

UNIVERSIDADE FEDERAL DE SÃO CARLOS  
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

MODELOS E ALGORITMOS PARA A OTIMIZAÇÃO DO PLANEJAMENTO  
DA PRODUÇÃO DE GRÃOS ELETROFUNDIDOS

JOSÉ ROBERTO DALE LUCHE

SÃO CARLOS - SP

2011

**MODELOS E ALGORITMOS PARA A OTIMIZAÇÃO DO  
PLANEJAMENTO DA PRODUÇÃO DE GRÃOS ELETROFUNDIDOS**

UNIVERSIDADE FEDERAL DE SÃO CARLOS  
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

MODELOS E ALGORITMOS PARA A OTIMIZAÇÃO DO PLANEJAMENTO  
DA PRODUÇÃO DE GRÃOS ELETROFUNDIDOS

JOSÉ ROBERTO DALE LUCHE

Tese submetida ao Programa de Pós-Graduação em Engenharia de Produção como requisito parcial à obtenção do título de Doutor em Engenharia de Produção.

Orientadora: Prof. Dra. VITÓRIA PUREZA

Co-Orientador: Prof. Dr. REINALDO MORABITO

SÃO CARLOS - SP

2011

**Ficha catalográfica elaborada pelo DePT da  
Biblioteca Comunitária/UFSCar**

L936ma

Luche, José Roberto Dale.

Modelos e algoritmos para a otimização do planejamento da produção de grãos eletrofundidos / José Roberto Dale Luche. -- São Carlos : UFSCar, 2012.  
140 f.

Tese (Doutorado) -- Universidade Federal de São Carlos, 2011.

1. Programação da produção. 2. Dimensionamento de lotes. 3. Grãos eletrofundidos. 4. Programação (Matemática). 5. Heurística. 6. Meta-heurística. I. Título.

CDD: 658.53 (20<sup>a</sup>)



## FOLHA DE APROVAÇÃO

Aluno(a): José Roberto Dale Luche

TESE DE DOUTORADO DEFENDIDA E APROVADA EM 12/12/2011 PELA  
COMISSÃO JULGADORA:

Prof.ª Dr.ª Vitória Maria Miranda Pureza  
Orientador(a) PPGE/UFSCar

Prof. Dr. Reinaldo Morabito Neto  
PPGE/UFSCar

Prof.ª Dr.ª Débora Pretti Ronconi  
POLI/USP

Prof. Dr. Miguel Cezar Santoro  
POLI/USP

Prof. Dr. Néocles Alves Pereira  
PPGE/UFSCar

Prof. Dr. Marcos Nereu Arenales  
ICMC/USP

---

Prof. Dr. Mário Otávio Batalha  
Coordenador do PPGE

*Dedicado a Adilson Dale Luche (in memoriam).*



*"Compreender que há outros pontos de vista é o início da sabedoria"*  
*Joseph Campbell*

## AGRADECIMENTOS

---

A minha orientadora Vitória Pureza, amiga e incentivadora, pela orientação, confiança e atenção disposta ao trabalho.

Ao meu co-orientador Reinaldo Morabito, pelas colocações precisas, incentivo e comprometimento.

Aos funcionários da empresa estudada, em especial ao Pedro Grecco e Diego J. Padula, por toda a ajuda prestada.

Aos professores membros da banca, Débora Ronconi, Marcos Arenales, Miguel Santoro, Néocles A. Pereira, pela ajuda, sugestões e correções.

A professora Eli A. Toso, que contribuiu com sugestões importantes.

Aos amigos do grupo de pesquisa pelo coleguismo e troca de informações.

A minha mãe Anna Maria Dale Luche (*in memoriam*), minha protetora.

Ao meu pai Roberto Dale Luche, por quem devo parte de qualquer sucesso.

Aos colegas e todas as pessoas que me ajudaram de qualquer forma, meu eterno agradecimento.



## RESUMO

---

O número de aplicações bem sucedidas que utilizam modelos de otimização têm acompanhado a evolução dos computadores, tanto em *hardware*, com máquinas mais poderosas, como em *software*, com algoritmos mais inteligentes. Devido à importância da modelagem como ferramenta de apoio à tomada de decisão, muitos trabalhos que exploram formas de representação de problemas e técnicas de solução de modelos vêm sendo desenvolvidos. Este trabalho apresenta uma descrição detalhada das operações envolvidas no planejamento e controle da produção na indústria de grãos eletrofundidos e propõe o uso de modelos e métodos exatos e heurísticos para apoio à tomada de decisões nesta atividade, em particular, na programação da produção. Várias visitas foram realizadas a empresas do setor, e em uma dessas empresas foi empreendido um estudo de caso com o objetivo de formular alternativas para aumento da produtividade e a melhoria do nível de serviço aos clientes. Otimizar a programação da produção de grãos eletrofundidos não é uma tarefa simples, principalmente devido à grandeza dos tempos de preparação dos equipamentos, à diversidade de produtos e às limitações dos prazos de entrega da carteira de pedidos. Com base no estudo de caso, modelos de programação linear inteira mista que combinam modelos clássicos de seleção de processos e dimensionamento de lotes monoestágio foram desenvolvidos, e uma heurística construtiva, duas variantes de busca local, e um algoritmo GRASP foram propostos para resolver um dos modelos. Resultados computacionais com uma instância real e conjuntos de instâncias geradas aleatoriamente indicam que tanto os métodos exatos como heurísticos propostos são capazes de gerar programações da produção tão boas ou melhores do que as atualmente empregadas pela empresa estudada.

Palavras chaves: Programação da produção, dimensionamento de lotes, grãos eletrofundidos, programação matemática, heurísticas, meta-heurísticas.

## ABSTRACT

---

The number of successful applications that use optimization models has followed the evolution of the computers, as much in hardware, with more powerful machines, as in software, with more intelligent algorithms. Due to importance of the modeling as a decision support tool, much effort has been made to mathematically describe systems of interest and devise techniques for solving such models. This work presents a detailed description of the operations involved in production planning and control of the electrofused grain industry and proposes the use of exact and heuristic methods to support decisions in such activities, particularly in production scheduling. Several visits were made to companies in this sector and a case study was carried out one of these companies in order to formulate alternatives to increase productivity and improve customer service. Optimizing the production scheduling of electrofused grains is not a simple task mainly because of the scale of the equipment setup times, the diversity of the products, and the narrow orders due dates. Based on the case study, mixed linear programming models that combine known models of process selection and single-stage lot sizing were developed, and a constructive heuristic, local search variants, and a GRASP algorithm were proposed to solve one of the models. Computational results with a real instance and randomly generated instance sets show that the exact methods as well as the heuristics can produce as good or better production scheduling than the ones currently employed by the studied company.

Keywords: Production scheduling, lot sizing, electrofused grains, mathematical programming, heuristics, meta-heuristics.

# SUMÁRIO

---

---

<b>1. INTRODUÇÃO.....</b>	<b>12</b>
1.1. Objetivos e justificativas.....	13
1.2. Metodologia.....	14
1.3. Organização do trabalho.....	15
<b>2. A INDÚSTRIA DE GRÃOS ELETROFUNDIDOS.....</b>	<b>16</b>
2.1. Matéria-primas sintéticas.....	16
2.2. Refratários.....	17
2.3. Abrasivos.....	18
2.4. Cerâmica de alta tecnologia/ cerâmica avançada.....	20
2.5. Empresas produtoras de eletrofundidos.....	20
2.6. O processo de produção de carbetos de silício.....	22
2.7. O processo de produção de óxido de alumínio (ALO).....	28
2.7.1. Alumina eletrofundida branca (ALO branco).....	31
2.7.2. Alumina eletrofundida marrom (ALO marrom).....	34
2.8. O Sistema de produção.....	35
<b>3. REVISÃO BIBLIOGRÁFICA.....</b>	<b>41</b>
3.1. Contexto do trabalho.....	41
3.2. Problema de seleção de processos.....	42
3.3. Problemas de dimensionamento de lotes.....	44
3.3.1. Características de sistemas de produção na classificação de problemas de dimensionamento de lotes.....	44
3.3.2. Problema de dimensionamento de lotes não capacitado ( <i>uncapacitated lot sizing problem</i> - ULSP):.....	47
3.3.3. Problema de dimensionamento de lotes capacitado ( <i>capacitated lot-sizing problem</i> - CLSP):.....	49
3.3.4. Problema discreto de dimensionamento e programação de lotes ( <i>discrete lot-sizing and scheduling problem</i> - DLSP):.....	50

3.3.5.	Problema de programação e dimensionamento de lotes contínuo (continuous setup lot sizing problem – CSLP): .....	53
3.3.6.	Problema de programação e dimensionamento de lotes compartilhado (proportional lot sizing and scheduling problem – PLSP):.....	55
3.3.7.	Problema de dimensionamento e sequenciamento de lotes generalizado ( <i>general lot sizing and scheduling problem</i> - GLSP):.....	56
<b>4.</b>	<b>MODELAGEM MATEMÁTICA</b> .....	<b>59</b>
4.1.	Discussão preliminar .....	59
4.2.	O modelo MFP .....	60
<b>5.</b>	<b>HEURÍSTICAS APLICADAS AO PROBLEMA</b> .....	<b>66</b>
5.1.	Uma heurística construtiva (HC) .....	66
5.2.	Uma heurística de busca local ( <i>k-opt</i> ).....	69
5.2.1.	Pré-processamento .....	71
5.3.	Um algoritmo GRASP .....	72
5.3.1.	Fase de construção .....	73
5.3.2.	Fase de melhoria .....	74
5.4.	Minimização da falta e do excesso (estoque) .....	75
<b>6.</b>	<b>RESULTADOS COMPUTACIONAIS</b> .....	<b>77</b>
6.1.	Experimentos com a instância real .....	78
6.2.	Experimentos com instâncias geradas aleatoriamente.....	79
6.2.1.	Variação de demanda.....	79
6.2.2.	Variação do número de períodos .....	82
6.3.	Experimentos em um ambiente computacional mais eficiente.....	83
6.3.1.	Aplicação de heurísticas de busca local <i>k-opt</i> .....	85
6.3.2.	Algoritmo GRASP.....	89
6.4.	Experimentos com processamento paralelo.....	93
6.5.	Variações das heurísticas visando à minimização da falta e do excesso de produção .	94
<b>7.</b>	<b>MODELAGEM MATEMÁTICA - EXTENSÕES</b> .....	<b>98</b>
7.1.	Modelo discreto com custos de falta, excesso e <i>setup</i> (D-FES) .....	98
7.2.	Modelo com <i>setup</i> contínuo CS-FES .....	103
7.3.	Modelo compartilhado P-FES .....	107
7.4.	Modelo geral G-FES.....	110
7.5.	Modelo G-FES com custos de <i>setup</i> nulos no início do período de trabalho.....	112

<b>8. CONCLUSÕES .....</b>	<b>114</b>
<b>9. REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>118</b>
Apêndice I - Modelo MFP implementado no GAMS .....	128
APÊNDICE II - Modelo D-FES implementado no GAMS.....	129
APÊNDICE III - Modelo CS-FES implementado no GAMS .....	131
APÊNDICE IV - Modelo P-FES implementado no GAMS.....	133
APÊNDICE V - Modelo G-FES implementado no GAMS .....	135
Apêndice VI - Experimentos com parâmetros modificados.....	137
ÍNDICE REMISSIVO DE REFERÊNCIAS .....	139

## LISTA DE FIGURAS

---

---

Figura 2.1. Aplicação das matérias-primas sintéticas na indústria cerâmica .....	17
Figura 2.2. (a) Edward Acheson, (b) Acheson em laboratório.....	23
Figura 2.3. Fornos Acheson, à esquerda um forno em operação e à direita outro resfriando o material. ....	24
Figura 2.4. Forno Acheson .....	24
Figura 2.5. Conjunto de peneiras para classificação dos grãos eletrofundidos .....	25
Figura 2.6. Diagrama de fluxo de material na produção de carbetos de silício. ....	25
Figura 2.7. Amostras de carbetos de silício cristalizado e parcialmente cristalizado (Ferrocarbo).....	26
Figura 2.8. Carbetos de silício verde.....	26
Figura 2.9. Bauxita. ....	28
Figura 2.10. Amostras de ALO branco (a) e marrom (b).....	30
Figura 2.11. Forno Higgins .....	32
Figura 2.12. Bloco de ALO branco. ....	32
Figura 2.13. Quebra do material. ....	33
Figura 2.14. Diagrama de fluxo de material na fábrica de ALO. ....	33
Figura 2.15. Amostras de ALO branco: (a) grãos rejeitados e (b) grãos de alta qualidade.....	34
Figura 2.16. Curvas de produção de grãos nos moinhos. ....	36
Figura 2.17. Representação de modelos de malhas de fios metálicos. ....	37
Figura 4.1. Exemplos de programação da produção do item EC31_150. ....	61
Figura 5.1. Passos da heurística HC. ....	68
Figura 5.2. Passos da heurística $k$ -opt. ....	70
Figura 5.3. Exemplos de vizinhança da heurística $k$ -opt. ....	70
Figura 5.4. Passos da heurística $k$ -opt com vizinhança parcial. ....	71
Figura 5.5. Passos gerais de um algoritmo GRASP para um problema de minimização.....	73
Figura 5.6. Passos da fase de construção do algoritmo GRASP (heurística HC modificada)..	74
Figura 6.1. Demanda por período para produção em cada instância.....	80

Figura 6.3. Desvios percentuais médios das soluções com  $HC_{ext}$ , algoritmo  $BL_E$ , e algoritmo GRASP, em relação aos resultados com CPLEX11: (a) variação de demanda, e (b) variação de períodos..... 92

## LISTA DE QUADROS

---

---

Quadro 2.1. Dureza em Mohs para abrasivos minerais naturais).....	19
Quadro 2.2. Produtos ofertados pela Treibacher e Emas. ....	21
Quadro 2.3. Propriedades do SiC. ....	27
Quadro 2.4. Composição química do SiC. ....	27
Quadro 2.5. Propriedades do óxido de alumínio. ....	31
Quadro 2.6. Composição química do ALO branco da classe EK8R.....	34
Quadro 2.7. Composição química do ALO Marrom da classe EC31R.....	35
Quadro 6.1. Características dos algoritmos de busca local. ....	86
Quadro 6.2. Combinações de variações paramétricas utilizados no algoritmo GRASP. ....	90
Quadro 7.1. Tonelada por dia produzida do item $i$ utilizando o processo $j$ .....	100
Quadro 7.2. Demanda original do produto $i$ no período $t$ .....	100
Quadro 7.3. Demanda forçada do produto $i$ no período $t$ . ....	106



## LISTA DE TABELAS

---

---

Tabela 2.1. Quantidades (kg) de grãos eletrofundidos produzidos por dia pelos processos. ...	38
Tabela 6.1. Resultados obtidos com uma instância real. ....	78
Tabela 6.2. Resultados com variação de demanda: resolução do modelo e aplicação da heurística HC versão padrão e estendida. ....	81
Tabela 6.3. Resultados com variação do número de períodos: resolução do modelo e aplicação da heurística HC versão padrão e estendida. ....	82
Tabela 6.4. Resultados com variação de demanda: resolução do modelo com CPLEX 7 e CPLEX 11 e heurística HC <sub>ext</sub> . ....	84
Tabela 6.5. Resultados com variação do número de períodos: resolução do modelo com CPLEX 7, CPLEX 11 e heurística HC. ....	85
Tabela 6.6. Resultados com variação de demanda: algoritmos de busca local. ....	87
Tabela 6.7. Resultados com variação de períodos: algoritmos de busca local. ....	87
Tabela 6.8. Sumário dos resultados computacionais: resolução do modelo, HC <sub>ext</sub> e algoritmo BL <sub>D</sub> e BL <sub>E</sub> . ....	89
Tabela 6.9. Resultados com variação de demanda: resolução do modelo, HC <sub>ext</sub> , algoritmo BL <sub>E</sub> , e algoritmo GRASP. ....	91
Tabela 6.10. Resultados com variação de períodos: resolução do modelo, HC <sub>ext</sub> , algoritmo BL <sub>E</sub> , e algoritmo GRASP. ....	91
Tabela 6.11. Resultados com variação do número de períodos: resolução do modelo com CPLEX 11 com um e quatro <i>threads</i> . ....	94
Tabela 6.12. Resultados com variação de demanda: resolução do modelo, HC <sub>ext</sub> , algoritmo BL <sub>E</sub> , e algoritmo GRASP com a função objetivo (4.1'). ....	96
Tabela 6.13. Resultados com variação de período: resolução do modelo, HC <sub>ext</sub> , algoritmo BL <sub>E</sub> , e algoritmo GRASP com a função objetivo (4.1'). ....	97
Tabela 7.1. Exemplo de programação da produção para oito períodos segundo premissas do modelo D-FES com <i>backlog</i> . ....	101

Tabela 7.2. Exemplo de programação da produção para oito períodos segundo premissas do modelo D-FES sem <i>backlog</i> .....	103
Tabela 7.3. Exemplo de programação da produção para oito períodos segundo premissas do modelo CS-FES com <i>backlog</i> . ....	105
Tabela 7.4. Programação da produção modelos D-FES e CS-FES com <i>backlog</i> . ....	106
Tabela 7.5. Exemplo de programação da produção para oito períodos segundo premissas do modelo P-FES com <i>backlog</i> .....	109

## LISTA DE ABREVIATURAS

---

ALO	Óxido de alumínio
CS-FES	Modelo de <i>setup</i> contínuo com custos de falta, excesso e <i>setup</i>
CLSP	<i>Capacitated lot sizing problem</i>
CSLP	<i>Continuous setup lot sizing problem</i>
D-FES	Modelo discreto com custos de falta, excesso e <i>setup</i>
DLSP	<i>Discrete lot sizing problem</i>
ERP	<i>Enterprise Resource Planning</i>
G-FES	Modelo geral com custos de falta, excesso e <i>setup</i>
GLSP	<i>General lot sizing problem</i>
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
HC	Heurística construtiva
HC <sub>ext</sub>	Heurística construtiva estendida
HC <sub>ext</sub> BL <sub>E</sub>	Heurística construtiva estendida com busca local (conjunto E)
k-opt	Parâmetro da heurística de busca local
MFP	Modelo para minimizar a falta de produção
MFEP	Modelo para minimizar a falta e o excesso de produção
P-FES	Modelo compartilhado com custos de falta, excesso e <i>setup</i>
PLSP	<i>Proportional lot sizing problem</i>
S <sub>i</sub> C	Carbeto de silício
ULSP	<i>Uncapacitated lot sizing problem</i>

**INTRODUÇÃO**

---

O estudo desenvolvido neste trabalho é dirigido a um problema de planejamento e controle da produção encontrado na indústria de grãos eletrofundidos. Empresas de grãos eletrofundidos produzem e comercializam vários tipos de produtos para aplicação nas indústrias de cerâmicos, abrasivos e refratários, tendo um papel bastante relevante na economia da região Sudeste do Brasil. O setor integra o grupo das Indústrias Químicas e Farmacêuticas, apresentando diversas unidades produtivas na região, em particular, no estado de São Paulo.

O planejamento da produção de grãos eletrofundidos envolve várias decisões importantes, em particular, com respeito aos níveis de estoques de matérias-primas e produtos intermediários, e à programação dos fornos e processos subsequentes (britagem, moagem e classificação) em função da demanda de produtos finais. Por esta razão, gerentes de produção em empresas do setor de grãos eletrofundidos têm, em geral, dificuldades para programar a produção. Somado a essas dificuldades, essas empresas produzem dezenas de milhares de toneladas de grãos por ano, o que envolve o consumo de quantidades de energia elétrica de grande magnitude. A complexidade das decisões e os altos gastos envolvidos tornam-se, portanto, uma fonte de estímulo para a pesquisa de alternativas que otimizem o processo e o planejamento da produção.

O desenvolvimento do projeto envolveu visitas a empresas do setor a fim de se delinear as características do processo e do planejamento da produção. Este estudo foi aprofundado na Treibacher Schleifmittel Brasil (Salto, SP), cuja capacidade nominal de produção é de 60.000 toneladas/ano de grãos. Cada produto consiste de grãos de um dado material e faixa de granularidade (diâmetro). O objetivo da empresa é o de maximizar sua produtividade, atendendo a demanda dos clientes por diferentes produtos nas datas solicitadas.

Neste sentido, esta pesquisa trata o problema de definir os processos utilizados em cada período de um horizonte de planejamento. Por processo entende-se o conjunto de peneiras (por sua vez, previamente associadas a decisões dos equipamentos a montante, como, por exemplo, moinhos e britadores) utilizado para classificação dos grãos em diferentes faixas de granularidade. Cada processo define o rendimento dos produtos finais, ou seja, tem um impacto direto sobre o atendimento da demanda nas datas requeridas.

O problema em questão tem natureza combinatória, pertencendo à classe dos problemas chamados de “dimensionamento de lotes” (*lot-sizing problems*). Segundo a teoria da complexidade computacional, a maioria dos problemas desta classe são NP-Difíceis (HSU, 1983) e, em geral, formulados por modelos envolvendo variáveis que representam a produção de cada produto em cada período.

Para adequação à realidade do planejamento e programação da produção de grãos eletrofundidos, este trabalho discute e apresenta modelos de programação inteira mista que combinam formulações conhecidas de seleção de processos e de dimensionamento de lotes monoestágio. Desta forma, os modelos podem ser vistos como de dimensionamento de lotes que, ao invés de “lotes de produtos”, utilizam “lotes de processos” para produzir um conjunto de produtos.

Além dos modelos, são também propostos métodos heurísticos para resolução de um dos modelos. O desempenho dos métodos heurísticos é comparado com as soluções do pacote de otimização GAMS/CPLEX obtidas para uma instância real, e conjuntos de instâncias geradas aleatoriamente com base nos dados reais.

## **1.1. Objetivos e justificativas**

Este trabalho tem por objetivo propor modelos e métodos de solução que auxiliem na tomada de decisões em uma das principais etapas da fabricação, ou seja, na decisão da escolha e do sequenciamento dos processos a serem utilizados. Em particular, o modelo de programação linear inteira mista proposto em Luche *et al.* (2009) é revisitado e extensões deste modelo são propostas para contemplar outras possíveis condições operacionais.

A indústria de matérias-primas granulares eletrofundidas é reconhecidamente uma das grandes consumidoras de energia elétrica, o que reforça a necessidade de tornar o processo e o planejamento da produção mais eficiente. O impacto de tais melhorias não se

restringe ao aspecto econômico, mas também à menor degradação do meio ambiente pela redução de substâncias tóxicas eliminadas pelo processo de fabricação inerente a esta indústria.

Neste sentido, técnicas de pesquisa operacional podem ser utilizadas para fomentar a eficiência das decisões do planejamento, em particular, no tratamento de problemas de grande complexidade como o aqui tratado. Além disso, e conforme discutido em Luche *et al.* (2009), é comum um programa de produção precisar ser modificado várias vezes devido a imprevistos ou pedidos urgentes, o que reforça a importância de ferramentas capazes de gerar em tempo razoável, programas de produção eficientes.

## **1.2. Metodologia**

Neste trabalho, metodologias diferentes e complementares são empregadas. Foram estudadas duas empresas produtoras de grãos eletrofundidos, especificamente, a Treibacher Schleifmittel Brasil e a Elfusa Geral de Eletrofusão. Diversas visitas foram realizadas com vistas ao entendimento do problema e ao acompanhamento das operações. Simultaneamente, o problema de planejamento da produção foi estudado sob a ótica da pesquisa operacional, por meio da leitura de artigos, livros e outros materiais. A combinação das informações derivadas dos estudos de caso e da pesquisa bibliográfica em métodos quantitativos resultaram no desenvolvimento dos modelos matemáticos e implementações computacionais. No estudo de caso realizado na Treibacher foi realizada a coleta de dados no sistema de produção e da demanda. Os resultados obtidos pelo Planejamento e Controle da Produção (PCP) da empresa são comparados com os resultados obtidos pelo modelo algébrico e as técnicas heurísticas empregadas para a resolução do problema.

De acordo com Bertrand & Fransoo (2002) e Morabito & Pureza (2010), esta pesquisa é empírica por ter sido motivada por uma aplicação real encontrada em visitas a processos de produção de grãos eletrofundidos que ainda não se encontravam documentados na literatura. A pesquisa também é considerada normativa, porque aborda soluções para o problema e compara o desempenho de diferentes estratégias de resolução.

### 1.3. Organização do trabalho

O trabalho é organizado como se segue. No Capítulo 2 são apresentadas as principais matérias-primas empregadas e produtos associados ao setor de eletrofundidos. De forma breve, é descrito o processo de produção de grãos. Cabe ressaltar que apesar da descrição ser baseada no estudo de caso realizado, ela representa de forma fidedigna outras empresas do setor. Em particular, é discutido o problema de classificação dos grãos associado ao problema de dimensionamento de lotes de processos.

O Capítulo 3 revisa modelos matemáticos e métodos de resolução para apoio a decisões do planejamento e controle da produção em sistemas de manufatura. A revisão não é extensiva pois procura enfatizar o material relevante ao propósito deste trabalho. São também discutidas as características de sistemas de produção sob a perspectiva de problemas de dimensionamento de lotes.

O Capítulo 4 discute como modelos de seleção de processos e de dimensionamento de lotes monoestágio podem ser combinados para representar o problema de seleção de lotes de processos da empresa estudada.

O Capítulo 5 descreve os métodos heurísticos propostos para a solução do modelo de programação inteira mista que visa minimizar a falta e atraso de produção da empresa (modelo MFP). Uma heurística construtiva seleciona processos com base em uma função de avaliação gulosa. A mesma função de avaliação é também utilizada em duas variantes de heurísticas de busca local e em uma implementação da meta-heurística GRASP (*Greedy Randomized Adaptive Search Procedure*).

No Capítulo 6 são apresentados e discutidos os resultados computacionais obtidos com a resolução exata do modelo da empresa e com os métodos heurísticos sob duas plataformas computacionais distintas.

No Capítulo 7 são propostas extensões do modelo tratado no Capítulo 4 para contemplar outros objetivos e condições operacionais, e que podem ser utilizadas no planejamento e controle da produção da indústria de grãos eletrofundidos.

Finalmente, no Capítulo 8, são apresentadas as conclusões da pesquisa.

## A INDÚSTRIA DE GRÃOS ELETROFUNDIDOS

Empresas de grãos eletrofundidos compõem parte da indústria de transformação mineral que produz e comercializa vários tipos de produtos para aplicação nas indústrias de cerâmicos, abrasivos e refratários. No Brasil, estas empresas estão situadas principalmente na região Sudeste do estado de São Paulo. Neste capítulo, discute-se de forma breve as principais matérias-primas, produtos, e empresas em operação deste setor. São também apresentados os processos de manufatura utilizados em uma planta localizada em Salto, SP, discussão esta aplicável a outras empresas. O capítulo é encerrado com a apresentação de um problema comum de planejamento e controle da produção deste setor, o qual consiste na classificação dos grãos eletrofundidos por tamanho e quantidade.

### 2.1. Matéria-primas sintéticas

O setor de matérias-primas sintéticas é constituído por empresas produtoras de matérias-primas que não ocorrem ou que ocorrem em pequena escala na natureza e que devido a isto, são obtidas sinteticamente. Matérias-primas sintéticas são aquelas que individualmente ou em mistura foram submetidas a um tratamento térmico (calcinação, sinterização, fusão e fusão/redução), e/ou produzidas por processos químicos. As mais utilizadas industrialmente incluem alumina (óxido de alumínio) sob diferentes formas (calcinação, eletrofundida e tabular), carbetos de silício, e produtos químicos inorgânicos diversos (ANFACER, 2001).

No Brasil existem pouco mais de dez empresas produtoras de matérias-primas sintéticas, produzindo cerca de 190.000 toneladas por ano e com um faturamento médio da ordem de US\$ 115 milhões (ABC, 2009). Em termos de aplicação, as matérias-primas



sintéticas são utilizadas nas indústrias de produtos refratários, cerâmicas, de abrasivos e metalúrgicas, conforme apresentado na Figura 2.1. Um breve resumo sobre estes produtos é apresentado nas seções seguintes.

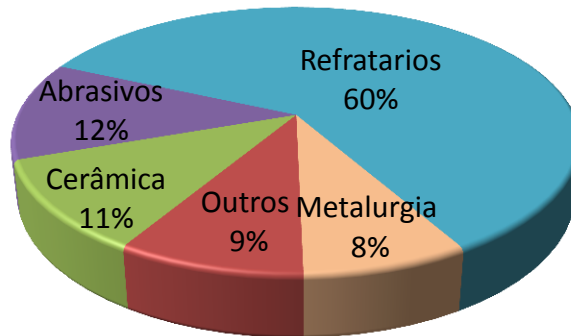


Figura 2.1 – Aplicação das matérias-primas sintéticas na indústria cerâmica. Fonte: ABC (2009)

## 2.2. Refratários

O entrelaçamento histórico entre os processos térmicos de manufatura e a tecnologia dos refratários começa com a descoberta do fogo. Os primeiros refratários foram fornecidos pela natureza, ou seja, cacinhos de rocha onde metais eram amolecidos para a confecção das primeiras ferramentas primitivas. Ao dominar o fogo, o homem descobriu que a argila como matéria-prima, quando queimada, poderia gerar utensílios de uso geral, caracterizados por elevada resistência mecânica.

Atualmente, vários são os produtos que compõem o grupo de materiais refratários, que como finalidade principal, deve suportar altas temperaturas apresentadas nos processos e operações em equipamentos industriais. Em geral, esforços mecânicos, ataques químicos, variações bruscas de temperatura e outras solicitações são também aplicados neste tipo de material. Além de boa resistência mecânica a baixa e a alta temperatura, é importante que seja levado em conta sua densidade e porosidade. A porosidade é uma variável microestrutural que deve ser controlada para produzir um tijolo refratário adequado. Ela provém também da porosidade dos grãos utilizados e dos vazios criados pelo conjunto de grãos. Se o grão é compacto, a porosidade é determinada em primeiro lugar pela granulometria adotada para as matérias primas e em menor escala, pela forma do grão e pela porcentagem de fundentes (CRUZ, 1982).

Refratários do tipo básico consistem fundamentalmente de magnésia (MgO), óxido de cálcio (CaO) e cromita, um óxido duplo de ferro e cromo ( $\text{FeCr}_2\text{O}_4$ ). No seu conjunto, esses materiais têm densidades aparentes elevadas, altos pontos de fusão e boa resistência ao ataque químico por óxidos e escórias básicas, porém, têm custo mais elevado (SMITH, 1996).

Dentre os refratários denominados ácidos, os refratários de sílica têm alta refratariedade, elevada resistência mecânica e rigidez a temperaturas próximas dos seus pontos de fusão. Os refratários de alumina contêm entre 50 e 99% de alumina e apresentam temperaturas de fusão superiores às dos refratários de argila. Podem ser usados em fornos que trabalham em condições mais extremas e a temperaturas mais altas do que os refratários de argila, porém, têm custo mais elevado.

