),t} \quad n = 1, \dots, N; t = 1, \dots, T$$

$$\Phi_{0,t} - \Gamma_t = \Phi_{0,t-1} + \chi_{0,t} - \Gamma_{t-1} - \Lambda_t \quad t = 1, \dots, T$$

$$\chi_{n,t} = \min \left\{ \begin{array}{l} \sigma_{n,0} + \Phi_{n,0} - \Phi_{n,t-1}, \\ \beta_{n,t}, \\ \left[(\Phi_{q,t-1} + \chi_{q,t} \theta_n) / E_{q,n} \right] q \in P(n) \end{array} \right. \quad \begin{array}{l} n = 0, \dots, N; \\ t = 1, \dots, T; \end{array}$$

*** Obtenção do valor de $Q(\sigma_0)$**

Foi mostrado no Capítulo 2 (Proposição 2.1), que uma vez especificado os valores para as variáveis de decisão $\sigma_{n,0}$, $n = 0, \dots, N$, então as variáveis de recurso $\Phi_{n,t}$, $\sigma_{n,t}$, Γ_t e $\chi_{n,t}$ são unicamente determinadas (vide Apêndice A). Portanto, o cálculo de $Q(\sigma_0)$ é imediato.

*** Um Limitante Inferior (L) para $Q(\sigma_0)$**

Como a função $Q(\sigma_0) \geq 0$ então $L = 0$ é um limitante inferior válido.

*** Obtenção do Corte de Otimalidade**

Seja $y_{n,i} = 1$, se $i \in S$, $y_{n,i} = 0$, se $i \notin S$ a m -ésima solução factível e $\lambda_m = Q(\sigma_0)$ o valor esperado do segundo estágio. A partir de (3.5) com $L = 0$, obtém-se o seguinte corte de otimalidade

$$\lambda \geq \lambda_m \left(\sum_{i \in S} y_{n,i} - \sum_{i \notin S} y_{n,i} \right) - \lambda_m (|S| - 1). \quad (3.7)$$

*** Corte de Factibilidade**

O modelo estocástico para um sistema kanban de produção tem sua infactibilidade associada ao não atendimento da demanda devido às restrições de capacidade de produção. Como o não atendimento da demanda é permitido, então o problema é sempre factível e portanto, é desnecessária a geração de cortes de factibilidade. Assim o **passo 4** pode ser eliminado do método.

*** A Escolha da Variável**

Dado o problema candidato, relaxa-se a integralidade das variáveis e um problema de programação linear é resolvido. Dentre as variáveis ótimas que assumem valores não inteiros é necessário escolher uma delas para ser dicotomizada. Esta escolha é muito importante, pois geralmente há algumas

variáveis que devem ser fixadas em seus valores inteiros para que as outras tornem-se inteiras na resolução do problema de programação linear.

A regra mais comum de escolha utiliza prioridades, onde um ordenamento das variáveis é previamente especificado como entrada de dados. Laporte *et al.* (1994) utilizam o algoritmo anterior para a resolução de um problema de localização com demanda estocástica, tendo obtido bons resultados ao escolher como variável a ser dicotomizada aquela mais próxima de um.

Outra possibilidade para a escolha da variável envolve penalidade, que é uma estimativa do acréscimo do valor ótimo do problema relaxado ao se forçar uma das variáveis a assumir um determinado valor inteiro. Considerando variáveis binárias, U_i é o acréscimo ao se forçar a variável a assumir o valor 1, e D_i o acréscimo ao se forçar a variável a assumir o valor 0. Tanto U_i como D_i são apenas um limitante inferior para o acréscimo da solução do problema relaxado, pois podem ser necessárias várias iterações no dual simplex para restaurar a factibilidade do problema.

Beale (1979) e Geoffrion *et al.* (1972) propõem diferentes regras para a escolha da variável usando penalidades e as utilizadas neste trabalho são as seguintes:

- $\text{MAX}(\text{MIN}(U_i, D_i))$, que evita problemas de degenerescência dual. O primeiro problema a ser examinado será aquele onde a variável possui a menor penalidade. Assim, a lista dos problemas candidatos conterá provavelmente, problemas com soluções inteiras pobres que poderão ser sondados com maior rapidez.
- $\text{MAX}(\text{MAX}(U_i, D_i))$, que determina a variável com maior degradação. O problema com maior penalidade será o primeiro a ser examinado, tentando assim sondar rapidamente este ramo da árvore.

A utilização de penalidades na escolha da variável, permite que o cálculo do limitante inferior para um determinado problema candidato seja aprimorado no sentido de tornar o seu valor mais próximo do verdadeiro, tornando o algoritmo mais eficiente. Nemhauser e Wolsey (1988) sugerem:

- $\hat{z}_{-K+1} = z_K + U_{K,j} + \sum_{i \in G/j} \min(U_{K,i}, D_{K,i})$, se a variável escolhida está fixada em um;
- $\hat{z}_{-K+1} = z_K + D_{K,j} + \sum_{i \in G/j} \min(U_{K,i}, D_{K,i})$, se a variável escolhida está fixada em zero;

onde,

- k = k -ésimo problema que está sendo pesquisado;
- j = variável a ser dicotomizada;
- z_K = o valor da solução ótima do k -ésimo problema candidato relaxado;
- G = conjunto das variáveis inteiras com valores fracionários
- \hat{z}_{-K} = valor estimado da solução ótima inteira do problema k ;

* O Problema Candidato a ser Pesquisado

A cada dicotomização de uma variável inteira que possui na iteração valor real, gera-se dois novos problemas (nós) que são considerados filhos do nó ativo (problema candidato que está sendo pesquisado) e irmãos entre si.

Dada a lista de problemas candidatos, existem duas opções básicas para a escolha do nó. Uma delas são as regras à priori que determinam previamente a ordem na qual a árvore de resolução será desenvolvida. A outra, são as regras adaptativas que escolhem um problema candidato através de informações disponíveis.

Para este trabalho duas diferentes estratégias adaptativas são utilizadas. A primeira é uma estratégia proposta por Nemhauser e Wolsey (1988) que sugerem o uso simultâneo de várias regras: se o nó ativo não foi sondado, escolha entre seus dois filhos usando a regra descrita na escolha da variável. Caso contrário, verifique o irmão. Se este não foi sondado, então é o escolhido. Senão, para os nós disponíveis, calcule o valor estimado da solução ótima do nó. Aquele com menor valor estimado será o próximo a ser pesquisado. A segunda estratégia consiste em escolher o problema, na lista, com menor limitante inferior.

Através da combinação das regras de escolha do problema candidato e da variável a ser dicotomizada, 5 estratégias são elaboradas para a utilização no algoritmo *branch-and-cut*.

Destas cinco estratégias, três utilizam a regra proposta por Nemhauser e Wolsey para a determinar o problema candidato (nó ativo). Destas três estratégias, duas aplicam penalidade através das regras descritas anteriormente para a escolha da variável, e a terceira dicotomiza a variável que estiver mais próxima de um.

A quarta e quinta estratégias determinam o problema candidato através do menor limitante inferior. A quarta estratégia escolhe a variável através da regra de penalidade $\text{MAX}(\text{MIN}(U_j, D_j))$ enquanto a quinta utiliza a regra $\text{MAX}(\text{MAX}(U_j, D_j))$. A tabela 3.1 apresenta estas estratégias.

TABELA 3.1 - ESTRATÉGIAS PARA A EXECUÇÃO DO MÉTODO ÓTIMO

Estratégia	Escolha do Problema Candidato (Nó Ativo)	Escolha da Variável a ser dicotomizada
1 -	Estratégia proposta por Nemhauser e Wolsey .	Penalidade com a regra $\text{MAX}(\text{MIN}(U_i, D_i))$.
2 -	Estratégia proposta por Nemhauser e Wolsey .	Penalidade com a regra $\text{MAX}(\text{MAX}(U_i, D_i))$.
3 -	Estratégia proposta por Nemhauser e Wolsey .	A variável escolhida é aquela mais próxima de um.
4 -	Menor Limitante Inferior	Penalidade com a regra $\text{MAX}(\text{MIN}(U_i, D_i))$.
5 -	Menor Limitante Inferior	Penalidade com a regra $\text{MAX}(\text{MAX}(U_i, D_i))$.

3.3.1 O ALGORITMO L-SHAPED INTEIRO PARA O MODELO ESTOCÁSTICO DE UM SISTEMA KANBAN

Para uma dada iteração do método considere o problema candidato (PCR):

$$\begin{aligned}
 &\text{Minimizar } z = \lambda \\
 &\text{sujeito a } \sum_{i=1}^{\bar{K}_n} 2^{i-1} y_{n,i} \geq 0 && n = 0, \dots, N \\
 &\lambda \geq \lambda_r \left(\sum_{i \in S} y_{n,i} - \sum_{i \notin S} y_{n,i} \right) - \lambda_r (|S| - 1) && r = 1, 2, \dots, R \\
 &0 \leq y_{n,i} \leq 1, y_i \notin F && n = 0, \dots, N
 \end{aligned}$$

- Passo 1:** Inicialização
 $r = v = 0$; $\bar{z} = \infty$;
 O único nó pendente corresponde ao problema candidato inicial.
- Passo 2:** Selecione um nó pendente na lista, se não existir **pare**.
- Passo 3:** Faça $v \leftarrow v + 1$;
 Resolva o (PCR) e seja (λ_v) uma solução ótima;
 (Se $r=0$, faça λ igual a $-\infty$)
- Passo 4:** Se $\lambda_v > \bar{z}$, sonde o nó e retorne ao **passo 2**.
- Passo 5:** Verifique a integralidade das variáveis binárias.
 Se esta é violada, crie dois novos nós e adicione-os na lista de nós pendentes e volte para o **Passo 2**.
- Passo 6:** Calcule $Q(\sigma_0)$ e $z_v = Q(\sigma_0)$
 Se $z_v < \bar{z}$ então atualize $\bar{z} \leftarrow z_v$.
- Passo 7:** Se $\lambda_v \geq Q(\sigma_0)$ sonde o problema atual e volte para o **passo 2**;
 Caso contrário, crie um corte de otimalidade (16), faça $r \leftarrow r + 1$ e retorne a **passo 3**.

A Figura 3.1 representa o fluxograma deste algoritmo.

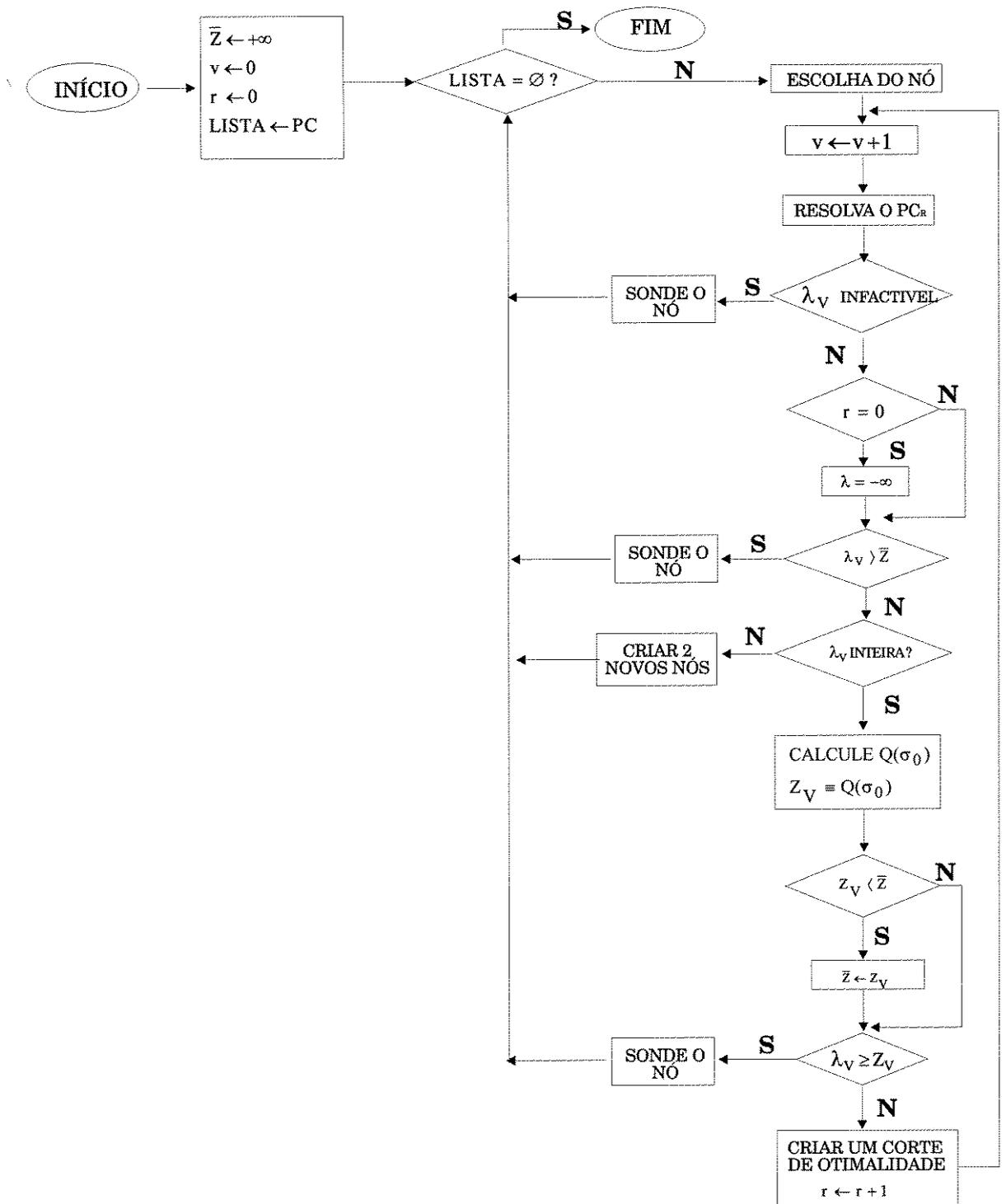


FIGURA 3.1 - FLUXOGRAMA DO ALGORITMO L-SHAPED PARA O MODELO ESTOCÁSTICO DE UM SISTEMA KANBAN

3.4 IMPLEMENTAÇÃO E ANÁLISE

3.4.1 GERAÇÃO DOS PROBLEMAS

A limitação em termos de tempo computacional e de memória associado ao método *branch-and-cut* permite somente a resolução de problemas pequenos. Cada problema pode ser caracterizado por dois tipos de parâmetros, discutidos a seguir, e pelas diferentes estruturas de produção mostradas na Figura 3.2. O horizonte de planejamento considerado é de 4 períodos.

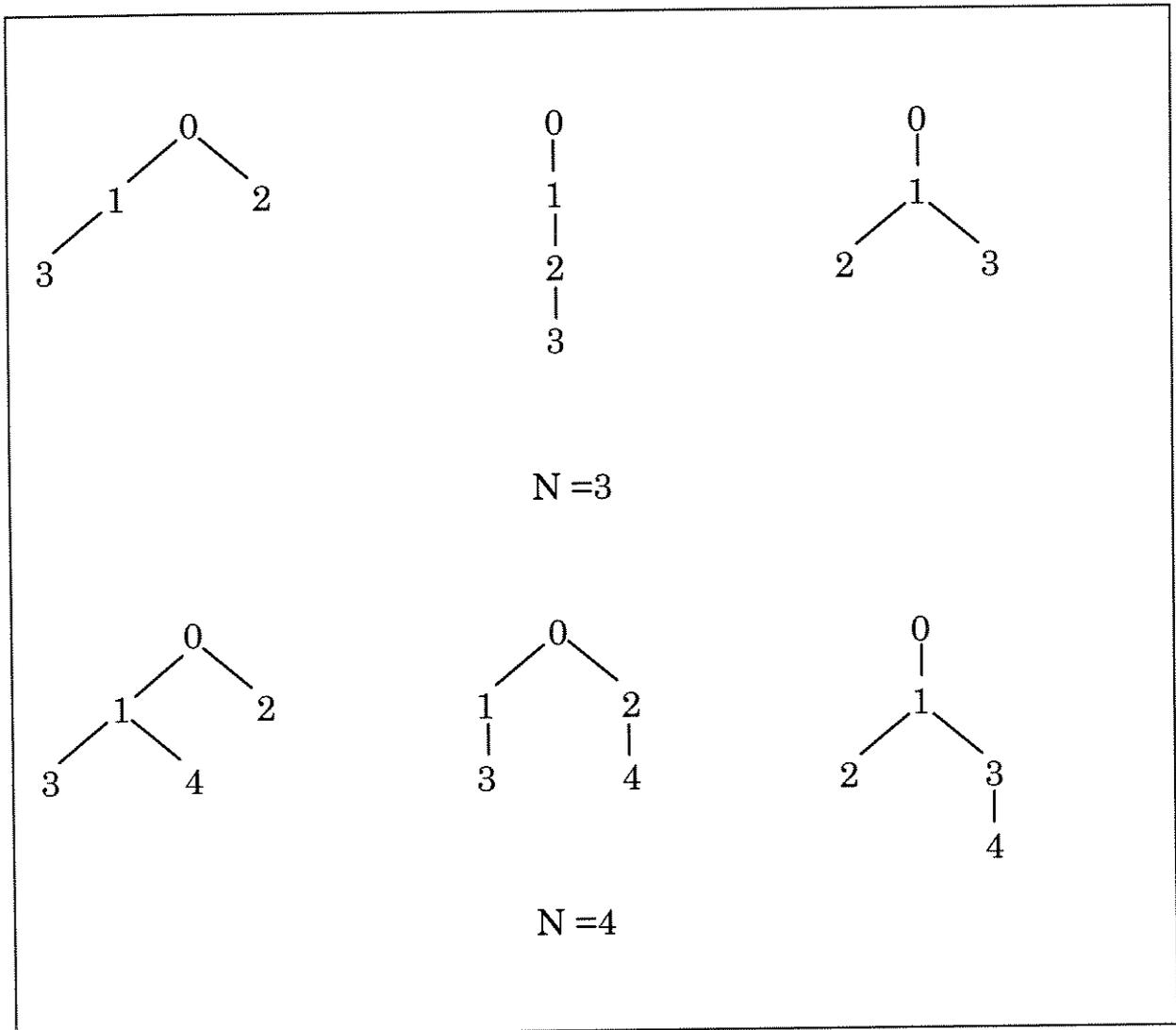


FIGURA 3.2 - ESTRUTURAS PEQUENAS DE PRODUÇÃO

- parâmetros determinísticos: são gerados aleatoriamente através de uma distribuição uniforme nos intervalos:

- $\Phi_{n,0} \in [0; 3]$;
- $E_{n,s(n)} \in [1; 2]$;
- $\theta_n \in [0,7; 1]$;
- $K_0 \in [5; 10]$,

$$K_n = g * E_{n,0}, n = 1, \dots, N, \text{ onde } g \in [5; 10].$$

O parâmetro K_n é proporcional a $E_{n,0}$, dado que este representa o número de containers do item n necessários para a produção de um container do item final.

- os custos unitários de estoque (h_n) e de demanda não atendida (p) foram também gerados considerando três possíveis casos:
 - h_n e p gerados na mesma faixa: $[5; 10]$,
 - h_n maior que p : $h_n \in [100; 500]$ e $p \in [5; 10]$,
 - h_n menor que p : $h_n \in [5; 10]$ e $p \in [100; 500]$.

- parâmetros aleatórios:

- capacidade de produção ($\beta_{n,t}$).

Entende-se por capacidade de produção o número de containers disponíveis em um estágio e período. Em um determinado período de tempo, um estágio pode sofrer redução de recursos de produção acarretando a diminuição de um container na capacidade de produção. Supõe-se que a redução de um container no horizonte T ocorre em $t_1 = (T/2 - b)$ ou $t_2 = (T/2 + b)$, onde b é gerado aleatoriamente no intervalo $[1; T/2]$. Os estágios onde ocorrem a redução em t_1 e t_2 são também obtidos aleatoriamente.

A capacidade de produção ($\beta_{n,t}$), é gerada por uma distribuição uniforme nos intervalos:

$$\beta_{0,t} \in [1; 10]$$

$$\beta_{n,t} = f * E_{n,0}, \text{ onde } f \in [1; 10].$$

A justificativa para a definição do parâmetro $\beta_{n,t}$ é idêntica à de K_n

– demanda (Λ_t)

Considera-se três tipos de demanda durante o horizonte T: constante, crescente e decrescente. Para cada período de tempo, 5 possíveis valores de demanda são gerados aleatoriamente dentro de uma faixa de variação de 50% em torno de uma reta ideal que representa a demanda esperada. A Figura 3.3 representa o caso de demanda crescente.

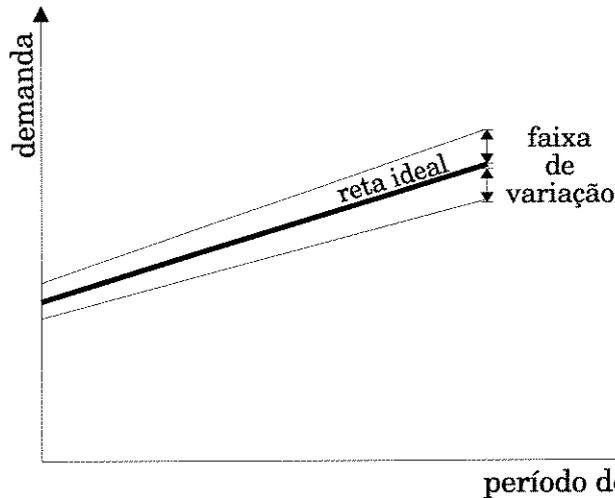


FIGURA 3.3 . FAIXA DE VARIAÇÃO PARA GERAÇÃO ALEATÓRIA DA DEMANDA

A reta ideal é gerada a partir dos valores da demanda esperada do primeiro e do último período de tempo do horizonte de planejamento. Considerando os diferentes tipos de demanda, os valores para a obtenção da reta ideal são:

crescente: início:1, final:10;
 constante: início:10, final:10;
 decrescente: início:10, final:1.

Através da combinação destas situações de capacidade de produção e demanda, 10 cenários são considerados na resolução dos problemas.

Na execução do procedimento ótimo, o problema de programação linear (problema candidato) foi resolvido através do MINOS, que dentre os softwares disponíveis era aquele que melhor lidava com a instabilidade numérica e esparsidade.

O método ótimo tem como solução inicial a solução obtida por um método heurístico que é discutido no próximo capítulo.

Os programas foram codificados em linguagem C, no ambiente Free BSD (UNIX) em um PC Pentium com 48Mb de RAM e 88MHz.

3.4.2 ANÁLISE DOS RESULTADOS

Foram considerados problemas pequenos ($N=3, T=4; N=4, T=4$). Cada problema pode ser caracterizado pelos parâmetros de custo, demanda e pelas diferentes estruturas, mostrados na Figura 3.2. Para cada configuração foram geradas 2 replicações totalizando $2*3*3*3*2 = 108$ problemas.

Seja:

- NW, a regra adaptativa para escolha do nó proposta por Nemhauser e Wolsey.
- LI, a regra do menor limitante inferior na escolha do problema candidato.
- MAX(MIN()) e MAX(MAX()) as regras de penalidade para a escolha da variável.
- $\cong 1$, a regra que escolhe a variável mais próxima de um

As Tabelas 3.2 e 3.3 apresentam os resultados obtidos pela utilização das cinco diferentes estratégias. O corte de otimalidade, para este problema não é de boa qualidade, sendo necessário um grande número de cortes para a obtenção da solução ótima. Apesar das diferenças entre os resultados serem pequenas, note que os piores valores são obtidos pela terceira estratégia.

As estratégias que determinam o problema candidato através do menor limitante inferior atingem o ótimo mais rápido, demorando porém, para sondar os nós restantes na árvore. Considerando o número médio de nós e iterações a estratégia quatro é a que se comportou melhor, entretanto, a estratégia um é em média 5% mais rápida.

TABELA 3.2 - AVALIAÇÃO DAS ESTRATÉGIAS PARA 3 ESTÁGIOS

N=3, T=4					
	- 1 - NW MAX(MIN())	- 2 - NW MAX(MAX())	- 3 - NW $\cong 1$	- 4 - LI MAX(MIN())	- 5 - LI MAX(MAX())
Nº. MÉDIO DE NÓS	7045	7192	7500	6993	7136
Nº. MÉDIO DE ITERAÇÕES	10873	11021	11431	10870	10965
ITERAÇÃO MÉDIA ÓTIMA	6918	5084	7023	5042	5199
Nº. MÉDIO DE CORTES	3829	3780	3927	3877	3828
TEMPO MÉDIO DE EXECUÇÃO	1209.65	1229.56	1418.91	1263.23	1277.80

TEMPO MÉDIO DE EXECUÇÃO EM SEGUNDOS

TABELA 3.3 - AVALIAÇÃO DAS ESTRATÉGIAS PARA 4 ESTÁGIOS

N=4, T=4					
	- 1 - NW MAX(MIN())	- 2 - NW MAX(MAX())	- 3 - NW $\cong 1$	- 4 - LI MAX(MIN())	- 5 - LI MAX(MAX())
Nº. MÉDIO DE NÓS	8858	8962	9021	8838	8939
Nº. MÉDIO DE ITERAÇÕES	13835	13940	14108	13818	13913
ITERAÇÃO MÉDIA ÓTIMA	4032	3748	4237	2205	2008
Nº. MÉDIO DE CORTES	4977	4873	5051	4980	4910
TEMPO MÉDIO DE EXECUÇÃO	1602.55	1615.75	1775.33	1681.40	1696.50

TEMPO MÉDIO DE EXECUÇÃO EM SEGUNDOS

CAPÍTULO 4

O MÉTODO HEURÍSTICO E A BUSCA TABU

4.1 INTRODUÇÃO

Este capítulo apresenta um método heurístico e uma implementação de busca tabu para obter soluções de melhor qualidade para o modelo estocástico.

Estruturas maiores de produção são geradas e através de testes computacionais estes métodos são avaliados. Todos os programas foram codificados em linguagem C, no ambiente FREE BSD (Unix) em um PC Pentium com 48Mb de RAM e 88MHz.

4.2 O MÉTODO HEURÍSTICO

Em cada estágio do sistema de produção, o número máximo de kanbans a ser inserido é dado por $(K_n - \Phi_{n,0})$, que proporciona um alto custo de estoque e um baixo custo por demanda não atendida. O número mínimo de kanbans a ser injetado é zero com o qual se obtém um custo baixo de estoque e um custo alto de demanda não atendida.

A heurística inicia com $\sigma_{n,0}$, o número de kanbans a ser inserido no estágio n do sistema de produção, em $(K_n - \Phi_{n,0})$. Para cada estágio, o valor de $\sigma_{n,0}$ é reduzido enquanto o valor da função objetivo diminuir. Quando não ocorrem mais alterações com o estágio fixado, um novo estágio passa a ser pesquisado.

Note que se o número máximo de kanbans for muito grande em relação ao seu valor mínimo, a heurística se torna lenta no processo de retirada unitária de kanbans e no cálculo do valor da função objetivo, principalmente para grandes sistemas de produção e/ou grandes horizontes de planejamento. Optou-se então por usar busca por bipartição (ou busca binária) para encontrar os melhores valores de $\sigma_{n,0}$. Inicialmente, para cada item $n \in \{0,1,\dots,N\}$, o limite superior da busca é $(K_n - \Phi_{n,0})$ e o limite inferior é zero.

Considerou-se duas formas de realizar a busca binária no vetor $\sigma_{n,0}$, uma a partir do item final ($n=0$) até os estágios sem predecessores, e a outra no sentido contrário. Observou-se que as duas abordagens podem levar a soluções distintas, dependendo do problema. Tendo em vista que a heurística proposta é rápida, optou-se por executar as duas formas e considerar o melhor valor obtido entre elas.

Uma vez identificada a melhor forma da busca binária, passa-se para uma busca local utilizando-se a mesma abordagem. Considerando o valor de $\sigma_{n,0}$ encontrado na busca binária, avalia-se o valor da solução em uma vizinhança determinada através de uma análise experimental entre as soluções ótima e heurística (sem busca local) de vários problemas pequenos. Dado o ponto $\sigma_0 = (\sigma_{0,0}, \sigma_{1,0}, \dots, \sigma_{i,0}, \dots, \sigma_{n,0})$ os pontos vizinhos são descritos por $\sigma_0 = (\sigma_{0,0}, \sigma_{1,0}, \dots, \tilde{\sigma}_{i,0}, \dots, \sigma_{n,0})$ onde $\tilde{\sigma}_{i,0}$ pertence ao intervalo $[0, K_i - \Phi_{i,0}]$ se i é o estágio final e ao intervalo $[\sigma_{i,0} \pm 2 * E_{i,0}]$ para os demais estágios, onde $E_{i,0}$ é o número de containers do item i necessários para produzir um container do item final.

Dado um valor de σ_0 , determina-se para cada cenário os valores das variáveis de estoque e demanda não atendida através das equações descritas no capítulo anterior. O valor esperado do custo, $Q(\sigma_0)$, é então calculado como a média dos valores da função objetivo $\left\{ \sum_{t=1}^T \left\{ \sum_{n=0}^N h_n \Phi_{n,t} + p \Gamma_t \right\} \right\}$.

A seguir apresenta-se o algoritmo do método heurístico.

Passo 1: Inicialize $\sigma_{n,0}$ com o valor $(K_n - \Phi_{n,0})$, para $n = 0, 1, \dots, N$.

Determine o valor da solução incumbente.

Passo 2: Enquanto teste $\neq 0$, faça

teste = 0

Para $i = 0$ até N faça

Através de uma busca binária verifique se existe entre $(K_i - \Phi_{i,0})$ e 0, algum valor que melhore o valor da solução incumbente. Caso isto ocorra atualize o valor da solução incumbente, altere o valor de $\sigma_{i,0}$ e faça teste=1.

fim do loop i

Passo 3: Solução_auxiliar = solução incumbente.

$$\sigma_{n,0_aux} = \sigma_{n,0}.$$

Repita os passos 1 e 2 modificando a ordem de busca nos estágios, ou seja, de N à 0.

Passo 4: Se solução_auxiliar < solução incumbente

então início = 0, fim = N;

solução incumbente = solução_auxiliar;

$$\sigma_{n,0} = \sigma_{n,0_aux}.$$

senão início = N, fim = 0;

Passo 5: Para i = início até fim faça

Se i = 0 então LI = 0 e LS = $(K_i - \Phi_{i,0})$.

senão LI = $\max\{0, (\sigma_{i,0} - E_{i,0} * 2)\}$,

LS = $\min\{(K_i - \Phi_{i,0}), (\sigma_{i,0} + E_{i,0} * 2)\}$.

Para j = LI até LS faça

$$aux = \sigma_{i,0}.$$

$$\sigma_{i,0} = j.$$

Calcule o valor da solução atual. Se a solução atual é pior que a incumbente faça $\sigma_{i,0}$ igual a aux.

Senão atualize a solução incumbente.

fim do loop j

fim do loop i

No Apêndice B mostra-se que a complexidade da heurística proposta é $O(N^3 * K^2 * C * T)$ onde $K_{max} = \max_n K_n$.

4.3 BUSCA TABU

A heurística apresentada é baseada em busca local e termina em um ótimo local. Soluções de melhor qualidade podem ser encontradas com a utilização de meta-heurísticas que procuram superar a limitação da otimalidade local através de estratégias que exploram diferentes regiões do espaço de soluções.

A busca tabu, proposta por Glover(1989,1995), é uma destas meta-heurísticas. Possui como principais características a utilização de memória adaptativa e estratégias de busca. É iniciada através de uma solução qualquer. A cada iteração um conjunto de movimentos vizinhos é avaliado e o melhor deles é selecionado. Uma nova solução é então obtida e qualquer movimento que altere determinadas características dessa nova solução fica proibido de acordo com as regras de restrição tabu. Através do critério de aspiração um movimento tabu pode ser escolhido quando for particularmente atrativo. A busca tabu termina após um número pré determinado de iterações, ou após um determinado período de tempo ou ainda após um certo número de iterações sem melhoria.

Inicialmente, foi considerada uma vizinhança em que, dado um ponto $\sigma_0 = (\sigma_{0,0}, \sigma_{1,0}, \dots, \sigma_{i,0}, \dots, \sigma_{n,0})$ os pontos vizinhos são descritos por $\sigma_0 = (\sigma_{0,0}, \sigma_{1,0}, \dots, \tilde{\sigma}_{i,0}, \dots, \sigma_{n,0})$, onde $|\tilde{\sigma}_{i,0} - a_i| = 1, \forall i \in \{0, \dots, N\}$. Testes iniciais realizados com esta vizinhança mostraram que o tempo computacional para obter boas soluções é extremamente alto.

Optou-se então por uma vizinhança semelhante à da busca local utilizada no método heurístico, onde $\tilde{\sigma}_{i,0}$ pertence ao intervalo $[0, K_i - \Phi_{i,0}]$ para i fixo. Uma iteração da busca tabu consiste em examinar todos os pontos vizinhos para $i=0, 1, \dots, N$ e escolher o de menor custo.

Os componentes da busca tabu são definidos a seguir:

movimento: corresponde à inserção ou retirada de kanbans de um estágio;

vizinhança: são todos os possíveis números de kanbans obtidos pela inserção ou retirada de kanbans em um estágio;

restrição tabu: proíbe-se que o número de kanbans volte a assumir o valor que possuía na solução sobre a qual é aplicado o movimento atual;

duração tabu: consiste em um número de iterações em que um certo número de kanbans é proibido. Vários testes foram realizados para a definição do valor para duração tabu e os melhores resultados foram

obtidos através de um valor aleatório dependente do número de estágios do problema que é dado pela expressão $N \cdot g$, $g \in U[7,13]$;

critério de aspiração: o número de kanbans tabu pode ser liberado se o valor da função objetivo diminuir em relação ao valor da solução incumbente.

A busca tabu é iniciada com a solução fornecida pelo método heurístico. Os movimentos realizados na busca binária e local, passos 2 e 5 do método heurístico, são incluídos na lista tabu. A busca é realizada a partir do item final (estágio zero) até os estágios sem predecessores.

4.3.1 TESTES PRELIMINARES

A obtenção de bons resultados para a busca tabu dependem da determinação adequada dos parâmetros:

- número máximo de iterações (IT_MÁXIMO);
- escolha do movimento.

Testes realizados mostraram que o melhor valor para IT_MÁXIMO é obtido por uma relação envolvendo o número de estágios e períodos de tempo. Experimentalmente, obteve-se para IT_MÁXIMO a expressão $N \cdot T \cdot 10$.

Os últimos ajustes para a busca tabu foram realizados em relação a escolha do movimento a ser executado. Duas possibilidades foram testadas neste caso:

- escolher dentre os possíveis movimentos aquele que representa a maior melhoria (melhor movimento);
- escolher o primeiro movimento que apresente melhoria (primeiro movimento).

Resultados experimentais mostraram que a escolha do primeiro movimento, apesar de ser computacionalmente mais rápida, apresentou um desempenho ruim e por isto utilizou-se o melhor movimento.

4.4 IMPLEMENTAÇÃO E ANÁLISE DOS RESULTADOS

4.4.1 UM LIMITANTE INFERIOR

A qualidade das soluções fornecidas pela heurística e busca tabu pode ser avaliada através das soluções ótimas para problemas de pequenas estruturas de produção. Entretanto, para problemas com estruturas maiores esta avaliação só é possível através de um limitante inferior do valor ótimo.

Esse limitante consiste na relaxação linear do problema estocástico considerando todas as restrições associadas a cada cenário. Para tal, é necessário que a restrição de produção,

$$\chi_{n,t} = \min \left\{ \begin{array}{l} \sigma_{n,0} + \Phi_{n,0} - \Phi_{n,t-1}, \\ \beta_{n,t}, \\ \left\lfloor \left(\Phi_{q,t-1} + \chi_{q,t} \theta_n \right) / E_{q,n} \right\rfloor \quad q \in P(n) \end{array} \right. \quad \begin{array}{l} n = 0, \dots, N; \\ t = 1, \dots, T; \end{array}$$

seja linearizada. Para cada termo desta equação duas restrições disjuntivas são criadas com a utilização de variáveis binárias.

Sejam $\gamma_{n,t}$, $\xi_{n,t}$, $\delta_{n,t}$ variáveis binárias associadas, respectivamente, ao primeiro, segundo e terceiro termo. Então, esta restrição é modificada para:

$$\chi_{n,t} \leq \left(\sigma_{n,0} + \Phi_{n,0} \right) - \Phi_{n,t-1} \quad n = 0, \dots, N; t = 1, \dots, T,$$

$$\chi_{n,t} \geq \left(\sigma_{n,0} + \Phi_{n,0} \right) - \Phi_{n,t-1} - K_n \left(1 - \gamma_{n,t} \right) \quad n = 0, \dots, N; t = 1, \dots, T,$$

$$\chi_{n,t} \leq \beta_{n,t} \quad n = 0, \dots, N; t = 1, \dots, T,$$

$$\chi_{n,t} \geq \beta_{n,t} \xi_{n,t} \quad n = 0, \dots, N; t = 1, \dots, T,$$

$$\chi_{n,t} \leq \left(\Phi_{q,t-1} + \chi_{q,t} \right) / E_{q,n} \quad n = 0, \dots, N; t = 1, \dots, T; q \in P(n),$$

$$\chi_{n,t} \geq \left(\Phi_{q,t-1} + \chi_{q,t} \right) / E_{q,n} - K_n \left(1 - \delta_{q,n,t} \right) \quad n = 0, \dots, N; t = 1, \dots, T; q \in P(n),$$

e

$$\gamma_{n,t} + \xi_{n,t} + \sum_{q \in P(n)} \delta_{q,n,t} = 1 \quad n = 0, \dots, N; t = 1, \dots, T,$$

$$\gamma_{n,t}, \xi_{n,t}, \delta_{q,n,t} \in \{0,1\} \quad n = 0, \dots, N; t = 1, \dots, T; q \in P(n)$$

A relaxação linear é então obtida ao se desconsiderar a integralidade do vetor σ_0 e das variáveis binárias

4.4.2 ESTRUTURAS PEQUENAS DE PRODUÇÃO

Para avaliar a qualidade das soluções heurística, tabu e do limitante inferior define-se:

- **GAP1:** diferença do valor percentual da solução ótima em relação ao limitante inferior:

$$GAP1 = \frac{Z^* - LI}{LI} * 100$$

onde,

Z^* = valor da solução ótima obtido pela resolução do método ótimo;
 LI = valor do limitante inferior.