Refratários especiais, por sua vez, contêm materiais cerâmicos que são usados em aplicações consideradas específicas. Alguns desses materiais consistem em óxidos com pureza relativamente alta, muitos dos quais podem ser produzidos com pouca porosidade. Nesse grupo estão alumina, sílica, magnesita ( $\text{MgCO}_3$ ), berília ( $\text{BeO}$ ), zircônia ( $\text{ZrO}_2$ ) e mulita sintética ( $3\text{Al}_2\text{O}_3 \cdot 2\text{SiO}_2$ ). Outros materiais incluem compostos à base de carbetto de silício, além do carbono e da grafita (CALLISTER, 2006).

O carbetto de silício, produto sintético obtido por eletrofusão e discutido em mais detalhes na seção 2.8, caracteriza-se por muitas propriedades interessantes, das quais as mais notáveis são sua alta resistência mecânica, seu baixo coeficiente de expansão térmica e sua elevada condutibilidade térmica, e também, elevada resistência à abrasão.

Cabe ressaltar que a indústria de refratários brasileira cobre praticamente todos os principais campos de aplicação de refratários e consegue atender satisfatoriamente as necessidades do mercado brasileiro (SEAE, 2011).

### **2.3. Abrasivos**

A abrasão é definida como sendo a operação de remover partículas de um material por atrito contra outro material que será quase sempre mais duro que o primeiro. A importância nas operações de abrasão reside no fato de que elas corrigem os defeitos das operações precedentes, como, por exemplo, nos processos de usinagem (SAINT-GOBAIN, 2011). Os abrasivos são, portanto, ferramentas destinadas aos processos de abrasão.

Muitas aplicações requerem a utilização de grãos abrasivos com características controladas. As pesquisas neste campo tiveram os primeiros sucessos no desenvolvimento em laboratório do carbetto de silício (SiC) e do óxido de alumínio ( $\text{Al}_2\text{O}_3$ ). Este segmento é o principal cliente das empresas estudadas neste trabalho.

Grãos abrasivos incluem o óxido de alumínio marrom, óxido de alumínio branco, carbetto de silício preto, carbetto de silício verde, abrasivos zirconados, diamantes naturais e artificiais e abrasivos especiais. Segundo Lawrence (1985), as partículas cerâmicas têm de ser duras e com arestas de corte aguçadas. A ferramenta abrasiva deve também possuir certa quantidade de porosidade, a fim de criar canais para passagem de ar ou líquido através da estrutura. O Quadro 2.1 apresenta uma classificação em dureza para abrasivos encontrados na natureza.

Quadro 2.1 - Dureza em Mohs para abrasivos minerais naturais (MORTENSEN, 2006).

<i>Material Mineral</i>	<i>Escala de dureza em Mohs<sup>1</sup></i>
Talco ( $\text{Mg}_3\text{Si}_4\text{O}_{10}(\text{OH})_2$ )	1
Gesso ( $\text{CaSO}_4 \frac{1}{2} \text{H}_2\text{O}$ )	2
Calcita ( $\text{CaCO}_3$ )	3
Fluorita ( $\text{CaF}_2$ )	4
Apatita ( $\text{Ca}_3(\text{PO}_4)_2(\text{OH}, \text{F}, \text{Cl})$ )	5
Feldspato ((Na, K) Al Si <sub>2</sub> O <sub>3</sub> , Ca Al <sub>2</sub> SiO <sub>3</sub> , Ba AlSi <sub>2</sub> O <sub>3</sub> )	6
Quartzo ( $\text{SiO}_2$ )	7
Topázio ( $\text{Al}_2[(\text{F}, \text{OH})_2\text{SiO}_4]$ )	8
Corundum ( $\text{Al}_2\text{O}_3$ )	9
Diamante (carbono)	10

Até o século XIX, existia à disposição apenas abrasivos naturais, a referência era a pedra de esmeril, mineral de cor preta, constituído de 40% de óxido de ferro e 60% de óxido de alumínio com dureza inferior a 9 Mohs<sup>1</sup>. No século XX, surgiram os abrasivos artificiais silicosos e aluminosos, de alta dureza.

Abrasivos silicosos são constituídos de carbetto de silício, feitos em fornos elétricos, com 9,6 Mohs de dureza, sendo recomendados para metais de fraca resistência a tração (ferro

<sup>1</sup> A Escala de Mohs quantifica a dureza dos minerais, isto é, a resistência que um determinado mineral oferece ao risco em sua superfície.

fundido, latão, cobre, alumínio e materiais não metálicos). Segundo dados da SEAE (2011), este mercado é de âmbito nacional, devido à falta de competitividade de abrasivos importados cujos preços são cerca de 38% superiores àqueles praticados internamente. Apenas 13% dos abrasivos utilizam SiC como insumo. O grupo Saint Gobain detém participação próxima a 60% no mercado de lixas e rebolos à base de SiC.

Os abrasivos aluminosos, por sua vez, são obtidos pela fusão da bauxita (minério de óxido de alumínio, silício e ferro) em fornos elétricos, obtendo dureza de 9,4 Mohs. São recomendados para metais mais resistentes a tração, como o aço e o bronze fosforoso.

#### **2.4. Cerâmica de alta tecnologia/ cerâmica avançada**

O aprofundamento dos conhecimentos da ciência dos materiais proporcionou o desenvolvimento de novas tecnologias e o aprimoramento das existentes nas mais diferentes áreas, como a aeroespacial, eletrônica, nuclear e muitas outras, e que passaram a exigir materiais de mais alta qualidade. Tais materiais passaram a ser desenvolvidos a partir de matérias-primas sintéticas de altíssima pureza e por meio de processos rigorosamente controlados (ABC, 2009).

Ao contrário dos cerâmicos tradicionais, que se baseiam principalmente na argila, os cerâmicos técnicos ou cerâmicos de engenharia são fundamentalmente compostos puros, ou quase puros, de óxidos, carbeto ou nitreto. A alumina ( $\text{Al}_2\text{O}_3$ ), o nitreto de silício ( $\text{Si}_3\text{N}_4$ ), o carbeto de silício (SiC) e a zirconia ( $\text{ZrO}_2$ ), combinados com outros óxidos refratários, são alguns dos mais importantes cerâmicos técnicos (SMITH,1996). Em particular, materiais cerâmicos à base de alumina são extensamente usados em uma variedade de campos da engenharia devido a sua excelente estabilidade química, boas propriedades elétricas e mecânicas, além de custo relativamente baixo no processo de manufatura (YOSHIZAWA *et al.*, 2004).

#### **2.5. Empresas produtoras de eletrofundidos**

##### **Treibacher – EMAS**

Entre os anos de 1952 e 1953, o Brasil iniciou a produção dos refratários fabricados por eletrofusão; até então, somente o processo de obtenção do vidro era feito pela fusão dos materiais. A Eletro Metalúrgica Abrasivos Salto (EMAS), localizada na cidade de

Salto, SP, iniciou a produção de alumina eletrofundida para refratários, com bauxita calcinada procedente de Poços de Caldas. Várias indústrias de refratários desenvolveram novas tecnologias com o enriquecimento em alumina de alguns produtos, permitindo a fabricação de refratários com até 95% de  $Al_2O_3$ , abrindo ao mercado a oferta de produtos com muito melhor desempenho. Posteriormente, a empresa passou a produzir carbetos de silício (TREIBACHER, 2011). A EMAS pertenceu também à Carborundum, que passou a produzir refratários acabados para as indústrias consumidoras. Posteriormente a EMAS foi vendida para a Alcoa.

A Treibacher Schleifmittel Brasil iniciou suas atividades no país em outubro de 2002 com a aquisição da EMAS. O Grupo Treibacher Schleifmittel, com sua sede na cidade de Villach, Áustria, é controlado pelo Grupo Imerys, com sede em Paris, França. O grupo possui onze plantas industriais em nove países, e está presente nos continentes europeu, americano e asiático. O Quadro 2.2 apresenta os produtos ofertados pela Treibacher e pela divisão EMAS.

A Treibacher é líder mundial em tecnologia e produção de óxidos de alumínio eletrofundidos. Estes materiais são utilizados nos mercados de abrasivos, polimento, refratários, cerâmica, jateamento e aplicações especiais. Além dos óxidos de alumínio eletrofundidos, a Treibacher produz o carbetos de silício, mas apenas na planta de Salto, SP (TREIBACHER, 2011).

Quadro 2.2 - Produtos ofertados pela Treibacher e Emas. Fonte: SEAE (2011)

<i>Produto</i>		<i>Treibacher e Grupo Imerys</i>	<i>Divisão EMAS</i>
Óxido de alumínio	Branco	Abrasivo	■
		Refratário	■
	Marrom	Abrasivo	■
		Refratário	■
	Carbetos de silício		■
	Micronização de alumina calcinada e hidratada		■
	Zircônio de alumina	■	
	Pigmentos e aditivos	■	
	Produtos de cerâmica	■	
	Caulim	■	
	Carbonato de cálcio para diversas aplicações	■	

### **Elfusa Geral de Eletrofusão**

Com uma linha de produtos semelhante a da Treibacher, a empresa Elfusa Geral de Eletrofusão se instalou na cidade de São João da Boa Vista, SP, inicialmente constituída pelo Grupo Curimbaba e Sivat. Atualmente a empresa pertence em sua totalidade ao Grupo Curimbaba.

A Elfusa é uma empresa de processamento mineral que atua há mais de 40 anos no mercado. Possui como principais clientes, indústrias de artigos refratários, abrasivos, pisos cerâmicos, dentre outras. Destina 60% de sua produção ao mercado interno, exportando os outros 40% para mais de 25 países. Na sua linha de produtos encontram-se os óxidos de alumínio eletrofundido marrom, branco, cimentos aluminosos, espinelas, mulitas eletrofundidas, wolastonita, ferro silício, alumina-zircônia e crômia-alumina. A empresa não produz o carbetto de silício, embora pretenda produzi-lo futuramente (ELFUSA, 2011).

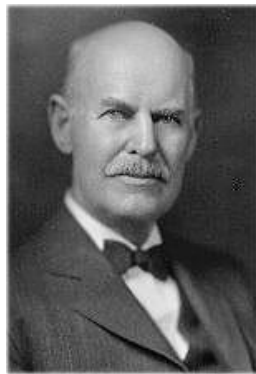
### **Saint Gobain / CASIL**

Em 1999, o grupo francês Saint Gobain adquiriu o controle acionário da empresa CASIL, situada em Barbacena, MG. A CASIL é uma empresa brasileira que atua na indústria de produtos minerais não metálicos, especificamente na produção de carbetto de silício. Na estimativa da empresa, a CASIL detém 60% do mercado de produção deste produto. Após a aquisição, com a regularização da produção, possível graças à estabilização do fornecimento de matérias-primas em volumes e prazos adequados, houve redução do consumo específico de energia elétrica. Segundo a empresa, a economia alcançou cerca de 6% por tonelada de carbetto de silício produzido. O intercâmbio com as demais fábricas de SiC do grupo Saint Gobain (todas no exterior) permitiu a absorção de tecnologia na área de processamento do produto bruto, possibilitando a introdução de novos itens à linha de produtos da CASIL. Além da CASIL, o grupo Saint Gobain também adquiriu a Carborundum, outra importante empresa desse segmento.

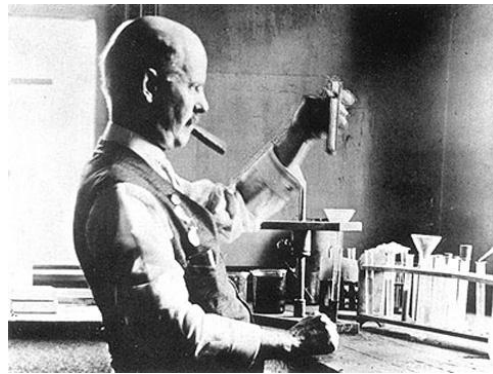
## **2.6. O processo de produção de carbetto de silício**

Primeiro abrasivo artificial, o carbetto de silício é produzido por meio do processo Acheson. O processo foi concebido pelo engenheiro norte-americano Edward Acheson (Figura 2.2.), o qual posteriormente fundou "The Carborundum Company" no estado da Pensilvânia, EUA (CHF, 2009).

Em razão de sua grande dureza (9,0 a 9,6 na escala de Mohs) e de sua boa condutibilidade, térmica e elétrica, o carbetto de silício é utilizado em grande escala para a fabricação de abrasivos (lixas e rebolos), os quais são utilizados no acabamento superficial de mármore, granitos, vidros e outros materiais. O carbetto de silício pode ainda ser utilizado em composições metálicas ferrosas como agente desoxidante e carburante, e na composição de elementos de aquecimento para fornos elétricos e de produtos para indústria de refratários.



(a)

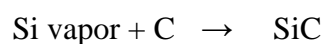
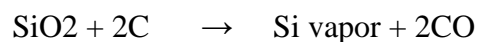


(b)

Figura 2.2 - (a) Edward Acheson, (b) Acheson em laboratório. Fonte: (CHF, 2009).

O processo de fabricação do carbetto de silício é essencialmente o mesmo até hoje. A matéria-prima consiste de uma mistura de areia silicosa, tão pura quanto possível (o teor de  $\text{SiO}_2$  não deve ser inferior a 97%) e coque de petróleo com um ligeiro excesso de carbono, aproximadamente 2% de cloreto de sódio (a fim de eliminar parte das impurezas sob a forma de cloretos metálicos voláteis) e cerca de 10% do total em serragem para facilitar a liberação do monóxido de carbono produzido durante a reação (ABC, 2009).

Tal mistura segue para fornos de formato retangular (fornos Acheson) com paredes contendo eletrodos de grafite ligados a um gerador elétrico através de barramentos de cobre (Figuras 2.3 e 2.4). Alma de pó de grafite é alocada ao centro da mistura para conduzir a eletricidade entre os eletrodos e provocar o aquecimento e transformação da matéria-prima. A mistura é então levada a uma temperatura superior a 2000 °C durante aproximadamente 36 horas. As principais reações químicas do processo consistem em:



O material próximo à grafite é o de melhor qualidade por ser onde se formam os melhores cristais; à medida que se afasta da alma de grafite, os cristais são menores e tidos como de baixa qualidade. Na zona mais fria do forno (abaixo de 2000 °C) encontram-se as camadas de estruturas diferentes, tais como: SiC amorfo e uma crosta constituída por materiais que não reagiram. O carбето de silício é constituído de 96 a 99% de SiC, o restante sendo silício, sílica livre, carbono livre, assim como óxido de cálcio, de ferro e de alumínio.



Figura 2.3 - Fornos Acheson, à esquerda um forno em operação e à direita outro resfriando o material.  
Fonte: ABC (2009)

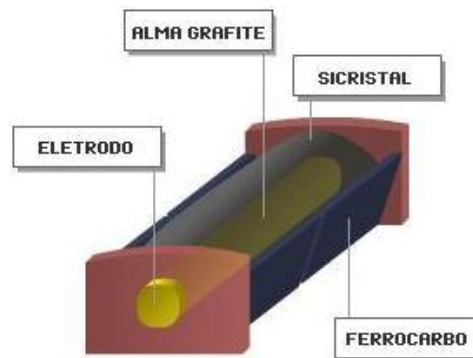


Figura 2.4. - Forno Acheson. Fonte: TREIBACHER (2011)

Em seguida, o material resultante passa por uma etapa de britagem e moagem; posteriormente é feita uma separação magnética, onde os grãos com alto teor de ferro são separados para reprocesso. Finalmente, a última etapa desse processo de transformação classifica os grãos por meio de um conjunto de peneiras vibratórias retangulares que definem a faixa granulométrica do produto (Figura 2.5).



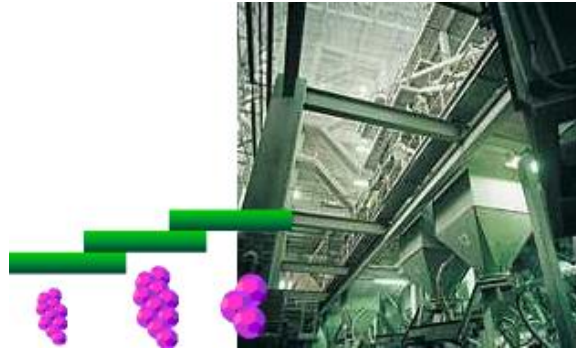


Figura 2.5 – Conjunto de peneiras para classificação dos grãos eletrofundidos. Fonte: TREIBACHER (2011)

As etapas descritas podem ser visualizadas no diagrama de fluxo de material ilustrado na Figura 2.6. Mais informações sobre o tamanho de partículas (grãos) e regulagem dos moinhos podem ser encontradas em Allen (1990). Cabe ressaltar que Pileggi *et al.* (1998) desenvolveram e aplicaram um *software* na ALCOA-EMAS que automatiza o processo de combinação de matérias-primas na obtenção de produtos cerâmicos.

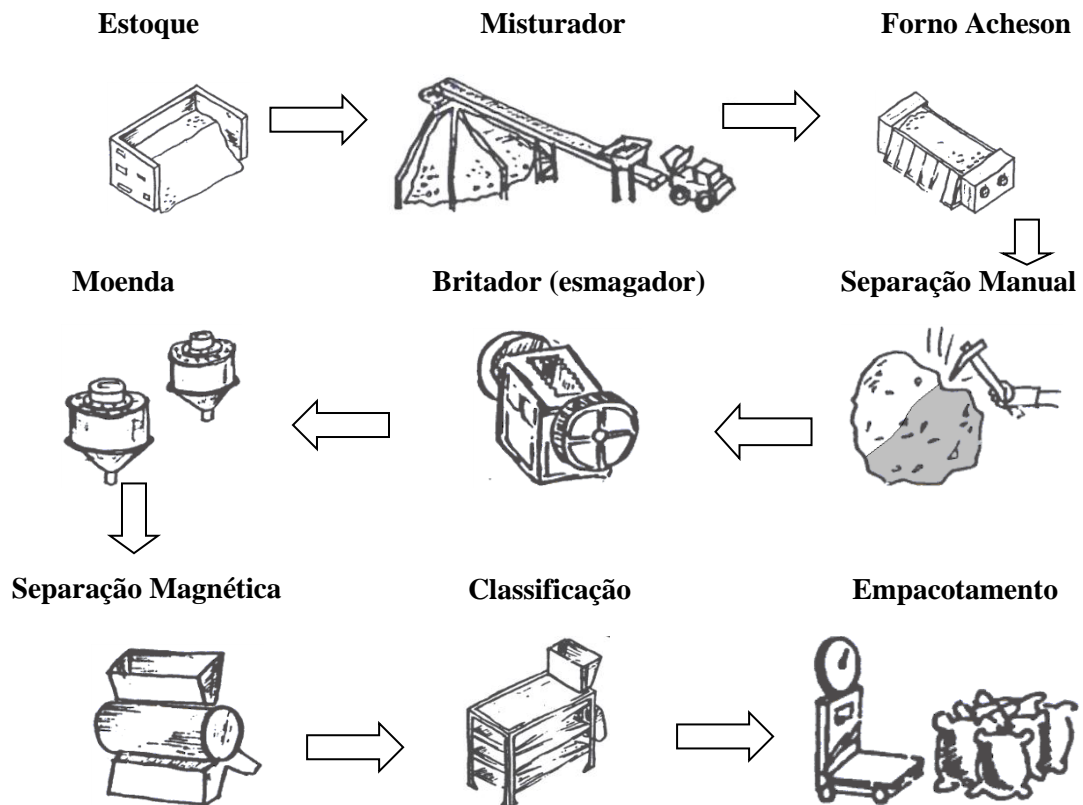


Figura 2.6 – Diagrama de fluxo de material na produção de carbetto de silício.

O carbeto de silício perde em dureza apenas para o diamante, mas tem pouca resistência a impactos devido à baixa ductilidade. As empresas produtoras dispõem do carbeto de silício totalmente cristalizado (SiC), e o parcialmente cristalizado (Ferrocarbo ou SiC de grau metalúrgico) (Figura 2.7).



Figura 2.7 - Amostras de carbeto de silício cristalizado e parcialmente cristalizado (Ferrocarbo). Fonte: TREIBACHER (2011)

A coloração do carbeto de silício cristalizado varia do verde claro mais ou menos transparente, ao preto com reflexos metálicos. Estas colorações dependem de inclusões de sílica, de carbono e principalmente de alumínio. A coloração preta indica um teor mais elevado de carbono livre, finamente repartido ou a um pequeno teor de alumínio ou de silício absorvido, enquanto que a cor verde (Figura 2.8) se deve à presença de teores de ferro.



Figura 2.8. - Carbeto de silício verde.

Em função de suas diferentes características químicas e físicas, o carbeto de silício de grau metalúrgico e o carbeto de silício cristalizado são utilizados em aplicações distintas. O carbeto de silício verde encontra-se na camada mais interna do bloco fundido e é caracterizado pelo elevado grau de pureza. É utilizado na produção de abrasivos devido a sua alta resistência física e condutividade elétrica. O carbeto de silício preto encontra-se na camada intermediária do bloco fundido e também possui elevado grau de pureza pouco inferior ao carbeto de silício verde. É utilizado na produção de abrasivos e principalmente

refratários. O carbetto de silício de uso metalúrgico, menos cristalizado, encontra-se na camada mais externa e é utilizado pela indústria metalúrgica como agente desoxidante.

Empresas que produzem o carbetto de silício metalúrgico podem produzir com a mesma mistura e no mesmo forno o carbetto de silício cristal (preto e verde), uma vez que os fatores determinantes para o tipo de produto são o tempo que o material permanece aquecido e a temperatura utilizada.

Quanto ao uso de carbetto de silício, a substituição entre metalúrgico e cristal, bem como entre o carbetto de silício preto e verde, não é interessante em razão dos diferentes usos, clientes, características físicas e químicas e, principalmente, os preços dos produtos. O carbetto de silício metalúrgico e o carbetto de silício cristal pertencem a mercados relevantemente distintos. No Quadro 2.3 são apresentadas suas propriedades, já sua composição química é apresentada no Quadro 2.4.

Quadro 2.3 - Propriedades do SiC.

<i>Propriedade</i>	<i>Valor</i>
SiC	98,5%
Dureza	9,0 - 10,0 Mohs
Peso específico (MEA)	3,21 g/cm <sup>3</sup>
Resistência Elétrica Específica	2,13 Ω /cm
Ponto de dissociação	acima de 2500°C (não chega à fusão)
Alta Refratariedade	sim
Extrema dureza	sim
Estável até 2500°C	sim, quando aquecido em condições redutoras
Alta resistência a abrasão	sim
Semicondutor	sim
Resistente ao Choque Térmico	sim

Quadro 2.4 - Composição química do SiC.

<i>Variáveis de Controle</i>	<i>Especificação</i>	<i>Típico ( % )</i>
Carbono Livre	0,60 Máx.	0,35
Si + SiO <sub>2</sub>	1,50 Máx.	1,10
Fe	0,50 Máx.	0,25
Al	0,40 Máx.	0,15
SiC	96,50 Mín.	97,87

## 2.7. O processo de produção de óxido de alumínio (ALO)

O óxido de alumínio ( $\text{Al}_2\text{O}_3$ ) é também conhecido como corindo artificial, como referência ao corundum ou corindo natural, por sua vez, um cristal de óxido de alumínio escavado em minas e considerado o segundo composto natural mais duro da natureza. Uma aplicação do corindo natural é servir como abrasivo; entretanto, este mineral é bastante raro e quando surge em concentração alta (em gemas) é bastante procurado pela indústria de lapidação. A variante vermelha do corindo natural é conhecida como "rubi" e as demais variantes são denominadas "safiras" (SEAE, 2011).

O óxido de alumínio fundido é produzido diretamente a partir do minério bauxita (Figura 2.9) ou da alumina, a qual já é o óxido de alumínio. Na prática comercial é encontrada uma grande variedade de tipos de alumina que recebem uma série de adjetivos, tais como calcinada, baixa soda, hidratada, gama, tabular, eletrofundida e outras. A base para a produção dessas aluminas é principalmente o processo Bayer.

No processo Bayer, a bauxita é enviada da mina à refinaria, onde o óxido de alumínio é isolado por meio da retirada dos outros componentes (óxidos de ferro e de silício). Na refinaria, é feito o tratamento da bauxita com hidróxido de sódio em tanques pressurizados e aquecidos a  $145\text{ }^\circ\text{C}$  (digestores), resultando em uma solução de aluminato de sódio e uma lama vermelha insolúvel, onde se concentram as impurezas. Depois que o óxido de ferro e outras impurezas são precipitados, acrescenta-se cristais de alumínio hidratado. Formam-se cristais pesados, posteriormente triturados para expelir a água remanescente, restando um pó branco fino (alumina) que é encaminhado para a redução. De quatro a seis toneladas de minério de bauxita são produzidas aproximadamente duas toneladas de alumina.



Figura 2.9 – Bauxita.

A seguir, são descritos os principais produtos do segmento:

- Hidróxido de Alumínio (ou Alumina tri-hidratada ou hidrato) – É um produto inorgânico de origem sintética, com elevada pureza e grande consistência físico-química. Apresenta a característica de se decompor quando submetido a temperaturas acima de 230 graus Celsius. Suas principais aplicações são: (i) fonte de alumínio para as indústrias químicas, tais como sulfato de alumínio, aluminato de sódio, fluoreto de alumínio e catalisadores; (ii) fonte de alumínio para a indústria de vidros, tais como vidros de embalagem, ópticos e fibra de vidro; (iii) retardante de chama, supressor de fumaça e também utilizado nas indústrias de plástico reforçado com fibra de vidro, mármore sintético, fios e cabos elétricos, e isolantes elétricos.
- Óxido de Alumínio (ou Alumina ou Alumina calcinada) – Por meio de um processo controlado de calcinação do hidróxido de alumínio, obtém-se uma diversificada linha de aluminas calcinadas que são empregadas em diversas indústrias. O óxido de alumínio é considerado um óxido refratário pela sua capacidade de resistir a elevadas temperaturas sem sofrer alterações significativas na sua natureza química. É principalmente aplicado na indústria de refratários e fibras cerâmicas, na indústria vidreira, esmaltes e revestimentos cerâmicos, como matéria prima para a produção de aluminas, mulita, e na indústria de abrasivos e de polimento.
- Aluminas Eletrofundidas (Corindons) – O ALO branco e o ALO marrom (coridons) têm propriedades físico-químicas muito similares às do carbetto de silício (alta dureza, alta refratariedade e alta resistência química), com aplicações na indústria de refratários, abrasivos, revestimentos cerâmicos e isoladores elétricos. Os coridons apresentam-se tanto na cor marrom (eletrofusão de bauxita calcinada e aditivos) como na cor branca (eletrofusão da alumina calcinada).
- Mulita eletrofundida Branca – A mulita é um silicato de alumínio encontrado também na natureza, porém muito raramente. É obtida industrialmente também por fusão de matérias-primas num forno elétrico, tal como os coridons. Este material é destinado à indústria de materiais refratários para siderurgia, indústria de vidros, cerâmica e outras.

- Produtos Ultrafinos – Algumas aplicações demandam a utilização de partículas ultrafinas de modo a ter maior reatividade e/ou dar melhor acabamento ao produto final. As duas empresas visitadas durante este trabalho possuem equipamentos que permitem que esta cominuição (fragmentação a pedaços muito pequenos) seja feita levando as partículas a microns sem interferir nas propriedades químicas dos materiais.

No restante desta seção, a discussão está restrita ao ALO branco e ao ALO marrom. Estes segmentos receberam maior atenção no estudo devido a maior complexidade de seus sistemas de produção e mercados. De fato, a grande variedade de aplicações a que ambos os produtos se destinam, e a existência de carteiras de pedidos com demandas distribuídas em várias datas, torna a tarefa de programação da produção bastante difícil.

A diferenciação entre o ALO branco e o ALO marrom reside na matéria-prima empregada e suas aplicações distintas. A variedade marrom é produzida a partir da bauxita calcinada; durante o processo produtivo, limalhas de ferro são adicionadas como agente redutor de impurezas (tal como sílica) contidas na bauxita. Já a variedade branca é produzida a partir da alumina calcinada e não requer a utilização de agentes redutores. O ALO branco (Figura 2.10.a) possui grau de pureza acima de 97,5%, enquanto que o ALO marrom (Figura 2.10.b) possui grau de pureza entre 95% e 97,5%. O Quadro 2.5 apresenta algumas propriedades físicas e químicas relevantes destas duas variantes de óxido de alumínio.

O setor de abrasivos utiliza o óxido de alumínio fundido a fim de garantir rigidez, dureza e resistência ao desgaste em produtos como lixas, esmeris, utensílios para lapidação, utensílios para polimento de superfícies e jateamento. O setor de refratários utiliza o óxido de alumínio fundido para obter estabilidade térmica e química, cerâmica técnica, e cerâmica para construção civil.



Figura 2.10 - Amostras de ALO branco (a) e marrom (b). Fonte: TREIBACHER (2011)

Nas empresas visitadas, a fabricação de óxido de alumínio é dividida em duas seções, uma responsável pela produção de óxido de alumínio branco e a outra, pela produção de óxido de alumínio marrom. Embora de maneira geral os processos produtivos de ambos os produtos sejam similares, em particular, na moagem e na seleção, estes são manufacturados em linhas de produção separadas e independentes. O motivo é evitar a contaminação dos produtos entre si. Produzir os dois itens em uma mesma linha de produção em revezamento seria fonte geradora de altos custos/tempos de *setup* necessários à limpeza do maquinário.

Quadro 2.5 - Propriedades do óxido de alumínio.

<i>Propriedade</i>	<i>ALO Branco</i>	<i>ALO Marrom</i>
% $Al_2O_3$	99,5%	96,70 %
Dureza escala Mohs	9,0	9,0
Peso específico	3,65 g/cm <sup>3</sup>	3,85 g/cm <sup>3</sup>
Ponto de fusão	2040 °C	2000 °C
Alta Refratariedade	sim	sim
Alta dureza	sim	sim
Alta resistência a abrasão	sim	sim
Resistente ao impacto	-	sim
Pureza Química	sim	-

No processo produtivo destes produtos, a grande diferença consiste na classificação do grão. O material para abrasivos é selecionado em classes granulométricas, de acordo com padrões internacionais rigorosos e bastante específicos; já o material para refratários é selecionado em grupos mais abrangentes, principalmente de acordo com as necessidades dos consumidores. É importante frisar que o processo de seleção de material para abrasivos é mais trabalhoso, rigoroso e oneroso que o de material para refratários. Isto se aplica tanto para o óxido de alumínio branco como para o marrom.

### 2.7.1. Alumina eletrofundida branca (ALO branco)

Para a produção do ALO branco, parte-se da alumina calcinada de alta pureza. Ela é depositada em fornos do tipo Higgins; cada forno possui três eletrodos que aquecem a matéria-prima a uma temperatura entre 2000 e 3000° C, levando-a a fase líquida (Figura 2.11).



Figura 2.11 - Forno Higgins. Fonte: TREIBACHER (2011)

Após o processo de liquefação, o material fica em repouso até alcançar o estado sólido, sendo então transportado para uma área de resfriamento no qual permanece por 30 horas. O resfriamento lento confere alta estabilidade química e mecânica ao produto. Ao final do resfriamento, são obtidos blocos (Figura 2.12) de aproximadamente 20 toneladas.



Figura 2.12 - Bloco de ALO branco. Fonte: ELFUSA (2011).

O transporte nesta fábrica é realizado por uma ponte rolante, responsável pela primeira quebra do material (Figura 2.13); em seguida, o material passa por uma etapa de britagem e moenda de rolo, e posteriormente é feita a separação magnética. A última etapa desse processo de transformação classifica os grãos por meio de um conjunto de peneiras vibratórias circulares que definem a faixa granulométrica do produto. A vibração da peneira é instalada na placa do lado da caixa de tela; impulsionada pelo motor, a tela gira junto com o eixo comum; então, a força de inércia centrífuga produzida durante o processo de rotação irá forçar a tela a vibrar.





Figura 2.13 - Quebra do material. Fonte: TREIBACHER (2011)

As etapas citadas podem ser visualizadas no diagrama de fluxo de material ilustrado na Figura 2.14.

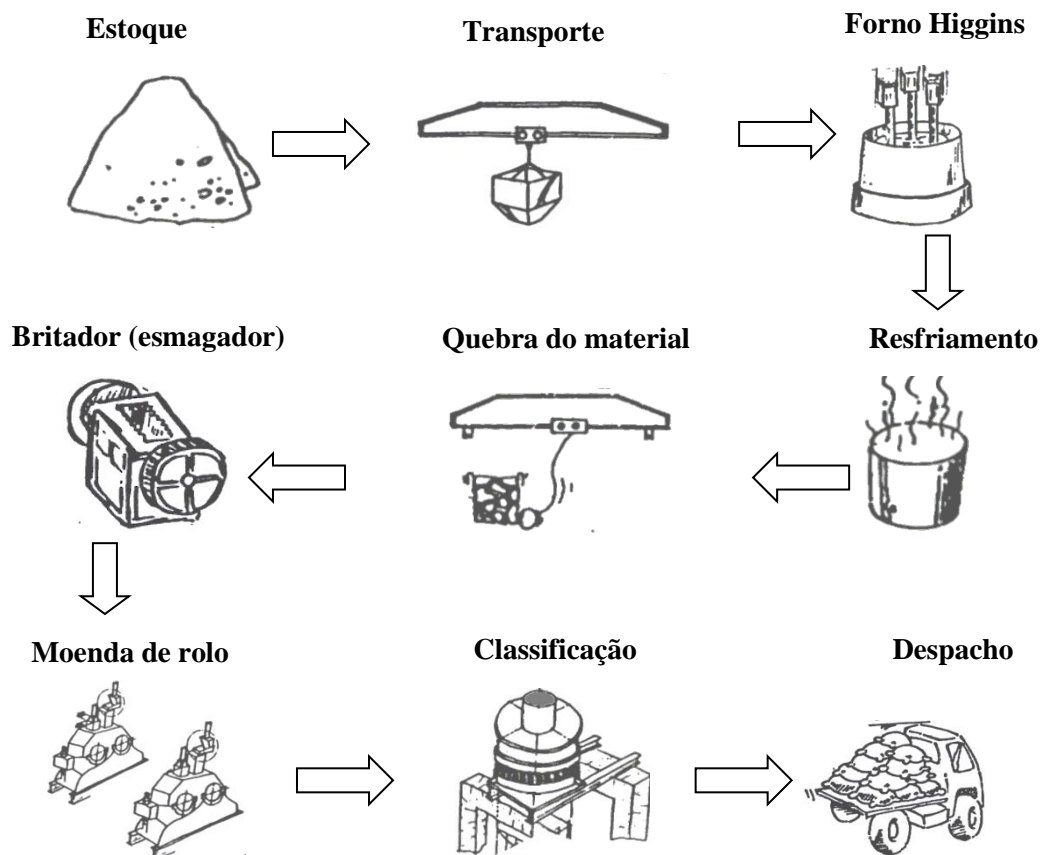


Figura 2.14 – Diagrama de fluxo de material na fábrica de ALO.

O controle acurado do processo garante um material extremamente branco no interior do bloco, contudo, nas regiões mais externas e constituídas de material parcialmente fundido e sinterizado, apresentam produtos de coloração escura (Figura 2.15), que são separados manualmente para evitar contaminação no produto branco de primeira qualidade. Como a separação manual é bastante imperfeita, cerca de 20% dos blocos acabam sendo destinados a um produto de segunda linha, no qual a coloração não é fator preponderante. (ELFUSA, 2011). O Quadro 2.6 apresenta a composição química do ALO branco.



Figura 2.15 - Amostras de ALO branco: (a) grãos rejeitados e (b) grãos de alta qualidade. Fonte: ELFUSA (2011).

Dentre as aplicações do ALO branco, incluem-se os concretos convencionais, baixo-cimento e anti-erosivos, rolos refratários, placas de válvulas gavetas, e insertos e esmaltes/fritas.

Quadro 2.6 - Composição química do ALO branco da classe EK8R.

<i>Variáveis de Controle</i>	<i>Especificação</i>	<i>Típico (%)</i>
SiO <sub>2</sub>	0,02 Máx.	0,02
Na <sub>2</sub> O	0,60 Máx.	0,38
Fe <sub>2</sub> O <sub>3</sub>	0,15 Máx.	0,09
Al <sub>2</sub> O <sub>3</sub>	99,00 Mín.	99,51

### 2.7.2. Alumina eletrofundida marrom (ALO marrom)

O processo produtivo do ALO marrom é quase idêntico ao do ALO branco; uma diferença, além das apresentadas no Quadro 2.5, se dá durante o processo de eletrofusão. Os óxidos metálicos contidos na carga, com exceção do óxido de alumínio, são reduzidos a metais elementares pelo carbono; com o ferro adicionado na carga, tais metais formam ligas que por serem mais densas que a alumina fundida se depositam no fundo do forno. A principal liga formada é o ferrosilício.

A principal diferença entre o ALO marrom e o ALO branco está, portanto, na composição química. O Quadro 2.7 apresenta a composição química do ALO marrom, o qual apresenta cerca de 97,0% de  $Al_2O_3$  e 2,50% de  $TiO_2$ , enquanto o ALO branco apresenta 99,5% de  $Al_2O_3$ . Esta diferença na composição química tem influência direta nas aplicações dos produtos; como exemplo, o ALO marrom é aplicado em operações abrasivas que se deseja maior resistência ao impacto por parte do abrasivo.

Dentre as aplicações do ALO marrom (classe EC31R), incluem-se concretos convencionais e de baixo cimento, tijolos aluminosos de  $Al_2O_3$ - $SiO_2$ -C e argamassas refratárias.

Quadro 2.7 - Composição química do ALO Marrom da classe EC31R

<i>Variáveis de Controle</i>	<i>Especificação</i>	<i>Típico ( % )</i>
$SiO_2$	1,50 Máx.	0,90
$TiO_2$	3,0 Máx.	2,70
$Fe_2O_3$	1,00 Máx.	0,35
$Al_2O_3$	95,00 Mín.	96,05

## 2.8. O Sistema de produção

O sistema de produção na planta é intermitente e repetitivo: sempre ocorrem mudanças no tipo de produto fabricado, gerando grande variedade de produtos. O *layout* é definido por produto, ou seja, as máquinas são arranjadas de acordo com a sequência de operações pelas quais o produto passa. Como todos os produtos possuem a mesma sequência de operações nas diversas máquinas, o padrão de fluxo no sistema de produção desta empresa é basicamente um *flow-shop*.

Diversas decisões importantes estão envolvidas no planejamento da produção das fábricas de ALO branco e marrom, e na fábrica de SiC. Em particular, estas decisões dizem respeito aos níveis de estoques de matérias-primas e produtos intermediários, e à programação da produção dos fornos e processos subsequentes (britagem, moagem e classificação) em função da demanda de produtos finais.

No presente estudo de caso, a separação dos grãos é feita por um conjunto de peneiras e existem várias destas que poderão ser utilizadas para obtenção de boa classificação dos grãos. A programação da produção dos grãos na empresa não está apenas restrita a escolha das peneiras que serão utilizadas para formar o conjunto, mas também

depende da programação dos moinhos. É pela regulagem (*aperto*) dos moinhos que se define se existirá maior quantidade de grãos finos ou de grãos grossos, sabendo-se que para os moinhos existem várias curvas de produção possíveis de serem implementadas.

Na Figura 2.16, são apresentados dois gráficos de curvas de produção de grãos nos moinhos. Os gráficos, especificam a faixa de espessura de grãos produzidas para uma dada regulagem do moinho. O gráfico da esquerda, por exemplo, apresenta uma curva que representa a produção de maiores quantidades de grãos grossos (note que a curva é mais acentuada à esquerda do gráfico). Já o gráfico da direita representa uma curva de produção de maior quantidade de grãos finos.

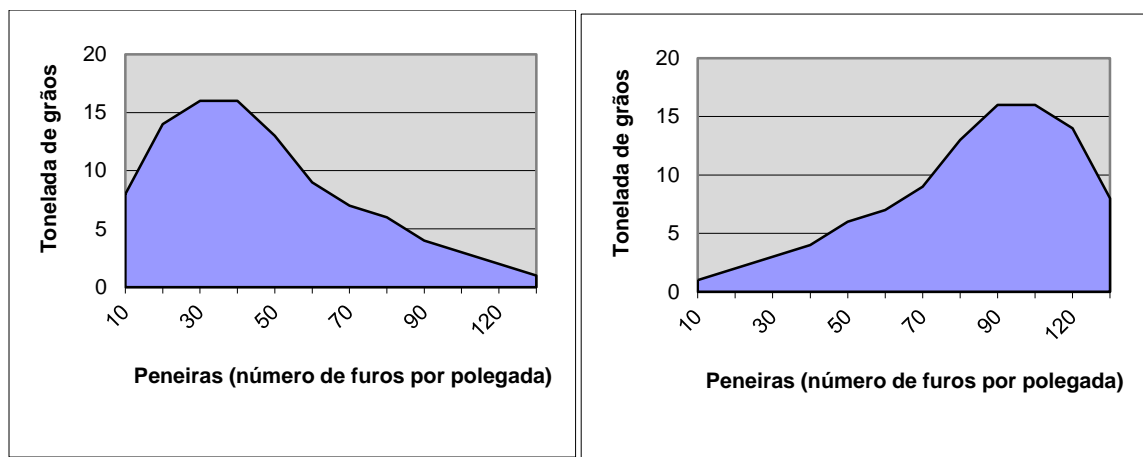


Figura 2.16 – Curvas de produção de grãos nos moinhos.

O tamanho do lote diário não é igual para todos os períodos. Se os grãos estão em uma curva de produção de grãos mais espessos, haverá volume alto de produção, enquanto que se estiverem em uma curva de produção de grãos finos o volume produzido será menor. Isso acontece devido ao ajuste feito nos moinhos. Ao apertá-los para que triturarem melhor o grão, serão obtidos grãos mais finos, mas devido a abertura para passagem dos grãos ser menor, o tamanho do lote também será reduzido. Portanto, quanto mais finos forem os grãos, menor será o tamanho do lote. A regulagem utilizada nos moinhos pode influenciar levemente no tempo de processamento, tornando-o maior quando de uma curva de produção maior de grãos finos.

É importante compreender então que o conjunto de peneiras define quais são os produtos selecionados (produzidos) e a regulagem dos moinhos define a quantidade de cada um destes produtos. Ou seja, num conjunto de peneiras poderá haver alta produção de grãos

do produto EC31\_120 e baixa do produto EC31\_20, mas em um outro período da produção, ao se alterar a regulagem dos moinhos (sem mudar o conjunto de peneiras), essa relação poderá se inverter ou tomar outra proporção.

A programação da classificação dos grãos deve ser feita em conjunto com a programação dos fornos, das britadeiras e das moendas, e envolve certas dificuldades. O conjunto de peneiras vibratórias utilizado para classificação dos grãos tem a finalidade de separar os grãos por tamanho (o tamanho é definido pela quantidade de furos por polegada quadrada na peneira) e pode ser montado com diferentes combinações de peneiras. Por exemplo, o produto chamado EC31\_120 é um grão que passa inicialmente por uma peneira de 100 furos por polegada quadrada e é retido em uma peneira de 120 furos (a quantidade de furos por polegada quadrada pode variar de 5 furos na peneira de grãos mais grossos, até 220 na peneira de grãos mais finos); representações de modelos de malhas (tela) para peneira podem ser vistos na Figura 2.17. Ainda é possível a mistura de várias faixas granulométricas formando um novo produto.

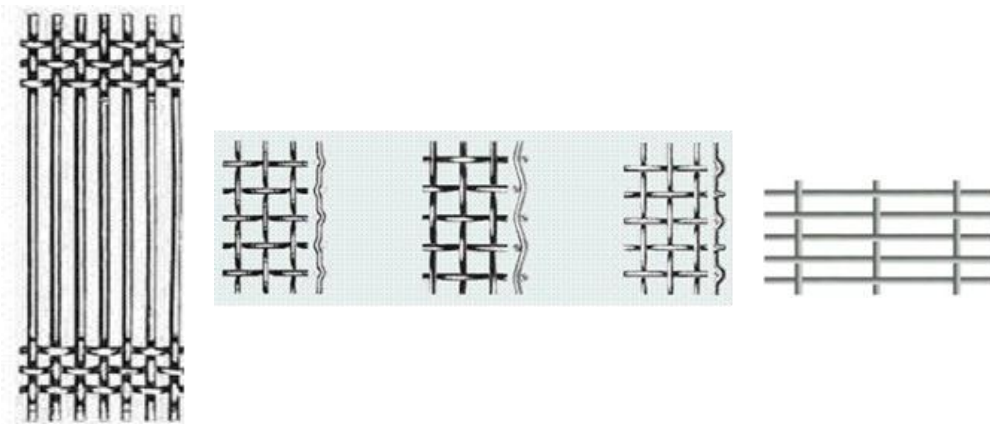


Figura 2.17 – Representação de modelos de malhas de fios metálicos. Fonte: PERALTA(2011).

Conforme inferido, cada produto possui uma faixa granulométrica diferente, e vários produtos podem ser demandados em um mesmo horizonte de planejamento da produção, tipicamente de um mês. A Treibacher trabalha sob pedido para alguns itens e para estoque para outros, programando sua produção no final do mês anterior, podendo efetuar mudanças no programa durante a produção em função de novos pedidos que chegam com data de entrega antes do final do mês programado. É importante enfatizar que um mesmo produto pode aparecer em vários pedidos com diferentes prazos de entrega no mês.

A combinação de um conjunto de peneiras e uma dada regulagem de moinhos e britadores define o que a empresa denomina de *processo*. A Tabela 2.1. ilustra um exemplo hipotético das quantidades em quilos diários (turno de 8 horas) de 23 produtos (grãos eletrofundidos) que podem ser produzidos por 10 processos diferentes. Por exemplo, o processo 1 é capaz de produzir 2.000 quilos do produto EC31\_36, 600 quilos do produto EC31\_120, 500 quilos do produto EC31\_150, e assim por diante, totalizando 27.700 quilos por dia (última linha da tabela).

Note que uma das principais características deste sistema produtivo é que um mesmo produto pode ser produzido por diferentes processos e em quantidades diferentes, sendo que alguns produtos produzidos podem não ter sido demandados. Desta forma, é comum carregar estoques de produtos ainda sem demanda por longos períodos de tempo até o final do horizonte planejado.

Tabela 2.1 - Quantidades (kg) de grãos eletrofundidos (linhas da tabela) produzidos por dia pelos processos 1-10 (colunas).

Produto	Processo									
	1	2	3	4	5	6	7	8	9	10
EC31_36	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000
EC31_120	600		600		600		1000		600	
EC31_150	500			300	300			300	300	
EC31_180	300		300		300		300			
EC31_220	300			300						
EG52_120		500		500		500		500		10000
EG52_150		300	300			200	500		800	
EG52_180		300		300				300	1000	
EC31R3-8_1-4	2000	2000	2000	2000	1000	2000	2000	2000	3600	10000
EC31R3-5_7	5000	5000	5000	5000	4000	5000	5000	5000	4800	
EC31R5-16_4	2000	2000	2000	2000	1000	2000	2000	2000	5000	5000
EC31R_08_F	5000	5000	10000	8000	7000	7000	10000	10000		
EC31R_08_F1	6000				800		1500	1000		
EC31R_08_F2		400		500		1000		1000		
EC31R_08_F3					500		1000	1000		2000
EC31R_08_F4	2000		2000			1000				
EC31R_08_F5		900					1000			
EC31R_08_G			1000	1000	1000					
EC31R_08_G1		1000	1000		1000					
EC31R_08_G2						1000		1000	8000	
EC31R_08_G3	2000			5000		4000				
EC31R_08_G4		8000			7000					
EC31R_08_G5			1200	800	500		1000			
<b>TOTAL</b>	<b>27700</b>	<b>27400</b>	<b>27400</b>	<b>27700</b>	<b>27000</b>	<b>25700</b>	<b>27500</b>	<b>26100</b>	<b>26100</b>	<b>27000</b>

Devido aos altos tempos de preparação dos conjuntos de peneiras (principal custo de *setup*), a empresa evita utilizar mais de um processo por dia. A troca de peneiras costuma

ser feita após o expediente normal, para serem utilizadas apenas no próximo dia de trabalho (é importante salientar que os turnos de trabalho não totalizam 24 horas por dia). A empresa trabalha com centenas de possíveis conjuntos de peneiras, e tem dificuldades para programar os fornos, as britadeiras e as moendas, e escolher os melhores conjuntos de peneiras e a sequência de utilização dos mesmos, de maneira a produzir, de forma eficiente e dentro dos prazos estabelecidos, a demanda de produtos em carteira.

Para cada período é necessário encontrar o processo que melhor atende a quantidade demandada de cada produto, sabendo-se que, por um lado, dificilmente todos os produtos demandados serão produzidos por este processo, e que, por outro lado, provavelmente outros produtos não demandados serão produzidos. A programação da produção de grãos está também condicionada às seguintes considerações:

- 1 O horizonte de programação adotado em geral é de um mês, o que se resume normalmente a 19 períodos (dias) em que há produção na empresa;
- 2 Novos pedidos de clientes podem ser aceitos ao longo de um horizonte de tempo já programado. Ou seja, dentro de uma programação definida, pode ser necessário refazer a programação para satisfazer a nova demanda. Isto pode ocorrer inclusive nos períodos "congelados" de um horizonte de tempo rolante (veja em Araujo *et al.* (2007), uma aplicação da técnica de horizonte rolante em uma pequena fundição de aço);
- 3 O estoque inicial dos produtos não é considerado, uma vez que este é decrementado da demanda destes produtos no primeiro período, antes de dar início à programação;
- 4 Os produtos são produzidos em monoestágio (demanda independente), portanto não existe ordem de precedência na produção de itens.
- 5 As restrições de capacidade das máquinas são consideradas no momento da elaboração dos processos de produção, ou seja, cada processo de produção leva em conta as restrições de capacidade dos equipamentos da linha de produção.
- 6 Existe ainda um tipo de produto que é designado por *faixa*, pois é um produto resultante da mistura de diversos outros produtos atingindo uma média granulométrica satisfatória para um produto de classe B.

O material de baixa qualidade que sai dos fornos e que ainda não passou pelo britador retorna ao forno para ser fundido novamente; produtos de boa qualidade que foram produzidos sem demanda, permanecem em estoque até que haja demanda ou oportunidade de utilizá-los em alguma faixa que tenha demanda. A manutenção dos estoques é necessária para que seja possível atender a demanda dos produtos e faixas, entretanto os excessos devem ser evitados devido aos custos, mesmo quando pequenos, de carregar e gerenciar estoques entre períodos.

Atualmente, a programação da produção, feita manualmente pelo departamento de planejamento e controle da produção da empresa, prioriza prazos de entrega e clientes (em geral apertados e negociados pelo departamento de vendas) e fornece um nível de serviço diferenciado a clientes preferenciais. Verifica-se também uma tendência entre os clientes de dividir suas demandas mensais em pedidos semanais ou quinzenais, o que resulta em um maior número de pedidos com quantidades demandadas de menor porte. Nos casos em que não é possível programar a produção de alguns itens de forma a atender completamente suas demandas em certo período, a empresa renegocia os prazos de entrega com os clientes a fim de minimizar o descumprimento dos mesmos (LUCHE *et al.* 2009). Portanto, uma melhor programação da produção reduz o número de renegociações de prazos de entrega entre a empresa e seus clientes. Além disso, também facilita a aplicação de certas técnicas de programação, como o uso de horizonte rolante (CLARK & CLARK, 2000).



**REVISÃO BIBLIOGRÁFICA**

---

Este capítulo revisa modelos de programação matemática clássicos assim como métodos de resolução elaborados com a finalidade de apoiar decisões do planejamento e controle da produção em sistemas de manufatura. A revisão é parcial e procura enfatizar o material relevante ao propósito deste trabalho, especificamente, problemas de seleção de processos e de dimensionamento de lotes.

**3.1. Contexto do trabalho**

Em sistemas de manufatura, ao serem formulados os objetivos, é necessário também, planejar como atingi-los, organizar os recursos humanos e físicos necessários para a execução, programar a ação dos recursos humanos sobre os recursos físicos e controlar esta ação para a correção de eventuais desvios. No âmbito da administração da produção, este processo é realizado pela função de Planejamento e Controle da Produção (PCP).

Segundo Contador & Contador (1997), o objetivo principal do PCP é propiciar informações adequadas para que os administradores possam gerenciar as atividades da operação produtiva e satisfazer, de forma contínua, à demanda dos consumidores. O gerenciamento se dá por meio do planejamento, onde é formalizado o que se espera que aconteça; por meio do controle, torna-se possível lidar com as variações que possam ocorrer no sistema, de forma que as metas planejadas sejam alcançadas (SLACK *et al.*, 2009). Pode-se entender também que o planejamento da produção estabelece as metas para um horizonte de planejamento, procurando otimizar a utilização dos recursos disponíveis para atender aos compromissos firmados em carteira de pedidos e estar melhor preparado para atender à demanda futura (JOHNSON & MONTGOMERY, 1974).

Segundo Toso (2008), dentre os objetivos específicos das atividades de PCP em sistemas de manufatura incluem-se: estabelecer as quantidades a produzir de cada produto (dimensionamento de lotes); definir as quantidades a produzir de cada produto por um processo ou por diversos processos alternativos (seleção de processos); determinar os níveis de produção para cada estação de trabalho (balanceamento de linha), e programar as operações produtivas (*scheduling*). Diversos modelos e métodos de resolução para o tratamento destes problemas podem ser encontrados na literatura de gerência da produção e pesquisa operacional, como, por exemplo, em Johnson & Montgomery (1974), Hax & Candea (1984), Winston (1991), Askin & Standridge (1993), Williams (1993), Graves *et al.* (1993), Gershwin (1994), Nahmias (1995), Colin (2007), Arenales *et al.* (2007) e Taha (2008).

Nas seções seguintes, é inicialmente apresentado o modelo clássico de seleção de processos; em seguida, é feita uma caracterização e breve revisão de modelos de dimensionamento de lotes, com ênfase aos que descrevem os problemas *Discrete lot Sizing Problem* (DLSP) e *Proportional Lot Sizing Problem* (PLSP). Estes últimos fornecem a base para o desenvolvimento dos modelos do Capítulo 7. A discussão inclui também uma revisão de métodos exatos e heurísticos descritos na literatura para resolução desses modelos clássicos.

### **3.2. Problema de seleção de processos**

Neste tipo de problema, as demandas de produtos são conhecidas previamente ao longo de um horizonte de planejamento. A produção de cada produto pode resultar de vários “caminhos” alternativos (rotas, processos) pelos quais ele pode ser produzido. Os custos unitários e os recursos de produção utilizados dependem do processo selecionado. Os recursos têm disponibilidade limitada em cada período (por exemplo, capacidade de produção em horas), de maneira que os produtos competem por esses recursos de acordo com o processo de produção escolhido. O problema consiste em determinar o quanto de cada produto deve ser produzido por cada processo em um período a fim de minimizar os custos de produção, sujeito às restrições impostas por limitações de recursos e de demanda.

Matematicamente, o problema de seleção de processos pode ser descrito por um modelo de programação linear, conforme apresentado em Johnson e Montgomery (1974):

## Formulação matemática

Índices :

$i$ : produto ( $i=1,\dots,m$ );

$j$ : processo ( $j=1,\dots,J$ );

$k$ : recurso ( $k=1,\dots,K$ );

Variáveis :

$q_{ij}$  quantidade do produto  $i$  produzida pelo processo  $j$ ;

Parâmetros:

$c_{ij}$  custo de produzir uma unidade do produto  $i$  por meio do processo  $j$ ;

$R_k$  quantidade disponível do recurso  $k$ ;

$p_{ijk}$  quantidade do recurso  $k$  necessário para produzir uma unidade do produto  $i$  pelo processo  $j$ ;

$d_i$  demanda do produto  $i$ .

$$\min z = \sum_{i=1}^m \sum_{j=1}^J c_{ij} q_{ij} \quad (3.1)$$

s.a.

$$\sum_{j=1}^J q_{ij} = d_i \quad \forall i \quad (3.2)$$

$$\sum_{i=1}^m \sum_{j=1}^J p_{ijk} q_{ij} \leq R_k, \quad \forall k \quad (3.3)$$

$$q_{ij} \geq 0, \text{ inteiro} \quad \forall i, j \quad (3.4)$$

Em (3.1), a função objetivo procura minimizar o custo total de produção; as equações (3.2) asseguram que a demanda de cada produto será satisfeita; as desigualdades (3.3) correspondem às restrições de disponibilidade de recursos; e (3.4) correspondem às restrições de domínio das variáveis.

### 3.3. Problemas de dimensionamento de lotes

Assim como no problema de seleção de processos, em problemas de dimensionamento de lotes cada produto pode ser produzido por diferentes processos alternativos e os custos e recursos de produção dependem do processo escolhido. Entretanto, outras características também consideradas incluem, por exemplo, a possibilidade de estocagem dos produtos a um dado custo em cada período. Além disso, o tempo e custo de preparação dos equipamentos envolvidos na produção de cada tipo de produto podem depender ou não da sequência programada (em seleção de processos não é realizado o sequenciamento dos processos escolhidos). O problema consiste em determinar o quanto produzir de cada produto em cada processo e em cada período, de maneira a minimizar os custos de produção e estocagem, sujeito às restrições de limitação de recursos e atendimento da demanda.

Enquanto o problema de seleção de processos é de fácil resolução (classe P), o mesmo não pode ser dito sobre a maioria dos problemas de dimensionamento de lotes. Tal motivação aliada a sua relevância prática, justifica o fato da literatura especializada ser bastante vasta. Em particular, Drexler & Kimms (1997), Karimi *et al.* (2003) e Staggemeier & Clark (2001), apresentam revisões de variações do problema. Outros *surveys* sobre o problema são encontrados em Zhu & Wilhelm (2006), Gao *et al.* (2008), Quadt & Kuhn (2008) e Robinson *et al.* (2009). Estudos de problemas de dimensionamento e sequenciamento de lotes e métodos de solução são também encontrados em Bitran & Matsuo (1986), Karmarkar *et al.* (1987), Matta & Guignard (1989), Kimms (1996a), Feng & Cheng (1998), Armentano *et al.* (1999), Clark & Clark (2000), Haase & Kimms (2000), Meyr (2000, 2002), Surie & Stadtler (2003), Fleszar & Hindi (2004), Toledo *et al.* (2007, 2009), Poltroniere *et al.* (2008), Ferreira *et al.* (2009), Molina *et al.* (2009) e Tonaki & Toledo (2010).

#### 3.3.1. Características de sistemas de produção na classificação de problemas de dimensionamento de lotes

Problemas de dimensionamento de lotes são delineados conforme as características do sistema de produção em que se inserem. Karimi *et al.* (2003) aponta as características descritas nos itens (i)-(viii), apresentados a seguir. Jans & Degraeve (2008) e Heuvel & Wagelmans (2008) consideram outras três características que têm sido estudadas

em dimensionamento de lotes somente em anos recentes, por sua vez, discutidas nos itens (ix)-(xi).

- (i.) **Horizonte de planejamento:** É o intervalo de tempo no qual o programa mestre de produção se estende, podendo ser finito ou infinito. Um horizonte de planejamento finito geralmente é acompanhado de demanda dinâmica, enquanto que o horizonte infinito por demanda estacionária (estática). Um sistema pode ser observado de forma contínua ou discreta, dividindo os problemas de dimensionamento de lotes nos tipos '*large bucket*' ou '*small bucket*'. Nos problemas '*large bucket*', os períodos de tempo são grandes o bastante para produzir vários itens, enquanto que em problemas '*small bucket*' os períodos são relativamente pequenos, sendo produzido apenas um item em cada período de tempo.
- (ii.) **Número de níveis (estágios de produção):** Os sistemas de produção podem ser classificados como sistemas monoestágio (único estágio) e sistemas multiestágio. Em sistemas monoestágio é esperado que as matérias primas sejam transformadas diretamente em produtos finais, os quais possuem demanda independente (veja item (vi)). Em sistemas multiestágio, existem duas ou mais etapas (estágios) de produção, as matérias primas são transformadas em produtos intermediários e em cada etapa há dependência do término da etapa anterior para que a atual seja iniciada (demanda dependente); o último estágio de produção é responsável pela montagem do produto final.
- (iii.) **Número de produtos:** O número de itens ou produtos finais em um sistema de produção caracteriza a dificuldade da modelagem e resolução do problema. Os sistemas são divididos em duas categorias: com um único item, e com dois ou mais itens diferentes, muitas vezes competindo pelos mesmos recursos, onde a complexidade desta última categoria é superior a do primeiro.
- (iv.) **Restrições de capacidade ou de recursos:** A capacidade e os recursos em um sistema produtivo incluem força de trabalho, equipamentos, máquinas, espaço, orçamento, dentre outros. Quando não há restrição de capacidade a ser considerada no problema, este é do tipo não capacitado; caso contrário, o problema é do tipo capacitado. As restrições de capacidade afetam diretamente a complexidade do problema, tornando sua resolução mais difícil.
- (v.) **Produto perecível:** Em casos em que os produtos são perecíveis, é necessário considerar restrições quanto ao tempo de armazenagem, outra característica que contribui para o aumento da dificuldade de resolução do problema.