- **GAP2:** representa a diferença percentual da solução heurística em relação a solução ótima:

$$GAP2 = \frac{Z_H - Z^*}{Z^*} * 100$$

onde,

Z_H = valor da solução obtido pela heurística.

- **GAP3:** é a melhoria percentual da solução obtida pela busca tabu em relação à solução heurística:

$$GAP3 = \frac{Z_H - Z_{BT}}{Z_{BT}} * 100;$$

onde,

Z_{BT} = valor da solução obtido pela busca tabu.

Dos 108 problemas executados pelo método ótimo, 93 atingiram a solução ótima sem alterar a solução inicial gerada pela heurística.

A Tabela 4.1 apresenta os resultados das diferenças percentuais entre as soluções obtidas de acordo com a relação entre o custo unitário de estoque (h_n) e o custo unitário de demanda não atendida (p). O valor da solução heurística coincide com o valor da a solução ótima em todos os problemas onde $h_n < p$, e o seu pior desempenho ocorreu para os problemas onde $h_n > p$. Através da aplicação da busca tabu o valor ótimo é obtido em todos os problemas, tornando desnecessário os cálculo do GAP 3. O GAP 1 indica que o limitante inferior é pobre e que os piores resultados ocorrem quando $h_n > p$. A Tabela 4.2 mostra que a heurística é insensível em relação ao tipo de demanda.

TABELA 4.1 - GAPS 1 E 2 PARA AS DIFERENTES RELAÇÕES DE CUSTOS ENTRE h_n E P

	$h_n < p$		$h_n e p \in [5,10]$		$h_n > p$	
	N=3, T=4	N=4, T=4	N=3, T=4	N=4, T=4	N=3, T=4	N=4, T=4
GAP1 MÉDIO	13,54%	21,88%	26,20%	36,63%	131,14%	105,66%
GAP1 MÁXIMO	78,24%	109,21%	75,16%	94,83%	316,64%	234,72%
GAP2 MÉDIO*	0,00%	0,00%	0,00%	1,55%	1,06%	5,05%
GAP2 MÁXIMO	0,00%	0,00%	0,00%	2,63%	1,15%	7,85%
PROBLEMAS COM SOLUÇÃO ÓTIMA \neq HEURÍSTICA	-	-	-	5	5	5
TOTAL DE PROBLEMAS	18	18	18	18	18	18

* MÉDIA CONSIDERANDO SOMENTE OS PROBLEMAS ONDE A SOLUÇÃO ÓTIMA É DIFERENTE DA HEURÍSTICA

TABELA 4.2 - GAPS 1 E 2 PARA AS DIFERENTES RELAÇÕES DE DEMANDA

	CRESCENTE		DECRESCENTE		CONSTANTE	
	N=3, T=4	N=4, T=4	N=3, T=4	N=4, T=4	N=3, T=4	N=4, T=4
GAP1 MÉDIO	54,79%	45,04%	49,34%	54,92%	66,76%	64,21%
GAP1 MÁXIMO	248,10%	145,29%	239,29%	234,72%	316,64%	183,72%
GAP2 MÉDIO*	1,08%	2,99%	1,04%	2,77%	1,07%	4,31%
GAP2 MÁXIMO	1,08%	4,52%	1,06%	4,39%	1,15%	7,85%
PROBLEMAS COM SOLUÇÃO ÓTIMA ≠ HEURÍSTICA	1	3	2	4	2	3
TOTAL DE PROBLEMAS	18	18	18	18	18	18

* MÉDIA CONSIDERANDO SOMENTE OS PROBLEMAS ONDE A SOLUÇÃO ÓTIMA É DIFERENTE DA HEURÍSTICA

A Tabela 4.3 apresenta os tempos médios de execução, em segundos, dos métodos e da relaxação linear.

TABELA 4.3 - TEMPOS DE EXECUÇÃO

	N=3	N=4
TEMPO MÉDIO DE EXECUÇÃO MÉTODO ÓTIMO	1209,65	1602,55
TEMPO MÉDIO DE EXECUÇÃO MÉTODO HEURÍSTICO	0,09	0,12
TEMPO MÉDIO DE EXECUÇÃO BUSCA TABU	2,23	4,18
TEMPO MÉDIO DE EXECUÇÃO LIMITANTE INFERIOR	34,59	53,65
TOTAL DE PROBLEMAS	54	54

4.4.3 NOVAS ESTRUTURAS DE PRODUÇÃO

Em uma estrutura de montagem, considera-se que o estágio final possui nível zero e o nível de qualquer outro estágio é um mais o nível de seu estágio sucessor. A profundidade é o seu maior nível. Foram elaboradas estruturas de produção abertas, com um número de nós maior por nível e uma profundidade menor; estruturas fechadas, com um número menor de nós por nível e profundidade maior e estruturas intermediárias. As Figuras 4.3 a 4.6 apresentam estas estruturas.

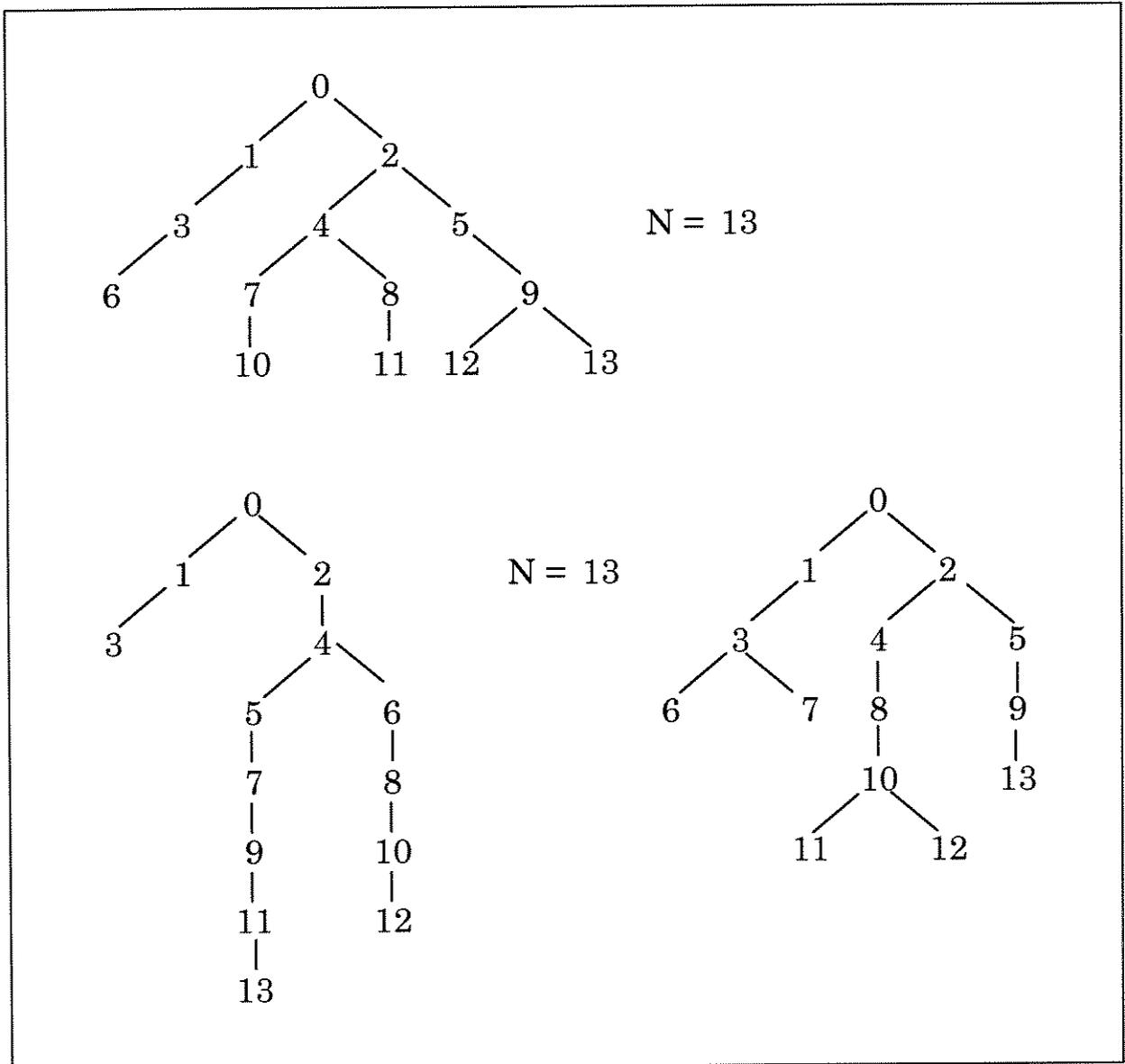


FIGURA 4.3 - ESTRUTURA ABERTA, FECHADA E INTERMEDIÁRIA PARA 13 ESTÁGIOS

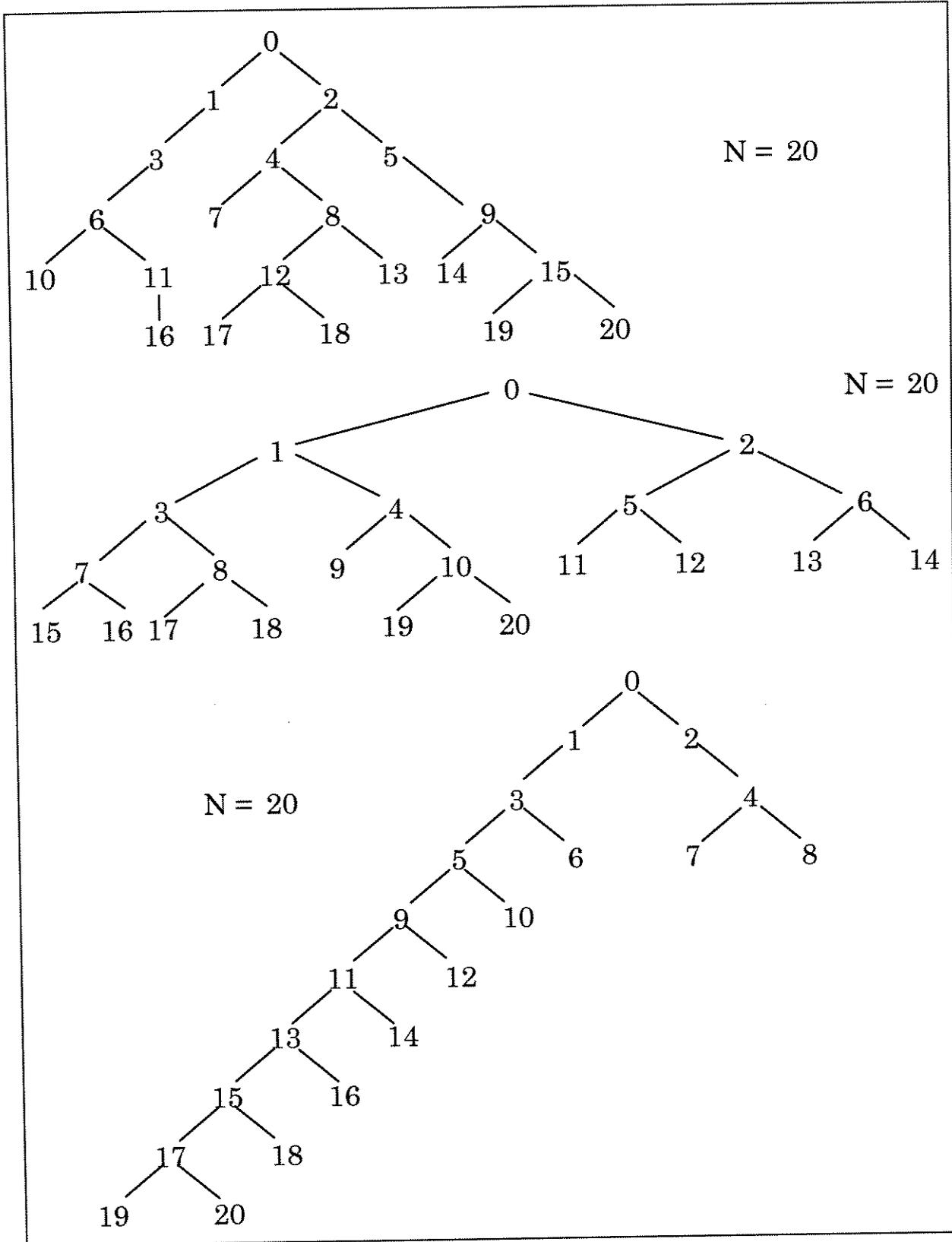
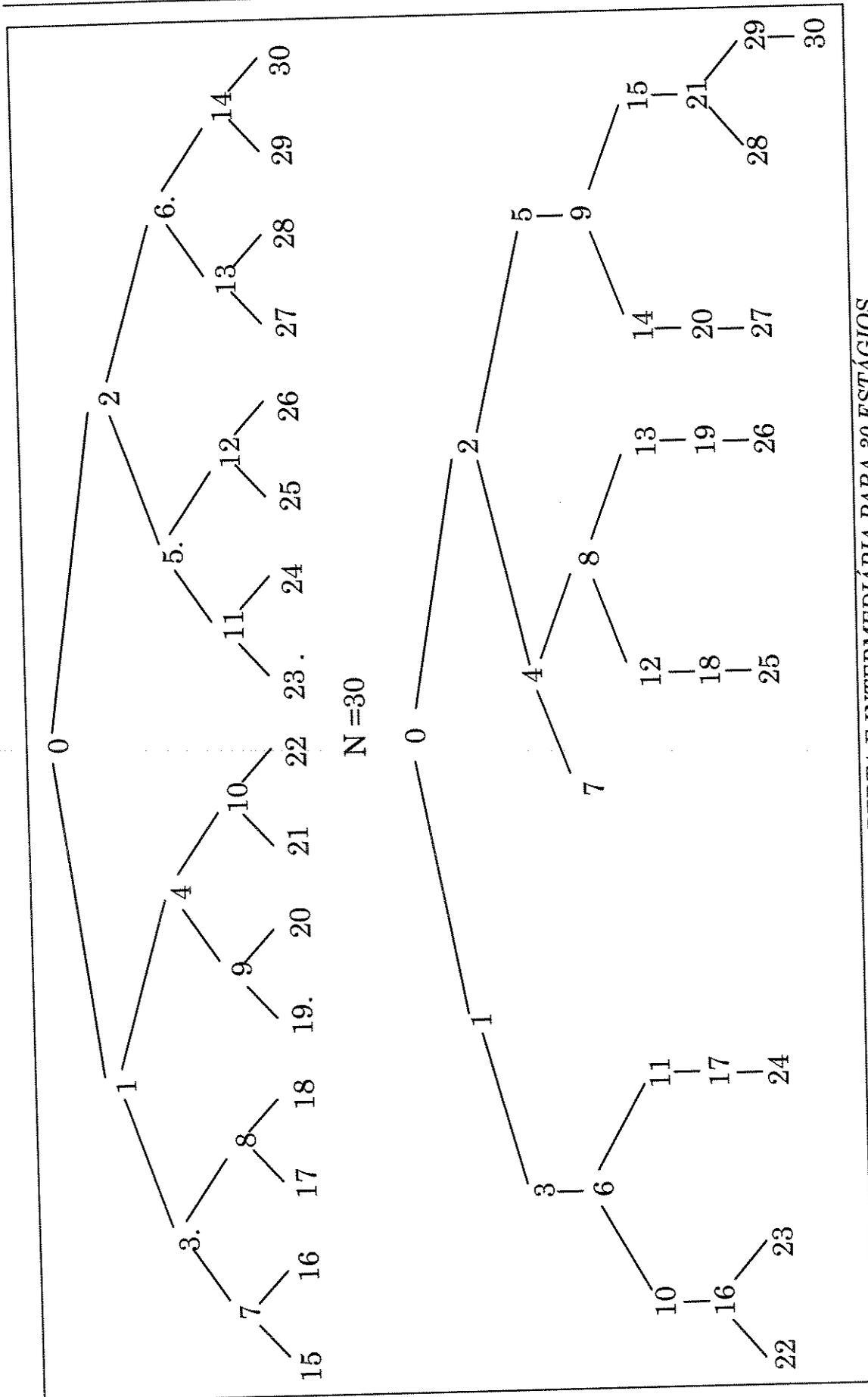


FIGURA 4.4 - ESTRUTURA INTERMEDIÁRIA, ABERTA E FECHADA PARA 20 ESTÁGIOS



N=30

FIGURA 4.5 - ESTRUTURA ABERTA E INTERMEDIÁRIA PARA 30 ESTÁGIOS

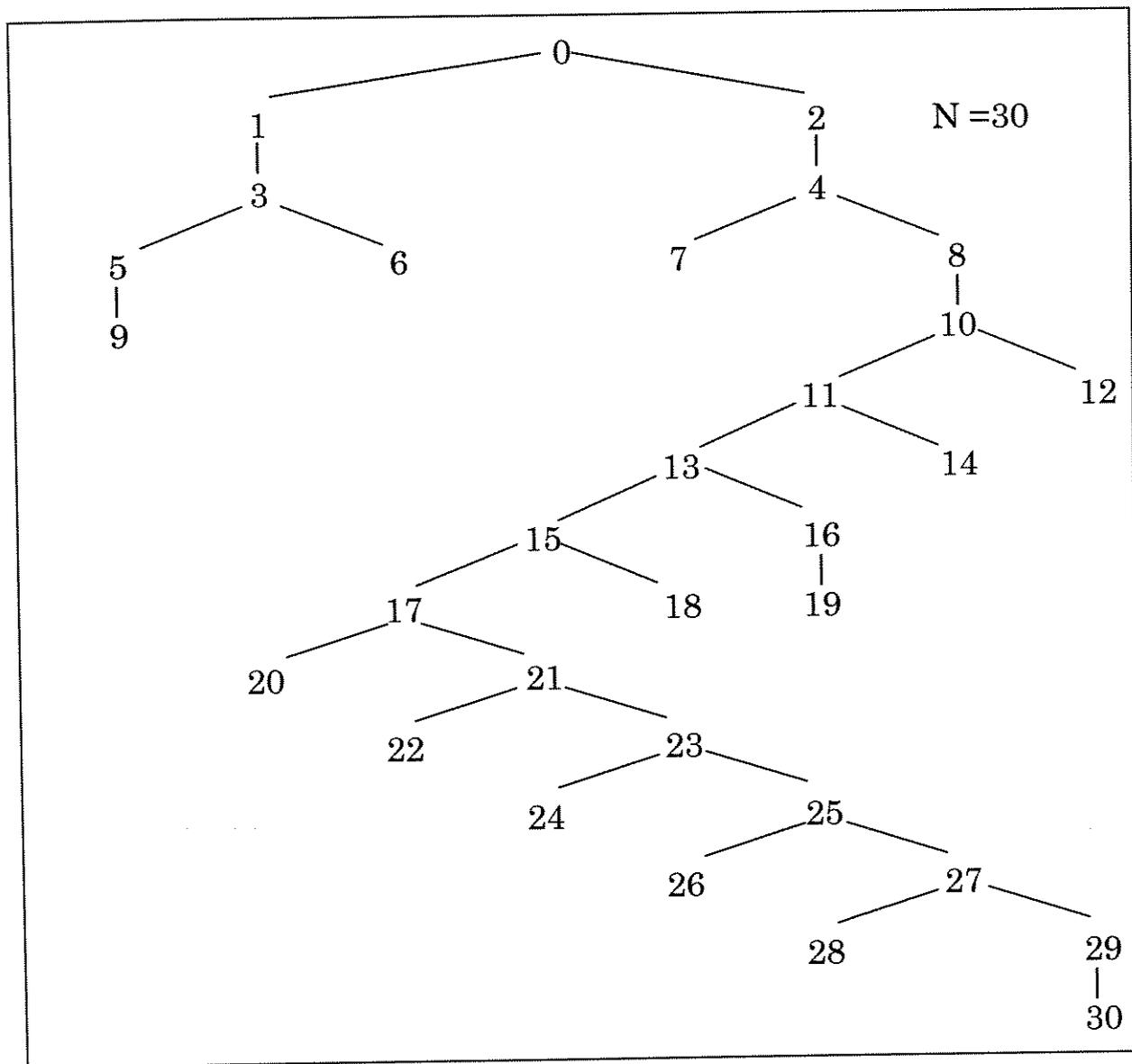


FIGURA 4.6 - ESTRUTURA FECHADA PARA 30 ESTÁGIOS

Foram gerados 945 problemas com as seguintes configurações: $N=6$, $T=6$; $N=13$, $T=6$; $N=13$, $T=10$; $N=20$, $T=6$; $N=20$, $T=10$; $N=30$, $T=6$ e $N=30$, $T=10$. Para cada combinação de custo, demanda e estrutura cinco replicações foram realizadas. Os valores dos parâmetros foram gerados segundo os intervalos definidos no capítulo anterior.