- (vi.) **Demanda:** A demanda, ou seja, a quantidade dos itens a ser produzida em cada período para atender aos pedidos, é um dado de entrada do problema. A demanda pode ser classificada como dinâmica, que muda com o tempo, ou estática, que não se altera com o tempo. A demanda que varia no tempo deve ser satisfeita pela produção total, de forma a minimizar os custos totais os quais podem ser compostos por custos de *setup* e de estoque. Em problemas em que a demanda é conhecida “*a priori*”, tem-se a demanda determinística; caso contrário, a demanda é estocástica. Na demanda independente, as necessidades de um item não dependem das decisões de dimensionamento de lote de algum outro item. Problemas com demanda dinâmica e dependente são mais complexos que problemas com demandas estáticas ou independentes. Do mesmo modo, problemas com demandas estocásticas são mais complexos que problemas com demandas determinísticas.
- (vii.) **Estrutura dos custos e tempos de *setup* (preparação):** Ocorrem quando a mudança da produção de um tipo de produto para outro tipo acarreta em custo (por exemplo, gastos com mão de obra) e/ ou tempo despendido (por exemplo, limpeza da linha). Existem duas classes de *setup*: o *setup* dependente e o *setup* independente da sequencia utilizada de processos ou itens antecessores. O primeiro torna a resolução do problema de dimensionamento de lotes mais complexa (Zhu & Wilhelm, 2006).
- (viii.) **Política de atendimento da demanda:** Algumas empresas conseguem operar com atrasos no atendimento da demanda (*backlog*), permitindo que a solicitação de um período  $t$  possa ser atendida em períodos futuros. É possível, em alguns casos, que essa demanda não seja totalmente atendida, implicando em vendas perdidas ou menor receita. Custos de atraso são adicionados à função objetivo do modelo matemático, a fim de minimizar sua ocorrência.
- (ix.) **Remanufatura:** A remanufatura de itens que tenham sido produzidos em períodos anteriores pode ser útil para atender em parte ou mesmo totalmente a demanda em certos períodos. Esse problema pode ser visto como logística reversa. Entre as aplicações, incluem-se paletes e contêineres já usados que podem ser reparados e estocados para satisfazer parte da demanda por esses itens. Em caso relacionado, algumas empresas utilizam o termo *refurbish* em selos de *notebooks* e outros aparelhos eletrônicos para diferenciá-los de artigos novos. Pan *et al.* (2009) utilizam a remanufatura de itens no problema de dimensionamento de lotes, e Beltrán & Krass (2002) consideram em seu trabalho o retorno de produtos do varejo como uma segunda fonte de fornecimento.

(x.) **Janelas de tempo:** Os estudos em problemas de *lot sizing*, em geral, consideram que todos os produtos podem ter sua produção iniciada em qualquer período do horizonte de produção, porém, em casos particulares, não são todos os produtos que podem ser iniciados no início do horizonte. Neste caso, a produção depende da disponibilidade de matéria-prima ou produtos intermediários (semi-acabados). É um problema mais recentemente considerado na literatura de *lot sizing*; dentre os poucos trabalhos encontrados tratando dessa característica, incluem-se os de Lee *et al.* (2001), Wolsey (2005), Brahimi *et al.* (2006a), e Jaruphongsa & Lee (2007).

(xi.) **Capacidade acumulada:** Em problemas clássicos de *lot sizing* capacitado, a quantidade produzida é limitada pelas restrições de capacidade, mas qualquer capacidade restante no período é considerada perdida. Um problema de *lot sizing* que considere capacidade acumulada irá transferir qualquer capacidade restante ao próximo período de produção. Este problema pode ocorrer se a restrições de capacidade não correspondem a uma capacidade perecível, tal como o tempo, mas preferivelmente a uma capacidade física sob a forma de um orçamento monetário ou de matérias-primas. Um estudo detalhado sobre o problema com capacidade acumulada é encontrado em Sargut & Romeijn (2007).

### 3.3.2. Problema de dimensionamento de lotes não capacitado (*uncapacitated lot sizing problem* - ULSP):

O problema de dimensionamento de lotes não capacitado é a mais simples variação desta classe de problemas. Ele é caracterizado como monoestagiado, leva em conta custos de *setup* sem *carryover*<sup>1</sup>, e horizonte de planejamento finito dividido em vários períodos discretos. A demanda de cada produto ou item é conhecida em cada período, podendo variar sobre o tempo, e limites de capacidade não são considerados.

O ULSP multi-item pode ser representado matematicamente pelo seguinte modelo de programação inteira mista:

---

<sup>1</sup> *setup carryover* consiste em aproveitar a preparação da linha em um próximo período sem incorrer em *setup* novamente

### Formulação matemática

#### Índices:

$t$  período de produção ( $t=1,\dots,T$ );

$i$  item ( $i=1,\dots,m$ );

#### Variáveis de decisão:

$I_{it}$  quantidade estocada do item  $i$  no final do período  $t$ ;

$x_{it}$  variável binária que indica se houve *setup* para o item  $i$  no período  $t$  ( $x_{it} = 1$ ) ou não ( $x_{it} = 0$ );

$q_{it}$  quantidade produzida do item  $i$  no período  $t$ .

#### Parâmetros:

$T$  quantidade de períodos no horizonte de planejamento;

$m$  quantidade de itens;

$h_{it}$  custo de estocagem do item  $i$  no período  $t$ ;

$s_{it}$  custo de *setup* do item  $i$  no período  $t$ ;

$d_{it}$  demanda do item  $i$  no período  $t$ ;

$c_{it}$  custo unitário de produção do item  $i$  no período  $t$ ;

$M_{it} = \sum_{k=t}^T d_{ik}$  limitante superior do volume de produção do item  $i$  no período  $t$ .

$$\text{Min } z = \sum_{i=1}^m \sum_{t=1}^T (s_{it} x_{it} + c_{it} q_{it} + h_{it} I_{it}) \quad (3.5)$$

s.a.

$$q_{it} + I_{it-1} - I_{it} = d_{it} \quad \forall i, t \quad (3.6)$$

$$q_{it} \leq M_{it} x_{it} \quad \forall i, t \quad (3.7)$$

$$x_{it} \in \{0,1\} \quad \forall i, t \quad (3.8)$$

$$I_{it}, q_{it} \geq 0 \quad \forall i, t \quad (3.9)$$

A função objetivo (3.5) procura minimizar os custos de *setup*, de produção e de estoque; nas equações (3.6) é representado o balanceamento de estoques em cada período  $t$ .



Em (3.7), um limitante superior do volume de produção para cada item é determinado pela demanda acumulada até o último período do horizonte de planejamento. As restrições (3.8) e (3.9) definem o domínio das variáveis.

O ULSP é facilmente resolvido em tempo polinomial do número de períodos pelo algoritmo de Wagner & Whitin (1958). O problema multi-item é decomposto em  $m$  subproblemas, um para cada item (o modelo [(3.5)-(3.9)] é reescrito eliminando-se o índice  $i$  associado aos produtos), e cada subproblema é resolvido utilizando programação dinâmica. Heuvel & Wagelmans (2005) comparam métodos de solução para o problema com horizonte rolante. Em Yaman (2009) é considerada a produção de dois itens, onde o item de maior qualidade pode substituir parte ou toda a demanda do item de menor qualidade. Outras aplicações com o modelo ULSP são encontradas em Basnet & Leung (2005), Heuvel & Wagelmans (2005), Aksent (2007) e Melo & Wolsey (2010). Revisões sobre modelos ULSP podem ser encontradas em Wolsey (1995), Karimi *et al.* (2003) e Brahim *et al.* (2006b).

### 3.3.3. Problema de dimensionamento de lotes capacitado (*capacitated lot-sizing problem - CLSP*):

Apresentado inicialmente por Manne (1958), o problema de dimensionamento de lotes capacitado é tido como uma extensão do ULSP com restrições de capacidade (veja abaixo, as desigualdades (3.10)), em que vários itens competem pelo mesmo recurso.

Parâmetros adicionais em relação ao modelo ULSP:

$R_t$  recurso (capacidade) disponível da máquina no período  $t$ .

$p_i$  recurso necessário para produzir uma unidade do item  $i$ ;

$$\sum_{i=1}^m p_i q_{it} \leq R_t \quad \forall t \quad (3.10)$$

Conhecido como *large bucket problem* por produzir vários itens em um mesmo período, no modelo CLSP cada período consiste tipicamente de uma semana e o horizonte de programação é usualmente de até seis meses. Como os itens competem pelo mesmo recurso, o modelo não pode ser decomposto em problemas menores, o que faz da classe NP-Difícil

(BITRAN & YANASSE, 1982). Se tempos de *setup* também forem incorporados ao modelo, encontrar uma solução factível já é um problema NP-Difícil (MAES *et al.*, 1991).

Diferentes métodos de solução têm sido propostos para o CLSP e variações do mesmo. Recentemente, Akbalik & Penz (2009) resolveram, por meio de um algoritmo de programação dinâmica pseudo-polinomial, um caso especial do CLSP, onde várias máquinas alternativas produzem um único item, e as máquinas têm capacidade de produção e custos diferentes. O modelo considera custos de *setup* em cada período, custos fixos e variáveis para a produção do lote, e custos de estoque; todos os custos variam com o tempo. O objetivo do problema é encontrar um plano de produção sem *backlogging*, enquanto os custos de produção e de estoque são otimizados. Exemplos da aplicação de métodos exatos incluem os trabalhos de Lozano *et al.* (1991), Miller *et al.* (2000), Dastidar & Nagi (2005), Lobo *et al.* (2007), Waterer (2007), Mateus *et al.* (2009) e Chang (2010).

Araujo *et al.* (2004) propõem um modelo de programação inteira mista e um método de solução baseado em busca local para o tratamento de um problema prático de dimensionamento e sequenciamento de lotes em uma fundição. Parveen & Haque (2007) utilizam uma heurística construtiva para um problema multi-item dinâmico, a qual constrói a solução período a período com base na prioridade dos itens. Em Berretta *et al.* (2005) é proposto o uso de meta-heurísticas (busca tabu e *simulated annealing*) para resolver o problema de *lot sizing* capacitado multinível, com custos e tempos de *setup* e *lead times*. Métodos heurísticos baseados em redes neurais também têm sido utilizados para resolução de variações do problema (Aarts *et al.* (2000) e Gaafar & Choueiki (2000)). Outras aplicações de métodos heurísticos para problemas de *lot sizing* incluem os trabalhos de Lambrecht & Vanderveken (1979), Maes & Van Wassenhove (1986), Maes & Van Wassenhove (1988), Brandimarte *et al.* (1998), Alidaee *et al.* (2001), Gopalakrishnan *et al.* (2001), Alfieri *et al.* (2002), Lin *et al.* (2002), Merce & Fontan (2003), Toledo *et al.* (2006), Federgruen *et al.* (2007). Revisões e comparações de heurísticas podem ser encontradas em Jans & Degraeve (2007) e Buschkühl *et al.* (2010).

### **3.3.4. Problema discreto de dimensionamento e programação de lotes (*discrete lot-sizing and scheduling problem* - DLSP):**

Um modelo DLSP combina decisões de dimensionamento e sequenciamento de lotes uma vez que prescreve que somente um item pode ser produzido por período, utilizando

toda a capacidade do período. Por este motivo, Schrage (1984) o caracteriza como um modelo “tudo ou nada”. Por produzir apenas um item por período, o modelo DLSP é conhecido como *small bucket problem*, onde os períodos de produção são discretizados em horas ou dias. Custos de *setup* ocorrem apenas no início de um dado período quando a produção de um item diferente do período anterior é iniciada.

Matematicamente, o DLSP pode ser representado por um modelo de programação inteira mista, conforme apresentado em Drexl & Kimms (1997):

### Formulação matemática

Variáveis de decisão:

- $I_{it}$  quantidade estocada do item  $i$  no final do período  $t$ ;
- $x_{it}$  variável binária que indica se houve *setup* para o item  $i$  no período  $t$  ( $x_{it} = 1$ ) ou não ( $x_{it} = 0$ );
- $q_{it}$  quantidade produzida do item  $i$  no período  $t$ .
- $y_{it}$  indica se a linha está preparada para o item  $i$  no início do período  $t$  ligando o processo ( $y_{it} = 1$ ) ou não ( $y_{it} = 0$ );

Parâmetros:

- $h_i$  custo de estocagem do item  $i$ ;
- $d_{it}$  demanda do item  $i$  no período  $t$ ;
- $y_{i0}$  indica se a linha está preparada para o item  $i$  no início do período 1 ( $y_{i0} = 1$ ) ou não ( $y_{i0} = 0$ );
- $s_i$  custo de *setup* do item  $i$ ;
- $R_t$  recurso (capacidade) disponível da máquina no período  $t$ .
- $p_i$  recurso necessário para produzir uma unidade do item  $i$ ;

$$\text{Min } z = \sum_{i=1}^m \sum_{t=1}^T (s_i x_{it} + h_i I_{it}) \quad (3.11)$$

s.a.

$$I_{it} = I_{i(t-1)} + q_{it} - d_{it} \quad \forall i, t \quad (3.12)$$

$$p_t q_{it} = R_t y_{it} \quad \forall i, t \quad (3.13)$$

$$\sum_{i=1}^m y_{it} \leq 1 \quad \forall t \quad (3.14)$$

$$x_{it} \geq y_{it} - y_{i(t-1)} \quad \forall i, t \quad (3.15)$$

$$y_{it} \in \{0,1\} \quad \forall i, t \quad (3.16)$$

$$I_{it}, q_{it}, x_{it} \geq 0 \quad \forall i, t \quad (3.17)$$

A função objetivo (3.11) procura minimizar os custos de *setup* e de estoque; nas equações (3.12) é representado o balanceamento de estoques para cada item  $i$  em cada período  $t$ . Nas equações (3.13) é implementado o conceito de “tudo ou nada” em que, diferentemente do CLSP, o recurso necessário de produção para a quantidade de um item num período, deve ser igual à capacidade de produção deste item para o período. As inequações (3.14) restringem a produção à no máximo um item por período; o início de um novo lote de um item diferente daquele do período anterior é identificado por meio das restrições (3.15), onde  $x_{it} = 1$  indica a mudança de item e irá cobrar o *setup* para a produção deste. Note que se  $y_{it} = 1$ , há produção do item  $i$  no período  $t$  indicando, portanto, que a linha estará preparada para produção deste item no início de  $t = t + 1$  (veja a descrição do parâmetro  $y_{i0}$ ). As restrições (3.16) e (3.17) definem o domínio das variáveis. É interessante observar que não há a necessidade de definir as variáveis  $x_{it}$  como binárias devido a (3.16). Note que  $y_{it} = 1$  representa a situação em que há produção do item  $i$  no período  $t$  e que a linha estará preparada para utilização deste processo no início de  $t+1$ .

Dentre os primeiros estudos sobre o DLSP estão as aplicações de Lasdon & Terjung (1971) na programação da produção de um fabricante de pneus, e de Schrage (1984) que explorou outras extensões do modelo. Salomon *et al.* (1997) apresentam uma versão do DLSP com custo e tempo de preparação dependente da sequência. Um algoritmo exato baseado em programação dinâmica é proposto; os resultados computacionais mostram que o método é capaz de encontrar a solução ótima em tempo computacional aceitável para problemas de tamanho moderado (10 itens e 20 períodos, por exemplo). Outras aplicações de

métodos exatos em modelos DLSP são encontradas nos trabalhos de Fleischmann (1990). Jordan & Drexler (1998), Wolsey (2002) e Gicquel *et al.* (2009).

Fleischmann (1994) resolve o DLSP considerando custo de preparação dependente da sequência. O problema é reformulado como um problema do caixeiro viajante com janelas de tempo, e a partir desta reformulação é proposta uma heurística e um novo procedimento para determinação dos limitantes inferiores por meio de relaxação Lagrangiana.

Dessouky *et al.* (1999) consideram o problema de programação da produção em uma indústria de processamento químico, procurando minimizar a penalidade por atraso e por antecipação da produção. As demandas são produzidas em lotes, cada lote pode atender a demanda de várias ordens de clientes, os lotes têm tamanho fixo e apenas um lote pode ser produzido por período. O tempo de *setup* é independente da sequência e o tempo de processamento dos lotes é idêntico. Foi desenvolvida uma heurística que consiste na repetição de dois passos. Dada uma atribuição inicial das ordens dos clientes aos lotes, a melhor sequência dos lotes é encontrada resolvendo-se um problema de designação (dos lotes aos períodos). A seguir, um problema de transporte é resolvido a fim de redistribuir as ordens dos clientes aos períodos, obtendo-se uma nova formação de lotes em cada período. Os dois passos da heurística são repetidos até que não se obtenha melhoria da solução.

Supithak *et al.* (2010) apresentam um algoritmo baseado no problema de designação para resolver problemas sem custo de *setup*, um algoritmo *backward* para resolução de instâncias pequenas, e um algoritmo genético para instâncias de grande porte.

Em Brüggerman & Jahnke (2000) são discutidos alguns estudos sobre o problema; uma crítica interessante é feita sobre o fato de o DLSP considerar que um item esteja disponível tão logo ele seja produzido, porém quando são considerados micro períodos, em muitos casos práticos um item estará disponível somente depois que o último item do lote for produzido. Para considerar essa nova restrição, um modelo modificado é apresentado e resolvido em duas fases utilizando a meta-heurística *simulated annealing*.

### **3.3.5. Problema de programação e dimensionamento de lotes contínuo (*continuous setup lot sizing problem* – CSLP):**

Um inconveniente do modelo DLSP (descrito na seção anterior) consiste em não permitir a utilização parcial da capacidade de produção em cada período (restrição (3.13)). De fato, o modelo orienta a produção de um lote do tamanho da capacidade do período, mesmo

não tendo tal volume de demanda para os produtos em questão, o que gera estoques desnecessários.

Nos casos em que os tamanhos de lote necessários para atendimento da demanda utilizem parcialmente a capacidade do período, o modelo CSLP é uma melhor alternativa em relação ao modelo DLSP. No modelo CSLP, a restrição (3.13) é simplesmente substituída por:

$$p_i q_{it} \leq R_t y_{it} \quad \forall i, t \quad (3.13')$$

Assim como no DLSP, o modelo CSLP utiliza um único processo por período, podendo deixar a linha ociosa caso o lote seja menor que a capacidade no período.

Aparentemente, a única diferença entre os modelos baseados no DLSP e no CSLP está na utilização da capacidade total (DLSP) e total ou parcial do (CSLP) no período. Entretanto há um aspecto importante que deve ser explorado: no DLSP, os custos de *setup* ocorrem sempre que um novo lote é iniciado. Suponha, por exemplo, que um lote do processo  $j$  é terminado no período  $t$ . Além disso, suponha que outro lote para o mesmo processo  $j$  está programado para o período  $t' > t$ . Considere agora o caso em que a linha de produção fique parada entre os períodos  $t+1, \dots, t'-1$ . No modelo DLSP, custos de *setup* incorrerão duas vezes, enquanto que no CSLP, tais custos incorrem apenas uma vez. Isso acontece porque no CSLP têm-se:

$$y_{i(t+1)} = \dots = y_{i(t'-1)} = 1,$$

o que não contradiz

$$q_{i(t+1)} = \dots = q_{i(t'-1)} = 0,$$

como no DLSP.

Note que isso acontece porque  $p_i q_{it}$  não precisa ser igual à capacidade do período, ou seja, pode-se ter  $q_{it} = 0$  com  $y_{it} = 1$  sem implicação no valor de  $z$ . Aplicações do modelo CSLP são encontradas em Hindi (1995) e Gupta & Magnusson (2005).

### 3.3.6. Problema de programação e dimensionamento de lotes compartilhado (*proportional lot sizing and scheduling problem – PLSP*):

O modelo PLSP tem como principal característica permitir que um item  $j$  seja produzido em períodos em que a capacidade não tenha sido totalmente utilizada pelo item  $i$  ( $i \neq j$ ) que iniciou a produção. Isso permite que a produção de  $j$  seja adiantada e evita capacidade excedente na linha de produção. A troca para  $j$  implica em custo de *setup* no período  $t$  em questão; note, porém, que se no período  $t+1$  o item  $j$  continuar sendo produzido, não se incorrerá em custo de *setup* no início de  $t+1$ . Se  $i$  e  $j$  são produzidos em um único período, é especificada a ordem com que estes itens são processados, o que implica em decisões adicionais de programação (*scheduling*).

A função objetivo e a maioria das restrições do modelo DLSP se mantêm no modelo PLSP, porém, as equações (3.13) dão lugar às inequações (3.18) que restringem a produção de um item em certo período somente quando a máquina estiver devidamente preparada, e permite que sejam produzidos no máximo dois diferentes itens num período, desde que um deles já esteja sendo produzido no período anterior (ou seja, já havia uma preparação para produzir o primeiro item). Como até dois itens podem ser produzidos em um mesmo período,  $y_{it} = 1$  se refere ao item produzido até o final do período, e as restrições (3.19) são introduzidas para restringir a soma dos recursos utilizados à capacidade máxima por período.

$$p_i q_{it} \leq R_t (y_{i(t-1)} + y_{it}) \quad \forall i, t \quad (3.18)$$

$$\sum_{i=1}^m p_i q_{it} \leq R_t, \quad \forall t \quad (3.19)$$

Em Drexl & Haase (1995) é apresentado um método de solução heurístico para o modelo PLSP. Neste método, a cada período é associado um custo de produção. O método de solução consiste na escolha aleatória de cada item com probabilidade de escolha diferente segundo seu custo no período, favorecendo a escolha dos itens com menor custo. Posteriormente, Drexl & Haase (1996) apresentam um algoritmo guloso baseado no mesmo método. Outros estudos com meta-heurísticas aplicadas ao PLSP são encontrados em Kimms (1996b, 1999) e Belvaux & Wolsey (2001).

### 3.3.7. Problema de dimensionamento e sequenciamento de lotes generalizado (*general lot sizing and scheduling problem - GLSP*):

No modelo GLSP, cada período  $t$  é subdividido em subperíodos de tamanho variável, fixando-se um número máximo de lotes a serem produzidos por período ( $P_t$ ). Cada lote está associado a um subperíodo (ou posição) no período  $t$ . Os subperíodos são indexados cronologicamente e de maneira contínua ao longo do horizonte de planejamento.

Segundo Fleischmann & Meyr (1997), o GLSP é um modelo geral porque diversos modelos para dimensionamento e sequenciamento de lotes diferem dele apenas por restrições adicionais que modificam a estrutura e esforço computacional para obtenção das soluções. Em outras palavras, o conjunto de soluções do GLSP contém o conjunto de soluções de vários outros modelos como o CLSP, o DLSP, o CSLP e o PLSP.

O GLSP pode ser representado por um modelo de programação inteira mista, conforme a seguinte adaptação do modelo discutido em Drexl e Kimms (1997):

#### Formulação matemática

Variáveis de decisão:

- $I_{it}$  quantidade estocada do item  $i$  no final do período  $t$ ;
- $q_{in}$  quantidade produzida do item  $i$  no subperíodo  $n$ ;
- $x_{in}$  variável binária que indica se ocorreu *setup* para o item  $i$  no subperíodo  $n$  ( $x_{in} = 1$ ) ou não ( $x_{in} = 0$ );
- $y_{in}$  variável binária que indica se a linha está pronta para produzir o item  $i$  no subperíodo  $n$  ( $y_{in} = 1$ ) ou ( $y_{in} = 0$ ).

Parâmetros:

- $s_i$  custo de *setup* do item  $i$ ;
- $R_t$  recurso (capacidade) disponível da máquina no período  $t$ .
- $p_i$  recurso necessário para produzir uma unidade do item  $i$ ;
- $N_t$  número máximo de subperíodos no período  $t$ ;
- $F_t$  primeiro subperíodo do período  $t$ :  $F_t = 1 + \sum_{i=1}^{t-1} N_i$  ;
- $L_t$  último subperíodo no período  $t$ :  $L_t = F_t + N_t - 1$ ;



$$\text{Min } z = \sum_{i=1}^m \sum_{n=1}^N s_i x_{in} + \sum_{i=1}^m \sum_{t=1}^T h_i I_{it} \quad (3.20)$$

s.a.

$$I_{it} = I_{i(t-1)} + \sum_{n=F_t}^{L_t} q_{in} - d_{it}, \quad \forall j, t \quad (3.21)$$

$$p_t q_{in} \leq R_t y_{in} \quad \forall t, n = F_t, \dots, L_t \quad (3.22)$$

$$\sum_{i=1}^m \sum_{n=F_t}^{L_t} p_t q_{in} \leq R_t, \quad \forall t \quad (3.23)$$

$$\sum_{i=1}^m y_{in} \leq 1, \quad \forall n \quad (3.24)$$

$$x_{in} \geq y_{in} - y_{i(n-1)}, \quad \forall i, n \quad (3.25)$$

$$y_{in} \in \{0,1\}, \quad \forall i, n \quad (3.26)$$

$$I_{it} \geq 0, \quad \forall i, t \quad (3.27)$$

$$q_{in}, x_{in} \geq 0, \quad \forall i, n \quad (3.28)$$

A função objetivo (3.20) visa a minimização dos custos totais de *setup* e de estoque. Note que as restrições do modelo são idênticas aos dos modelos DLSP e PLSP, com exceção do uso de subperíodos  $n$ . As variáveis  $x_{in}$  não precisam ser declaradas como binárias em (3.28), uma vez que tal domínio já é garantido em (3.25).

Estudos sobre o GLSP são apresentados em Belvaux & Wolsey (2001), Koclar (2005) e Menezes *et al.* (2010). Aplicações do modelo GLSP são encontradas em Toso *et al.* (2009) e Clark *et al.* (2010) na programação da produção de fábricas de nutrição animal, e em Ferreira (2009, 2010) na programação da produção de uma empresa de bebidas.

No trabalho de Dogan & Grossmann (2006) são propostos um modelo e uma heurística para resolver um problema de manufatura monoestagiado. O horizonte de programação é dividido em semanas e as datas de entrega são definidas para o final de cada semana. A sequência da produção em cada semana é importante devido aos custos de *setup* serem dependentes da sequência.

Sung & Maravelias (2008) apresentam uma versão do GLSP para um problema monoestagiado, multi-item, com *setups* independentes da sequência e carregados por vários

períodos (*setup carryover*). Vários *setups* podem ser realizados em um mesmo período e o tempo de *setup* pode ser maior que um período.

No capítulo a seguir são apresentados os modelos matemáticos desenvolvidos para a indústria de grãos eletrofundidos e baseados em alguns dos modelos de *lot sizing* aqui discutidos.

**MODELAGEM MATEMÁTICA**

Este capítulo apresenta modelos de programação matemática elaborados com a finalidade de apoiar decisões do planejamento e controle da produção em sistemas de manufatura na indústria de grãos eletrofundidos. Em particular, é mostrado como os problemas das empresas Treibacher (empresa A) e Elfusa (empresa B) e suas variantes podem ser representados pela combinação de modelos conhecidos de seleção de processos e de dimensionamento de lotes monoestágio. Alguns dos modelos combinados são resultados de um estudo anterior (Luche, *et al.* 2009), enquanto outros são aqui propostos.

**4.1. Discussão preliminar**

Conforme apresentado no Capítulo 2, na indústria de grãos eletrofundidos, cada processo produz lotes de diferentes produtos e cada produto é produzido por diferentes processos. Note que este conceito geral está presente no modelo de seleção de processos, descrito na seção 3.2 do capítulo anterior. Entretanto, no problema tratado as quantidades dos produtos a serem produzidas por cada processo são parâmetros de entrada e não variáveis, ou seja, as quantidades dos produtos produzidos por cada processo são pré-determinadas e fixas para o período de tempo em que o processo é mantido ligado (equivalente a 1 turno de trabalho de 8 horas). Além dessa diferença, o modelo de seleção de processos não contempla decisões de sequenciamento, é mono-periódico, não considerando possibilidades de atraso no atendimento da demanda (*backlogging*).

O modelo do problema discreto de dimensionamento e programação de lotes (DLSP) e descrito na seção 3.3.4 do capítulo anterior é, por sua vez, caracterizado pelo conceito de “tudo ou nada”, ou seja, apenas uma única decisão relativa à produção é adotada a

cada período. Este conceito também é encontrado no problema de planejamento da produção das empresas A e B estudadas. Entretanto, enquanto no modelo DLSP esta decisão recai no *produto* a ser produzido a cada período, no planejamento da produção das empresas de grãos eletrofundidos, ela prescreve qual *processo* deve ser adotado em cada período.

A análise motivou o desenvolvimento do modelo MFP (Luche *et al.*, 2009). Descrito na próxima seção, o modelo MFP consiste de uma combinação dos modelos de seleção de processos e o DLSP. Note que segundo a caracterização proposta por Karimi *et al.* (2003), o problema estudado na indústria de grãos eletrofundidos pode ser classificado como de dimensionamento de lotes capacitado, máquina única, monoestágio, multi-item, com horizonte de planejamento finito e multi-períodos, demanda dinâmica, determinística e independente, onde custos de preparação não são considerados.

Além do modelo MFP, uma variação que considera custos/tempos de *setup* é proposta. Cabe ressaltar que outras empresas do setor, como a empresa B, não necessariamente operam com um único processo ao longo de cada período de produção (dia), porém, assim como a empresa A, encontram grande dificuldade em estimar os custos/tempos de *setup* envolvidos na troca de processos.

#### **4.2. O modelo MFP**

Nas empresas estudadas, os custos/tempos de *setup* são muito grandes e difíceis de serem estimados por serem dependentes da sequência. A empresa A, por exemplo, faz a troca do processo de produção antes do início do novo período de produção, pois a troca de turno dos funcionários permite essa preparação. Para esta empresa, deseja-se encontrar um plano de produção com no máximo um processo por período. Custos de estoque também não são levados em consideração, uma vez que a empresa considera que os produtos produzidos em excesso podem ser misturados de forma a gerar uma faixa granulométrica com demanda.

Em vista disso, o modelo MFP (Minimizar a Falta de Produção) consiste em encontrar um programa de produção que minimize a falta no atendimento à demanda dos produtos, e faz uso de variáveis de falta e excesso nas restrições de demanda. Esta função objetivo é consistente com o objetivo da empresa, e o modelo incorpora a restrição de que apenas um processo por período deva ser utilizado. Deve-se ressaltar que faltas não implicam em perda de vendas, mas no atraso em satisfazer as demandas. Ressalta-se também que o

modelo considera que o estoque inicial do produto é nulo; em casos que haja estoque inicial, este é subtraído da demanda (sem deixá-la negativa) antes da execução do modelo.

Especificamente, o modelo MFP procura minimizar a quantidade de itens demandados não produzidos no prazo. Para cada período em que se carrega a falta de um produto, a quantidade de falta é registrada em uma matriz Produto  $\times$  Período. Por exemplo, supondo que o produto EC31\_150 tenha demanda de 2.000 kg para o quinto período, e que a empresa só conseguirá produzir 1.500 kg até o final deste período, faltarão 500 kg para cumprimento da demanda solicitada. Supondo também que essa falta só seja satisfeita no décimo período, a falta desse produto incorrerá também nos períodos 6 a 9. À função objetivo, será então incrementado 500 kg em cada um dos cinco períodos (i.e, de 5 a 9) em que é constatada a falta do produto; tal programação pode ser vista na Figura 4.1(a).

O valor da função objetivo deste modelo é, portanto, igual a 2.500 kg caso haja apenas a falta do produto EC31\_150. Note que neste modelo, é preferível dever 1.000 kg de um produto por dois períodos consecutivos (totalizando 2.000 kg de falta), do que dever 500 kg por cinco períodos consecutivos, onde o total é de 2500 kg (veja a programação na Figura 4.1(b)). Isso é válido para todos os produtos em todos os períodos, de maneira que o valor da função objetivo corresponde à soma dos valores armazenados na matriz Produto  $\times$  Período.

EC31_150	Período									
	1	2	3	4	5	6	7	8	9	10
(F.O.: Min z) Z=	0	0	0	0	500	1000	1500	2000	2500	<b>2500</b>
Falta real	0	0	0	0	500	500	500	500	500	0
Demanda	-	-	-	-	2000	-	-	-	-	-

(a) caso 1

EC31_150	Período									
	1	2	3	4	5	6	7	8	9	10
(F.O.: Min z) Z=	0	0	0	0	1000	2000	2000	2000	2000	<b>2000</b>
Falta real	0	0	0	0	1000	1000	0	0	0	0
Demanda	-	-	-	-	2000	-	-	-	-	-

(b) caso 2

Figura 4.1 - Exemplos de programação da produção do item EC31\_150.

## Formulação matemática

Índices :

- $i$  item  $\{1, \dots, m\}$ ;  
 $j$  processo  $\{1, \dots, J\}$ ;  
 $t$  período de produção  $\{1, \dots, T\}$ ;

Variáveis:

- $x_{jt}$  1 se o processo  $j$  é usado no período  $t$ , 0 caso contrário ( $j = 1, \dots, J$ ;  $t = 1, \dots, T$ );  
 $I_{it}^-$  falta do produto  $i$  ao final do período  $t$  ( $i = 1, \dots, m$ ;  $t = 1, \dots, T$ );

Parâmetros:

- $m$  quantidade de produtos;  
 $J$  quantidade de processos de produção disponíveis;  
 $T$  horizonte de programação (em períodos de produção);  
 $a_{ij}$  quantidade do produto  $i$  produzido pelo processo  $j$  no período  $t$  ( $i = 1, \dots, m$ ;  $j = 1, \dots, J$ );  
 $d_{it}$  demanda do produto  $i$  no período  $t$  ( $i = 1, \dots, m$ ;  $t = 1, \dots, T$ );

$$\text{Min } z = \sum_{t=1}^T \sum_{i=1}^m I_{it}^- \quad (4.1)$$

s.a.

$$\sum_{t=1}^t \sum_{j=1}^J a_{ij} x_{jt} + I_{it}^- \geq \sum_{t=1}^t d_{it}, \quad \forall i, t \quad (4.2)$$

$$\sum_{j=1}^J x_{jt} \leq 1 \quad \forall t \quad (4.3)$$

$$x_{jt} \in \{0,1\} \quad \forall j, t \quad (4.4)$$

$$I_{it}^- \geq 0 \quad \forall i, t \quad (4.5)$$

A função objetivo (4.1) minimiza a falta de produção dos produtos que possuem demanda. As restrições de demanda (4.2) incluem variáveis de falta para cada produto  $i$  em cada período  $t$  (a quantidade total produzida de um produto até um dado período  $t$  mais a falta, deve ser maior ou igual à demanda acumulada daquele produto até o período  $t$ ).

Note que é possível reescrever (4.2) em sua forma clássica (restrições 4.2'). Entretanto, testes preliminares mostraram que a convergência do método é em média melhor com a utilização da primeira alternativa.

$$I_{it-1}^+ - I_{it-1}^- + \sum_{j=1}^J a_{ij} x_{jt} - I_{it}^+ + I_{it}^- = d_{it} \quad \forall i, t \quad (4.2')$$

As restrições (4.3) impõem que apenas um processo possa ser utilizado por período, e as restrições (4.4) e (4.5) estabelecem a integralidade das variáveis  $x_{jt}$  e a não negatividade das variáveis  $I_{it}^-$ . Note que o modelo [(4.1)-(4.5)] pode ser visto como um modelo de dimensionamento de lotes que, ao invés de utilizar “lotes de produtos”, utiliza “lotes de processos” para produzir um conjunto de produtos. A implementação do modelo na linguagem algébrica *GAMS* pode ser vista no Apêndice I.

Note também que uma solução factível com  $I_{it}^- = 0$  para todo  $i$  e todo  $t$  é ótima para o modelo MFP. Embora a preocupação principal do gerente de produção da empresa A seja a de minimizar a falta de produção, o modelo poderia ser alterado para também minimizar o excesso de produção (estoque entre períodos), de forma a contemplar situações em que os resultados apresentam falta igual a 0 (todas as datas de vencimento da demanda satisfeitas na programação). As restrições (4.2) são reescritas para descrever as restrições do modelo clássico DLSP com variáveis de excesso conforme (4.2'):

#### Modelo MFEP (Minimizar a Falta e o Excesso de Produção)

Variáveis adicionais:

$I_{it}^+$  excesso (estoque) do item  $i$  ao final do período  $t$  ( $i = 1, \dots, m; t = 1, \dots, T$ );

$$\text{Min } z = \sum_{t=1}^T \sum_{i=1}^m I_{it}^- + \lambda \sum_{t=1}^T \sum_{i=1}^m I_{it}^+ \quad (4.1')$$

$$I_{it-1}^+ - I_{it-1}^- + \sum_{j=1}^J a_{ij} x_{jt} - I_{it}^+ + I_{it}^- = d_{it} \quad \forall i, t \quad (4.2')$$

$$\sum_{j=1}^J x_{jt} \leq 1 \quad \forall t \quad (4.3)$$

$$x_{jt} \in \{0,1\} \quad \forall j, t \quad (4.4)$$

$$I_{it}^-, I_{it}^+ \geq 0 \quad \forall i, j, t \quad (4.5')$$

$$\text{com } I_{i0}^- = I_{i0}^+ = 0.$$

O parâmetro  $\lambda$  em (4.1') deve representar uma fração do excesso produzido a ser considerada na função objetivo, com valor suficientemente pequeno de forma a garantir que o primeiro objetivo (minimizar a falta de produção) domine o segundo (minimizar o excesso de produção). As restrições de demanda (4.2') incluem variáveis de folga (excesso de produção) e de falta para cada produto  $i$  em cada período  $t$ . Assim, a quantidade total produzida de um produto em um período  $t$  subtraído de sua falta ou somado ao seu excesso no período  $t-1$  e adicionado a sua falta ou subtraído de seu excesso no período  $t$  deve ser igual a sua demanda no período  $t$ .

Ainda para o caso de falta nula, o número de períodos necessários para a produção requerida também poderá ser otimizado. Neste último caso, o ideal é produzir o mais cedo possível. Embora seja possível adiar a produção em um dado período, isto não é feito devido à oportunidade de receber novas ordens e ser capaz de produzi-las durante o horizonte planejado. Ao adiantar a produção de alguns itens, a empresa pode maximizar a produtividade da planta, porém estará sujeita à geração de estoques maiores. Uma maneira de viabilizar essa estratégia no modelo é redefinir a função objetivo (4.1) como:

$$\text{Min } z = \sum_{t=1}^T \sum_{i=1}^m I_{it}^- + \lambda \sum_{t=1}^T \sum_{j=1}^J tx_{jt} \quad (4.1'')$$

Em (4.1'') o parâmetro  $\lambda$  é também utilizado para garantir que o primeiro objetivo domine o segundo. O modelo também pode ser reescrito de forma que o valor da função objetivo em  $z$  corresponda ao número de períodos necessários (ao invés de uma função de penalidade como em (4.1'')). Ou seja:

$$\text{Min } z = \sum_{t=1}^T \sum_{i=1}^m I_{it}^- + \lambda \sum_{t=1}^T \sum_{j=1}^J x_{jt} \quad (4.6)$$

s.a.



$$\sum_{t'=1}^t \sum_{j=1}^J a_{ij} x_{jt'} + I_{it}^- \geq \sum_{t'=1}^t d_{it'} \quad \forall i, t \quad (4.2)$$

$$\sum_{j=1}^J x_{jt} \leq 1 \quad \forall t \quad (4.3)$$

$$\sum_{j=1}^J x_{jt} \geq \sum_{j=1}^J x_{j,t+1} \quad \forall t, t \neq T \quad (4.7)$$

$$x_{jt} \in \{0,1\} \quad \forall j, t \quad (4.4)$$

$$I_{it}^- \geq 0 \quad \forall i, j, t \quad (4.5)$$

As restrições (4.7) asseguram uma ordem na atribuição das variáveis  $x_{jt}$ , de modo que o período  $t+1$  não é utilizado sem que o período  $t$  tenha sido. Neste caso, a solução utiliza sempre os primeiros períodos disponíveis para produção. Note que ao puxar a produção, são gerados maiores volumes de estoque entre períodos. Por essa razão o excesso de produção não é considerado na função objetivo, pois se o fosse os objetivos conflitariam.

Outras modificações poderiam ser aplicadas aos modelos a fim de investigar possíveis soluções práticas que os gerentes de produção da empresa A não vêm explorando, e ainda pouco utilizadas pela empresa B. Por exemplo, o modelo [(4.1)-(4.5)] também pode ser estendido para considerar custos de estoque e de *setup*, este último muito importante quando os turnos passam a ser de 24 horas, o que é comum em algumas épocas do ano. O capítulo 7 discute essas alternativas com outros quatro modelos baseados na literatura de *lot-sizing*.

No próximo capítulo são apresentados os métodos heurísticos desenvolvidos neste trabalho para resolver o problema em estudo.

## HEURÍSTICAS APLICADAS AO PROBLEMA

---

Problemas NP-difíceis são uma constante motivação para o desenvolvimento de técnicas e algoritmos capazes de resolvê-los. Entretanto, determinar uma solução ótima para estes problemas em tempo computacional razoável pode ser muito difícil ou até impossível. A satisfação das condições que atestam a otimalidade de uma dada solução pode requerer um grande esforço computacional (tempo e memória), mesmo em instâncias com um número moderado de variáveis de decisão e restrições.

De fato, modelos de otimização realistas podem ter milhares de variáveis e restrições, caso do problema da programação da produção em empresas de grãos eletrofundidos. Como observado nos experimentos computacionais (Capítulo 6), o tamanho da instância do problema impõe limitações à resolução exata dos modelos descritos no Capítulo 4.

Em função dessas limitações, algoritmos heurísticos foram desenvolvidos para tratamento de instâncias de maior porte ou como uma alternativa ao uso de métodos exatos. As heurísticas tiveram como foco a resolução dos modelos MFP e MFEP da seção 4.2., ou seja, mantendo as mesmas restrições e objetivo dos modelos. Inicialmente é apresentada uma heurística construtiva, seguida de uma heurística de busca local e um algoritmo GRASP (*Greedy Randomized Adaptive Search Procedures* - Procedimento Guloso Aleatório Adaptativo).

### 5.1. Uma heurística construtiva (HC)

Nesta abordagem, aqui denominada HC, uma solução é construída sequencialmente a partir do primeiro período do horizonte de planejamento. A escolha do

processo para um período  $t$  particular é guiada por uma função que avalia quão bem cada processo atende as datas de entrega da demanda de todos os produtos do período  $t$  ao período  $T$ , partindo da premissa que nenhuma produção ocorrerá do período  $t+1$  até  $T$ . Nesta seção,  $I_{it} < 0$  corresponde a falta e  $I_{it} > 0$  ao excesso do produto  $i$  no período  $t$ , portanto,  $I_{it} = I_{it}^+ - I_{it}^-$  com  $I_{it}^+ \geq 0$  e  $I_{it}^- \geq 0$ .

Seja  $d'_{it}$  a demanda devida do produto  $i$  no período  $t$ , isto é, a demanda do produto no período  $t$  subtraída de seu estoque no fim do período  $t-1$  ( $d'_{it} = d_{it} - I_{i,t-1}$ ) (note que  $I_{i,t-1} < 0$  corresponde à falta acumulada de períodos anteriores). Para cada processo  $j$  ( $j = 1..n$ ), a função de avaliação  $F_{jt}$  é dada por:

$$F_{jt} = \sum_{i=1}^m \sum_{t''=t}^T \text{Min} \left\{ \frac{(a_{ij} - d'_{it''})}{(t''-t+1)^v}, 0 \right\} \quad (5.1)$$

Note que  $F_{jt}$  compreende somente termos associados aos períodos com falta de produção ( $a_{ij} - d'_{it''} < 0$ ) e o impacto de cada termo na avaliação diminui conforme a distância de  $t''$  aumenta em relação ao período atual  $t$ . Este impacto é controlado pelo parâmetro  $v$  ( $v \geq 0$ ). Quanto maior é o valor de  $v$ , menor é a contribuição da falta em  $F_{jt}$ .

Os  $p$  processos melhor avaliados (conjunto  $P$ ) em um dado período  $t$  são selecionados para análise. Cada processo em  $P$  é usado como um processo provisório (processo candidato) para o período  $t$ . Como exemplo, suponha que o tamanho do conjunto  $P$  seja 2 e que os dois processos mais bem avaliados para  $t=1$  sejam os processos 2 e 3 (i.e.,  $P = \{2, 3\}$ ). Iniciando com o processo 2 como processo candidato para  $t=1$ , os estoques e as demandas devidas de todos os produtos de  $t=1$  até  $T$  são atualizados. A função de avaliação é então usada para determinar os processos nos períodos seguintes  $t+1$  até  $T$ , um período por vez. Para cada período, o processo melhor avaliado é selecionado. Continuando com o exemplo, suponha que atribuir o processo 3 ao período  $t=2$  seja a melhor opção para o atendimento das demandas dos períodos  $t''=2-5$  (de acordo com (5.1)), admitindo que não haja produção nos períodos  $t''=3-5$ . O processo 3 é, portanto, associado ao período  $t=2$ , e os estoques e a demanda devida são atualizados de  $t=2$  à  $T$ . Continuando de forma semelhante, os processos para  $t=4$  e  $t=5$  são selecionados.

Note que neste ponto, uma solução provisória completa de  $t=1$  até  $T$  é gerada, sendo o processo 2 associado a  $t=1$ . Uma segunda solução provisória é obtida, desta vez,

inicialmente atribuindo-se o outro processo em  $P$  (processo 3) a  $t=1$ . É fácil perceber que quanto maior o número de processos em  $P$ , maior é o número de soluções provisórias e, conseqüentemente, maior é o esforço computacional requerido.

A solução provisória que apresentar a menor falta de produção define o processo a ser utilizado em  $t$ . Os estoques e as demandas devidas do período  $t$  à  $T$  são atualizadas de acordo com esta decisão, e os processos atribuídos aos períodos  $t+1$  a  $T$  na solução provisória correspondente são desconsiderados. Todo o procedimento é então repetido para o período  $t+1$  a fim de determinar o processo a ele atribuído e, continuando de forma similar, para cada período restante do horizonte até que seja obtida uma solução completa. Uma descrição mais formal da heurística é apresentada pelos seguintes passos na Figura 5.1.

1. Faça  $d'_{ii} = d_{ii}$  de cada produto  $i$  em cada período  $t$  ( $i=1..m; t=1..T$ ).
2. Para  $t1=1$  até  $T$ :
  - 2.1. Calcule  $F_{j,t1}$  (equação (5.1)) para cada processo  $j$  ( $j = 1..J$ ).
  - 2.2. Faça  $P =$  lista de  $p$  processos com as melhores avaliações.
  - 2.3. Para cada  $r \in P$ :
    - 2.3.1. Faça  $processo\_candidato(t1) = r$ .
    - 2.3.2. Para todos os produtos  $i$  ( $i=1..m$ ), atualize (provisoriamente) os estoques  $I_{i,t1}$  no período  $t1$  e as demandas devidas  $d'_{ii}$  em todos os períodos restantes  $t1+1$  até  $T$ .
    - 2.3.3. Para  $t2=t1+1$  até  $T$ :
      - 2.3.3.1. Calcule  $F_{j,t2}$  para cada processo  $j$  ( $j = 1..J$ ). Seja  $q$  o processo com melhor avaliação.
      - 2.3.3.2. Faça  $processo\_candidato(t2)=q$ .
      - 2.3.3.3. Para todos os produtos  $i$  ( $i=1..m$ ), atualize (provisoriamente) os estoques  $I_{i,t2}$  no período  $t2$  e as demandas devidas  $d'_{ii}$  em todos os períodos restantes  $t2+1$  até  $T$ .
    - 2.3.4. Seja  $f'_{r,t1,T}$  a falta de todos os produtos do período  $t1$  à  $T$  obtido com  $processo\_candidato(t1) = r$ .
  - 2.4. Atribua o processo  $j'$  para o qual  $f'_{r,t1,T}$  é o mínimo ( $j' = \operatorname{argmin}_{r \in P} \{f'_{r,t1,T}\}$ ) ao período  $t1$ . Para cada produto  $i$  ( $i=1..m$ ), atualize os estoques  $I_{i,t1}$  no período  $t1$  e as demandas devidas  $d'_{ii}$  em todos os períodos restantes  $t1+1$  à  $T$ .
3. Retorne a solução  $S$  resultante.

Figura 5.1. - Passos da heurística HC.

Cabe ressaltar que o algoritmo descrito na Figura 5.1 tem complexidade da ordem de  $O(J^3)$ , onde  $J$  é o número de processos. A fim de diminuir o tempo computacional dispendido, a função de avaliação (5.1) pode ser reescrita para considerar um número limitado de períodos à frente a serem considerados. Especificamente:

Novo Parâmetro:

$maxTF$  = quantidade máxima de períodos considerados à frente;

$$F_{jt} = \sum_{i=1}^m \sum_{t''=t}^{Min(t+maxTF, T)} Min \left\{ \frac{(a_{ij} - d'_{it'')}}{(t''-t+1)^v}, 0 \right\} \quad (5.1')$$

Note que um número  $maxTF$  de períodos à frente do período  $t$  reduz o tempo dispendido para o cálculo da função  $F_{jt}$ . Como  $F_{jt}$  é calculada muitas vezes durante a aplicação da heurística, esta redução pode afetar de forma significativa o tempo total da construção da solução, sendo portanto indicada para instâncias com grande número de períodos. Entretanto, para que a qualidade da solução não se deteriore, o valor de  $maxTF$  deve marcar um ponto de corte no horizonte de planejamento a partir do qual a influência da demanda é pouco relevante para a escolha do processo no período  $t$ .

## 5.2. Uma heurística de busca local ( $k$ -opt)

O procedimento de busca local  $k$ -opt parte da solução gerada pela heurística HC (solução  $S$ ), e a cada iteração gera uma nova solução (desde que esta implique em melhorias em relação a solução corrente) por meio de operações de substituição dos processos correntemente designados aos períodos. A vizinhança  $N(S)$  de uma dada solução corrente  $S$  consiste das soluções obtidas com a substituição da *sequência de processos* designados em  $S$  do período  $t$  até o período  $t + k - 1$  (onde  $k \leq T - t + 1$ ) por uma sequência de processos diversa, onde  $k$  é um parâmetro de entrada (veja a ilustração na Figura 5.3). Todas as possíveis sequências de processos nestes  $k$  períodos são testadas, e aquela que melhor substitui os processos retirados ( $S'$ ) passa a ser a nova solução corrente  $S$  se apresentar um valor de falta menor que o de  $S$ . Caso contrário,  $S$  é um ótimo local e o procedimento é finalizado. A Figura 5.2 sumariza os passos da heurística  $k$ -opt.

1. Aplique a heurística HC (descrita na Figura 5.1), obtendo uma solução inicial  $S$ .  
Faça  $\text{nova\_iteração} = \text{verdadeiro}$ .
2. Enquanto  $\text{nova\_iteração} = \text{verdadeiro}$ :
  - 2.1. Para  $t = 1$  até  $(T - k + 1)$ :
    - 2.1.1. Gere e avalie a vizinhança  $N(S)$  de  $S$  resultante da substituição da sequência de processos designados em  $S$  do período  $t$  até o período  $t + k - 1$  por uma sequência de processos diversa.
    - 2.1.2. Selecione a solução  $S'$  de  $N(S)$  tal que  $\text{falta}(S') \leq \text{falta}(S'')$ ,  $\forall S'' \in N(S)$ ;
    - 2.1.3. Se  $\text{falta}(S') < \text{falta}(S)$ , faça  $S = S'$  e  $\text{nova\_iteração} = \text{verdadeiro}$ . Caso contrário, faça  $\text{nova\_iteração} = \text{falso}$ .
3. Retorne  $S$ .

Figura 5.2. – Passos da heurística  $k$ -opt.

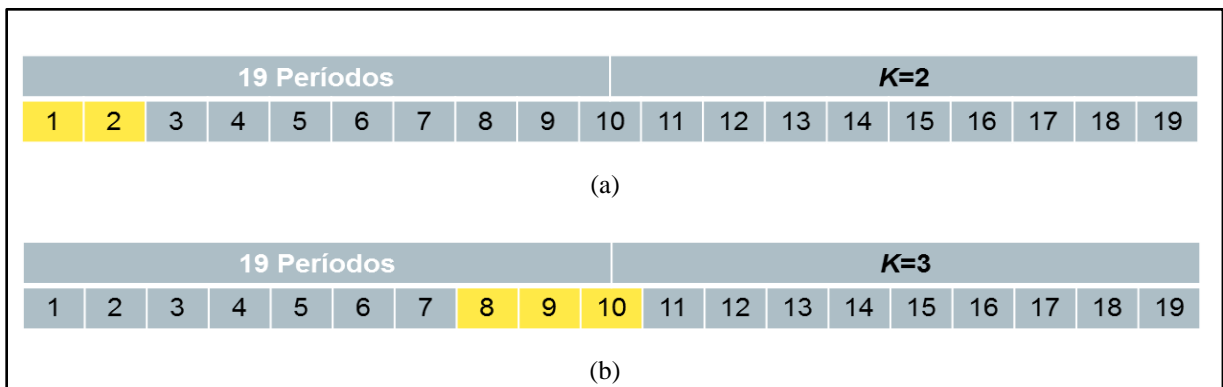


Figura 5.3. – Exemplos de vizinhança da heurística  $k$ -opt com (a)  $k=2$  e (b)  $k=3$ .

Quanto maior for o valor de  $k$ , maior qualidade terá o ótimo local produzido. Note, porém, que a busca local gera e avalia  $(T - k)J^k$  soluções vizinhas a cada iteração. Tomando como exemplo  $k=2$ ,  $J=159$  e  $T=19$ , este número corresponde a  $(19 - 2)159^2 = 429.777$  soluções vizinhas. Para  $k = 3$ , a vizinhança consiste de 64.314.864 soluções, cujo tempo de geração e de avaliação do valor de falta torna proibitivo o esforço computacional da busca.

Uma alternativa para que possam ser usados valores maiores em  $k$  é encontrar um intervalo de  $k$  períodos cuja substituição de processos seja potencialmente promissora, e limitar a geração da vizinhança a apenas este intervalo. Uma opção natural é selecionar os  $k - 1$  períodos que antecedem o período com maior falta acumulada de produção. Seja  $t$  o período com maior falta acumulada. Para  $k=3$  são investigadas todas as possíveis substituições dos processos dos períodos  $t - 2$ ,  $t - 1$  e  $t$ , gerando soluções vizinhas à partir de todas as combinações possíveis de trincas com os processos disponíveis. Neste caso, uma vizinhança parcial  $N_p(S)$  com apenas  $(J^k)$  soluções vizinhas é gerada. Tem-se, assim, a heurística  $k$ -opt com vizinhança parcial, cujos passos são apresentados na Figura 5.4.

1. Aplique a heurística HC (descrita na Figura 6.1), obtendo uma solução inicial  $S$ . Faça  $nova\_iteração = verdadeiro$ .
2. Enquanto  $nova\_iteração = verdadeiro$ :
  - 2.1. Obtenha o período  $t'$  para o qual  $\sum_{i=1}^m f_{it}$  é máximo ( $t' = arg \text{Max}_{t \leq T} \{ \sum_{i=1}^m f_{it} \}$ )
  - 2.2. Gere e avalie a vizinhança  $N_p(S)$  de  $S$  resultante da substituição da sequência de processos designados em  $S$  do período  $t' - k + 1$  até o período  $t'$  por uma sequência de processos diversa.
  - 2.3. Selecione a solução  $S'$  de  $N_p(S)$  tal que  $falta(S') \leq falta(S'')$ ,  $\forall S'' \in N_p(S)$ .
  - 2.4. Se  $falta(S') < falta(S)$ , faça  $S = S'$  e  $nova\_iteração = verdadeiro$ . Caso contrário, faça  $nova\_iteração = falso$ .
3. Retorne  $S$ .

Figura 5.4. – Passos da heurística  $k$ -opt com vizinhança parcial.

### 5.2.1. Pré-processamento

A fim de reduzir ainda mais o tempo computacional de execução de ambas as versões de  $k$ -opt sem deteriorar muito a qualidade das soluções obtidas, é criada uma lista limitada de processos elegíveis para substituir os processos correntes. A lista inclui apenas os  $n$  processos que mais apareceram no conjunto  $P$  (processos candidatos) ao longo da aplicação da heurística HC. Esse pré-processamento elimina processos que o histórico da construção da

solução inicial considerou pouco atraentes. Assim,  $k$ -opt e  $k$ -opt com vizinhança parcial geram respectivamente  $(T-k)n^k$  e  $n^k$  soluções vizinhas, onde  $n \leq J$ .

Heurísticas de busca local como  $k$ -opt fornecem soluções localmente ótimas, ou seja, soluções ótimas em uma região bastante limitada do espaço, e definida pelo escopo da vizinhança adotada e da solução inicial. A fim de obter soluções de melhor qualidade, foi investigada a utilização de meta-heurísticas, uma vez que estas possuem mecanismos que permitem a continuidade às explorações para além de ótimos locais. Em particular, foram considerados algoritmos genéticos (Holland, 1975) e a metaheurística GRASP (Feo & Resende, 1989). Algoritmos genéticos geram populações de soluções, o que confere um paralelismo intrínseco à busca de soluções. A escolha do GRASP, por sua vez, deve-se à rapidez em gerar soluções e a facilidade com que a heurística construtiva HC pode ser incorporada ao algoritmo.

A próxima seção está restrita à descrição do algoritmo GRASP proposto neste trabalho, uma vez que as implementações de algoritmos genéticos desenvolvidas não apresentaram resultados satisfatórios para o problema tratado.

### 5.3. Um algoritmo GRASP

GRASP é uma meta-heurística probabilística cuja aplicação consiste na repetição de duas fases. Na primeira fase (passo 2.1 da Figura 5.5), uma solução é construída com base em uma função de avaliação gulosa, ou seja, escolhendo o próximo componente da solução com base no benefício local que este traz. A diferença primordial em relação a uma heurística de construção tradicional é que, uma vez identificados os componentes candidatos mais atraentes com a função de avaliação gulosa e chamando-se este conjunto de LRC (Lista Restrita de Candidatos), um dos candidatos é selecionado de forma aleatória para inclusão na solução parcial. Ao final de cada inclusão, todos os componentes candidatos à inserção são reavaliados em decorrência do último elemento inserido. O tamanho de LRC é controlado por parâmetros que definem os componentes elegíveis com base em sua atratividade relativa para o objetivo do problema. Por exemplo, o componente da solução é eleito dentre os  $k$  primeiros, onde  $k$  é um valor especificado pelo usuário. Outra possível opção é eleger o componente dentre aqueles que se encontram dentro de uma faixa de valor  $[(1-\alpha)*f_{\max}, f_{\max}]$ , onde  $f_{\max}$  é



o valor do melhor componente e  $\alpha \in [0,1]$  é definido pelo usuário. Note que  $\alpha = 0$  implica em uma lista de tamanho 1, de forma que o algoritmo se comportará de forma puramente gulosa. Se  $\alpha = 1$ , a lista é composta por todos os componentes e a seleção é puramente aleatória. A segunda fase destes algoritmos (passo 2.2 da Figura 5.5) consiste na aplicação de melhorias a partir da solução inicial, geralmente por meio de buscas locais ou aplicação de outra meta-heurística. A aplicação das fases 1 e 2 constituem uma iteração GRASP.

1. Faça  $S^* = \emptyset$ ,  $f(S^*) = \infty$ , e  $it$  (número de iterações GRASP)=0.
2. Enquanto  $it < itmax$  (número máximo de iterações):
  - 2.1. (Fase 1) Gere aleatoriamente uma solução  $S_0$  com base na LRC corrente.
  - 2.2. (Fase 2) Aplique o procedimento de melhoria a  $S_0$ , obtendo a solução  $S$ . Se  $f(S) < f(S^*)$ , faça  $S^* = S$ .
  - 2.3. Faça  $it = it + 1$ .
3. Retorne  $S^*$ .

Figura 5.5. – Passos gerais de um algoritmo GRASP para um problema de minimização.

As próximas seções discutem a implementação das duas fases do algoritmo proposto para resolução do problema de planejamento da produção tratados neste trabalho.

### 5.3.1. Fase de construção

Assim como em HC, a solução é construída iterativamente (Figura 5.6). A função de avaliação (5.1) é utilizada para medir o benefício da atribuição do processo  $j$  ao período  $t$ , e um processo  $j$  é atribuído a cada período  $t$  por vez. A seleção do processo  $j$  para o período  $t$  é aleatória, porém considera apenas um subconjunto de processos com melhor avaliação armazenados em uma lista restrita de candidatos (LRC). O tamanho de LRC é controlado pelo parâmetro  $p$  discutido na descrição da heurística HC (seção 5.1). Note que se  $p=1$ , têm-se um comportamento puramente guloso e para  $p=J$ , um comportamento puramente aleatório. A escolha aleatória dentre processos com as melhores avaliações permite que soluções distintas, porém de boa qualidade, sejam geradas. Note que a fase de construção é igual a HC retirando-se os passos 2.3 a 2.3.4 da Figura 5.1 e modificando o passo 2.4 de forma que o processo

atribuído a um dado período  $t$  seja selecionado aleatoriamente de LRC. Por esta razão, a fase de construção foi denominada de heurística HC modificada.

1. Faça  $d'_{it} = d_{it}$  de cada produto  $i$  em cada período  $t$  ( $i=1..m$ ;  $t=1..T$ ).
2. Para  $t1=1$  até  $T$ :
  - 2.1. Calcule  $F_{j,t1}$  (equação (5.1)) para cada processo  $j$  ( $j = 1..J$ ).
  - 2.2. Faça LRC = lista de  $p$  processos com as melhores avaliações.
  - 2.3. Atribua aleatoriamente um processo  $p$  de LRC ao período  $t1$ . Para cada produto  $i$  ( $i=1..m$ ), atualize os estoques  $I_{i,t1}$  no período  $t1$  e as demandas devidas  $d'_{it}$  em todos os períodos restantes  $t1+1$  à  $T$ .
3. Retorne a solução  $S$  resultante.

Figura 5.6. - Passos da fase de construção do algoritmo GRASP (heurística HC modificada).

A média e a variância do valor das soluções construídas são diretamente afetadas pelo valor de  $p$ . Se  $p$  é pequeno, menor a variância, menor o espaço de soluções percorrido e maior a chance de aprisionar a busca local em um ótimo local pobre. Se  $p$  é grande, maior a variância, menor a possibilidade de o procedimento resultar em um ótimo local pobre, porém, maior é o tamanho da vizinhança explorada. Para a aplicação eficaz do método, é necessária, portanto, a definição de um intervalo de valores para  $p$  de forma a balancear a relação entre qualidade das soluções, quantidade de iterações necessárias e tamanho da vizinhança explorada.

Note que como HC, a heurística é adaptativa porque os benefícios associados com cada processo  $j$  em  $t$  são atualizados a cada iteração da fase de construção para refletir as mudanças ocorridas com a seleção dos  $t-1$  processos atribuídos anteriormente. O componente aleatório do método corresponde à maneira como o processo  $j$  é selecionado dentre os processos em LRC a cada  $t$ .