Resultados iniciais da busca tabu, mostraram que o critério de parada por número de iterações é inviável para as estruturas de 20 e 30 estágios. Para 20 estágios a utilização desse critério leva a tempos na ordem de 10.000 segundos. Portanto, foi adotado o critério de parada por tempo de execução e através de testes obteve-se que 1800 segundos apresenta uma boa relação

entre o esforço computacional e qualidade de solução. A Figura 4.7 mostra exemplos do comportamento da busca tabu ao longo do tempo de execução. Nos exemplos a) e d) houve melhoria em relação ao valor da solução heurística com a aplicação de busca tabu. Isto não ocorreu nos exemplos b) e c).

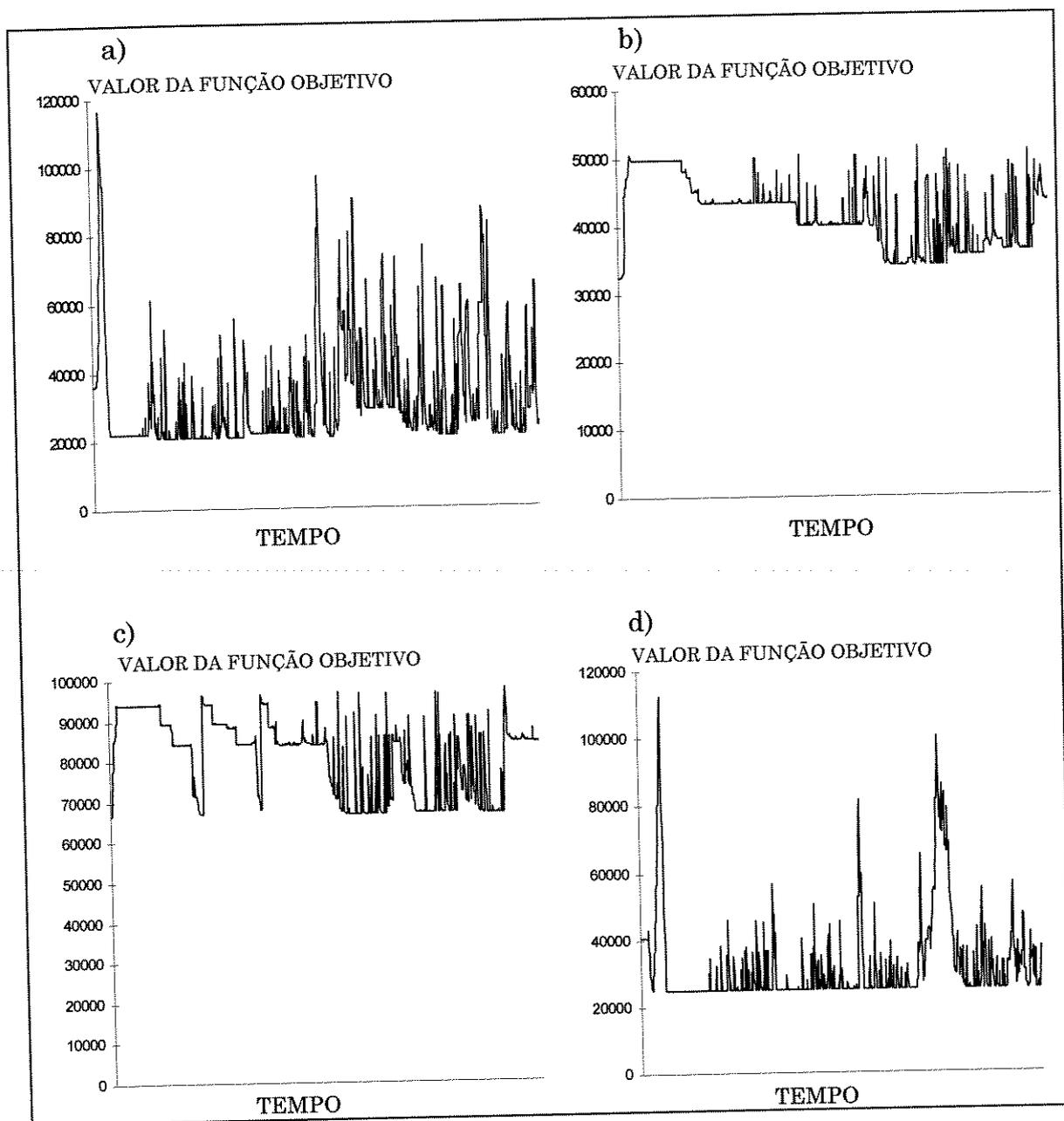


FIGURA 4.7 - COMPORTAMENTO DA BUSCA TABU AO LONGO DO TEMPO

Testes também foram realizados para avaliar a qualidade da solução da busca tabu tendo como solução inicial a heurística sem e com o procedimento de busca local (passo 5 do método heurístico). Para isto define-se:

- $\text{DESVIO 1} = \frac{Z_{\text{BTC}} - Z_{\text{BTS}}}{Z_{\text{BTS}}} * 100$, no caso em que $Z_{\text{BTC}} > Z_{\text{BTS}}$ e

onde,

Z_{BTC} = valor da solução obtido pela busca tabu **com** busca local,

Z_{BTS} = valor da solução obtido pela busca tabu **sem** busca local.

- $\text{DESVIO 2} = \frac{Z_{\text{BTS}} - Z_{\text{BTC}}}{Z_{\text{BTC}}} * 100$, no caso em que $Z_{\text{BTS}} > Z_{\text{BTC}}$

Observa-se pela Tabela 4.6 que com o aumento do tamanho dos problemas a melhoria da solução heurística com busca local diminui em relação à heurística sem busca local. Entretanto, o número de problemas em que esta melhoria ocorre, aumenta. Comparando-se os desvios, nota-se que o segundo produz melhores resultados, isto é, a busca tabu com busca local apresenta maiores ganhos. Nos problemas menores não existem praticamente diferenças entre os desvios, pois o critério de parada de 1800 segundos proporciona uma longa duração da busca tabu, reduzindo o efeito da solução de partida da busca. O tempo de execução da busca local mostrado na Tabela 4.7, corresponde em média a 19% do tempo total da heurística. Sendo assim optou-se pela utilização deste procedimento no método heurístico.

TABELA 4.6- COMPARAÇÕES ENTRE AS SOLUÇÕES SEM E COM BUSCA LOCAL

	N=6, T=4	N=13, T=6	N=20, T=10	N=30, T=10
MELHORIA DA HEURÍSTICA COM BUSCA LOCAL	2,11%	6,38%	0,82%	0,40%
N.º. TOTAL DE PROBLEMAS ONDE A SOLUÇÃO HEURÍSTICA COM BUSCA LOCAL É MELHOR	6	13	68	86
DESVIO 1*	0,00%	0,41%	0,05%	3,24%
DESVIO 2**	0,00%	0,00%	1,92%	17,60%
Nº. PROBLEMAS COM O MESMO VALOR DE SOLUÇÃO TABU	135	132	115	105
Nº. PROBLEMAS ONDE A BUSCA TABU SEM BUSCA LOCAL É MELHOR	0	3	10	3
Nº. PROBLEMAS ONDE A BUSCA TABU COM BUSCA LOCAL É MELHOR	0	0	10	27
TOTAL DE PROBLEMAS	135	135	135	135

* CONSIDERANDO SOMENTE OS PROBLEMAS ONDE BUSCA TABU SEM BUSCA LOCAL OBTIVE MELHORES RESULTADOS

** CONSIDERANDO SOMENTE OS PROBLEMAS ONDE BUSCA TABU COM BUSCA LOCAL OBTIVE MELHORES RESULTADOS

TABELA 4.7 TEMPO MÉDIO DE EXECUÇÃO DA HEURÍSTICA

	TEMPO DA BUSCA BINÁRIA	TEMPO DA BUSCA LOCAL	TEMPO TOTAL
N=03, T=04	0,07	0,02	0,09
N=04, T=04	0,09	0,03	0,12
N=06, T=04	0,24	0,08	0,33
N=13, T=06	2,57	0,74	3,30
N=20, T=10	16,89	4,31	21,20
N=30, T=10	64,62	14,41	79,04

TEMPO DE EXECUÇÃO EM SEGUNDOS

4.4.4 ANÁLISE DOS RESULTADOS

Nesta seção são analisados os resultados obtidos pela heurística e pela busca tabu para os problemas de maior tamanho. Para cada tipo de demanda e relação de custos, 45 problemas foram gerados

O limitante inferior, obtido através da programação linear, mostrou-se muito pobre não sendo possível inferir sobre a qualidade da solução heurística e tabu. Além disso, o grande número de variáveis e restrições para os problemas maiores trouxeram problemas de memória na sua resolução pelo MINOS.

Através das Tabelas 4.8, 4.9 e 4.10 observa-se que a diferença do GAP3 para o três tipos de demanda é da ordem de 1%. O tempo médio da heurística foi praticamente o mesmo para os três tipos de demanda.

TABELA 4.8 - AVALIAÇÃO DA BUSCA TABU EM RELAÇÃO A HEURÍSTICA PARA PROBLEMAS COM DEMANDA CRESCENTE

	N=6 T=4	N=13 T=6	N=13 T=10	N=20 T=6	N=20 T=10	N=30 T=6	N=30 T=10
TEMPO MÉDIO DE EXECUÇÃO HEURÍSTICA	0,34	3,31	6,00	11,92	21,11	44,96	79,51
GAP3 MÉDIO	3,48%	3,57%	9,93%	9,10%	6,62%	7,83%	2,54%
GAP3 MÁXIMO	30,63%	30,43%	42,54%	34,49%	39,59%	41,44%	39,86%
TEMPO MÉDIO DA MELHOR ITERAÇÃO*	5,73	157,15	234,09	498,23	806,73	993,52	787,85
N.º DE PROBLEMAS ONDE A BUSCA TABU ≠ HEURÍSTICA	15	21	32	32	23	23	17

* MÉDIA CONSIDERANDO SOMENTE OS PROBLEMAS ONDE A SOLUÇÃO TABU É DIFERENTE DA HEURÍSTICA

TABELA 4.9 - AVALIAÇÃO DA BUSCA TABU EM RELAÇÃO A HEURÍSTICA PARA PROBLEMAS COM DEMANDA DECRESCENTE

	N=6 T=4	N=13 T=6	N=13 T=10	N=20 T=6	N=20 T=10	N=30 T=6	N=30 T=10
TEMPO MÉDIO DE EXECUÇÃO HEURÍSTICA	0,33	3,29	5,92	11,87	21,09	44,77	78,65
GAP3 MÉDIO	2,89%	3,53%	8,48%	9,10%	5,70%	6,78%	2,39%
GAP3 MÁXIMO	29,97%	28,9%	39,90%	38,94%	39,58%	40,77%	39,19%
TEMPO MÉDIO DA MELHOR ITERAÇÃO*	4,40	168,21	215,55	611,91	735,82	1156,07	703,90
N.º DE PROBLEMAS ONDE A BUSCA TABU ≠ HEURÍSTICA	16	19	30	29	19	22	18

* MÉDIA CONSIDERANDO SOMENTE OS PROBLEMAS ONDE A SOLUÇÃO TABU É DIFERENTE DA HEURÍSTICA

TABELA 4.10 - AVALIAÇÃO DA BUSCA TABU EM RELAÇÃO A HEURÍSTICA PARA PROBLEMAS COM DEMANDA CONSTANTE

	N=6 T=4	N=13 T=6	N=13 T=10	N=20 T=6	N=20 T=10	N=30 T=6	N=30 T=10
TEMPO MÉDIO DE EXECUÇÃO HEURÍSTICA	0,33	3,29	5,82	11,86	21,39	44,70	78,94
GAP3 MÉDIO	2,76%	3,16%	7,25%	8,59%	5,28%	6,56%	2,37%
GAP3 MÁXIMO	29,37%	28,52%	39,37%	38,57%	40,14%	40,63%	39,68%
TEMPO MÉDIO DA MELHOR ITERAÇÃO*	3,31	161,05	197,53	588,24	745,23	1207,50	831,88
N.º DE PROBLEMAS ONDE A BUSCA TABU ≠ HEURÍSTICA	14	19	28	29	18	22	17

* MÉDIA CONSIDERANDO SOMENTE OS PROBLEMAS ONDE A SOLUÇÃO TABU É DIFERENTE DA HEURÍSTICA

Considerando as diferentes relações de custo, observa-se pelas Tabelas 4.11, 4.12 e 4.13 que os maiores valores para o GAP3 ocorrem quando $h_n > p$, o que indica que neste caso o desempenho da heurística é inferior, como ocorreu também para estruturas pequenas. Verifica-se também que para os problemas onde $h_n < p$ o ganho apresentado pela busca tabu é muito pequeno em relação ao valor da solução heurística. Os problemas onde h_n e $p \in [5,10]$ possuem um ganho intermediário e apresentam o maior tempo médio para a obtenção da melhor solução. O tempo de execução da heurística não apresentou diferenças discrepantes nas três relações de custo.

TABELA 4.11 - AVALIAÇÃO DA BUSCA TABU EM RELAÇÃO A HEURÍSTICA PARA $h_n > P$

	N=6 T=4	N=13 T=6	N=13 T=10	N=20 T=6	N=20 T=10	N=30 T=6	N=30 T=10
TEMPO MÉDIO DE EXECUÇÃO HEURÍSTICA	0,33	3,43	6,17	12,14	23,48	45,29	84,35
GAP3 MÉDIO	7,21%	7,48%	17,21%	18,52%	13,19%	14,86%	6,46%
GAP3 MÁXIMO	30,63%	30,43%	42,54%	38,94%	40,14%	41,44%	39,86%
TEMPO MÉDIO DA MELHOR ITERAÇÃO*	2,17	97,50	139,57	476,72	675,06	1164,36	886,48
N.º DE PROBLEMAS ONDE A BUSCA TABU \neq HEURÍSTICA	24	34	40	44	36	36	26

* MÉDIA CONSIDERANDO SOMENTE OS PROBLEMAS ONDE A SOLUÇÃO TABU É DIFERENTE DA HEURÍSTICA

TABELA 4.12 - AVALIAÇÃO DA BUSCA TABU EM RELAÇÃO A HEURÍSTICA PARA $h_n < P$

	N=6 T=4	N=13 T=6	N=13 T=10	N=20 T=6	N=20 T=10	N=30 T=6	N=30 T=10
TEMPO MÉDIO DE EXECUÇÃO HEURÍSTICA	0,32	3,16	5,34	11,38	18,84	43,54	75,30
GAP3 MÉDIO	0,00%	0,00%	1,42%	0,04%	0,00%	0,00%	0,01%
GAP3 MÁXIMO	0,00%	0,00%	16,92%	0,64%	0,00%	0,00%	0,14%
TEMPO MÉDIO DA MELHOR ITERAÇÃO*	-	-	284,14	530,21	-	-	43,06
N.º DE PROBLEMAS ONDE A BUSCA TABU \neq HEURÍSTICA	-	-	13	8	-	-	8

* MÉDIA CONSIDERANDO SOMENTE OS PROBLEMAS ONDE A SOLUÇÃO TABU É DIFERENTE DA HEURÍSTICA

TABELA 4.13 - AVALIAÇÃO DA BUSCA TABU EM RELAÇÃO A HEURÍSTICA PARA $h_n, e, p \in [5, 10]$

	N=6 T=4	N=13 T=6	N=13 T=10	N=20 T=6	N=20 T=10	N=30 T=6	N=30 T=10
TEMPO MÉDIO DE EXECUÇÃO HEURÍSTICA	0,34	3,30	6,13	12,13	21,27	45,60	77,46
GAP3 MÉDIO	1,92%	2,78%	5,97%	8,24%	4,42%	6,31%	0,84%
GAP3 MÁXIMO	16,63%	14,16%	21,63%	27,61%	22,15%	27,29%	11,24%
TEMPO MÉDIO DA MELHOR ITERAÇÃO*	6,98	206,75	295,97	662,28	901,98	961,58	934,04
N.º DE PROBLEMAS ONDE A BUSCA TABU \neq HEURÍSTICA	21	24	34	38	24	31	18

* MÉDIA CONSIDERANDO SOMENTE OS PROBLEMAS ONDE A SOLUÇÃO TABU É DIFERENTE DA HEURÍSTICA

Finalmente, através das tabelas anteriores é possível verificar que apesar do tempo computacional da heurística apresentar um comportamento pseudo-polinomial, seu tempo médio real é relativamente baixo. Observa-se também que ela é mais sensível ao aumento do número de estágios do que ao número de períodos.