### 5.3.2. Fase de melhoria

Uma vez construída uma solução inicial para o problema, esta é submetida à aplicação de buscas locais. Para isso, foi utilizada a heurística  $k$ -opt (discutida na seção 5.2)

com diferentes valores para  $k$ . A partir da solução resultante (ótimo local), opcionalmente pode-se reaplicar  $k$ -opt, desta vez, com valor de  $k$  maior que o aplicado na chamada anterior.

Os passos do algoritmo GRASP são apresentados na Figura 5.7.

1. Faça  $S^* = \emptyset$ ,  $falta(S^*) = \infty$  e  $it = 0$ .
2. Enquanto  $it \leq itmax$  (número máximo de iterações):
  - 2.1. Aplique a heurística de construção HC modificada (descrita na Figura 5.6),
  - 2.2. Aplique a heurística de busca local  $k$ -opt a partir da solução do passo anterior, obtendo a solução  $S$ . Se  $falta(S) < falta(S^*)$ , faça  $S^* = S$ .
  - 2.3. Faça  $it=it + 1$ .
3. (Passo opcional) Aplique a heurística de busca local  $k$ -opt a partir da solução  $S^*$  com novo  $k$  maior que o valor utilizado no passo 2.2. Faça  $S^*$  igual à solução resultante.
4. Retorne  $S^*$ .

Figura 5.7 – Passos do algoritmo GRASP.

#### 5.4. Minimização da falta e do excesso (estoque)

O modelo MFEP é revisitado nesta seção, com o propósito de se obter soluções que otimizem não só a falta mas também o excesso de produção. Para tal, é utilizada a função objetivo (4.1'), apresentada na seção 4.2:

$$\text{Min } z = \sum_{t=1}^T \sum_{i=1}^m I_{it}^- + \lambda \sum_{t=1}^T \sum_{i=1}^m I_{it}^+ \quad (4.1')$$

O peso do excesso de produção na função objetivo (parâmetro  $\lambda$ ) deve ser estipulado de maneira que a falta acumulada se mantenha como objetivo principal. Assim como no modelo, as heurísticas levam em conta o excesso com peso  $\lambda$  e a função de avaliação (5.1) passa a ser computada pela seguinte formula:

$$F_{jt} = \sum_{i=1}^m \sum_{t''=t}^T \text{Min} \left\{ \frac{(a_{ij} - d'_{it''})}{(t''-t+1)^v}, 0 \right\} + \text{Max} \left\{ \frac{(a_{ij} - d'_{it''})}{(t''-t+1)^v}, 0 \right\} * \lambda \quad (5.1'')$$

onde,

$$\text{Min} \left\{ \frac{(a_{ij} - d'_{it''})}{(t'' - t + 1)^v}, 0 \right\} \quad \text{representa a falta e}$$

$$\text{Max} \left\{ \frac{(a_{ij} - d'_{it''})}{(t'' - t + 1)^v}, 0 \right\} * \lambda \quad \text{representa o excesso com peso } \lambda.$$

A função de avaliação com número de períodos à frente limitado (5.1') também é modificada de forma similar:

$$F_{jt} = \sum_{i=1}^m \sum_{t''=t}^{\text{Min}(t+\text{max } TF, T)} \text{Min} \left\{ \frac{(a_{ij} - d'_{it''})}{(t'' - t + 1)^v}, 0 \right\} + \text{Max} \left\{ \frac{(a_{ij} - d'_{it''})}{(t'' - t + 1)^v}, 0 \right\} * \lambda \quad (5.1''')$$

No próximo capítulo são apresentados os experimentos computacionais realizados para resolução do modelo MFP e MFEP (descritos no capítulo 4), tanto de forma exata como pelas heurísticas propostas.

**RESULTADOS COMPUTACIONAIS**

Neste capítulo, os experimentos com a resolução exata do modelo MFP fizeram uso da linguagem de modelagem Gams (BROOKE *et al.*, 1992) com o *solver* de programação inteira mista CPLEX (ILOG, 2009) baseado no método *branch-and-cut* (Belvaux & Wolsey, 2000). As heurísticas, por sua vez, foram codificadas em linguagem Pascal orientada a objetos Delphi 7.0 (Borland). Ambos os tipos de método foram aplicados a uma instância com dados reais coletados da empresa Treibacher, assim como os conjuntos de instâncias geradas aleatoriamente a partir dos dados reais.

Nas seções 6.1 e 6.2, os resultados da resolução do modelo são uma transcrição do que é reportado em Luche *et al.* (2009). No artigo, os experimentos se restringem à aplicação do método exato e da heurística construtiva HC, tendo sido realizados em um computador Pentium IV 3.0 GHz com 2.0 Gb RAM. O modelo foi resolvido com a versão 7 do CPLEX com parâmetros padrão, *gap* zero de otimalidade e tempo de execução limitado a três horas (10.800 segundos). A aplicação da heurística HC, por sua vez, envolveu múltiplas execuções, cada qual com um par diferente de valores dos parâmetros  $v$  e  $p$  utilizados na função de avaliação (5.1). Estes valores foram definidos inicialmente como  $v=\{2,3,4\}$  e  $p=\{3,4,5,6\}$ . Ou seja, a heurística retorna a melhor solução obtida nas  $3 \times 4 = 12$  execuções. Os valores de  $v$  e  $p$  foram então estendidos ( $HC_{ext}$ ) para  $v=\{2,3,4,5\}$  e  $p=\{3,4,5,6,7,8\}$ , sendo retornada a melhor solução obtida nas  $4 \times 6 = 24$  execuções. Em ambas as combinações de variações paramétricas (HC e  $HC_{ext}$ ), o tempo de obtenção da solução é a soma dos tempos computacionais das execuções realizadas.

Na seção 6.3, os mesmos experimentos são realizados desta vez sob um diferente ambiente computacional, e incluindo os resultados da aplicação das variações de heurísticas de busca local e do algoritmo GRASP. Resultados com o modelo obtidos com processamento

paralelo são apresentados na seção 6.4. Por fim, na seção 6.5 são apresentados os resultados das variações do modelo e das heurísticas visando minimização da falta e do excesso.

### 6.1. Experimentos com a instância real

A instância real é caracterizada por  $J=159$  processos de produção disponíveis para a programação de  $m=50$  itens em  $T=19$  períodos (dias) de produção. Segundo informações da empresa, a programação realizada pelo PCP da empresa resultou em uma falta total de 13.450 kg. Apesar de não haver registros do tempo requerido para a obtenção desta solução, em geral os programadores da empresa necessitam de várias horas (até mesmo dias) para encontrar um programa de produção satisfatório para uma dada carteira de pedidos (Luche *et al.*, 2009).

A heurística construtiva HC, por outro lado, encontrou uma solução com falta total de 12.100 kg, obtida em 9,5 segundos de execução. Com a busca local (combinação  $BL_E$  definida no Quadro 6.1 com  $k=2$ , pré-processamento, vizinhança limitada em  $\{5..19\}$  períodos e 15 soluções geradas), obteve-se uma falta de 11.800 kg em 904 segundos, enquanto que o GRASP (combinação 9 definida no Quadro 6.3 com  $p=\{3..8\}$ ,  $v=\{2,3,4\}$ ,  $maxTF=11$ ,  $itmax=20$  e  $k=\{2,3\}$ ) obteve uma solução com falta de 10.475 kg em 1.281 segundos. A solução obtida com a resolução exata do modelo MFP (com 3.021 variáveis inteiras) apresentou falta total de 10.475 kg em um tempo computacional de 26 segundos. As soluções apresentadas pelo GRASP e pela resolução exata do modelo MFP representam uma redução de 22% em relação ao valor da falta na solução da empresa, o que pode ser considerado bastante significativo. O resumo desses resultados é apresentado na Tabela 6.1.

Tabela 6.1 - Resultados obtidos com uma instância real.

<i>Instância Real</i>	<i>Falta</i>	<i>Tempo</i>
Empresa	13.450	várias horas
Modelo MFP (CPLEX)	10.475	26 s
Heurística Construtiva (HC)	12.100	9,5 s
Busca Local ( <i>k-opt</i> )	11.800	904 s
GRASP	10.475	1.281 s

## 6.2. Experimentos com instâncias geradas aleatoriamente

A construção de conjuntos de instâncias geradas aleatoriamente a partir do exemplo real teve como objetivo verificar como o desempenho dos métodos é afetado ao se (i) incrementar ou decrementar a demanda em porcentagens determinadas (variação de demanda), e ao se (ii) incrementar o número de períodos de produção (variação de períodos). Para cada um dos testes foram gerados conjuntos de dez instâncias cada.

Os resultados médios com cada conjunto de instâncias nos testes de variação de demanda e do número de períodos são apresentados, respectivamente, nas Tabelas 6.2 e 6.3. Em cada tabela, a terceira coluna apresenta a porcentagem de soluções com certificados de otimalidade (CO), ou seja, a porcentagem do total de instâncias em que foi comprovada a otimalidade das soluções resultantes. Na quarta coluna, encontra-se a média dos desvios percentuais máximos da otimalidade (*Gap*) encontradas com o GAMS/CPLEX. A sexta e a oitava coluna fornecem a média dos desvios percentuais da solução da heurística HC e de sua aplicação com variação paramétrica estendida ( $HC_{ext}$ ) em relação aos resultados com o modelo. A quinta, a sétima e a nona coluna apresentam o tempo computacional médio (em segundos) utilizado pelos métodos em cada conjunto de instâncias.

### 6.2.1. Variação de demanda

A Tabela 6.2 apresenta os resultados dos experimentos para 11 conjuntos de 10 exemplos (totalizando 110 exemplos) gerados aleatoriamente a partir do exemplo da empresa a fim de considerar mudanças nas demandas originais. Cada exemplo contém os mesmos  $m=50$  produtos,  $J=159$  processos e  $T=19$  períodos (dias) do exemplo fornecido, totalizando 3.021 variáveis inteiras no modelo MFP.

A geração das instâncias se inicia com o conjunto S (*standard*). As instâncias deste conjunto foram obtidas somando-se inicialmente a quantidade total demandada para cada produto da instância real em todos os períodos, ou seja, obtendo-se a demanda total de cada produto no horizonte de planejamento. A demanda total de cada produto foi então redistribuída escolhendo aleatoriamente cada período a ter demanda do item, assim como a quantidade requerida do mesmo. Para evitar períodos com demanda muito pequena por um dado item, se a quantidade atribuída a um período  $t$  fosse menor que um valor pré-definido,

esta quantidade era transferida para qualquer outro período com demanda já atribuída àquele item.

No gráfico da Figura 6.1 pode ser observada a dispersão da demanda nos 19 períodos (em quilos), onde cada linha do gráfico representa a quantidade total dos itens com demanda (eixo Y) em cada período (eixo X) para cada uma das dez instâncias do conjunto S. No gráfico da Figura 6.2 observa-se a média da demanda dos dez conjuntos de instâncias em cada período. Nos conjuntos I10-I50, as demandas dos produtos nas instâncias do conjunto S foram incrementadas em 10, 20, 30, 40 e 50%, respectivamente. A variação da demanda ocorre de forma similar para os conjuntos D10-D50, onde as demandas do conjunto S são subtraídas em 10, 20, 30, 40 e 50% de seu valor, respectivamente.

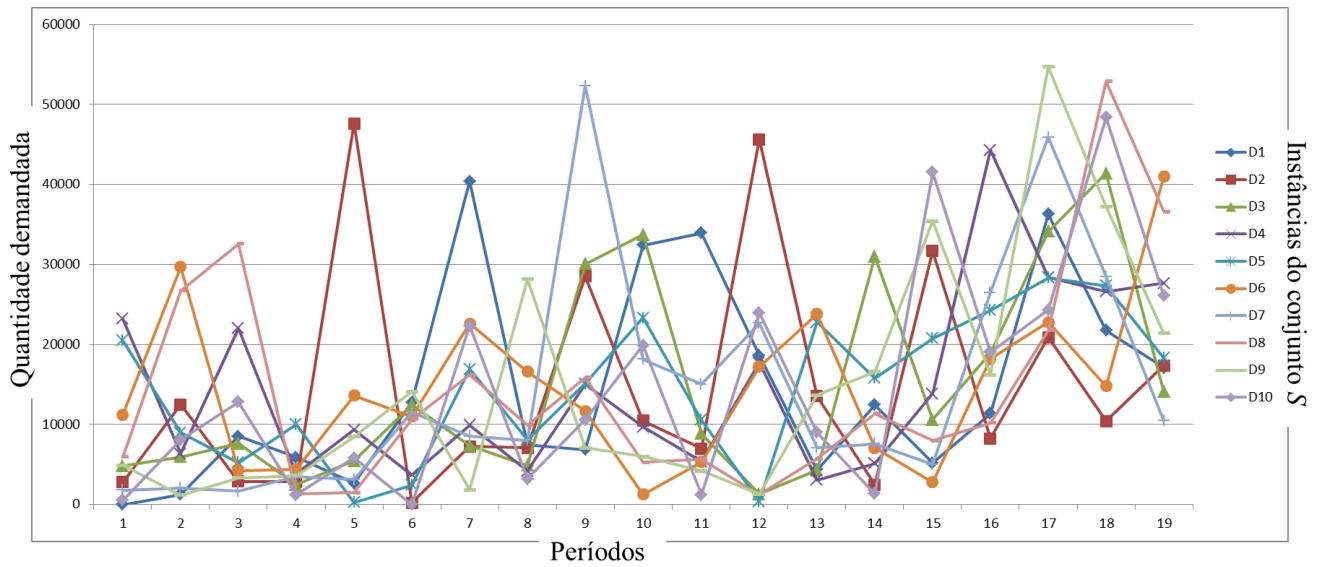


Figura 6.1 - Demanda (em quilos) por período para produção em cada instância.

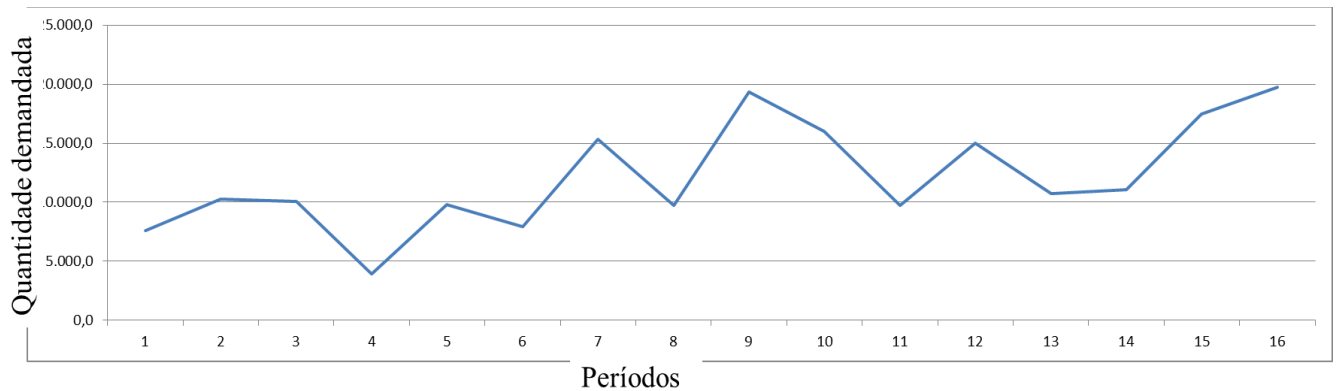


Figura 6.2 – Média da demanda (em quilos) das instâncias por período para produção.



Note que ao se reduzir a demanda dos produtos (conjuntos D10-D50), os exemplos se tornam, em média, mais fáceis de serem resolvidos otimamente pelo modelo com o CPLEX (tempos computacionais decrescem com a porcentagem de redução). Contrariamente, ao se aumentar a demanda dos produtos (conjuntos I10-I50), a resolução dos exemplos se torna mais difícil. De fato, a otimalidade de grande parte dos exemplos destes conjuntos não pode ser provada dentro do tempo máximo estipulado de 10.800 segundos.

Tabela 6.2 - Resultados com variação de demanda: resolução do modelo e aplicação da heurística HC versão padrão e estendida.

Conjunto	Variação de demanda (%)	Modelo MFP			HC		HC <sub>ext</sub>	
		CO (%)	Gap (%)	Tempo (s)	DPM (%)	Tempo (s)	DPM (%)	Tempo (s)
D50	-50	100	0	17	1,8	60	0,8	145
D40	-40	100	0	36	5,7	60	5,4	146
D30	-30	100	0	87	15,3	60	13,9	147
D20	-20	100	0	706	12,7	63	11,2	150
D10	-10	90	0,1	1.851	4,2	62	4,2	153
S	0	90	0,1	2.153	16,7	64	15,7	156
I10	+10	60	2,4	6.375	10,9	67	10,4	163
I20	+20	60	5,6	5.995	10,4	72	9,6	174
I30	+30	30	5,5	8.617	7,3	75	5,5	178
I40	+40	20	3,9	8.825	5,5	73	5,3	177
I50	+50	50	1,8	6.529	6,1	74	4,6	187
Média			1,8	3.745	8,8	66	7,9	161

Embora o tempo médio do modelo (3.745 s) tenha sido muito superior ao requerido pelas heurísticas HC (66 s) e HC<sub>ext</sub> (161 s), os resultados em termos da qualidade da solução foram superiores. O modelo obteve um *gap* de otimalidade de 1,8% (o que não implica que as soluções obtidas não sejam ótimas), ao passo que as heurísticas HC e HC<sub>ext</sub> obtiveram desvios percentuais em relação ao modelo de 8,8 e 7,9%, respectivamente. Levando em conta que se trata de heurísticas construtivas, os resultados das mesmas podem ser considerados de boa qualidade.

### 6.2.2. Variação do número de períodos

Na Tabela 6.3 são apresentados os resultados obtidos com 4 conjuntos de 10 exemplos (totalizando 40 exemplos) gerados aleatoriamente com base nas instâncias do conjunto S, discutido na seção anterior. Para cada instância de S, o número de períodos foi multiplicado por 2, 3, 4 e 5, resultando, respectivamente, nos conjuntos T2, T3, T4 e T5. A demanda do horizonte de planejamento original é então repetida exatamente em cada bloco de 19 períodos. Os conjuntos T2, T3, T4 e T5 totalizam no modelo MFP, um número de variáveis inteiras igual a 6.042, 9.063, 12.084 e 15.105, respectivamente.

Tabela 6.3 - Resultados com variação do número de períodos: resolução do modelo e aplicação da heurística HC versão padrão e estendida.

Conjunto	Variação de períodos (×)	Modelo MFP			HC		HC <sub>ext</sub>	
		CO (%)	Gap (%)	Tempo (s)	DPM (%)	Tempo (s)	DPM (%)	Tempo (s)
T2	2	50	3,5	6.313	19,3	308	18,4	796
T3	3	10	16,4	9.809	6,7	1.102	5,2	2.735
T4	4	10	20,3	10.697	2,5	2.173	1,2	5.572
T5	5	0	31,2	10.800	-9,4	3.821	-10,5	9.680
Média			17,8	9.405	4,8	1.851	3,6	4.696

Conforme esperado, os resultados mostram que se torna mais difícil resolver otimamente o modelo MFP por meio do CPLEX conforme o número de períodos do horizonte de planejamento cresce. Isso é verificado pela redução substancial de certificados de otimalidade e o aumento do *gap* (atingindo 31,2% no conjunto T5) e o aumento do tempo computacional. Estes resultados ilustram como a complexidade aumenta com o tamanho da instância. Resultados semelhantes são esperados ao se aumentar o número de processos disponíveis para produção.

Para estes conjuntos de instâncias, a heurística HC apresenta um desvio percentual médio de 4,8% em relação ao modelo. O desempenho inferior no conjunto T2 (DP = 19,3%) é compensado pelo bom desempenho no conjunto T5 ao alcançar um desvio percentual de -9,4%. HC<sub>ext</sub>, como esperado, supera HC com um desvio percentual de apenas 3,6% em relação ao modelo. Note que HC<sub>ext</sub> requer um tempo computacional mais de duas vezes superior ao tempo de HC e duas vezes inferior ao tempo do modelo. Estes resultados

revelam claramente que na medida que o horizonte de planejamento cresce, a qualidade das soluções providas pelo modelo se deterioram a uma taxa maior que aquelas providas pelas heurísticas.

### 6.3. Experimentos em um ambiente computacional mais eficiente

Acompanhando a evolução tecnológica do *software* e *hardware* ao longo deste projeto, tornou-se interessante investigar possíveis mudanças no desempenho do *solver* CPLEX versão 7 (discutido nas seções 6.1 e 6.2) com a disponibilização de sua versão 11, ambos aplicados em um computador mais potente. A versão 11 do CPLEX traz melhorias no método *branch-and-cut* utilizado para programação inteira mista (característica do problema tratado neste trabalho) e também dá suporte à programação paralela, esta última discutida na seção 6.4.

A Tabela 6.4 apresenta os resultados obtidos nos experimentos de variação de demanda com ambas as versões do CPLEX. Os resultados da heurística HC e sua versão estendida são também incluídos a fim de se observar mudanças em seu desempenho relativo frente ao CPLEX 11. Os experimentos foram, desta vez, realizados em um computador equipado com processador Intel Core2Quad com 4 Gb de RAM e sistema operacional Windows XP. Como esta máquina é consideravelmente mais rápida que a utilizada na obtenção dos resultados da Tabela 6.2, diferenças em tempo computacional necessário para resolver as instâncias são observadas.

Os resultados das colunas 2-4 da Tabela 6.4 indicam que a utilização da nova máquina permitiu ao CPLEX 7 provar a otimalidade de cinco instâncias a mais em relação ao obtido anteriormente (Tabela 6.2). Com esse resultado, o *gap* médio baixou de 1,8% para 1,5%, com uma redução do tempo médio de execução de 3.745 para 3.023 segundos. Melhor que sua versão mais antiga, o CPLEX 11 provou a otimalidade de todas as instâncias destes conjuntos (colunas 5-7 da Tabela 6.4) em um tempo médio de apenas 518 segundos.

Quanto ao desempenho das heurísticas HC e HC<sub>ext</sub>, a única alteração em relação à Tabela 6.2 incorreu sobre os tempos computacionais requeridos para resolução das instâncias. Os resultados quanto à falta naturalmente não mudam, mas como agora são comparados aos resultados obtidos com o CPLEX 11, o desvio médio percentual teve um pequeno aumento, de 8,8% para 9,5% para HC, e de 7,9% para 8,6% para HC<sub>ext</sub>, mantendo-se HC<sub>ext</sub> um pouco superior à HC. O motivo do crescimento do *gap* ter sido pequeno deve-se ao CPLEX 11 ter

provado a otimalidade da solução das instâncias onde o CPLEX 7, apesar de obter o mesmo resultado, não foi capaz de provar a otimalidade no tempo disponível. Portanto, o maior número de certificados de otimalidade obtidos com o CPLEX 11 não significa melhorias na qualidade das soluções encontradas com o CPLEX 7.

Tabela 6.4 – Resultados com variação de demanda: resolução do modelo com CPLEX 7 e CPLEX 11 e heurística HC<sub>ext</sub>.

Conjunto	Modelo MFP			Modelo MFP			HC		HC <sub>ext</sub>	
	CPLEX 7			CPLEX 11			DPM (%)	Tempo (s)	DPM (%)	Tempo (s)
	CO (%)	Gap (%)	Tempo (s)	CO (s)	Gap (%)	Tempo (s)				
D50	100	0	4	100	0	5	1,8	21	0,8	51
D40	100	0	7	100	0	8	5,7	22	5,4	55
D30	100	0	27	100	0	10	15,3	23	13,9	57
D20	100	0	68	100	0	27	12,7	24	11,2	59
D10	100	0	846	100	0	128	4,2	25	4,2	62
S	90	0,4	1.351	100	0	617	16,7	18	15,7	44
I10	80	2,1	3.748	100	0	1.134	10,9	28	10,4	69
I20	60	4,6	5.100	100	0	682	13,4	29	12,6	71
I30	40	4,6	7.682	100	0	593	10,1	30	8,3	74
I40	30	3,3	8.494	100	0	1.053	7,1	30	6,9	75
I50	50	1,1	5.929	100	0	1.443	6,9	30	5,4	75
Média		1,5	3.023		0	518	9,5	25	8,6	63

A Tabela 6.5 apresenta os resultados obtidos nos experimentos com variação do número de períodos (conjuntos de instâncias T2-T5) e executados na nova máquina com ambas as versões do CPLEX e a heurística HC e sua versão estendida. Note que a versão 11 do *solver* obteve *gap* de otimalidade de apenas 1,9%, melhorando consideravelmente os resultados obtidos com a versão 7. O baixo *gap* obtido pelo modelo na versão 11 do *solver* fez com que a média dos desvios das heurísticas em relação ao modelo fossem consideravelmente mais altos, ultrapassando 20 pontos percentuais. Para a heurística HC, o desvio percentual médio varia de 20,6% a 22,8% com tempos computacionais médios entre 123 a 1.740 segundos. Para a heurística estendida, o desvio percentual médio atinge 21,2%, enquanto os tempos computacionais médios chegam a 4.302 segundos.

É importante observar que os tempos computacionais obtidos pelas heurísticas devem justificar sua aplicação em instâncias de maior porte, uma vez que os tempos médios com o CPLEX 11 (entre 2.477 e 7.372 segundos) podem tornar sua utilização proibitiva em algumas situações práticas.

Tabela 6.5 - Resultados com variação do número de períodos: resolução do modelo com CPLEX 7, CPLEX 11 e heurística HC.

Conjunto	Modelo MFP CPLEX 7			Modelo MFP CPLEX 11			HC		HC <sub>ext</sub>	
	CO	Gap	Tempo	Tempo	Gap	Tempo	DPM	Tempo	DPM	Tempo
	(%)	(%)	(s)	(s)	(%)	(s)	(%)	(s)	(%)	(s)
T2	60	2,8	5.103	90	1,0	2.477	22,8	123	19,3	298
T3	20	11,0	9.103	80	1,6	3.378	20,6	389	20,3	955
T4	20	17,7	9.340	80	2,3	5.357	21,5	902	21,2	2.219
T5	10	11,2	10.517	60	2,9	7.372	21,5	1.740	21,2	4.302
Média		10,7	8.516		1,9	4.646	21,6	788	20,5	1.943

Estes experimentos revelam claramente a marcante diferença de desempenho entre as duas versões do *solver*; a versão 11 consegue um número muito maior de certificados de otimalidade em um tempo médio de execução bastante inferior à versão 7. Das 150 instâncias executadas, a versão 11 garantiu a otimalidade de 141 soluções contra apenas 95 na versão 7. O tempo médio computacional para resolver os conjuntos com variação de demanda foi quase 6 vezes menor na versão 11 e para a variação de períodos, a versão 7 precisou de quase o dobro do tempo da versão 11.

### 6.3.1. Aplicação de heurísticas de busca local $k$ -opt

Nesta seção são descritos os resultados da aplicação dos procedimentos de busca local (seção 5.2) a partir das soluções providas pela heurística HC<sub>ext</sub>. Três valores do parâmetro  $k$  foram utilizados para criar cinco diferentes variações de algoritmos de busca local a partir da heurística  $k$ -opt, e descritas no Quadro 6.1. A primeira coluna do quadro fornece a sigla que identifica o algoritmo; a segunda coluna indica os valores de  $k$  utilizados no algoritmo; a terceira coluna é marcada com ‘Sim’ se a busca local leva em conta apenas os

processos selecionados para o conjunto P durante a execução da  $HC_{ext}$  (pré-processamento, apresentado na seção 5.2.1); a quarta coluna é marcada com ‘Sim’ se a função de avaliação (5.1’) empregada no algoritmo utiliza um número limitado de períodos à frente. Nos casos em que o número de períodos é limitado (variações  $BL_D$  e  $BL_E$ ), os números entre parênteses significam a faixa de períodos para os quais são geradas vizinhanças de busca. Tomando como exemplo o algoritmo  $BL_D$ , é gerada uma solução com a vizinhança que considera a 5 períodos adiante, outra solução com a vizinhança que considera a 6 períodos adiante, e assim por diante, até se gerar uma solução com a vizinhança que considera 13 períodos adiante. Desta forma é gerado um total de 9 soluções (quinta e última coluna). As soluções geradas são comparadas e a melhor é selecionada para a próxima iteração da busca local.

A justificativa para os algoritmos  $BL_D$  e  $BL_E$  se baseia no fato de que testes preliminares com diferentes números de períodos à frente não indicaram um número de períodos comum a todas as instâncias que gerasse a melhor solução. Os testes ainda sugeriram que entre 5 e 19 períodos se concentram a maioria das soluções de boa qualidade, mesmo para as instâncias com até 95 períodos de produção.

Quadro 6.1 – Características dos algoritmos de busca local.

Algoritmo	$k$	Pré-processamento	Limite de número de períodos em $f$	Quantidade de soluções geradas
$BL_A$	1	Não	Não	1
$BL_B$	3	Sim	Não	1
$BL_C$	2	Sim	Não	1
$BL_D$	2 e 3	Sim	Sim (5-13)	9
$BL_E$	2	Sim	Sim (5-19)	15

A Tabela 6.6 apresenta os desvios percentuais médios das soluções dos algoritmos  $BL_A$ – $BL_E$  em relação às soluções do modelo (com o *solver* CPLEX 11) para os conjuntos de instâncias com variação de demanda. Os resultados para os conjuntos com variação de períodos são apresentados na Tabela 6.7.

Tabela 6.6 – Resultados com variação de demanda: algoritmos de busca local.

Conjunto	BL <sub>A</sub>		BL <sub>B</sub>		BL <sub>C</sub>		BL <sub>D</sub>		BL <sub>E</sub>	
	<i>DPM</i> (%)	Tempo (s)	<i>DPM</i> (%)	Tempo (s)	<i>DPM</i> (%)	Tempo (s)	<i>DPM</i> (%)	Tempo (s)	<i>DPM</i> (%)	Tempo (s)
D50	0,8	112	0,8	123	0,4	137	7,2	515	0,3	952
D40	5,3	115	5,3	125	4,8	141	0,2	522	0,2	941
D30	13,5	118	13,5	128	13,0	147	4,4	537	2,8	945
D20	9,4	120	9,4	130	9,1	167	2,5	542	1,7	966
D10	4,2	122	4,2	130	3,8	174	2,5	553	2,5	921
S	13,4	122	15,0	198	11,1	175	7,0	538	5,0	968
I10	8,4	129	10,0	178	6,9	175	4,5	548	3,6	1.043
I20	11,3	131	12,4	176	10,6	175	3,3	521	3,5	1.076
I30	7,8	135	7,9	176	6,0	186	3,4	539	3,2	1.099
I40	6,3	135	6,8	171	5,5	184	2,2	533	2,6	1.111
I50	4,9	134	5,3	170	3,9	185	1,8	529	1,8	1.075
Média	7,8	111	8,2	155	6,8	168	3,5	534	2,5	1.009

Tabela 6.7 - Resultados com variação de períodos: algoritmos de busca local.