4.4.5 UM NOVO PONTO DE PARTIDA PARA O MÉTODO HEURÍSTICO

A análise das tabelas 4.11, 4.12 e 4.13 mostrou que a heurística é de boa qualidade para os problemas com $h_n < p$ e que o seu pior desempenho ocorre nos problemas onde $h_n > p$. Estes resultados podem estar associados com o ponto de partida $(K_n - \Phi_{n,0})$ no passo 2 do algoritmo heurístico.

Para $h_n < p$ as melhores soluções estão ligadas a um número maior de kanbans que apesar de elevar o nível de estoque, tende a diminuir a existência de demanda não atendida que é mais cara, enquanto que para $h_n > p$ um número pequeno de kanbans proporciona menores custos já que demanda não atendida é mais barata que o estoque. Portanto, um ponto de partida a ser pesquisado é zero, ou seja, não inserir kanbans no sistema de produção e manter somente aqueles já existentes no estoque inicial.

O método heurístico foi executado a partir destes dois pontos e a tabela 4.14 apresenta a inicialização que obteve melhor a solução e de quanto é, em média, esta melhoria. Por exemplo, para $N=20$ e $T=06$, o ponto de partida $(K_n - \Phi_{n,0})$ apresenta para $h_n < p$ soluções em média 53,83% melhores em relação à aquelas obtidas através do ponto inicial zero. Entretanto para $h_n > p$, o ponto inicial zero é, em média, 18,33% melhor.

Portanto, através da tabela 4.14 é possível observar que o método heurístico apresenta um melhor desempenho para $h_n > p$ quando o valor zero é utilizado como ponto de partida.

TABELA 4.14 - MELHORES PONTOS DE PARTIDA PARA A HEURÍSTICA NAS DIFERENTES RELAÇÕES DE CUSTO

	$h_n < p$	$h_n > p$	$h_n \text{ e } p \in [5,10]$
N=03, T=04	HK - 68,67%	H0 - 0,29%	HK - 15,53%
N=04, T=04	HK - 52,84%	H0 - 1,40%	HK - 5,90%
N=06, T=04	HK - 66,88%	H0 - 15,83%	HK - 5,19%
N=13, T=06	HK - 37,60%	H0 - 3,85%	HK - 5,17%
N=13, T=10	HK - 71,49%	H0 - 15,79%	HK - 10,28%
N=20, T=06	HK - 53,83%	H0 - 18,33%	H0 - 1,85%
N=20, T=10	HK - 56,62%	H0 - 12,60%	HK - 6,91%
N=30, T=06	HK - 44,58%	H0 - 26,63%	H0 - 8,88%
N=30, T=10	HK - 48,25%	HK - 1,61%	HK - 8,01%

HK = HEURÍSTICA COM PONTO DE PARTIDA ($K_n - \Phi_{n,0}$)

H0 = HEURÍSTICA COM PONTO DE PARTIDA ZERO

A busca tabu foi executada para os problemas com estruturas maiores de produção e demanda constante, considerando como solução inicial o valor obtido pela heurística através dos diferentes pontos de partida.

Pela tabela 4.15 é possível observar que a solução da busca tabu tendo como solução inicial o valor heurístico obtido a partir de $(K_n - \Phi_{n,0})$ apresentou melhores resultados em relação à solução da busca tabu tendo como solução o valor heurístico obtido a partir de zero. Note que os resultados tornam-se significativos à medida que o tamanho das configurações aumenta. Verifica-se também que estas soluções são obtidas com maior rapidez.

TABELA 4.15 - AVALIAÇÃO DA BUSCA TABU PARA $h_n < p$ TENDO COMO SOLUÇÃO INICIAL A SOLUÇÃO HEURÍSTICA COM DIFERENTES PONTOS DE PARTIDA

	N=06 T=04	N=13 T=06	N=13 T=10	N=20 T=06	N=20 T=10	N=30 T=06	N=30 T=10
MELHOR SOLUÇÃO TABU	0,00%	TK 0,18%	TK 0,01%	TK 1,25%	TK 20,43%	TK 12,51%	TK 51,10%
TEMPO MÉDIO DA MELHOR ITERAÇÃO PARA PONTO DE PARTIDA: $(K_n - \Phi_{n,0})$	-	-	60,39	48,94	-	-	3,65
TEMPO MÉDIO DA MELHOR ITERAÇÃO PARA PONTO DE PARTIDA: 0	11,07	221,42	216,82	952,26	1265,50	857,38	549,92
NÚMERO TOTAL DE PROBLEMAS	15	15	15	15	15	15	15

TK = BUSCA TABU TENDO COMO SOLUÇÃO INICIAL A SOLUÇÃO DA HEURÍSTICA COM PONTO DE PARTIDA $(K_n - \Phi_{n,0})$

T0 = BUSCA TABU TENDO COMO SOLUÇÃO INICIAL A SOLUÇÃO DA HEURÍSTICA COM PONTO DE PARTIDA ZERO

"-" A BUSCA TABU NÃO MELHOROU A SOLUÇÃO FORNECIDA PELA HEURÍSTICA
TEMPO MÉDIO DA MELHOR ITERAÇÃO EM SEGUNDOS

Através das tabelas 4.16 e 4.17 verifica-se que para $h_n > p$ e h_n e $p \in [5,10]$ não existem diferenças discrepantes entre os valores obtidos pela busca tabu a partir da heurística com diferentes pontos iniciais.

TABELA 4.16 - AVALIAÇÃO DA BUSCA TABU PARA $h_n > p$ TENDO COMO SOLUÇÃO INICIAL A SOLUÇÃO HEURÍSTICA COM DIFERENTES PONTOS DE PARTIDA

	N=06 T=04	N=13 T=06	N=13 T=10	N=20 T=06	N=20 T=10	N=30 T=06	N=30 T=10
MELHOR SOLUÇÃO TABU	0,00%	0,00%	0,00%	TK 0,59%	TK 0,78%	T0 0,65%	T0 3,41%
TEMPO MÉDIO DA MELHOR ITERAÇÃO PARA PONTO DE PARTIDA: $(K_n - \Phi_{n,0})$	0,85	93,88	108,98	510,14	504,52	1066,47	497,66
TEMPO MÉDIO DA MELHOR ITERAÇÃO PARA PONTO DE PARTIDA: 0	0,97	87,42	169,78	442,64	647,18	97,12	172,24
NÚMERO TOTAL DE PROBLEMAS	15	15	15	15	15	15	15

TK = BUSCA TABU TENDO COMO SOLUÇÃO INICIAL A SOLUÇÃO DA HEURÍSTICA COM PONTO DE PARTIDA $(K_n - \Phi_{n,0})$

T0 = BUSCA TABU TENDO COMO SOLUÇÃO INICIAL A SOLUÇÃO DA HEURÍSTICA COM PONTO DE PARTIDA ZERO

TABELA 4.17 - AVALIAÇÃO DA BUSCA TABU PARA $h_n E p \in [5,10]$ TENDO COMO SOLUÇÃO INICIAL A SOLUÇÃO HEURÍSTICA COM DIFERENTES PONTOS DE PARTIDA

	N=06 T=04	N=13 T=06	N=13 T=10	N=20 T=06	N=20 T=10	N=30 T=06	N=30 T=10
MELHOR SOLUÇÃO TABU	0,00%	TK 0,17%	TK 0,65%	T0 0,03%	T0 0,56%	T0 0,82%	T0 2,03%
TEMPO MÉDIO DA MELHOR ITERAÇÃO PARA PONTO DE PARTIDA: $(K_n - \Phi_{n,0})$	2,24	110,02	199,36	578,20	389,76	704,53	441,48
TEMPO MÉDIO DA MELHOR ITERAÇÃO PARA PONTO DE PARTIDA: 0	11,17	177,41	289,48	665,91	716,94	474,07	218,84
NÚMERO TOTAL DE PROBLEMAS	15	15	15	15	15	15	15

TK = BUSCA TABU TENDO COMO SOLUÇÃO INICIAL A SOLUÇÃO DA HEURÍSTICA COM PONTO DE PARTIDA $(K_n - \Phi_{n,0})$

T0 = BUSCA TABU TENDO COMO SOLUÇÃO INICIAL A SOLUÇÃO DA HEURÍSTICA COM PONTO DE PARTIDA ZERO

CAPÍTULO 5

UM MODELO DETERMINÍSTICO PARA O SISTEMA KANBAN DE PRODUÇÃO

5.1 INTRODUÇÃO

Neste capítulo apresenta-se um modelo determinístico para o sistema kanban de produção. Soluções são obtidas através de um método heurístico e de busca tabu. Os resultados computacionais são analisados.

5.2 UM MODELO DETERMINÍSTICO

O modelo aqui discutido é semelhante ao apresentado por Bitran e Chang (1987) e considera a demanda Λ_t e capacidade de produção $\beta_{n,t}$ como parâmetros determinísticos. Demanda não atendida não é permitida em nenhum estágio. Todos os itens produzidos em um estágio podem ser usados pelo seu predecessor no mesmo período de tempo, ou seja, $\theta_n = 1$. O objetivo é determinar o número de kanbans em cada estágio a ser inserido pela gerência no início do horizonte de planejamento que minimize o custo de estoque em todos os estágios. Inafectibilidade pode ocorrer devido às restrições de capacidade e a proibição de demanda não atendida.

Bitran e Chang (1987) consideram o estágio final como apenas uma operação de montagem, não havendo a necessidade de kanbans. A produção deste estágio em cada período é a própria demanda externa do período, e conseqüentemente não existe estoque. A função objetivo consiste na somatória ponderada do número de kanbans em cada estágio e pode ser interpretada como um limitante superior do valor do estoque no sistema em qualquer período.

Os autores utilizam um quarto termo na restrição de produção para garantir que a produção de cada item ao longo do horizonte de planejamento não exceda sua demanda efetiva. Este termo também é introduzido na restrição de produção do modelo aqui proposto.

Seja Q_n a demanda efetiva, em termos do número de containers cheios do item n , imposta no estágio n durante todo o horizonte de planejamento:

$$\begin{cases} Q_0 = \sum_{t=1}^T \Lambda_t \\ Q_n = \max \left\{ 0, \left(E_{n,s(n)} Q_{s(n)} - \Phi_{n,0} \right) \right\} \quad n = 1, \dots, N \end{cases}$$

A restrição a ser inserida na equação da produção é,

$$\chi_{n,t} \leq Q_n - \sum_{r=1}^{t-1} \chi_{n,r},$$

que garante que somente haverá produção no período t se a demanda efetiva não foi atendida pela produção nos períodos anteriores.

Portanto, o modelo determinístico é formulado como:

$$\text{Minimizar } \left\{ \sum_{t=1}^T \left\{ \sum_{n=0}^N h_n \Phi_{n,t} \right\} \right\}$$

Sujeito a

$$\sigma_{0,t} = \sigma_{0,t-1} - \chi_{0,t} + \Lambda_t \quad (1)$$

$$\sigma_{n,t} = \sigma_{n,t-1} - \chi_{n,t} + E_{n,s(n)} \chi_{s(n),t} \quad (2)$$

$$\Phi_{0,t} = \Phi_{0,t-1} + \chi_{0,t} - \Lambda_t \quad t = 1, \dots, T \quad (3)$$

$$\Phi_{n,t} = \Phi_{n,t-1} + \chi_{n,t} - E_{n,s(n)} \chi_{s(n),t} \quad n = 1, \dots, N; t = 1, \dots, T \quad (4)$$

$$\chi_{n,t} = \min \left\{ \begin{aligned} &\sigma_{n,t-1}, \\ &\beta_{n,t} \\ &\left[\left(\Phi_{q,t-1} + \chi_{q,t} \right) / E_{q,n} \right] q \in P(n) \end{aligned} \right\} \quad (5)$$

$$Q_n - \sum_{r=1}^{t-1} \chi_{n,r} \quad n = 0, \dots, N; t = 1, \dots, T$$

$$\sigma_{n,0} \in \left\{ 0, 1, \dots, (K_n - \Phi_{n,0}) \right\} \quad (6)$$

De modo análogo ao discutido no capítulo 2 para o modelo estocástico tem-se que $\sigma_{n,0} + \Phi_{n,0} = \sigma_{n,t} + \Phi_{n,t}, \forall t$ e portanto as restrições (1) e (2) do modelo podem ser omitidas. O primeiro termo da restrição (5) é substituído por $\sigma_{n,0} + \Phi_{n,0} - \Phi_{n,t-1}$.

Este modelo é um problema de programação inteira não linear que pode ser linearizado através da introdução de variáveis binárias como realizado no capítulo 4. Devido a limitação do tamanho dos problemas resolvidos por um algoritmo *branch and bound*, optou-se pela utilização de métodos heurísticos para a obtenção de soluções sub-ótimas.

*** O Método Heurístico**

O método heurístico é o mesmo utilizado no modelo estocástico, com as devidas alterações associadas ao caso determinístico. Soluções infactíveis podem ocorrer durante seu processo de execução e são desconsideradas.

*** Busca Tabu**

As modificações na busca tabu são basicamente as mesmas realizadas no método heurístico e as soluções infactíveis também são desconsideradas.

Bitran e Chang (1987) apresentam também um teste de factibilidade que procura encontrar uma solução factível tal que cada estágio satisfaça a demanda efetiva de seu sucessor, produzindo o mais tarde possível. Este teste de factibilidade pode ser utilizado como um teste inicial para que o método heurístico e busca tabu sejam executados somente quando o problema é factível. O algoritmo de factibilidade é descrito a seguir:

Passo 1: Para $t=1, \dots, T$ faça

$$\bar{\chi}_{0,t} \leftarrow \Lambda_t$$

$$n \leftarrow 0$$

Passo 2: se $n=0$ então

$$Q_{0,1} \leftarrow \max\{0, \bar{\chi}_{0,1} - \Phi_{0,0}\}$$

$$\left\{ \begin{array}{l} Q_{0,t} \leftarrow \max\{0, \sum_{r=1}^t \bar{\chi}_{0,r} - \Phi_{0,0}\} - \sum_{r=1}^{t-1} Q_{0,r} \\ \text{para } t = 2, 3, \dots, T \end{array} \right.$$

senão

$$Q_{n,1} \leftarrow \max\{0, [E_{n,s(n)} \bar{\chi}_{s(n),1} - \Phi_{n,0}]\}$$

$$\left\{ \begin{array}{l} Q_{n,t} \leftarrow \max\{0, E_{n,s(n)} \sum_{r=1}^t \bar{\chi}_{n,r} - \Phi_{n,0}\} - \sum_{r=1}^{t-1} Q_{n,r} \\ \text{para } t = 2, 3, \dots, T \end{array} \right.$$

$$t \leftarrow T$$

Passo 3: $\chi_{n,t} \leftarrow \min\{Q_{n,t}, \beta_{n,t}\}$

Se $t=1$ então vá para o **passo 4**

$$\text{senão } \left\{ \begin{array}{l} Q_{n,t-1} \leftarrow Q_{n,t-1} + \max\{0, Q_{n,t} - \beta_{n,t}\} \\ t \leftarrow t - 1, \text{ volte para o início do } \textit{passo 3} \end{array} \right.$$

Passo 4: Se $\beta_{n,1} < Q_{n,1}$

então o problema é infactível pare

senão, se $n=N$ então o problema é factível e pare

$$\text{senão } \left\{ \begin{array}{l} n \leftarrow n + 1 \\ \text{volte para o } \textit{passo 2} \end{array} \right.$$

Os passos 2 e 3 deste algoritmo calculam $\chi_{n,t}$. e através destes valores é possível obter $\sigma_{n,0}$ a partir da restrições (1) e (2). De (2) segue-se que:

$$\sigma_{n,t} = \sigma_{n,0} - \sum_{r=1}^t \chi_{n,r} + E_{n,s(n)} \sum_{r=1}^t \chi_{s(n),r} \geq 0$$

e portanto,

$$\sigma_{n,0} = \max_t \left(E_{n,s(n)} \sum_{r=1}^t \chi_{s(n),r} - \sum_{r=1}^t \chi_{n,r} \right) \text{ para } n=1,\dots,N. \quad (7)$$

Para $n=0$, $\sigma_{0,0}$ é obtido da mesma forma trocando-se o primeiro termo de (7) por $\sum_{r=1}^t \Lambda_r$.

5.3 IMPLEMENTAÇÃO E ANÁLISE DOS RESULTADOS

Foram consideradas as estruturas de 13, 20 e 30 estágios apresentadas no capítulo 4 e os parâmetros são gerados aleatoriamente através de uma distribuição uniforme nos intervalos:

- $\Phi_{n,0} \in [0; 10]$;
- $\Lambda_t \in [1;15]$;
- $E_{n,s(n)} \in [1; 2]$;
- $h_n \in [5; 10]$;
- $\beta_{0,t} \in R^*g$,

$\beta_{n,t} \in R * g * E_{n,0}$, $n=1,\dots,N$ onde $g \in [10; 25]$ e $R=1,0$ ou $R=1,1$;

O fator $R=1,0$ está associado a uma capacidade de produção folgada e o fator $R=1,1$ a uma capacidade apertada. Estes valores foram obtidos experimentalmente. O intervalo $[10;25]$ foi escolhido de forma compatível com o intervalo de geração da demanda.

- $K_n = \max_t \{ \beta_{n,t} \}$

Esta definição de K_n procura evitar a geração de grande número de problemas infactíveis.

Foram gerados 180 problemas com as seguintes configurações: N=13, T=6; N=13, T=10; N=20, T=6; N=20, T=10; N=30, T=6; N=30, T=10. Para cada combinação de capacidade de produção e estrutura cinco replicações foram realizadas.

Testes computacionais foram realizados para determinar o tempo de execução para busca tabu. Novamente, 1800 segundos é o que apresenta uma boa relação entre o esforço computacional e qualidade de solução.

Testes também foram realizados para comparar soluções de partida para o método heurístico considerando o valor da solução do teste de factibilidade e o número máximo de kanbans que pode ser inserido no sistema $(K_n - \Phi_{n,0})$. Os resultados mostraram que dos 180 problemas executados 12 são infactíveis. Considerando somente os problemas factíveis, observou-se que a heurística tendo a solução do teste de factibilidade como partida obteve em 20% dos problemas soluções em média 3% melhores que aquelas onde o ponto de partida foi o número máximo de kanbans. Quando este último é utilizado obteve-se um ganho médio de 44% em 67% do problemas em relação ao ponto de partida dado pelo teste de factibilidade.

Em relação à busca tabu as soluções utilizando o teste de factibilidade como partida apresentaram em média um ganho de 4% em 25% dos problemas, enquanto que 62% dos problemas obtiveram uma solução 25% em média melhor com o número máximo de kanbans como solução de partida.

Observou-se também, a ocorrência de infactibilidade na solução obtida com o número máximo de kanbans em dois problemas. A aplicação da heurística e busca tabu geraram soluções de má qualidade quando comparadas com aquelas obtidas pelo teste de factibilidade como ponto de partida. Assim, optou-se pelo número máximo de kanbans como solução de partida quando esta solução é factível e pelo teste de factibilidade, caso contrário.

Para a análise dos resultados obtidos utiliza-se o GAP 3, já definido no capítulo 3, que representa a melhoria percentual da solução obtida pela busca tabu em relação à solução heurística:

$$\text{GAP 3} = \frac{Z_H - Z_{BT}}{Z_H} * 100.$$

As Tabelas 6.1 e 6.2 apresentam este GAP para as diferentes categorias de capacidade de produção. Para cada categoria 90 problemas foram gerados. Observa-se por estas tabelas que o tempo de execução do teste de factibilidade é rápido para as duas categorias. A solução heurística apresenta tempos maiores de execução quando a capacidade de produção é folgada. Não houve discrepâncias entre as duas categorias em relação ao GAP 3. Além

disso, deve-se notar que o ganho médio da solução tabu em relação à solução obtida pelo teste de factibilidade é de 70%.

TABELA 6.1 - AVALIAÇÃO DA BUSCA TABU EM RELAÇÃO A HEURÍSTICA PARA CAPACIDADE DE PRODUÇÃO APERTADA

	N=13, T=6	N=13, T=10	N=20, T=6	N=20, T=10	N=30, T=6	N=30, T=10
TEMPO MÉDIO DE EXECUÇÃO DO TESTE DE FACTIBILIDADE	0,02	0,02	0,03	0,03	0,04	0,04
TEMPO MÉDIO DE EXECUÇÃO HEURÍSTICA	1,53	2,80	4,46	8,78	27,42	49,78
GAP3 MÉDIO	2,41	1,30	1,82	3,38	0,22	-
GAP3 MÁXIMO	11,18	10,24	12,20	19,33	1,72	-
TEMPO MÉDIO DA MELHOR ITERAÇÃO*	383,49	70,57	937,91	591,49	773,18	-
N.º DE PROBLEMAS INFECTÍVEIS	-	3	1	-	4	3

* MÉDIA CONSIDERANDO SOMENTE OS PROBLEMAS ONDE A SOLUÇÃO TABU É DIFERENTE DA HEURÍSTICA

TABELA 6.1 - AVALIAÇÃO DA BUSCA TABU EM RELAÇÃO A HEURÍSTICA PARA CAPACIDADE DE PRODUÇÃO COM MAIOR FOLGA

	N=13, T=6	N=13, T=10	N=20, T=6	N=20, T=10	N=30, T=6	N=30, T=10
TEMPO MÉDIO DE EXECUÇÃO DO TESTE DE FACTIBILIDADE	0,02	0,02	0,03	0,03	0,04	0,04
TEMPO MÉDIO DE EXECUÇÃO HEURÍSTICA	1,52	2,88	4,97	9,70	41,38	78,33
GAP3 MÉDIO	1,62	0,43	2,06	0,50	1,40	0,73
GAP3 MÁXIMO	14,01	4,90	16,72	3,39	7,61	4,38
TEMPO MÉDIO DA MELHOR ITERAÇÃO*	130,03	396,41	340,70	731,99	242,73	699,71
N.º DE PROBLEMAS INFACÍVEIS	-	-	-	-	1	-

* MÉDIA CONSIDERANDO SOMENTE OS PROBLEMAS ONDE A SOLUÇÃO TABU É DIFERENTE DA HEURÍSTICA

CAPÍTULO 6

CONCLUSÕES

A proposta deste trabalho foi a elaboração de um modelo estocástico para o sistema kanban de produção e o desenvolvimento de métodos para a sua resolução. Até o momento, não é de nosso conhecimento a existência de trabalhos semelhantes nesta área.

Inicialmente, o modelo estocástico para o sistema kanban foi resolvido através do método L-shaped inteiro para programação estocástica inteira com recurso. Foram realizados testes computacionais com estruturas de produção pequenas e os resultados mostram que o corte de otimalidade não é de boa qualidade, pois um número muito grande de cortes é necessário para obtenção de soluções ótimas. Dessa forma foi constatado que o método ótimo não é aplicável a problemas práticos devido ao porte destes e à necessidade de rápidas decisões.

O passo seguinte do trabalho realizado foi o desenvolvimento de uma heurística para o problema em estudo e a comparação do desempenho dessa heurística com o algoritmo ótimo. O teste de desempenho constituiu em utilizar as soluções obtidas pela heurística como soluções iniciais para o método ótimo e medir o erro médio entre a solução inicial e a solução ótima. Os resultados deste teste mostraram que em 86% dos problemas resolvidos, a solução inicial heurística é ótima e, nos demais problemas, o erro médio ficou abaixo de 5%. Deve-se observar que o tempo computacional utilizado pela heurística é pequeno.

Após essa análise de desempenho da heurística seguiu-se a implementação da busca tabu para melhorar a qualidade de sua solução. Duas abordagens foram testadas. A primeira delas parte da solução heurística como solução inicial. Na segunda abordagem utiliza-se como solução inicial aquela obtida pelo método heurístico sem o procedimento de busca local. Testes computacionais mostram que a primeira abordagem é mais eficiente levando a soluções finais de melhor qualidade.

Comparando as soluções obtidas após a aplicação de busca tabu com as soluções obtidas pelo método ótimo, verificou-se que embora tenham sido implementados somente os componentes de busca tabu correspondentes ao uso de memória de curto prazo, o procedimento mostrou-se eficaz. A heurística seguida de busca tabu gerou soluções ótimas para todos os problemas testados utilizando um baixo tempo computacional.

O desempenho da heurística e da busca tabu revelou-se insensível aos diferentes tipos de estruturas. A solução heurística é de boa qualidade para o caso em que o custo de estoque é menor que o custo de demanda não atendida. De fato, a heurística obteve soluções ótimas para todos os problemas de pequeno porte testados que possuíam esta característica de custo e nos problemas maiores, praticamente, não houve melhoria da solução com aplicação de busca tabu. Aparentemente, a heurística tem um desempenho pior quando o custo de estoque é maior que o custo de demanda não atendida pois, neste caso, a busca tabu apresenta os maiores ganhos com relação a solução inicial.

A complexidade da heurística mostra que no pior caso o tempo computacional apresenta um comportamento pseudo-polinomial com relação aos dados de entrada. O tempo computacional médio observado, entretanto, é relativamente baixo, indicando que a heurística pode ser utilizada em problemas médios e grandes quando o fator tempo for prioritário. Uma característica importante do tempo computacional é sua maior sensibilidade ao aumento no número de estágios que ao número de períodos.

Um modelo determinístico para o sistema kanban também foi apresentado. O método heurístico e a busca tabu do modelo estocástico foram alterados para sua resolução. Verificou-se que o tempo gasto para a execução da heurística é maior quando a capacidade de produção é folgada. A busca tabu não trouxe melhorias significativas em relação à solução heurística

Finalmente, destaca-se que este trabalho abriu um leque de opções para novas pesquisas, dentre elas é possível citar:

- estudos direcionados para obtenção de cortes de otimalidade de melhor qualidade, aumentando o porte dos problema a serem resolvidos otimamente;
- elaboração de vizinhanças menores para busca tabu visando obter a mesma qualidade de solução porém com um tempo computacional menor;
- incorporação de estratégias de diversificação e intensificação com o objetivo de explorar novas regiões e incentivar a busca em regiões consideradas promissoras. Esses componentes de longo prazo poderiam melhorar a qualidade da solução heurística;
- adaptar o método heurístico e busca tabu aqui propostos para:
 - . modelo com containers parcialmente cheios na produção e no estoque;
 - . modelo de vários itens finais e demanda externa para todos os estágios de produção;

- desenvolver uma linearização para o modelo determinístico que não envolva variáveis binárias de modo análogo ao proposto por Bitran e Chang.

APÊNDICE A

DEMONSTRAÇÕES

Este Apêndice apresenta as demonstrações das Proposições 1 e 2.

* Proposição 1

Dadas as equações:

$$\Phi_{n,t} = \Phi_{n,t-1} + \chi_{n,t} - E_{n,s(n)} \chi_{s(n),t} \quad n = 1, \dots, N; t = 1, \dots, T \quad (1)$$

$$\sigma_{n,t} = \sigma_{n,t-1} - \chi_{n,t} + E_{n,s(n)} \chi_{s(n),t} \quad n = 1, \dots, N; t = 1, \dots, T \quad (2)$$

$$\Phi_{0,t} = [\Phi_{0,t-1} + \chi_{0,t} - \Lambda_t - \Gamma_{t-1}]^+ \quad t = 1, \dots, T \quad (4),$$

$$\Gamma_t = -[\Phi_{0,t-1} + \chi_{0,t} - \Lambda_t - \Gamma_{t-1}]^- \quad t = 1, \dots, T \quad (5),$$

$$\sigma_{0,t} = \sigma_{0,t-1} - \chi_{0,t} + \min\{\Phi_{0,t-1} + \chi_{0,t}, \Lambda_t + \Gamma_{t-1}\} \quad t = 1, \dots, T \quad (6)$$

$$\chi_{n,t} = \min\left\{\begin{array}{l} \sigma_{n,t-1}, \\ \beta_{n,t}, \\ \left[(\Phi_{q,t-1} + \chi_{q,t} \theta_n) / E_{q,n} \right]_{q \in P(n)} \end{array}\right\}_{\substack{n=0, \dots, N \\ t=1, \dots, T}} \quad (7)$$

i. O número total de kanbans em cada estágio é invariante no tempo.

Demonstração:

Pela adição das equações (1) e (2) obtém-se para $n=1, \dots, N$ e $t=1, \dots, T$ que

$$\sigma_{n,t} + \Phi_{n,t} = \sigma_{n,t-1} + \Phi_{n,t-1}.$$

Então,

$$\sigma_{n,t} + \Phi_{n,t} = \sigma_{n,0} + \Phi_{n,0}.$$

Para o estágio zero existem duas situações:

1. $\Phi_{0,t} > 0$. Neste caso,

$$\min\{\Phi_{0,t-1} + \chi_{0,t}, \Lambda_t + \Gamma_{t-1}\} = \Lambda_t + \Gamma_{t-1}.$$

Pela soma das equações (4) e (6) obtém-se para $t=1, \dots, T$

$$\sigma_{0,t} + \Phi_{0,t} = \sigma_{0,t-1} + \Phi_{0,t-1}.$$

Então,

$$\sigma_{0,t} + \Phi_{0,t} = \sigma_{0,0} + \Phi_{0,0}.$$

2. $\Phi_{0,t} = 0$. Então,

$$\min\{\Phi_{0,t-1} + \chi_{0,t}, \Lambda_t + \Gamma_{t-1}\} = \Phi_{0,t-1} + \chi_{0,t}.$$

Somando novamente as equações (4) e (6) obtém-se para $t=1, \dots, T$

$$\sigma_{0,t} + \Phi_{0,t} = \sigma_{0,t-1} + \Phi_{0,t-1},$$

ii. $\Phi_{n,t}, \chi_{n,t}, \Gamma_t, \sigma_{n,t}$ são inteiros e não negativos

Demonstração:

Suponha que toda as variáveis estão determinadas até o período $t-1$ e que assumem valores inteiros e não negativos. Para os estágios sem predecessores de (7) obtém-se que $\chi_{n,t} = \min\{\sigma_{n,t-1}, \beta_{n,t}\}$, portanto, $\chi_{n,t} \geq 0$ e inteiro. Conseqüentemente $\chi_{n,t} \geq 0$ e inteiro para os estágios com predecessores. Desde que $\chi_{n,t} \geq 0$ e inteiro, segue-se de (4) e (5) que $\Phi_{0,t}, \Gamma_t$ são não-negativos e inteiros.

De (7) obtém-se que:

$$\chi_{s(n),t} \leq \left\lfloor \left(\Phi_{n,t-1} + \chi_{n,t} \theta_{s(n)} \right) / E_{n,s(n)} \right\rfloor \leq \left(\Phi_{n,t-1} + \chi_{n,t} \theta_{s(n)} \right) / E_{n,s(n)}$$

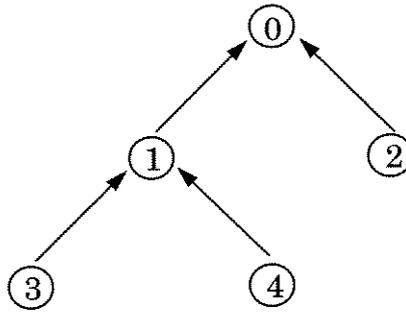
Assim, $E_{n,s(n)} \chi_{s(n),t} \leq \Phi_{n,t-1} + \chi_{n,t}$ e de (1) segue-se que $\Phi_{n,t} \geq 0$ e inteiro.

De (7) tem-se que $\chi_{n,t} \leq \sigma_{n,t-1}$. Então, (2) e (6) implicam que $\sigma_{n,t} \geq 0$ e inteiro.

iii. Uma vez que $\sigma_{n,0}$, $n = 0, \dots, N$ assume valores específicos, então as variáveis $\Phi_{n,t}$, $\sigma_{n,t}$, Γ_t , $\chi_{n,t}$ são unicamente determinadas

Demonstração:

Para ilustrar a demonstração considere a estrutura de montagem abaixo com $T=2$.



Para $t=1$, os valores de $\chi_{4,1}$, $\chi_{3,1}$ e $\chi_{2,1}$ são determinados, pois

$$\chi_{4,1} = \min\{\sigma_{4,0}, \beta_{4,1}\}, \chi_{3,1} = \min\{\sigma_{3,0}, \beta_{3,1}\}, \chi_{2,1} = \min\{\sigma_{2,0}, \beta_{2,1}\},$$

Dos valores de $\chi_{4,1}$ e $\chi_{3,1}$ pode-se determinar $\chi_{1,1}$ e todas as variáveis dos estágios 3 e 4. A partir dos valores de $\chi_{2,1}$ e $\chi_{1,1}$ determina-se $\chi_{0,1}$ e todas as variáveis associadas aos estágios 0, 1 e 2 (incluindo Γ_1). Para $t=2$, o procedimento é análogo.

*** Proposição 2**

Dadas as equações:

$$\varepsilon_{n,t} = \varepsilon_{n,t-1} + \chi_{n,t} - \alpha_n \left[(\chi_{n,t} + \varepsilon_{n,t-1}) / \alpha_n \right] \quad n = 0, \dots, N; t = 1, \dots, T \quad (8)$$

$$\Psi_{n,t} = \alpha_n \left[(e_{n,s(n)} \chi_{s(n),t} - \Psi_{n,t-1}) / \alpha_n \right] - (e_{n,s(n)} \chi_{s(n),t} - \Psi_{n,t-1}) \quad n = 1, \dots, N; t = 1, \dots, T \quad (9)$$

$$\Phi_{n,t} = \Phi_{n,t-1} + \left[(\chi_{n,t} + \varepsilon_{n,t}) / \alpha_n \right] - \left[(e_{n,s(n)} \chi_{s(n),t} - \Psi_{n,t-1}) / \alpha_n \right] \quad n = 1, \dots, N; t = 1, \dots, T \quad (10)$$

$$\sigma_{n,t} = \sigma_{n,t-1} + \left[(e_{n,s(n)} \chi_{s(n),t} - \Psi_{n,t-1}) / \alpha_n \right] - \left[(\chi_{n,t} + \varepsilon_{n,t-1}) / \alpha_n \right] \quad n = 1, \dots, N; t = 1, \dots, T \quad (11)$$

$$\Phi_{0,t} = \left[\Phi_{0,t-1} + \left[(\chi_{0,t} + \varepsilon_{0,t}) / \alpha_0 \right] - \Lambda_t - \Gamma_{t-1} \right]^+ \quad t = 1, \dots, T \quad (12)$$

$$\Gamma_t = - \left[\Phi_{0,t-1} + \left[(\chi_{0,t} + \varepsilon_{0,t}) / \alpha_0 \right] - \Lambda_t - \Gamma_{t-1} \right]^- \quad t = 1, \dots, T \quad (13)$$

$$\begin{aligned} \sigma_{0,t} = & \sigma_{0,t-1} - \left[(\chi_{0,t} + \varepsilon_{0,t-1}) / \alpha_0 \right] \\ & + \min \left\{ \Phi_{0,t-1} + \left[(\chi_{0,t} + \varepsilon_{0,t-1}) / \alpha_0 \right], \Lambda_t + \Gamma_{t-1} \right\} \quad t = 1, \dots, T \end{aligned} \quad (14)$$

$$\begin{aligned} \chi_{n,t} = \min \{ & \alpha_n \sigma_{n,t-1} - \varepsilon_{n,t-1}, \\ & \beta_{n,t}, \\ & \left[(\alpha_q \Phi_{q,t-1} + (\chi_{q,t} + \varepsilon_{q,t-1}) \theta_n + \Psi_{q,t-1}) / e_{q,n} \right] \quad q \in P(n) \} \quad n = 0, \dots, N; \\ & t = 1, \dots, T \end{aligned} \quad (15)$$

$$\text{iv. } \varepsilon_{n,t} \leq \alpha_n - 1 \quad \begin{array}{l} n = 0, \dots, N \\ t = 1, \dots, T \end{array}$$

Demonstração:

Suponha $(\chi_{n,t} + \varepsilon_{n,t-1})/\alpha_n$ inteiro, então

$$\chi_{n,t} + \varepsilon_{n,t-1} = \alpha_n \lfloor (\chi_{n,t} + \varepsilon_{n,t-1})/\alpha_n \rfloor,$$

e $\varepsilon_{n,t} = 0$.

Se $(\chi_{n,t} + \varepsilon_{n,t-1})/\alpha_n$ não é inteiro,

$$\chi_{n,t} + \varepsilon_{n,t-1} = \alpha_n \left(\lfloor (\chi_{n,t} + \varepsilon_{n,t-1})/\alpha_n \rfloor + R \right),$$

onde R é inteiro e $0 < R \leq \alpha_n - 1$

Definindo $\varepsilon_{n,t} \hat{=} R$ obtém-se

$$\varepsilon_{n,t} = \varepsilon_{n,t-1} + \chi_{n,t} - \alpha_n \lfloor (\chi_{n,t} + \varepsilon_{n,t-1})/\alpha_n \rfloor, \quad 0 \leq \varepsilon_{n,t} \leq \alpha_n - 1.$$

$$\text{v. } \Psi_{n,t} \leq \alpha_n - 1 \quad \begin{array}{l} n = 0, \dots, N \\ t = 1, \dots, T \end{array}$$

Demonstração:

Suponha $(e_{n,s(n)}\chi_{n,t} - \Psi_{n,t-1})/\alpha_n$ inteiro, então

$$\alpha_n \lfloor (e_{n,s(n)}\chi_{n,t} - \Psi_{n,t-1})/\alpha_n \rfloor = e_{n,s(n)}\chi_{n,t} - \Psi_{n,t-1},$$

e $\Psi_{n,t} = 0$.