Conjunto	BL <sub>A</sub>		BL <sub>B</sub>		BL <sub>C</sub>		BL <sub>D</sub>		BL <sub>E</sub>	
	<i>DPM</i> (%)	Tempo (s)	<i>DPM</i> (%)	Tempo (s)	<i>DPM</i> (%)	Tempo (s)	<i>DPM</i> (%)	Tempo (s)	<i>DPM</i> (%)	Tempo (s)
T2	17,6	897	18,1	1.089	13,9	1.171	7,9	1.284	5,1	3.053
T3	19,5	3.017	20,21	3.237	15,6	3.658	7,9	2.115	5,1	4.940
T4	21,2	7.097	21,1	6.935	20,7	7.959	8,0	2.973	5,1	7.086
T5	21,2	10.800	21,0	10.800	18,3	10800	9,3	3.887	5,9	9.394
Média	19,9	6.645	20,1	5.515	17,1	5.897	8,3	2.564	5,3	6.118

Note que BL<sub>A</sub>, BL<sub>B</sub> e BL<sub>C</sub> não utilizam vizinhança limitada, o que implica em analisar todos os períodos de  $t$  à  $T$  para cada  $t$ . Apesar de BL<sub>B</sub> utilizar  $k=3$  e BL<sub>C</sub> utilizar  $k=2$ , os tempos são próximos porque, conforme discutido na seção 5.2, devido ao esforço computacional da utilização de  $k=3$  ser muito maior, apenas uma pequena parte da solução é investigada ( $k-1$  períodos que antecedem o período com maior falta acumulada de produção), ao contrário de  $k=2$ , que investiga todos os períodos. Em BL<sub>D</sub>, a partir do ótimo local obtido

com  $k=2$  é reaplicada a busca local desta vez com  $k=3$ . Apesar de  $BL_E$  utilizar apenas  $k=2$ , ele demanda maior tempo computacional, pois gera seis soluções a mais que  $BL_D$ , e para essas soluções adicionais, a vizinhança é maior porque percorre um número maior de períodos à frente.

Como mostrado na seção anterior, os resultados da heurística construtiva apresentaram, nos conjuntos de instâncias com variação de demanda e de períodos, desvios percentuais de cerca de 9% e 20%, respectivamente. Nota-se que a aplicação de algoritmos de busca local a partir destas soluções forneceu melhorias de maior ou menor magnitude, sendo os algoritmos  $BL_D$  e  $BL_E$  aqueles que efetivamente contribuíram para reduções de desvios percentuais. Nos conjuntos com variação de demanda (Tabela 6.6), são obtidos desvios percentuais médios de 3,5% e 2,5%, e para os conjuntos com variação de períodos (Tabela 6.7) com respectivos desvios atingem 8,3% e 5,3%.

Os resultados mostram que soluções fornecidas pela heurística  $HC_{ext}$  podem ser consideravelmente melhoradas com a aplicação de buscas locais. Note que enquanto o algoritmo  $BL_E$  obteve em média as melhores soluções, o algoritmo  $BL_D$  requereu menor tempo computacional. Assim como ocorreram melhorias nos resultados de  $HC_{ext}$ , também ocorreu um aumento considerável no tempo computacional necessário para executar as heurísticas de busca local. O tempo computacional necessário para obtenção da solução final é diretamente proporcional ao número de trocas, tamanho da vizinhança e quantidade de soluções geradas.

Para efeitos de comparação, são apresentados na Tabela 6.8 os resultados obtidos com o modelo, a heurística  $HC_{ext}$  e os algoritmos  $BL_D$  e  $BL_E$  considerando os 15 conjuntos de instâncias tratadas. Como esperado, o tempo médio da heurística construtiva  $HC_{ext}$  é o menor dos quatro métodos aplicados, porém, o desvio médio percentual em relação ao modelo é de 11,8%. Apesar de um tempo médio computacional superior a quatro vezes o tempo médio exigido por  $HC_{ext}$  e quase duas vezes o tempo médio usado pelo CPLEX, a heurística de melhoria  $BL_E$  baixou o desvio médio percentual em relação ao CPLEX para apenas 3,2%. Os resultados obtidos por  $BL_E$  se tornam interessantes considerando que o método exato obteve um *gap* de otimalidade de apenas 0,5%. Com uma pequena piora de 1,6 pontos no *gap* em relação à  $BL_E$ ,  $BL_D$  precisou, em média, de apenas 1.075 segundos, menos que a metade do tempo utilizado por  $BL_E$ .



Tabela 6.8 – Sumário dos resultados computacionais: resolução do modelo,  $HC_{ext}$  e algoritmo  $BL_D$  e  $BL_E$ .

Conjunto	Modelo MFP CPLEX 11		$HC_{ext}$		$BL_D$		$BL_E$	
	Gap (%)	Tempo (s)	$DPM$ (%)	Tempo (s)	$DPM$ (%)	Tempo (s)	$DPM$ (%)	Tempo (s)
D50	0	5	0,8	51	7,2	515	0,3	952
D40	0	8	5,4	55	0,2	522	0,2	941
D30	0	10	13,9	57	4,4	537	2,8	945
D20	0	27	11,2	59	2,5	542	1,7	966
D10	0	128	4,2	61	2,5	553	2,5	921
S	0	617	15,6	44	7,0	538	5,0	968
I10	0	1.134	10,4	69	4,5	548	3,6	1.043
I20	0	682	12,6	71	3,3	521	3,5	1.076
I30	0	593	8,3	74	3,4	539	3,2	1.099
I40	0	1.053	6,9	75	2,2	533	2,6	1.111
I50	0	1.443	5,4	75	1,8	529	1,8	1.075
T2	0,9	2.477	19,3	298	7,9	1.284	5,1	3.053
T3	1,6	3.378	20,3	955	7,9	2.115	5,1	4.940
T4	2,3	5.357	21,2	2.219	8,0	2.973	5,1	7.086
T5	2,8	7.372	21,2	4.302	9,3	3.887	5,9	9.394
Média	0,5	1.618	11,8	564	4,8	1.075	3,2	2.371

### 6.3.2. Algoritmo GRASP

Nos experimentos realizados, foram analisados os resultados de dez combinações de variações de valores de parâmetros utilizados no algoritmo GRASP, descrito na seção 5.3.2. Estas variações recaem tanto sobre parâmetros intrínsecos ao método –  $p$  (tamanho da lista restrita de candidatos LRC) e  $itmax$  (número de iterações ou soluções a serem geradas) – como também sobre o parâmetro  $v$  da função de avaliação  $f$  descrita na seção 5.1, o parâmetro  $k$  da fase de busca local e o limite de períodos a serem investigados adiante para cada período  $t$  ( $maxTF$ ). O Quadro 6.3 descreve as características das combinações estudadas.

Quadro 6.2. – Combinações de variações paramétricas utilizados no algoritmo GRASP.

Combinação	$p$	$v$	$maxTF$	$itmax$	$k$
1	8	4	11	20	1
2	6	4	11	20	1
3	8	5	15	20	1
4	6	4	11	80	1
5	8	4	11	80	1
6	6	4	11	20	2
7	8	4	11	20	2
8	6	4	11	20	1 e 3
<b>9</b>	<b>8</b>	<b>4</b>	<b>11</b>	<b>20</b>	<b>2 e 3</b>
10	8	5	T	20	1

Os resultados da combinação de variação paramétrica de melhor desempenho (combinação 9) para os conjuntos de instâncias com variação de demanda são apresentados nas colunas 10 e 11 da Tabela 6.9, onde os tempos computacionais correspondem ao tempo total das  $itmax$  iterações. Observa-se que o desvio percentual médio do algoritmo GRASP em relação ao modelo resolvido com o *solver* CPLEX 11 varia entre 0,4% e 3,8%, com tempos de execução entre 1.193 e 1.815 segundos em cada conjunto de instâncias. O algoritmo GRASP conseguiu reduzir o desvio percentual médio das soluções de  $BL_E$  em 1% em relação ao CPLEX, ficando em média, à apenas 1,5% do ótimo. Porém, o tempo médio computacional, pouco maior que o exigido pela  $BL_E$ , é quase três vezes maior que o do CPLEX.

A fim de observar o desempenho das heurísticas frente ao CPLEX com instâncias de ainda maior porte, foi gerado o conjunto T6 ( $6 \times 19 = 114$  períodos), o qual contabiliza no modelo 18.126 variáveis. Os resultados obtidos com o algoritmo GRASP (combinação 9) para os conjuntos de instâncias com variação de períodos são apresentados nas colunas 10 e 11 da Tabela 6.10.

Tabela 6.9 – Resultados com variação de demanda: resolução do modelo, HC<sub>ext</sub>, algoritmo BL<sub>E</sub>, e algoritmo GRASP.

Conjunto	Modelo MFP CPLEX 11		HC <sub>ext</sub>		BL <sub>E</sub>		GRASP	
	<i>Gap</i>	Tempo	<i>DPM</i>	Tempo	<i>DPM</i>	Tempo	<i>DPM</i>	Tempo
	(%)	(s)	(%)	(s)	(%)	(s)	(%)	(s)
D50	0	5	0,8	51	0,3	952	1,1	1.193
D40	0	8	5,4	55	0,2	941	0,4	1.206
D30	0	10	13,9	57	2,8	945	3,8	1.272
D20	0	27	11,2	59	1,7	966	0,5	1.374
D10	0	128	4,2	61	2,5	921	1,2	1.412
S	0	617	15,6	44	5,0	968	1,5	1.572
I10	0	1.134	10,4	69	3,6	1.043	1,0	1.695
I20	0	682	12,6	71	3,5	1.076	2,4	1.685
I30	0	593	8,3	74	3,2	1.099	1,7	1.724
I40	0	1.053	6,9	75	2,6	1.111	1,8	1.775
I50	0	1.443	5,4	75	1,8	1.075	1,3	1.815
Média	0	518	8,6	63	2,5	1.009	1,5	1.520

Tabela 6.10 - Resultados com variação de períodos: resolução do modelo, HC<sub>ext</sub>, algoritmo BL<sub>E</sub>, e algoritmo GRASP.

Conjunto	Modelo MFP CPLEX 11		HC <sub>ext</sub>		BL <sub>E</sub>		GRASP	
	<i>Gap</i>	Tempo	<i>DPM</i>	Tempo	<i>DPM</i>	Tempo	<i>DPM</i>	Tempo
	(%)	(s)	(%)	(s)	(%)	(s)	(%)	(s)
T2	1,0	2.477	19,3	298	5,1	3.053	1,4	3.711
T3	1,6	3.378	20,3	955	5,1	4.940	1,4	6.588
T4	2,3	5.357	21,2	2.219	5,1	7.086	1,6	10.758
T5	2,9	7.372	21,2	4.302	5,9	9.394	1,6	10.800
T6	19,4	10.800	15,5	10.800	1,8	10.800	<b>-2,0</b>	10.800
Média	5,4	5.877	19,5	3.715	4,6	7.055	0,8	8.531

Conforme observado na Tabela 6.10, o desvio percentual médio apresentado pelo algoritmo GRASP em relação às soluções com o modelo é de 0,8% contra 4,6% do

algoritmo  $BL_E$ , e 19,5% com  $HC_{ext}$ . Considerando que o *gap* médio do modelo é de apenas 5,4%, isso significa que assim como nas instâncias com variação da demanda, o algoritmo GRASP alcançou resultados razoavelmente próximos do ótimo. Ao incluir o conjunto T6 nos testes, o GRASP obteve um desvio médio de -2,0% em relação ao CPLEX, confirmando que as heurísticas melhoram seu desempenho em relação ao modelo conforme o problema fica mais difícil de ser resolvido. Os tempos computacionais para os conjuntos T2 e T3 foram superiores aos do CPLEX e, mais uma vez, menores que os utilizados pelo algoritmo  $BL_E$ . Note que o tempo computacional médio de aplicação do algoritmo GRASP com o conjunto T5 e T6 respeita o limite imposto ao modelo de 10.800 segundos.

Em resumo, das quatro estratégias discutidas nos experimentos realizados com as 160 instâncias, o modelo resolvido pelo *solver* CPLEX 11 em geral obteve os melhores resultados em termos de qualidade de solução e tempo computacional, porém o algoritmo GRASP mostrou-se uma boa opção às ferramentas comerciais. O desempenho de cada método heurístico em relação ao CPLEX 11 pode também ser observado nos gráficos da Figura 6.3.

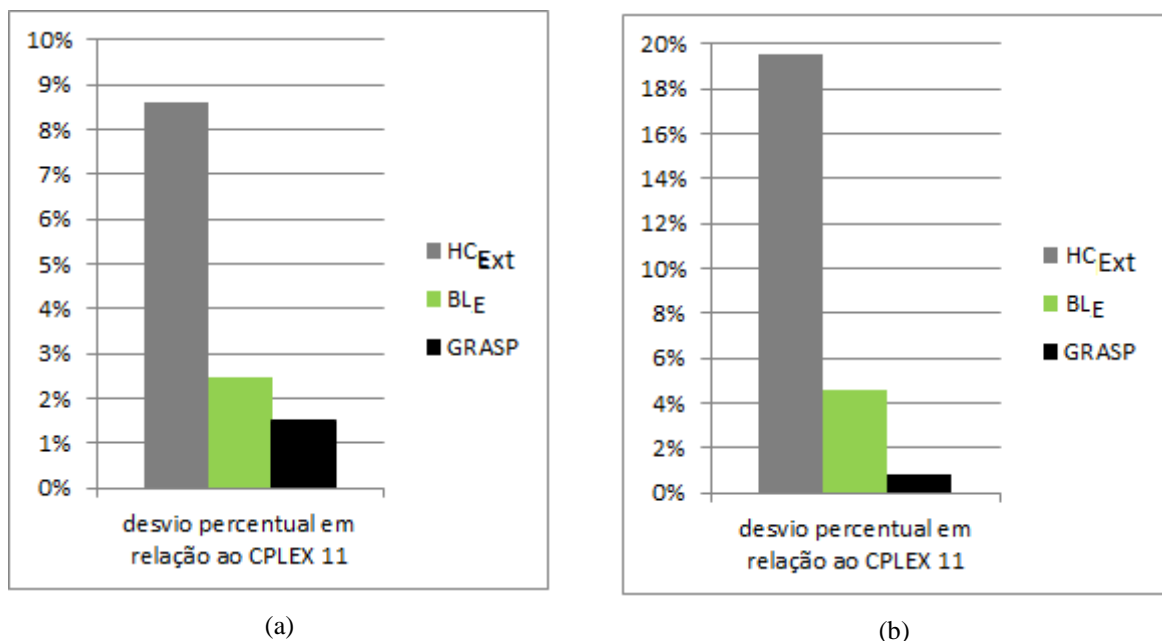


Figura 6.3 - Desvios percentuais médios das soluções com  $HC_{ext}$ , algoritmo  $BL_E$ , e algoritmo GRASP, em relação aos resultados com CPLEX11: (a) variação de demanda, e (b) variação de períodos.

#### 6.4. Experimentos com processamento paralelo

Computadores com apenas um núcleo de processamento executam suas tarefas sequencialmente, uma após a outra, sendo por isso chamados de computadores sequenciais. Tais máquinas têm um limite físico de aperfeiçoamento que é a velocidade da luz no vácuo, comprometendo assim o uso desses modelos como critério de dimensionamento (CARRIJO, 2001).

Era esperado que o desenvolvimento dos computadores alcançasse um patamar tal que contornasse essa limitação física, comum aos computadores pessoais e aos supercomputadores. De fato, foram criados computadores com arquitetura paralela, que diferentemente das máquinas sequenciais, possibilitam que vários processadores trabalhem concomitantemente na execução de uma mesma tarefa. Processadores atuais possuem até oito núcleos de processamento, o que é quase como ter oito máquinas realizando a tarefa.

O conceito de paralelismo representou uma grande evolução em relação às máquinas seriais, rompendo com a necessidade de se executar tarefas sequencialmente. Essa arquitetura permite que se alcance o desempenho dos atuais supercomputadores com um custo bem inferior em *hardware*, porém é necessário que o sistema operacional reconheça, distribua e administre a tarefa entre os processadores e uso da memória. Além disso, é necessário escrever algoritmos para processamento paralelo. Muitos algoritmos atualmente utilizados foram escritos para processamento sequencial, impedindo que sejam utilizados todos os recursos disponíveis.

Seguindo esta tendência de paralelização, a versão 11 do CPLEX permite que até quatro *threads* (linhas de execução) sejam executadas simultaneamente para o mesmo modelo em processadores a partir da geração Core2quad. A fim de observar o aumento da eficiência da execução paralela com o uso de *threads* em relação ao processamento sequencial (um *thread*), foram realizados experimentos com os conjuntos de instâncias que apresentaram maior dificuldade de resolução, ou seja, os conjuntos T2-T6. A Tabela 6.11 ilustra os resultados obtidos.

Tabela 6.11 - Resultados com variação do número de períodos: resolução do modelo com CPLEX 11 com um e quatro *threads*.

Conjunto	Modelo MFP CPLEX 11 / 1 <i>thread</i>			Modelo MFP CPLEX 11/ 4 <i>threads</i>		
	CO (%)	Gap (%)	Tempo (s)	CO (%)	Gap (%)	Tempo (s)
T2	90	1,0	2.477	90	0,2	1.419
T3	80	1,6	3.378	80	0,9	2.820
T4	80	2,3	5.357	90	1,8	3.418
T5	60	2,9	7.372	70	2,0	5.345
T6	0	19,4	10.800	30	12,2	10.142
Média	60	5,4	5.877	70	3,4	4.629

Os resultados da tabela mostram que o *gap* médio e o tempo médio utilizado na resolução dos quatro conjuntos de instâncias diminuí consideravelmente ao se utilizar 4 *threads*. A melhora em relação ao *gap* médio foi de 37% e o tempo médio utilizado diminuiu em 21%. Ressalta-se o maior número de certificados de otimalidade nos conjuntos T4, T5 e T6, o que sugere que seu uso é particularmente benéfico na resolução de instâncias difíceis.

No caso de GRASP, o processamento paralelo não melhorou a qualidade das soluções apresentadas nas seções anteriores. Para as instâncias tratadas, GRASP não demora para convergir para uma solução de boa qualidade, sugerindo que o uso de processamento paralelo irá aumentar o número de soluções geradas dentro de um tempo limite, porém, não encontrará solução de melhor qualidade neste conjunto estendido de soluções.

No Apêndice VI encontra-se uma discussão sobre o impacto que parâmetros modificados do CPLEX 11 têm sobre o desempenho na resolução do modelo MFP com o conjunto de instâncias tratadas.

### 6.5. Variações das heurísticas visando à minimização da falta e do excesso de produção

Conforme discutido na seção 4.2, o modelo MFP pode ser modificado com a utilização da função objetivo (4.1') a fim de se obter soluções que minimizem a falta e o excesso de produção da demanda. O peso do excesso na função objetivo (parâmetro  $\lambda$ ) deve ser definido de forma que a falta acumulada se mantenha como objetivo principal. Experimentos preliminares com as instâncias tratadas mostraram que o impacto do excesso na

função objetivo passa a ser influente nos resultados obtidos com  $\lambda = 0,001$ , ou seja, contabilizando 0,1% do excesso acumulado na função objetivo.

De igual forma, as heurísticas propostas podem ser alteradas com a utilização da função de avaliação (5.1'') discutida na seção 5.4:

$$F_{jt} = \sum_{i=1}^m \sum_{t'=t}^T \text{Min} \left\{ \frac{(a_{ij} - d'_{it'})}{(t''-t+1)^v}, 0 \right\} + \text{Max} \left\{ \frac{(a_{ij} - d'_{it'})}{(t''-t+1)^v}, 0 \right\} * \lambda \quad (5.1'')$$

onde,

$$\text{Min} \left\{ \frac{(a_{ij} - d'_{it'})}{(t''-t+1)^v}, 0 \right\} \quad \text{representa a falta e}$$

$$\text{Max} \left\{ \frac{(a_{ij} - d'_{it'})}{(t''-t+1)^v}, 0 \right\} * \lambda \quad \text{representa o excesso com peso } \lambda.$$

Também conforme discutido na seção 5.4, caso a variação da heurística utilize uma função de avaliação com um número limitado de períodos à frente, a função de avaliação passa a ser:

$$F_{jt} = \sum_{i=1}^m \sum_{t'=t}^{\text{Min}(t+\text{max TF}, T)} \text{Min} \left\{ \frac{(a_{ij} - d'_{it'})}{(t''-t+1)^v}, 0 \right\} + \text{Max} \left\{ \frac{(a_{ij} - d'_{it'})}{(t''-t+1)^v}, 0 \right\} * \lambda \quad (5.1''')$$

Os resultados da utilização da função objetivo (4.1') com  $\lambda = 0,001$  para os conjuntos com variação de demanda são apresentados na Tabela 6.12. Para cada um dos métodos heurísticos analisados, a qualidade média das soluções é aferida pelos desvios percentuais médios da falta ( $DP_F$ ) e do excesso ( $DP_I$ ) em relação aos resultados obtidos com CPLEX 11.

Para os testes realizados, o GRASP foi configurado com a combinação 7, ao invés da combinação 9 (Quadro 6.3), por utilizar menor tempo computacional. Isso se fez necessário, uma vez que o cálculo da função (5.1'') demanda maior esforço que o da função (5.1).

Tabela 6.12 – Resultados com variação de demanda: resolução do modelo,  $HC_{ext}$ , algoritmo  $BL_E$ , e algoritmo GRASP com a função objetivo (4.1’).

Conjunto	CPLEX 11		$HC_{ext}$				$BL_E$				GRASP			
	Gap (%)	Tempo (s)	$DP_I$ (%)	$DP_F$ (%)	DP (%)	Tempo (s)	$DP_I$ (%)	$DP_F$ (%)	DP (%)	Tempo (s)	$DP_I$ (%)	$DP_F$ (%)	DP (%)	Tempo (s)
D50	0	18	3,3	24,0	6,5	142	1,3	18,8	5,0	702	1,5	19,4	5,6	528
D40	0	33	11,7	-0,1	5,6	142	6,5	-1,2	2,1	757	7,1	-1,2	2,2	525
D30	0	55	2,3	12,6	6,5	140	0,9	3,3	2,3	703	1,0	0	0,4	514
D20	0	99	-0,9	4,3	3,5	142	0,6	2,7	1,8	685	0,7	2,7	1,9	527
D10	0	967	1,3	12,6	8,1	141	2,0	4,3	4,4	676	1,2	3,5	3,4	549
S	0	499	3,4	15,0	12,9	150	-1,7	4,9	4,3	668	-0,1	2,1	2,0	538
I10	0	1.791	1,6	10,1	9,3	148	-0,1	5,1	4,6	739	-0,3	5,1	4,6	528
I20	0,6	3.911	0,9	11,7	11,3	151	0,3	3,7	3,7	639	1,3	3,9	3,7	533
I30	0,6	4.259	2,7	8,3	8,2	147	0,1	2,8	2,7	640	-1,2	2,8	2,7	526
I40	0,7	5.217	5,2	8,0	8,0	151	4,2	2,5	2,5	623	3,3	3,0	3,0	524
I50	0	2.019	2,9	5,2	5,2	151	5,1	1,7	1,7	610	5,8	1,7	1,7	529
Média	0,2	1.715	3,1	10,2	7,7	146	1,7	4,4	3,2	676	1,8	3,9	2,8	529

Assim como nos experimentos anteriores, o modelo resolvido pelo *solver* CPLEX 11 obteve, em média, melhores resultados do que as técnicas heurísticas empregadas. Note, entretanto, que o GRASP apresenta desvio médio percentual baixo (2,8%), e obtidos em cerca de 30% do tempo computacional requerido pelo método exato. A tabela ainda mostra que para alguns conjuntos, as heurísticas fornecem  $DP_I$  ou  $DP_F$  negativo, mas não ambos negativos para o mesmo conjunto, permanecendo DP positivo para todos os conjuntos.

Os resultados da utilização da função objetivo (5.1’) com  $\lambda = 0,001$  para os conjuntos com variação de períodos são apresentados na Tabela 6.13.



Tabela 6.13 – Resultados com variação de período: resolução do modelo,  $HC_{ext}$ , algoritmo  $BL_E$ , e algoritmo GRASP com a função objetivo (4.1’).

Con- Junto	CPLEX 11		$HC_{ext}$				$BL_E$				GRASP			
	<i>Gap</i> (%)	Tempo (s)	$DP_I$ (%)	$DP_F$ (%)	$DP$ (%)	Tempo (s)	$DP_I$ (%)	$DP_F$ (%)	$DP$ (%)	Tempo (s)	$DP_I$ (%)	$DP_F$ (%)	$DP$ (%)	Tempo (s)
T2	1,8	8.000	8,6	22,2	15,2	1.026	5,0	5,1	4,5	2.825	4,3	3,6	3,8	1.665
T3	6,1	10.800	23,7	16,0	13,8	3.303	12,8	2,0	5,1	5.660	11,5	0,1	2,5	3.670
T4	13,2	10.800	28,2	9,3	11,5	7.157	10,7	-1,5	2,1	8.444	12,3	-2,8	1,6	6.756
T5	21,9	10.800	25,8	0,5	4,3	10.800	6,2	-7,6	<b>-3,2</b>	10.800	7,5	-8,7	<b>-3,5</b>	10.800
Média	10,8	10.100	21,6	12,0	11,2	5.571	8,7	-0,5	2,1	6.932	8,9	-2,0	1,1	5.722

Mais uma vez, o modelo resolvido pelo *solver* CPLEX 11 obteve, em média, melhores resultados do que as técnicas heurísticas empregadas. Cabe ressaltar que  $BL_E$  e o algoritmo GRASP obtiveram algumas melhores soluções no critério falta em relação ao CPLEX com impactos claros no desvio percentual médio (2,1 e 1,1%, respectivamente). Em particular, no conjunto T5 foram obtidos desvios percentuais iguais a -3,2% e -3,5% para  $BL_E$  e o algoritmo GRASP respectivamente. Este último obteve em média 1,1% de desvio médio percentual em relação ao CPLEX. Os resultados ainda mostram que a inclusão do excesso na função objetivo dificultou consideravelmente a obtenção de boas soluções com o CPLEX 11. Em média, o *gap* mostrou-se alto (10,8%) com tempo médio de execução de 10.100 segundos. Cabe lembrar que considerando apenas a falta na função objetivo, o CPLEX 11 obteve apenas 1,9% de *gap* a um tempo computacional de 4.646 segundos.

No próximo capítulo são propostas extensões dos modelos MFP e MFEP tratados no capítulo 4 para contemplar outros objetivos e condições operacionais, e que podem ser utilizadas no planejamento e controle da produção da indústria de grãos eletrofundidos.

**EXTENSÕES DA MODELAGEM MATEMÁTICA**

Neste capítulo são apresentadas extensões dos modelos MFP e MFEP discutidos no capítulo 4, e que poderiam ser utilizadas pelas empresas a fim de investigar possíveis soluções práticas ainda pouco exploradas. Por exemplo, e conforme comentado na seção 4.2, as empresas A e B ainda têm dificuldades em quantificar custos de estoque e *setup*; parte dessa dificuldade decorre do fato de pouco utilizarem essa prática. Entretanto, uma vez que passem a medir tais custos, será possível aplicar um modelo mais completo para o PCP da fábrica, visto que em algumas épocas do ano os turnos somam até 24 horas diárias.

**7.1. Modelo discreto com custos de falta, excesso e *setup* (D-FES)**

Seguindo a literatura de *lot-sizing*, uma variação dos modelos MFP e MFEP pode ser definida com base no modelo DLSP apresentado no Capítulo 3. A fim de considerar custos de *setup*, falta e excesso (estoque), o modelo D-FES, apresentado a seguir, considera a falta carregada (atraso na entrega ou *backlog*) e custos de excesso de produção e *setup*.

**Formulação matemática**

Variáveis:

- $x_{jt}$  variável binária que indica se houve *setup* para o item  $i$  no período  $t$  ( $x_{jt} = 1$ ) ou não ( $x_{jt} = 0$ );
- $I_{it}^+$  excesso (estoque) do item  $i$  ao final do período  $t$  ( $i = 1, \dots, m; t = 1, \dots, T$ );
- $I_{it}^-$  falta do item  $i$  ao final do período  $t$  ( $i = 1, \dots, m; t = 1, \dots, T$ );

$y_{jt}$  indica se a linha está preparada para o processo  $j$  no início do período  $t$  ligando o processo ( $y_{jt} = 1$ ) ou não ( $y_{jt} = 0$ );

Parâmetros:

$s_j$  custo de *setup* do processo  $j$ ;

$g_{it}$  custo unitário de falta do item  $i$  no período  $t$ ;

$h_{it}$  custo unitário de excesso de produção do item  $i$  no período  $t$ .

$a_{ij}$  quantidade do produto  $i$  produzido pelo processo  $j$  no período  $t$  ( $i = 1, \dots, m; j = 1, \dots, J$ );

$d_{it}$  demanda do produto  $i$  no período  $t$  ( $i = 1, \dots, m; t = 1, \dots, T$ );

$y_{j0}$  indica se a linha está preparada para o processo  $j$  no início do período 1 ( $y_{j0} = 1$ ) ou não ( $y_{j0} = 0$ );

$$\text{Min } z = \left( \sum_{i=1}^m \sum_{t=1}^T h_{it} I_{it}^+ + g_{it} I_{it}^- \right) + \left( \sum_{t=1}^T \sum_{j=1}^J s_j x_{jt} \right) \quad (7.1)$$

s.a.

$$I_{it-1}^+ - I_{it-1}^- + \left( \sum_{j=1}^J a_{ij} y_{jt} \right) - I_{it}^+ + I_{it}^- = d_{it} \quad \forall i, t \quad (7.2)$$

$$\sum_{j=1}^J y_{jt} \leq 1, \quad \forall t \quad (7.3)$$

$$x_{jt} \geq y_{jt} - y_{j(t-1)}, \quad \forall j, t \quad (7.4)$$

$$x_{jt}, y_{jt} \in \{0,1\}, \quad \forall j, t \quad (7.5)$$

$$I_{it}^+, I_{it}^- \geq 0 \quad \forall i, j, t \quad (7.6)$$

Na função objetivo (7.1),  $\sum_{i=1}^m \sum_{t=1}^T h_{it} I_{it}^+ + g_{it} I_{it}^-$  é a soma dos custos de excesso de produção e de falta, e  $\sum_{t=1}^T \sum_{j=1}^J s_j x_{jt}$  é soma dos custos de *setup*. As equações (7.2) são restrições de balanceamento de estoques; em (7.3) combinado à (7.5) são descritas as restrições “tudo ou nada” que fazem com que quando há produção no período, seja utilizada

toda sua capacidade, ao mesmo tempo em que se restringe a operação a no máximo 1 processo por período.

A necessidade de *setup* para a operação de um processo em um determinado período é definida em (7.4), e (7.5) e (7.6) correspondem às restrições de domínio das variáveis. É interessante observar que não há a necessidade de definir as variáveis  $x_{jt}$  como binárias, em função de (7.4). Note que se  $y_{jt} = 1$ , há produção com o processo  $j$  no período  $t$ , indicando também, que a linha estará preparada para utilização deste processo no início de  $t_{+1}$ .