Se $(e_{n,s(n)}\chi_{n,t} - \Psi_{n,t-1})/\alpha_n$ não é inteiro,

$$e_{n,s(n)}\chi_{n,t} - \Psi_{n,t-1} = \alpha_n \left(\lfloor (e_{n,s(n)}\chi_{n,t} - \Psi_{n,t-1})/\alpha_n \rfloor + R \right),$$

onde R é inteiro e $0 < R \leq \alpha_n - 1$

Definindo $\Psi_{n,t} \hat{=} \alpha_n - R$ obtém-se

$$\Psi_{n,t} = \alpha_n \left[\left(e_{n,s(n)} \chi_{s(n),t} - \Psi_{n,t-1} \right) / \alpha_n \right] - \left(e_{n,s(n)} \chi_{s(n),t} - \Psi_{n,t-1} \right), \quad 0 \leq \Psi_{n,t} \leq \alpha_n - 1.$$

i. O número total de kanbans em cada estágio é invariante no tempo

Demonstração:

Pela adição das equações (10) e (11) obtém-se para $n=1,\dots,N$ e $t=1,\dots,T$ que

$$\sigma_{n,t} + \Phi_{n,t} = \sigma_{n,t-1} + \Phi_{n,t-1}.$$

Então,

$$\sigma_{n,t} + \Phi_{n,t} = \sigma_{n,0} + \Phi_{n,0}.$$

Para o estágio zero existem duas situações:

1. $\Phi_{0,t} > 0$. Neste caso,

$$\min \left\{ \Phi_{0,t-1} + \left[(\chi_{0,t} + \varepsilon_{0,t-1}) / \alpha_0 \right], \Lambda_t + \Gamma_{t-1} \right\} = \Lambda_t + \Gamma_{t-1}.$$

Pela soma das equações (12) e (14) obtém-se para $t=1,\dots,T$

$$\sigma_{0,t} + \Phi_{0,t} = \sigma_{0,t-1} + \Phi_{0,t-1}.$$

Então,

$$\sigma_{0,t} + \Phi_{0,t} = \sigma_{0,0} + \Phi_{0,0}.$$

2. $\Phi_{0,t} = 0$. Então,

$$\min \left\{ \Phi_{0,t-1} + \left[(\chi_{0,t} + \varepsilon_{0,t-1}) / \alpha_0 \right], \Lambda_t + \Gamma_{t-1} \right\} = \Phi_{0,t-1} + \left[(\chi_{0,t} + \varepsilon_{0,t-1}) / \alpha_0 \right].$$

Somando novamente as equações (12) e (14) obtém-se para $t=1,\dots,T$

$$\sigma_{0,t} + \Phi_{0,t} = \sigma_{0,t-1} + \Phi_{0,t-1},$$

ii. $\Phi_{n,t}, \sigma_{n,t}, \Gamma_t, \varepsilon_{n,t}, \Psi_{n,t}$ e $\chi_{n,t}$ são inteiros e não negativos

Demonstração:

Suponha que toda as variáveis estão determinadas até o período t-1 e que assumem valores inteiros e não negativos. Para os estágios sem predecessores de (15) obtém-se que $\chi_{n,t} = \min\{(\alpha_n \sigma_{n,t-1} - \varepsilon_{n,t-1}), \beta_{n,t}\}$. Como $\sigma_{n,t}$ é definido como o número de kanbans no quadro de kanbans e **na produção**, quando $\varepsilon_{n,t} > 0$, então $\sigma_{n,t} > 0$. Como $0 \leq \varepsilon_{n,t} \leq \alpha_n - 1$, e

$$\alpha_n \sigma_{n,t-1} - \varepsilon_{n,t-1} \geq 0.$$

Portanto, $\chi_{n,t} \geq 0$ e inteiro. Conseqüentemente, $\chi_{n,t} \geq 0$ e inteiro para os estágios com predecessores.

Como

$$\chi_{n,t} + \varepsilon_{n,t-1} \geq \alpha_n \left\lceil (\chi_{n,t} + \varepsilon_{n,t-1}) / \alpha_n \right\rceil$$

obtém-se de (8) que $\varepsilon_{n,t} \geq 0$ e inteiro.

De (15) obtém-se que:

$$\begin{aligned} \chi_{s(n),t} &\leq \left\lceil \left(\alpha_n \Phi_{n,t-1} + (\chi_{n,t} + \varepsilon_{n,t-1}) \theta_n + \Psi_{n,t-1} \right) / e_{n,s(n)} \right\rceil \\ &\leq \left(\alpha_n \Phi_{n,t-1} + (\chi_{n,t} + \varepsilon_{n,t-1}) \theta_n + \Psi_{n,t-1} \right) / e_{n,s(n)} \\ &\leq \left(\alpha_n \Phi_{n,t-1} + (\chi_{n,t} + \varepsilon_{n,t-1}) + \Psi_{n,t-1} \right) / e_{n,s(n)} \end{aligned}$$

Assim,

$$\begin{aligned} \left(e_{n,s(n)} \chi_{s(n),t} - \Psi_{n,t-1} \right) / \alpha_n &\leq \Phi_{n,t-1} + (\chi_{n,t} + \varepsilon_{n,t-1}) / \alpha_n, \\ \left\lceil \left(e_{n,s(n)} \chi_{s(n),t} - \Psi_{n,t-1} \right) / \alpha_n \right\rceil &\leq \Phi_{n,t-1} + \left\lceil (\chi_{n,t} + \varepsilon_{n,t-1}) / \alpha_n \right\rceil, \end{aligned}$$

e de (10) segue-se que $\Phi_{n,t} \geq 0$ e inteiro.

Suponha que $(e_{n,s(n)}\chi_{s(n),t} - \Psi_{n,t-1}) \geq 0$. Então $\lceil (e_{n,s(n)}\chi_{s(n),t} - \Psi_{n,t-1}) \rceil \geq (e_{n,s(n)}\chi_{s(n),t} - \Psi_{n,t-1})$ e $\Psi_{n,t} \geq 0$ e inteiro. Caso contrário, isto é, $(e_{n,s(n)}\chi_{s(n),t} - \Psi_{n,t-1}) < 0$ e como $0 \leq \Psi_{n,t} \leq \alpha_n - 1$, então

$$-1 < ((e_{n,s(n)}\chi_{s(n),t} - \Psi_{n,t-1})/\alpha_n) < 0.$$

Portanto $\lceil (e_{n,s(n)}\chi_{s(n),t} - \Psi_{n,t-1})/\alpha_n \rceil = 0$ e $\Psi_{n,t} \geq 0$ e inteiro.

De (15) tem-se que

$$\chi_{n,t} \leq (\alpha_n \sigma_{n,t-1} - \varepsilon_{n,t-1})$$

ou,

$$\sigma_{n,t-1} \geq (\chi_{n,t} + \varepsilon_{n,t-1})/\alpha_n,$$

como $\sigma_{n,t-1}$ inteiro e não negativo,

$$\sigma_{n,t-1} \geq \lceil (\chi_{n,t} + \varepsilon_{n,t-1})/\alpha_n \rceil.$$

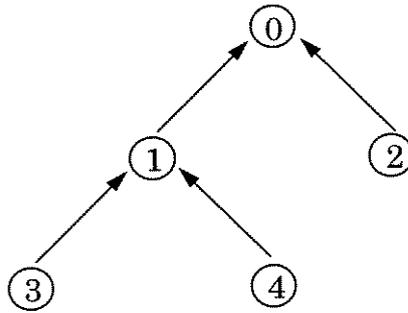
Então de (14) segue-se que $\sigma_{0,t} \geq 0$ e inteiro. Foi mostrado acima que $\lceil (e_{n,s(n)}\chi_{s(n),t} - \Psi_{n,t-1})/\alpha_n \rceil \geq 0$ e portanto de (11) tem-se que $\sigma_{n,t} \geq 0$ e inteiro.

As variáveis $\Phi_{0,t}$ e Γ_t são não-negativas e inteiras por definição (vide (12) e (13)).

iii. Uma vez que $\sigma_{n,0}, n = 0, \dots, N$ assume valores específicos, então as variáveis $\Phi_{n,t}, \sigma_{n,t}, \Gamma_t, \chi_{n,t}, \varepsilon_{n,t}, \Psi_{n,t}$ são unicamente determinadas

Demonstração:

Para ilustrar a demonstração considere a estrutura de montagem abaixo com $T=2$.



Para $t=1$, os valores de $\chi_{4,1}, \chi_{3,1}$ e $\chi_{2,1}$ são determinados, pois

$$\chi_{4,1} = \min\left\{\left(\sigma_{4,0} - \varepsilon_{4,0}\right), \beta_{4,1}\right\}, \chi_{3,1} = \min\left\{\left(\sigma_{3,0} - \varepsilon_{3,0}\right), \beta_{3,1}\right\}, \chi_{2,1} = \min\left\{\left(\sigma_{2,0} - \varepsilon_{2,0}\right), \beta_{2,1}\right\}$$

Dos valores de $\chi_{4,1}$ e $\chi_{3,1}$ pode-se determinar $\chi_{1,1}$ e todas as variáveis dos estágios 3 e 4. A partir dos valores de $\chi_{2,1}$ e $\chi_{1,1}$ determina-se $\chi_{0,1}$ e todas as variáveis associadas aos estágios 0, 1 e 2 (incluindo Γ_1). Para $t=2$, o procedimento é análogo.

APÊNDICE B

COMPLEXIDADE DA HEURÍSTICA

Neste apêndice é apresentada a análise de complexidade do pior caso para a heurística proposta, considerando o algoritmo principal e os procedimentos auxiliares.

Dado o algoritmo principal:

```
Passo 1: Para  $n \leftarrow 0, \dots, N$  faça:  
           $\{\sigma_{n,0} \leftarrow K_n - \Phi_{n,0}$   
  
Passo 2:  $Z \leftarrow 0$   
          Para  $i \leftarrow 1, \dots, C$  faça:  
          [    Calcula_Produção_Estoque( $\sigma, \Phi, \chi, \Gamma, \beta$ )  
               $aux \leftarrow 0$   
              Calcula_Valor_Solução( $h, \Phi, P, \Gamma, aux$ )  
               $Z \leftarrow Z + aux$   
           $Z \leftarrow Z / C$   
           $Z_{inicial} \leftarrow Z$   
  
Passo 3: Para  $n \leftarrow 0, \dots, N$  faça  
           $\{\sigma_{aux_n} \leftarrow \sigma_{n,0}$   
  
Passo 4: Faça  
          [    teste  $\leftarrow 0$   
              Para  $n \leftarrow 0, \dots, N$  faça  
                   $\{\text{calcula\_}\sigma(\text{teste}, Z)$   
              enquanto (teste  $\neq 0$ )
```

```

Passo 5:   Z_1 ← Z
              Z ← Z_inicial
              Para n ← 0,...,N faça
                  { σ_aux_n ← σ_n,0

Passo 6:   Faça
                [ teste ← 0
                  Para n ← N,...,0 faça
                      { calcula_σ(teste, Z)
                enquanto (teste ≠ 0)

Passo 7:   Se (Z_1 < Z)
                então [ Z ← Z_1
                       início ← 0
                       fim ← N
                       Para n ← 0,...,N faça
                           { σ_n,0 ← σ_aux_n
                senão [ início ← 0
                       fim ← N

Passo 8:   Busca_Local (início, fim, Z)
    
```

PASSO 1:

A finalidade deste passo é somente a inicialização das variáveis de σ_0 . Assim, a sua complexidade é $O(N)$.

PASSO 2:

A complexidade, aqui, está relacionada com o comando de repetições que executa **C** vezes dois procedimentos: **Calcula_Produção_Estoque** e **Calcula_Valor_Solução**.

Procedimento Calcula_Produção ($\sigma, \Phi, \chi, \Gamma, \beta$)

Passo A: Para $n \leftarrow 0, \dots, N$ faça
 $\{\chi_{n,t} \leftarrow 0, \Phi_{n,t} \leftarrow 0; \quad t = 1, \dots, T$

Passo B: Para $t \leftarrow 1, \dots, T$ faça

$$\chi_{n,t} \leftarrow \min \left\{ \left(\sigma_{n,0} + \Phi_{n,0} \right) - \Phi_{n,t-1}, \right.$$

$$\left. \left[\left(\theta_n \chi_{q,t} + \Phi_{q,t-1} \right) / E_{q,n} \right] \quad q \in P(n), \right.$$

$$\left. \beta_{n,t} \right\} \quad n \leftarrow N, \dots, 0$$

Passo C: $\Phi_{0,t} \leftarrow \Phi_{0,t-1} + \chi_{0,t} - \Lambda_t$
 Se $(\Phi_{0,t} < 0)$

$$\left[\begin{array}{l} \text{então} \quad \left\{ \begin{array}{l} \Phi_{0,t} \leftarrow 0 \\ \Gamma_t \leftarrow -1 * \Phi_{0,t} \end{array} \right. \\ \text{senão} \quad \Gamma_t \leftarrow 0 \end{array} \right.$$

$n \leftarrow 1$

Passo D: $\Phi_{n,t} \leftarrow \Phi_{n,t-1} + \chi_{n,t} - E_{n,s(n)} \chi_{s(n),t}$

$$\left[\begin{array}{l} \text{se } n \neq N \\ \text{então} \quad \left\{ \begin{array}{l} n \leftarrow n + 1 \\ \text{volte para o passo D} \end{array} \right. \end{array} \right.$$

O procedimento Calcula_Produção_Estoque determina os valores das variáveis de produção e estoque, $\chi_{n,t}$ e $\Phi_{n,t}$ respectivamente, envolvendo dois comandos de repetição e possui complexidade $O(N*T)$.

Procedimento Calcula_Valor_Solução (h, Φ, P, Γ, aux)

Passo A: Para $t \leftarrow 1, \dots, T$ faça
 $aux \leftarrow aux + (\Gamma_t * p)$
 Para $n \leftarrow 0, \dots, N$ faça
 $aux \leftarrow aux + (\Phi_{n,t} * h_t)$

Como o próprio nome sugere, este procedimento calcula o valor da função objetivo (armazenado em Z) e como também envolve dois comandos básicos de repetição sua complexidade é $O(N*T)$

Portanto o passo 2 da heurística possui complexidade $O(C*N*T)$

PASSO 3:

A complexidade deste passo é de $O(N)$ pois ele realiza somente a atribuição de valores para as variáveis σ_{aux_n} .

PASSO 4:

O objetivo, nesse momento da heurística, é alterar os valores de σ_0 de forma a melhorar o valor da função objetivo. Para tanto é feita uma busca binária nos intervalos em que as variáveis de σ_0 são definidas. Essa busca é repetida até que modificações nos valores de σ_0 não acarretem melhorias no valor da função objetivo

Procedimento Calcula_σ (teste, Z)

Passo A: $melhor_{\sigma} \leftarrow \sigma_{aux_n}$
 $\sigma_{valor} \leftarrow \sigma_{aux_n}$
 $\sigma_{up} \leftarrow \sigma_{aux_n}$
 $\sigma_{down} \leftarrow 0$

Passo B: Enquanto $(\sigma_{up} \geq \sigma_{down})$

$\sigma_{med} \leftarrow (\sigma_{up} + \sigma_{down})/2$
 $\sigma_{aux_n} \leftarrow \sigma_{med}$

Passo C: $Z_{aux} \leftarrow 0$

Para $i \leftarrow 1, \dots, C$ faça

$Calcula_Produção_Estoque(\sigma, \Phi, \chi, \Gamma, \beta)$
 $aux \leftarrow 0$
 $Calcula_Valor_Solução(h, \Phi, P, \Gamma, aux)$
 $Z_{aux} \leftarrow Z_{aux} + aux$

$Z_{aux} \leftarrow Z_{aux} / C$

Passo D: Se $(Z_{aux} \leq Z)$

então $\sigma_{up} \leftarrow \sigma_{med} - 1$
 $Z \leftarrow Z_{aux}$

senão $\sigma_{down} \leftarrow \sigma_{med} + 1$

Passo E: $\sigma_{aux} \leftarrow melhor_{\sigma}$

Passo F: Se $(\sigma_{valor} \neq \sigma_{aux_n})$
então { teste $\leftarrow 1$

No passo 4, existem 2 comandos de repetição. O comando externo realiza a função de controle da busca binária, determinando se ela deve ser executada (teste $\neq 0$) ou se a condição de parada (teste = 0) é satisfeita. O comando de repetição interno consiste de (N+1) chamadas ao procedimento Calcula_σ, que realiza a busca binária propriamente dita.

Para calcular a complexidade do passo 4 é preciso determinar, inicialmente, a complexidade do procedimento Calcula_σ. A cada vez que esse procedimento é aplicado sobre uma variável de σ_0 , a busca binária é executada no intervalo $[0, K_n - \Phi_{n,0} - \mu_n]$ que compreende todos os valores que σ_0 pode assumir. O valor de μ_n depende de quantas vezes a busca binária já foi aplicada anteriormente sobre σ_0 ao longo das repetições do passo 4.

A busca binária pode ser realizada sobre um intervalo genérico [a,b], $b > a$, em $O(\log(b-a))$ passos, sendo que a complexidade de cada passo depende do esforço computacional decorrente da chamada de rotinas de avaliação de solução necessárias para guiar a busca. No caso do procedimento Calcula_σ, a cada passo da busca binária, há C chamadas aos procedimentos Calcula_Produção_Estoque e Calcula_Valor_Solução, ambos de complexidade $O(N*T)$. Portanto, a complexidade da busca binária no intervalo $[0, K_n - \Phi_{n,0} - \mu_n]$, para a variável $\sigma_{n,0}$ é dada por $O(\log(K_n - \Phi_{n,0} - \mu_n) * C * N * T)$

Como $K_{\max} = \max_n K_n$ e $(\Phi_{n,0} - \mu_n) \geq 0, \forall n$ a soma para as (N+1) chamadas ao procedimento Calcula_σ, no comando de repetições interna é:

$$\begin{aligned} \sum_{i=0}^N \log\left(\left(\overline{K}_n - \Phi_{n,0} - \mu_n\right) * C * (N + 1) * T\right) &\leq C * (N + 1) * T \sum_{i=0}^N \log(K_n) \\ &\leq C * (N + 1) * T \sum_{i=0}^N K_n \leq C * (N + 1) * T \sum_{i=0}^N K_{\max} = C * (N + 1) * T * (N + 1) * K_{\max} \end{aligned}$$

Portanto, a complexidade do comando interno de repetições é $O(N^2 * K_{\max} * C * T)$

Passando agora para as repetições do nível externo é necessário observar que seu número depende de um teste baseado nos resultados do comando interno de repetições. Enquanto houver mudanças nos valores dos elementos de σ_0 , haverá repetições e, portanto, o número máximo de repetições está associado ao número máximo de mudanças possíveis no vetor σ_0 . Para realizar

este cálculo é preciso supor que os valores iniciais e finais de σ_0 são conhecidos e que representam o pior caso possível. Claramente esse valores são:

- Inicial: $\sigma_{n,0} = (K_n - \Phi_{n,0})$ para $\forall n \in N$ e
- Final: $\sigma_{n,0} = 0$ para $\forall n \in N$

Além disso, suponha que as alterações dos valores de cada elemento sejam feitas da forma mais lenta possível. Isso significa que, a cada repetição do comando externo, somente um dos elementos do vetor possa ter seu valor diminuído e que essa diminuição seja unitária. Dessa forma, cada elemento do vetor σ_0 será alterado $(K_n - \Phi_{n,0})$ vezes, o que significa um total de

$\sum_{n=0}^N (K_n - \Phi_{n,0})$ repetições. Então,

$$\sum_{n=0}^N (K_n - \Phi_{n,0}) \leq \sum_{n=0}^N K_n \leq \sum_{n=0}^N K_{\max} \leq (N + 1) * K_{\max}$$

Portanto, a complexidade do passo 4 é $O(N^3 * K_{\max}^2 * C * T)$

Os **passos 5, 6** são idênticos, respectivamente, aos passos 3, 4 tendo como diferença a ordem dos estágios nos comandos de repetição. Sendo as complexidades desses passos:

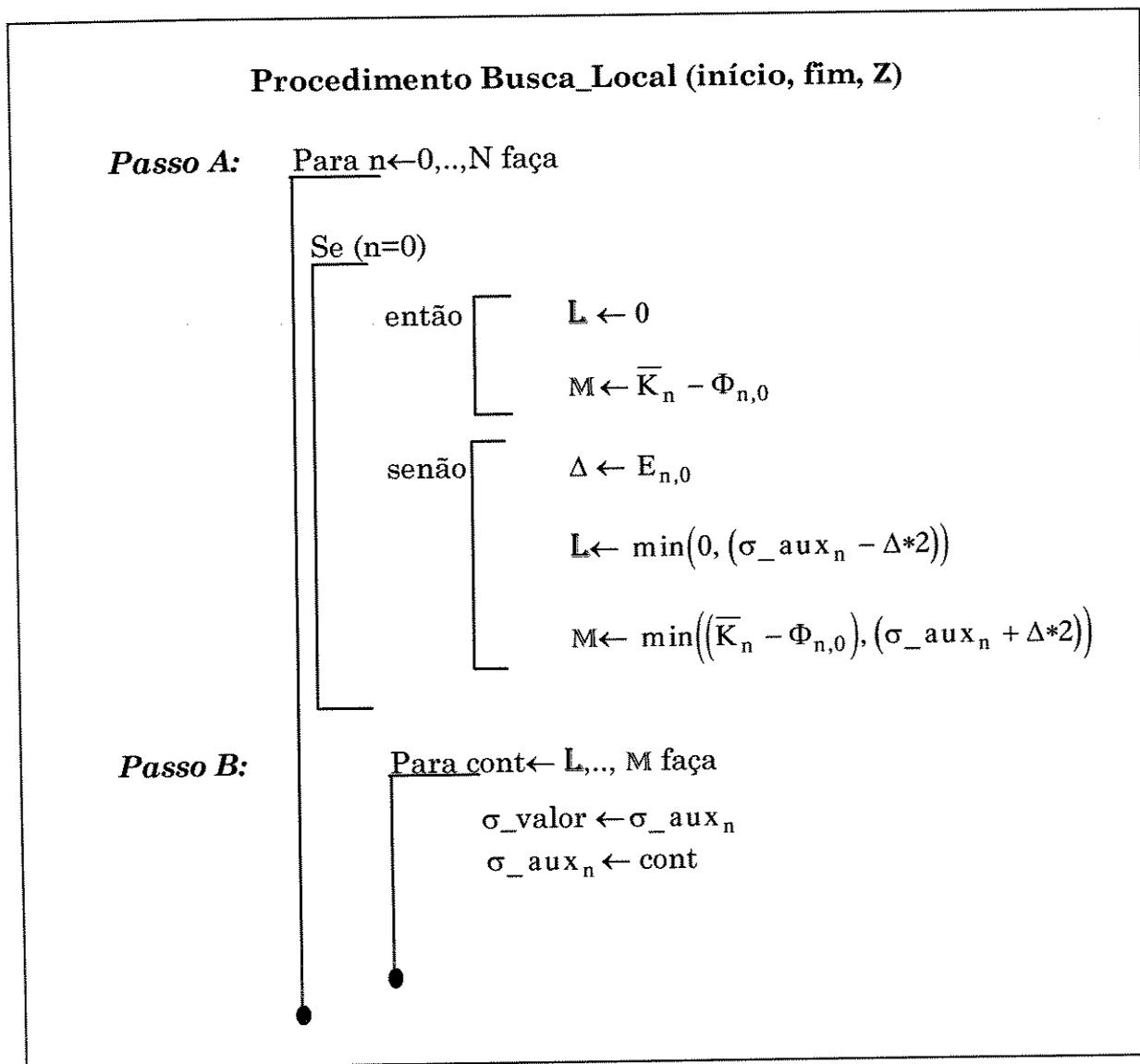
- **Passo 5:** $O(N)$
- **Passo 6:** $O(N^3 * K_{\max}^2 * C * T)$

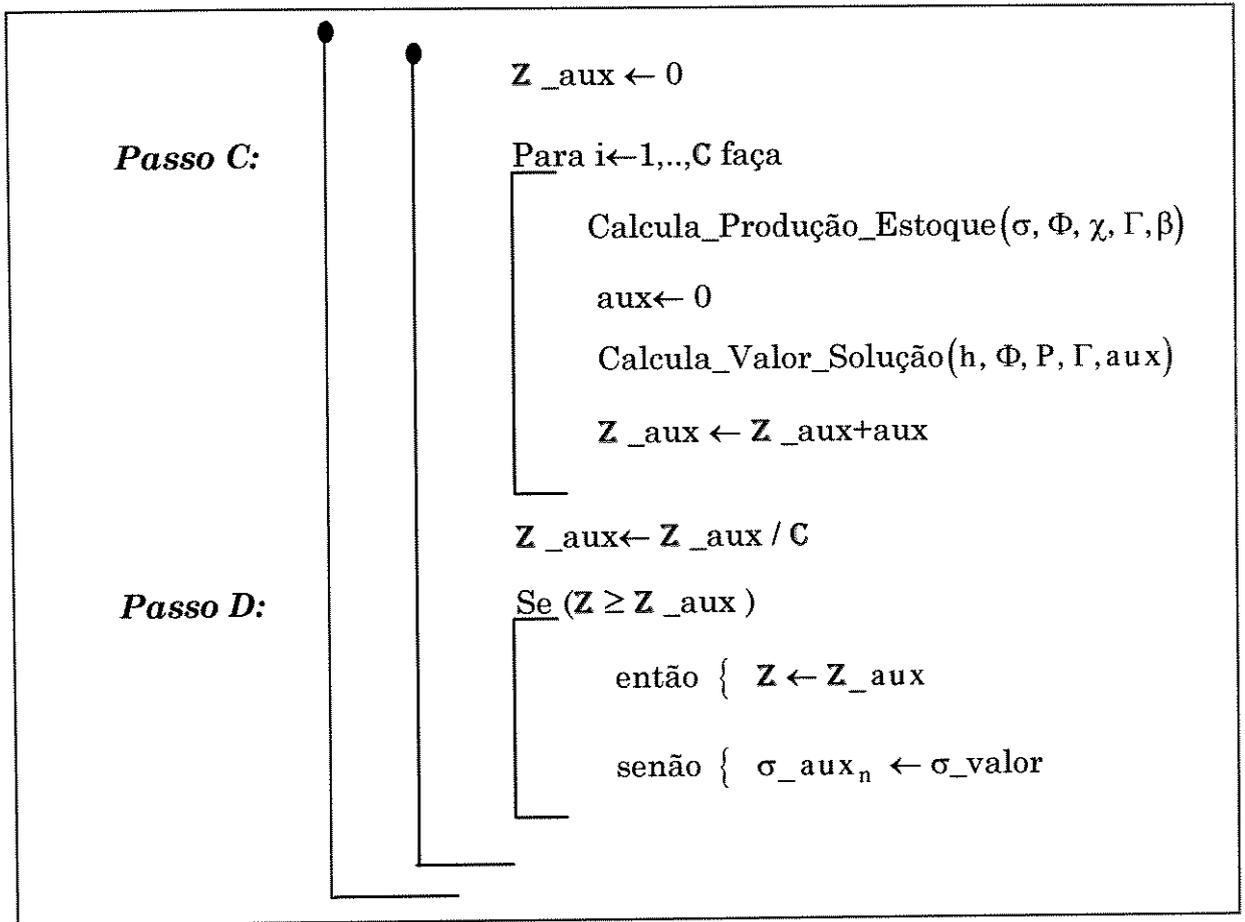
PASSO 7:

Este passo constitui na determinação da melhor solução entre as obtidas pelos passos 4 e 6 e a sua complexidade é $O(N)$ devido a atualização do vetor $\sigma_{n,0}$

PASSO 8:

O passo 8 constitui-se de um procedimento de busca local que realiza uma busca completa, em um determinado intervalo dos elementos de σ_0 , visando melhorar a solução obtida no final do passo 7.





O pior caso para esse procedimento é quando se permite que a busca seja realizada até $(K_n - \Phi_{n,0}) \leq K_{\max}$ para todos os elementos de σ_0 .

Dessa forma, o número de operações realizadas no passo 8 pode ser calculado como

$$\begin{aligned}
 & \sum_{n=0}^N n * K_{\max} * C * (N + 1) * T = \\
 & = K_{\max} * C * (N + 1) * T \frac{(N + 1) * (N + 1)}{2} = K_{\max} * C * T \frac{(N + 1)^2 * (N + 1)}{2} \\
 & \leq K_{\max} * C * (N + 1)^3 * T
 \end{aligned}$$

Portanto a complexidade desse passo é dado por:

$$O(K_{\max} * C * N^3 * T)$$

Como todos os passos são realizados de forma seqüencial, aquele que apresenta maior complexidade determina a complexidade da heurística. Toma-se então, como complexidade dominante a do passo 4 (ou 6, equivalentemente). Dessa forma a complexidade da heurística proposta pode ser escrita como:

$$O(K_{\max}^2 * N^3 * C * T)$$

BIBLIOGRAFIA

- Askin, G.R., Mitwasi, M.G. e Goldbeg, J.B. (1993), "Determining the Number of Kanbans in Multi-item Just-In-Time Systems", *IIE Transactions*, 25(1), 89-97
- Bard, J.F. e Golany, B. (1991) , "Determining the Number of Kanbans in a Multiproduct, Multistage Production System", *International Journal of Production Research*, 29(5), 881-895.
- Beale, M. L. (1979) , "Branch and Bound Methods for Integer Programming", *Annals of Discrete Mathematics*, 5, 201-219
- Benders, J.F. (1962), "Partition Procedures for Solving Mixed-Variables Programming Problems", *Numeric Mathematics*, 4, 238-252
- Bitran, G.R. e Chang, L. (1987), "A Mathematical Programming Approach to a Deterministic Kanban System", *Management Science*, 33(4), 427-441.
- Chu, Chao-Hsien e Shih, Wei-Ling (1992), "Simulation Studies in JIT Production", *International Journal of Production Research*, 30(11), 2573-2586.
- Co, H.C. e Jacobson, S.H.(1994), "The Kanban Assignment Problem in Serial Just-in-Time Production Systems", *IIE Transactions*, 26(2), 76-85.
- Corrêa, Henrique L. e Gianesi, Irineu G.N. (1993), *Just in Time MRP e OPT Um Enfoque Estratégico*, Editora Atlas, São Paulo
- Dantzig, G.B. (1955), "Linear Programming under uncertainty", *Management Science*, 1(3), 197-206.
- Deleersnyder, J.L., Hodgson, T.J., Muller, H. e O'Grady, P.J. (1989), "Kanban Controlled Pull Systems: An Analytic Approach", *Management Science*, 35(9), 1079-1091.
- Deleersnyder, J.L., Hodgson, T.J., King, R. E., O'Grady, P.J. e Savva, A. (1992), "Integrating Kanban Type Pull Systems and MRP Push Systems: Insights from a Markovian Model", *IIE Transactions*, 24(3), 43-56
- Ermoliev, Y. e Wets, R. (1988), *Stochastic Programming, an Introduction , in Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin.
- Flapper, S.D.P., Miltenburg, G.J. e Wijngaard, J.(1991), "Embedding JIT into MRP", *International Journal of Production Research*, 29(5), 881-895.

- Geofrion, A. M. e Mastern, R. E.(1972), "Integer Programming Algorithms: A Framework and State of the Art Survey", *Management Science*, 18(9).
- Glover, F.W., (1989), "Tabu Search, Part I", *Orsa Journal on Computing*, 1(3), 190-206
- Glover, F.W., (1995), *TABU SEARCH: FUNDAMENTALS AND USES*, Graduate School of Business, University of Colorado, CO, USA
- Golhar, Damodar Y. e Stamm, Carol Lee (1991), "The Just-in-Time Philosophy: A Literature Review", *International Journal of Production Research*, 29(4), 657-676.
- Goyal, Suresh K. e Deshmukh, S. G. A. (1992), "Critique of the Literature on Just-in-Time Manufacturing", *International Journal of Operation & Production Management*, 12(1), 18-28.
- Gravel, M. e Price, W.L. (1991), "Visual Interactive Simulation shows how to use the Kanban Method in Small Business", *Interfaces*, 21(5), 22-33.
- Gravel, M., Martel, J.M., Nadeau, R., Price, W. e Tremblay, R. (1992), "A Multicriterion View of Optimal Resource Allocation in Job-Shop Production", *European Journal of Operation Research*, 1771-1777.
- Hansotia, Behram J. (1990), "Stochastic Linear Programming with Recourse: A Tutorial", *Decision Sciences* (11), 151-180.
- Kimura, O. e Terada, H. (1981), "Design and Analysis of a Pull System, a Method of Multi-Stage Production Control", *International Journal of Production Research*, 19(3), 241-253.
- Laporte, G., Louveaux, F.V. e Mercure, H el ene (1989), "Models and exact solutions for a class of stochastic location-routing problems", *European Journal of Operational Research*, 39, 71-78.
- Laporte, G., Louveaux, F.V. e Mercure, H el ene, (1992), "The Vehicle Routing Problem with Stochastic Travel Times", *Transportation Science*, 26(3), 161-169.
- Laporte, G. e Louveaux, F.V. (1993), "The integer L-shaped method for stochastic integer program with complete recourse", *Operations Research Letters*, 13, 133-142.
- Laporte, G., Louveaux, F.V. e Van Hamme, Luc (1994), "Exact Solution to a Location Problem with Stochastic Demands", *Transportation Science*, 28(2), 95-103.
- Li, A. e Co, H. C (1991), "A Dynamic Programming Model for the Kanban Assignment Problem in a Multistage Multiperiod Production System", *International Journal of Production Research*, 29(1), 1-16.

- Loveuax, F. V. e Vlerk, M.H.(1993), "Stochastic Programming with Simple Integer Recourse", *Mathematical Programming*, 61, 301-325.
- Miltenburg, J. e Wijngaard (1991), "Designing and Phasing in Just-in-Time Production Systems", *International Journal of Production Research*, 29(1), 115-131.
- Mitwasi, M.G. e Askin, R. G.(1994), "Production Planning for a Multi-Item, Single-Stage Kanban System", *International Journal of Production Research*, 30(1), 1-9
- Monden, Y. (1983), *Toyota Production Systems*, Industrial Engineering and Management Press, Institute of Industrial Engineers, Atlanta.
- Nahmias, Steven (1989), *Production and Operation Analysis*, Irwing, Boston, USA.
- Nemhauser, George L. e Laurence A. Wolsey (1988), *Integer and Combinatorial Optimization*, Jonh Wiley & Sons, Inc. United States of America.
- Ohno, K., Nakashima, K. e Kojima, M.(1995), "Optimal Numbers of Two Kinds Of Kanbans In A JIT Production System", *International Journal of Production Research*, 33(5), 1387-1401.
- Philipoom, P.R., Rees, L.P., Taylor, B.W. e Huang, P.Y. (1990), "A Mathematical Programming Approach for Determining Workcentre Lotsizes in a Just-in-Time Systems with Signal Kanban", *International Journal of Production Research*, 28(1), 1-15.
- Price, W., Gravel, M., Naskanda, A.L. e Cantin, F. (1995), "Modelling the Performance of a Kanban Assembly Shop", *International Journal of Production Research*, 33(5), 1171-1177.
- Rees, L.P. e Taylor, B. W. (1989), "A Comparative Analisis of an MRP Lot-For-Lot System and a Kanban System for a Multistage Production Operation", *International Journal of Production Research*, 27(8), 1427-1443.
- Sarker, B.R. e Harris R. D.(1988), "The Effect of Imbalance in a Just-in-time Production System: A Simulation Study", *International Journal of Production Research*, 26(1), 1-18.
- Sarker, B. R. e Fitzsimmons, J. A. (1989), "The Performance of Push and Pull Systems: A Simulation and Comparative Study", *International Journal of Production Research*, 27(10), 1715-1731.
- Schmenner, Roge W. (1990), *Production/Operations Management - Concepts and Situations*, Macmillan Publishing Company

- Schonberger, R.J. (1982), *Japanese Manufacturing Techniques*, The Free Press, Macmillan, New York.
- Smith, Spencer A. (1989), *Computer Based Production and Inventory Control*, Prentice Hall, New Jersey
- So, K.C. e Pinault, S.C. (1988), "Allocating Buffer Storages in a Pull System", *International Journal of Production Research*, 26(12), 1959-1980.
- Spearman, M. L. e Zazanis, M. A. (1992), "Push and Pull Production Systems: Issues and Comparisons", *Operations Research*, 40(3), 521-532
- Sugimori, Y., Kasunoki, K. e Uchikawa, S. (1977), "Toyota Production System and Kanban System: Materialization of Just-in-Time with Respect-for-Human System", *International Journal of Production Research*, 15(6), 553-564.
- Van Slyke. R.M. e Wets, R.(1969), "L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming", *Siam Journal Application Mathematics*, 4(17), 638-663
- Volmann, Thomas E., Berry, William L.B. e Whybark, Clay D. (1988), *Manufacturing Planning and Control Systems*, Second Edition, Dow Jones - Irwin
- Yavuz, I.H. e Satir, A. (1995), "A Kanban-Based Simulation Study of a Mixed Model Just-in-Time Manufacturing Line", *International Journal of Production Research*, 33(4), 1027-1048.
- Wang, H. e Wang, H.(1990), "Determining the Number of Kanbans: A Step Toward Non-Stock-Production", *International Journal of Production Research*, 28(11), 2101-2115.
- Wets, R.J.B. (1966a), "Programming under uncertainty: the complete problem", *Z. Wahrscheinlichkeitstheorie und. Verw. Gebiete*, (4), 316-339
- Wets, R.J.B. (1966b), "Programming under uncertainty: the equivalent convex problem", *Siam Journal Application Mathematics*, 14(1), 89-105
- Wets, R.J.B. (1966c), "Programming under uncertainty: the solution set", *Siam Journal Application Mathematics*, 14(5), 1143-1151
- Wets, R.J.B.e Walkup, David W (1967), "Stochastic Programs with Recourse", *Siam Journal Application Mathematics*, 5(15),1299-1314.
- Wets, R.J.B. (1989), Stochastic Programming (Capítulo 5) em *Handbooks in Operation Research and Management Science, Volume 1 , Optimization*, editado por Nemhauser, G. L., Kan, A.G.Rinnoy e Todd, M.J. - North Holland.