No caso dos excessos de produção, grãos maiores poderiam ser reprocessados a partir da moagem, a fim de transformar os grãos em uma granularidade menor que tenha demanda ainda não atendida. O modelo poderá fazer uso dos parâmetros  $h_{it}$  para atribuir maior penalidade aos excessos de pequenos grãos, que dificilmente seriam transformados para uma granularidade ainda menor e com demanda. É importante observar que atualmente, o preço da matéria prima não faz compensar o reprocesso do grão, e a empresa prefere misturá-lo a outras medidas a fim de criar um produto de categoria “b”, vendido no mercado com preço inferior.

Para melhor ilustrar o comportamento do modelo, foram realizados testes com instâncias idealizadas, considerando-se a existência de dez produtos ( $m=10$ ), dez processos de produção ( $J=10$ ) e um horizonte de até dez períodos ( $T=10$ ) e custo de *setup*  $s_j = 1$ . Utilizado em todos os experimentos a seguir, o Quadro 7.1 indica a capacidade de produção de cada um dos dez processos de produção (colunas) para cada um dos dez diferentes produtos (linhas).

Quadro 7.1 - Tonelada por dia produzida do item  $i$  utilizando o processo  $j$ .

	1	2	3	4	5	6	7	8	9	10
EK8A_16	0	1000	2000	0	500	0	0	0	1000	700
EK8A_20	0	1000	500	0	500	0	0	0	1000	700
EK8A_24	0	0	500	0	300	0	0	0	200	700
EK8A_30	0	0	500	0	300	0	0	600	200	0
EK8A_36	0	0	500	0	0	0	0	700	200	0
EK8A_46	0	0	0	1500	0	0	0	700	200	0
EK8A_54	0	0	0	800	0	0	0	700	200	0
EK8A_60	1500	0	0	1500	2000	3000	3000	0	1000	0
EK8A_80	1000	1500	500	1500	1500	2000	2500	0	1000	0
EK8A_100	300	300	0	500	300	700	800	0	1000	0

A Tabela 7.1 mostra os resultados obtidos pelo modelo D-FES considerando também o Quadro 7.2, que contém a demanda de cada um dos dez itens (linhas) em cada um

dos dez períodos (colunas) do horizonte definido. A implementação do modelo em *GAMS* com a demanda do Quadro 7.2, pode ser vista no Apêndice II.

Quadro 7.2 – Demanda original do produto  $i$  no período  $t$ .

	1	2	3	4	5	6	7	8	9	10
EK8A_16	0	0	0	0	0	0	0	0	0	0
EK8A_20	0	0	0	0	0	0	0	0	0	0
EK8A_24	0	0	1000	0	0	0	0	0	0	0
EK8A_30	0	0	0	0	0	0	0	0	0	0
EK8A_36	0	0	0	0	0	0	0	0	0	0
EK8A_46	0	0	0	0	0	0	0	0	0	0
EK8A_54	0	0	0	0	0	0	0	0	0	0
EK8A_60	0	0	0	2000	2000	0	0	0	0	0
EK8A_80	0	0	0	0	0	0	0	0	0	0
EK8A_100	0	0	0	0	0	0	0	0	1000	0

Na Tabela 7.1, para cada período  $t$  de produção, a primeira linha identifica o processo  $j$  utilizado em  $t$ , a segunda linha identifica o processo  $j$  para o qual é cobrado *setup*, e a terceira linha indica o valor em unidades monetárias (u.m.) cobrado pelo *setup*. A quarta e a quinta linhas apresentam, respectivamente, o excesso e a falta de produção em  $t$ .

Tabela 7.1 - Exemplo de programação da produção com o modelo [(7.1)-(7.6)] para oito períodos segundo premissas do modelo D-FES com *backlog*.

	Período ( $t$ )							
	1	2	3	4	5	6	7	8
$j$	-	8	4	6	6	8	6	3
$j$ com $x_{jt} > 0$	-	8	4	6	-	8	6	3
$S_j$	-	80	70	120	-	80	120	100
$\sum_{i=1}^m I_{it}^+$	-	140	410	400	410	290	130	100
$\sum_{i=1}^m I_{it}^-$	100	130	90	70	30	10	410	390

Na Tabela 7.1, observa-se que o modelo D-FES somente cobra *setup* quando um novo lote de produção for iniciado por um processo diferente do utilizado no período anterior. Assim, deve-se considerar que a linha pode se manter preparada para o mesmo processo de um período para o outro (*setup carryover*). Na indústria de grãos eletrofundidos, o *setup carryover* é empregado devido à facilidade de manter o ajuste dos moinhos e a combinação

das peneiras de um período para o outro, sem a necessidade de ajustes ou manutenção preventiva. No exemplo ilustrado na Tabela 7.1, isso acontece no período 5, onde não é cobrado *setup* do processo 6, pois o mesmo processo fora utilizado também no período anterior.

Note que o modelo é apropriado para jornadas de 24 horas, onde a cada troca de processos, é cobrado um custo de *setup* na função objetivo. A função objetivo ainda considera falta e excesso carregados entre os períodos, obtendo o valor da solução em 4.490 u.m. Este valor é encontrado a partir dos valores atribuídos aos parâmetros de custo de falta, excesso e *setup* que podem ser ajustados pela empresa. Note que o valor da solução seria maior caso fosse atribuído um processo qualquer ao período 1, pois apesar da oportunidade de diminuir ou até mesmo eliminar a falta naquele período, quaisquer um dos processos aumentaria o excesso e implicaria em *setup*, aumentando os custos gerais de maneira que superariam os custos com a “pequena quantidade” em falta. Portanto, em casos em que a falta de produção seja pequena, ligar um novo processo em um período ocioso a fim de diminuir essa falta poderá gerar custos com maior estocagem e/ou custos de *setup*, resultando em uma solução de pior qualidade.

A falta carregada (*backlog*) não é uma prática encontrada em todas as empresas do setor de grãos eletrofundidos. Isso porque há clientes que exigem a entrega na data combinada e o atraso significa a desistência da compra. Fazendo-se uma simples modificação nas equações (7.2), pode-se considerar o caso de “vendas perdidas”, ao invés de atraso no atendimento da demanda. Para este caso,  $g_{it}$  passa a representar o custo unitário de venda perdida do item  $i$  no período  $t$ , sendo esperado que este seja maior que o valor aplicado ao atraso:

$$I_{it-1}^+ + \left( \sum_{j=1}^J a_{ij} y_{jt} \right) - I_{it}^+ + I_{it}^- = d_{it} \quad \forall i, t \quad (7.2')$$

Na Tabela 7.2, são apresentados os resultados com o modelo [(7.1), (7.2'), (7.3)-(7.6)]. Neste experimento, os valores dos parâmetros  $g_{it}$  foram quadruplicados, a fim de que o custo unitário de venda perdida seja maior que o custo unitário de atraso considerado no modelo [(7.1)-(7.6)]. Um detalhe importante a ser considerado é que o custo de venda perdida é aplicado uma única vez à demanda não atendida do item, ao passo que o custo de atraso é aplicado em cada período em que a falta é carregada. É importante também ressaltar que

custos por venda perdida é função de vários fatores externos, o que torna difícil sua mensuração.

Tabela 7.2 - Exemplo de programação da produção com o modelo [(7.1), (7.2'), (7.3)-(7.6)] para oito períodos segundo premissas do modelo D-FES sem *backlog*, mas com vendas perdidas.

	Período (t)							
	1	2	3	4	5	6	7	8
$j$	-	4	6	6	8	7	7	8
$j$ com $x_{jt} > 0$	-	4	6	-	8	7	-	8
$s_j$	-	70	120	-	80	110	-	80
$\sum_{i=1}^m I_{it}^+$	-	-	200	220	350	200	170	20
$\sum_{i=1}^m I_{it}^-$	100	100	110	10	-	30	310	190

Por não considerar a falta carregada, o valor da função objetivo passa a ser 5.630 u.m. nesta nova solução. Excluída a possibilidade de atrasos, não há a necessidade de produzir a demanda não atendida em períodos anteriores, o que altera significativamente a programação. Os custos de vendas perdidas naturalmente acabam por onerar o valor da função objetivo obtido com o modelo [(7.1)-(7.6)].

O critério “tudo ou nada” presente nos modelos baseados no DLSP tende a aumentar os excessos e, conseqüentemente, os custos de manutenção destes ao obrigar a utilização da capacidade total de produção no período. Uma maneira de evitar esse aumento no custo de estocagem é permitir que lotes menores sejam utilizados no período. Conforme discutido no capítulo 3, o modelo que considera esta prática é o CSLP. A seção 7.2 discute como o modelo (D-FES) pode ser reescrito como um CSLP com custos de falta, excesso e *setup* (CS-FES). Este modelo é útil para épocas do ano em que a demanda geral por produtos é baixa, de maneira que os períodos de tempo em que não ocorre produção podem ser utilizados para atualização profissional dos funcionários e manutenções no chão de fábrica.

## 7.2. Modelo com *setup* contínuo CS-FES

Neste modelo, são utilizadas as variáveis  $q_{jt}$  (fração de um período de produção utilizado com o processo  $j$  no período  $t$ ).

### Formulação matemática:

Variáveis:

$q_{jt}$ : fração de um período de produção utilizado com o processo  $j$  no período  $t$  ( $0 \leq q_{jt} \leq 1$ );

$x_{jt}$  variável binária que indica se houve *setup* para o item  $i$  no período  $t$  ( $x_{jt} = 1$ ) ou não ( $x_{jt} = 0$ );

$I_{it}^+$  excesso do item  $i$  ao final do período  $t$  ( $i = 1, \dots, m; t = 1, \dots, T$ );

$I_{it}^-$  falta do item  $i$  ao final do período  $t$  ( $i = 1, \dots, m; t = 1, \dots, T$ );

$y_{jt}$  indica se a linha está preparada para o processo  $j$  no início do período  $t$  ( $y_{jt} = 1$ ) ou não ( $y_{jt} = 0$ );

Parâmetros:

$s_j$  custo de *setup* do processo  $j$ ;

$g_{it}$  custo unitário de falta do item  $i$  no período  $t$ ;

$h_{it}$  custo unitário de estocagem do item  $i$  no período  $t$ .

$a_{ij}$  quantidade do produto  $i$  produzido pelo processo  $j$  no período  $t$  ( $i = 1, \dots, m; j = 1, \dots, J$ );

$d_{it}$  demanda do produto  $i$  no período  $t$  ( $i = 1, \dots, m; t = 1, \dots, T$ );

$y_{j0}$  indica se a linha está preparada para o processo  $j$  no início do período 1 ( $y_{j0} = 1$ ) ou não ( $y_{j0} = 0$ );

$$\text{Min } z = \left( \sum_{i=1}^m \sum_{t=1}^T h_{it} I_{it}^+ + g_{it} I_{it}^- \right) + \left( \sum_{t=1}^T \sum_{j=1}^J s_j x_{jt} \right) \quad (7.1)$$

s.a.

$$I_{it-1}^+ - I_{it-1}^- + \left( \sum_{j=1}^J a_{ij} q_{jt} \right) - I_{it}^+ + I_{it}^- = d_{it} \quad \forall i, t \quad (7.7)$$

$$q_{jt} \leq y_{jt} \quad \forall j, t \quad (7.8)$$

$$\sum_{j=1}^J y_{jt} \leq 1, \quad \forall t \quad (7.3)$$



$$x_{jt} \geq y_{jt} - y_{j(t-1)}, \quad \forall j, t \quad (7.4)$$

$$x_{jt}, y_{jt} \in \{0,1\}, \quad \forall j, t \quad (7.5)$$

$$I_{it}^+, I_{it}^- \geq 0 \quad \forall i, j, t \quad (7.6)$$

A equação (7.7) utiliza  $q_{jt}$  no lugar de  $y_{jt}$  em relação ao modelo D-FES, determinando juntamente com as restrições (7.8) que a capacidade de produção de um processo não precisa ser utilizada em sua totalidade no período. A implementação do modelo no *GAMS* pode ser vista no Apêndice III. Os resultados obtidos com o modelo CS-FES são resumidos na Tabela 7.3.

Tabela 7.3 - Exemplo de programação da produção com o modelo [(7.1), (7.7), (7.8), (7.3)-(7.6)] para oito períodos segundo premissas do modelo CS-FES com *backlog*.

	Período (t)							
	1	2	3	4	5	6	7	8
$j$	8	4	4	6	6	6	7	7
$x_{jt}$	1	1	-	1	-	-	1	-
$q_{jt}$	1	0,33	0,67	1	0,67	1	1	0,50
$\sum_{i=1}^m I_{it}^+$	140	230	410	400	366,67	266,67	290	96,67
$\sum_{i=1}^m I_{it}^-$	30	30	90	70	30	110	263,33	340

O valor da função objetivo é de 4.238 u.m., ou seja, menor que os 4.490 u.m. obtidos com o modelo D-FES. Isso foi possível uma vez que se excedeu menos a demanda ao deixar de utilizar a capacidade total do período, e também por fazer 2 trocas de processos a menos.

Para demonstrar uma diferença importante entre o modelo DLSP e o modelo CSLP discutida na seção 3.3.5, a demanda original apresentada no Quadro 7.2 e utilizada nos exemplos que resultaram nas Tabelas 7.1, 7.2 e 7.3, foi alterada de forma a tornar interessante a existência de períodos de produção totalmente ociosos entre períodos que utilizam o mesmo

processo de produção. Para isso foram concentradas demandas nos primeiros e últimos períodos de produção (Quadro 7.3). Os resultados são apresentados na Tabela 7.4.

Quadro 7.3 – Demanda forçada do produto  $i$  no período  $t$ .

	1	2	3	4	5	6	7	8
EK8A_16	0	0	0	0	0	0	60	70
EK8A_20	0	0	0	0	0	0	60	70
EK8A_24	0	0	0	0	0	0	60	70
EK8A_30	50	0	0	0	0	0	0	0
EK8A_36	60	0	0	0	0	0	0	0
EK8A_60	0	150	0	0	140	140	0	0
EK8A_80	0	150	0	0	140	140	0	0
EK8A_100	0	50	0	0	40	40	0	0

Tabela 7.4 - Programação da produção segundo os modelos D-FES e CS-FES com *backlog*.

		<i>Período (t)</i>							
		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
<b>D-FES</b>	$j$	6	2	-	-	2	2	8	8
	$x_{jt}$	1	1	-	-	1	-	1	-
<b>CS-FES</b>	$j$	6	2	2	2	2	2	8	8
	$x_{jt}$	1	1	-	-	-	-	1	-
	$q_{jt}$	0,857	1	-	-	0,933	0,933	0,857	1

Note que enquanto no modelo D-FES,  $y_{jt}$  indica a utilização do processo  $j$  no período  $t$ , no modelo CS-FES esta variável indica apenas que a linha de produção está preparada para o processo  $j$  ( $q_{jt}$  é o que define a utilização do processo no período). Essa diferença resulta em cobrança de *setup* no modelo D-FES no período 5, enquanto que no modelo CS-FES não, pois  $y_{jt}$  manteve-se igual a 1 nos períodos 3, 4 e 5. Com as novas demandas, foi obtido um custo total de 830 u.m. com o modelo D-FES, contra 358 u.m. do modelo CS-FES, devido à redução dos excessos e da quantidade de *setups*.

Para a empresa A, poucas vezes é interessante reduzir os excessos encurtando o turno de trabalho. Não obstante, vale à pena analisar o impacto nas soluções de uma mudança na principal característica da empresa; ao invés de utilizar apenas um processo por período, o

tempo que a linha ficaria ociosa seria utilizado para ligar outro processo. Assim, adiantando-se a produção que seria considerada apenas no período seguinte, a demanda seria mais bem atendida e um menor volume seria excedido. A fim de representar tal situação, a seção 7.3 descreve o modelo *Proportional Lot sizing and Scheduling Problem* com custos de falta, excesso e *setup* (P-FES).

### 7.3. Modelo compartilhado P-FES

Neste modelo, permite-se a utilização de até dois processos diferentes por período (prescrevendo uma única parada para troca de processos) a fim de evitar períodos com capacidade ociosa. Além disso, assim como no modelo CS-FES, permite-se a preservação da preparação de um período para o seguinte; ou seja, não se incorre em tempos/custos de *setup* se o último produto produzido no período  $t-1$  é o primeiro a ser produzido no período  $t$ .

Assim como no modelo CSLP-FES, as variáveis  $y_{jt}$  indicam se a linha de produção está preparada para o processo  $j$  no período  $t$ . Como dois processos podem ser utilizados em um mesmo período, deve ser evidente a ordem em que esses processos são utilizados. Isto é feito interpretando o estado do *setup* da seguinte maneira:  $y_{jt}$  é o estado de *setup* do processo ao final do período, ou seja,  $y_{jt}=1$  significa que processo  $j$  foi o último utilizado no período  $t$  e caso o período  $t+1$  seja iniciado com este mesmo processo não é cobrado *setup*. Devido a possibilidade de troca de processos ao longo do período, são considerados tempos de *setup* para cada processo.

#### **Formulação matemática:**

Variáveis:

$q_{jt}$ : fração de um período de produção utilizado com o processo  $j$  no período  $t$  ( $0 \leq q_{jt} \leq 1$ );

$x_{jt}$  variável binária que indica se houve *setup* para o item  $i$  no período  $t$  ( $x_{jt} = 1$ ) ou não ( $x_{jt} = 0$ );

$I_{it}^+$  excesso do item  $i$  ao final do período  $t$  ( $i = 1, \dots, m; t = 1, \dots, T$ );

$I_{it}^-$  falta do item  $i$  ao final do período  $t$  ( $i = 1, \dots, m; t = 1, \dots, T$ );

$y_{jt}$  indica se a linha está preparada para o processo  $j$  no início do período  $t$  ( $y_{jt} = 1$ ) ou não ( $y_{jt} = 0$ );

Parâmetros:

$st_j$ : fração de um período de produção utilizado com o tempo de *setup* requerido pelo processo  $j$  ( $0 \leq st_j \leq 1$ );

$s_j$  custo de *setup* do processo  $j$ ;

$g_{it}$  custo unitário de falta do item  $i$  no período  $t$ ;

$h_{it}$  custo unitário de estocagem do item  $i$  no período  $t$ .

$a_{ij}$  quantidade do produto  $i$  produzido pelo processo  $j$  no período  $t$  ( $i = 1, \dots, m; j = 1, \dots, J$ );

$d_{it}$  demanda do produto  $i$  no período  $t$  ( $i = 1, \dots, m; t = 1, \dots, T$ );

$y_{j0}$  indica se a linha está preparada para o processo  $j$  no início do período 1 ( $y_{j0} = 1$ ) ou não ( $y_{j0} = 0$ );

$$\text{Min } z = \left( \sum_{i=1}^m \sum_{t=1}^T h_{it} I_{it}^+ + g_{it} I_{it}^- \right) + \left( \sum_{t=1}^T \sum_{j=1}^J s_j x_{jt} \right) \quad (7.1)$$

s.a.

$$I_{it-1}^+ - I_{it-1}^- + \left( \sum_{j=1}^J a_{ij} q_{jt} \right) - I_{it}^+ + I_{it}^- = d_{it} \quad \forall i, t \quad (7.7)$$

$$q_{jt} \leq y_{j,t-1} + y_{jt} \quad \forall j, t \quad (7.9)$$

$$\sum_{j=1}^J (st_j x_{jt} + q_{jt}) \leq 1 \quad \forall t \quad (7.10)$$

$$\sum_{j=1}^J y_{jt} \leq 1, \quad \forall t \quad (7.3)$$

$$x_{jt} \geq y_{jt} - y_{j(t-1)}, \quad \forall j, t \quad (7.4)$$

$$x_{jt}, y_{jt} \in \{0,1\}, \quad \forall j, t \quad (7.5)$$

$$q_{jt}, I_{it}^+, I_{it}^- \geq 0 \quad \forall i, j, t \quad (7.11)$$

As restrições de capacidade em (7.9) permitem a utilização de até dois processos no mesmo período, onde neste caso, o primeiro deles também é o último processo utilizado no período anterior. Em (7.10), as restrições de capacidade levam em conta o tempo de *setup* do processo caso seja necessário, e em (7.11) encontram-se as demais restrições quanto ao domínio das variáveis. Assim como no CS-FES, caso existam períodos de inatividade entre períodos de produção com o mesmo processo finalizando/iniciando os períodos, os custos/tempos de *setup* são cobrados apenas uma vez. A implementação do modelo no *GAMS* pode ser vista no Apêndice IV.

Cabe enfatizar que apenas um *setup* por período é permitido e não que não é obrigatória a utilização de dois processos. Note que quando um *setup* é feito para o processo que inicia o período, não é possível utilizar outro processo ainda no mesmo período, pois o *setup* do período já terá sido utilizado para ligar o processo inicial. Observe na Tabela 7.5, que descreve a programação obtida com o modelo utilizando a demanda do Quadro 7.2, que essa restrição ocorre logo no período 1, onde o processo 8 necessita de um *setup* para dar início à produção no período.

Tabela 7.5. Exemplo de programação da produção para oito períodos segundo premissas do modelo P-FES com *backlog*.

	Período (t)							
	1	2	3	4	5	6	7	8
$x_{jt}$ com $j =$	8	4	6	-	-	-	7	-
$q_{jt}$ com $y_{jt} = 0$ (1º processo)	-	$j = 8$ 0,35	$j = 4$ 0,67	$j = 6$ 1	$j = 6$ 0,66	$j = 6$ 0,97	$j = 6$ 0,20	$j = 7$ 1
$q_{jt}$ com $y_{jt} = 1$ (2º processo)	$j = 8$ 0,67	$j = 4$ 0,33	$j = 6$ 0,01	-	-	-	$j = 7$ 0,50	-
$\sum_{i=1}^m I_{it}^+$	94	234	414	404	370	264	144	112
$\sum_{i=1}^m I_{it}^-$	53	28	88	68	28	106	370	328

Ainda na Tabela 7.5, as variáveis  $q_{jt}$  assumem valores que, levando em conta (quando necessário) o tempo de *setup* do processo (como no período 1), não ultrapassam a capacidade do período e, em outro momento, por exemplo, no período 4, permite que o processo 6 ocupe toda a capacidade do período. O tempo de *setup* do processo não incorre

neste período devido ao mesmo já ter incorrido no período anterior ( $t=3$ ). O modelo P-FES resultou em um custo total de 4.233 u.m., pouco melhor que as 4.238 u.m. do modelo CS-FES e das 4.490 u.m. do modelo D-FES. Vale ainda destacar que o modelo P-FES desconta em cada período que houve *setup*, seu respectivo tempo de preparação, o qual não é cobrado nos modelos D-FES e CS-FES.

Para empresas que trocam mais de uma vez de processo ao longo do mesmo período de produção (caso possível na empresa B), é necessário reescrever o modelo na forma de um modelo *General Lot-sizing and Scheduling Problem* com custos de falta, excesso e *setup* (G-FES), conforme discutido na seção 7.4.

#### 7.4. Modelo geral G-FES

##### **Formulação matemática:**

Variáveis:

- $x_{jn}$  variável binária que indica se ocorreu *setup* para o processo  $j$  no subperíodo  $n$  ( $x_{jn} = 1$ ) ou não ( $x_{jn} = 0$ );
- $y_{jn}$  variável binária que indica se a linha está preparada para o processo  $j$  ser utilizado no subperíodo  $n$  ( $y_{jn} = 1$ ) ou não ( $y_{jn} = 0$ ).
- $q_{jn}$  fração de um período de produção utilizado com o processo  $j$  no subperíodo  $n$  ( $0 \leq q_{jn} \leq 1$ );
- $I_{it}^+$  excesso do item  $i$  ao final do período  $t$  ( $i = 1, \dots, m; t = 1, \dots, T$ );
- $I_{it}^-$  falta do item  $i$  ao final do período  $t$  ( $i = 1, \dots, m; t = 1, \dots, T$ );

Parâmetros adicionais em relação ao modelo P-FES:

- $N_t$  número máximo de subperíodos no período  $t$ .
- $F_t = 1 + \sum_{t'=1}^{t-1} N_{t'}$  utilizado para denotar o primeiro subperíodo no período  $t$ .
- $L_t = F_t + N_t - 1$  utilizado para denotar o último subperíodo no período  $t$ .
- $st_j$ : fração de um período de produção utilizado com o tempo de *setup* requerido pelo processo  $j$  ( $0 \leq st_j \leq 1$ );

- $s_j$  custo de *setup* do processo  $j$ ;  
 $g_{it}$  custo unitário de falta do item  $i$  no período  $t$ ;  
 $h_{it}$  custo unitário de estocagem do item  $i$  no período  $t$ .  
 $a_{ij}$  quantidade do produto  $i$  produzido pelo processo  $j$  no período  $t$  ( $i = 1, \dots, m; j = 1, \dots, J$ );  
 $d_{it}$  demanda do produto  $i$  no período  $t$  ( $i = 1, \dots, m; t = 1, \dots, T$ );

$$\text{Min } z = \left( \sum_{i=1}^m \sum_{t=1}^T h_{it} I_{it}^+ + g_{it} I_{it}^- \right) + \left( \sum_{j=1}^J \sum_{n=1}^N s_j x_{jn} \right) \quad (7.12)$$

s.a.

$$I_{it-1}^+ - I_{it-1}^- + \left( \sum_{n=F_t}^{L_t} \sum_{j=1}^J a_{ij} q_{jn} \right) - I_{it}^+ + I_{it}^- = d_{it} \quad \forall i, t \quad (7.13)$$

$$q_{jn} \leq y_{j(n-1)} + y_{jn} \quad \forall j, n \quad (7.14)$$

$$\sum_{n=F_t}^{L_t} \sum_{j=1}^J (s_j x_{jn} + q_{jn}) \leq 1 \quad \forall t \quad (7.15)$$

$$\sum_{j=1}^J y_{jn} \leq 1, \quad \forall n \quad (7.16)$$

$$x_{jn} \geq y_{jn} - y_{j(n-1)}, \quad \forall j, n \quad (7.17)$$

$$x_{jn}, y_{jn} \in \{0,1\}, \quad \forall j, n \quad (7.18)$$

$$q_{jn}, I_{it}^+, I_{it}^- \geq 0 \quad \forall i, j, t, n \quad (7.19)$$

A função objetivo (7.12) minimiza os custos de falta (com *backlog*), custos de estocagem e de *setup*. Em (7.13) é feito o balanceamento de estoques. Note que um determinado processo pode ser utilizado em vários subperíodos no período. As inequações (7.14) garantem que se o processo  $j$  for programado para a posição  $p$ , a linha estará preparada para ele. As restrições de capacidade são descritas por (7.15), com o somatório  $\sum_{p=F_t}^{L_t}$  do lado esquerdo de (7.15) restringindo a quantidade de subperíodos em cada período  $t$ . Com as restrições (7.16), apenas um processo está preparado para produção no subperíodo. O subperíodo  $p$  no qual o *setup* deve ser cobrado é determinado pelas inequações em (7.17), e as

restrições de domínio das variáveis são definidas em (7.18) e (7.19). A implementação do modelo no *GAMS* pode ser vista no Apêndice V.

### 7.5. Modelo G-FES com custos de *setup* nulos no início do período de trabalho

Com o intuito de encontrar uma formulação que possa ser mais apropriada para o problema da empresa A, e levando em conta que a mesma atualmente considera que os custos de *setup* na troca de processos são nulos quando realizado no início do período de produção, a equação (7.17) do modelo anterior é substituída por:

$$x_{jn} \geq y_{jn} - y_{jn-1} \quad \forall j, t, n \neq F_t \quad (7.17')$$

Esta modificação do domínio da restrição permite tratar o caso em que não existem custos de *setup* entre períodos, mas, somente para as trocas ao longo destes. Custos e tempos de *setup* foram admitidos como independentes da sequência.

Note que (7.17') faz com que  $y_{jF_t} = 0$  ( $\forall j, t$ ). Desta forma, a quantidade de períodos em que incorre custos/tempos de *setup* poderá diminuir consideravelmente, pois o modelo evita a troca de processos ao longo do período, preferindo fazer essa troca no primeiro subperíodo de produção dos períodos.

Com o modelo G-FES, concluem-se as variações criadas a partir dos modelos MFP e MFEP. Os modelos apresentados podem ainda ser reformulados para priorizar os produtos por contribuição de lucro, ordens urgentes e clientes preferenciais, entre outros (veja, por exemplo, Munhoz & Morabito, 2001).

Cabe mencionar que a instância real foi utilizada também na resolução dos modelos D-FES, CS-FES e P-FES. Devido à dificuldade em obter custos reais de excesso, falta e de *setup* da empresa, os valores foram estimados e podem ser vistos nos apêndices II a V. A solução ótima do modelo D-FES foi obtida em apenas 2 minutos, enquanto que o CPLEX requereu 7.200 segundos para a resolução do modelo CS-FES. Com o modelo P-FES, por sua vez, obteve-se um *gap* de 39% em 10.800 segundos. Estes resultados mais uma vez reforçam o fato de que ao serem adicionados custos de excesso, *setup* e tempos de *setup*, os modelos tornam-se difíceis de serem resolvidos otimamente.



Atualmente o modelo MFP representa razoavelmente bem a expectativa da gerência da empresa A referente a uma ferramenta computacional de apoio ao PCP da empresa. Porém, instituída uma cultura de análise de tempos/custos de *setup* e excessos de produção, acredita-se que o modelo irá evoluir dentro da empresa gradativamente na ordem apresentada neste capítulo:

D-FES  $\Rightarrow$  CS-FES  $\Rightarrow$  P-FES  $\Rightarrow$  G-FES

Para a empresa B, tem-se a expectativa da gerência empregar inicialmente, um modelo como o P-FES. Porém, a empresa ainda não conseguiu quantificar os custos/tempos de *setup* necessários para obter boas programações com a ferramenta computacional.

## CONCLUSÕES

---

### 8.1. Conclusões

Este trabalho abordou um problema de relevância prática no planejamento e controle da produção na indústria de grãos eletrofundidos. Com vistas à contextualização, foram inicialmente apresentadas as principais matérias-primas empregadas e produtos associados ao setor de eletrofundidos. Foram também descritos o processo de produção na indústria de grãos eletrofundidos e, em particular, foi discutido o problema de classificação dos grãos associado ao problema de dimensionamento de lotes de processos.

Foram também apresentados modelos matemáticos para apoio às decisões de planejamento e controle da produção em sistemas de manufatura com vistas à seleção de processos em indústrias de grãos eletrofundidos. Um modelo de minimização da falta de produção (modelo MFP) foi revisitado e estendido para situações diversas que ocorrem em empresas desse setor. O modelo foi resolvido utilizando-se diferentes conjuntos de exemplos gerados aleatoriamente e as versões 7 e 11 do *solver* GAMS/CPLEX, baseado no método *branch-and-cut*. Diferentes estratégias de produção foram também modeladas a fim de discutir possíveis cenários ainda não explorados pela empresa onde foi realizado o estudo de caso, tais como a consideração de custos e/ou tempos de *setup* de produção, e minimização de estoques, tendo sempre como meta principal o melhor atendimento da demanda dos clientes.

Uma heurística construtiva que seleciona processos com base em uma função de avaliação gulosa foi proposta e implementada para resolução do modelo de minimização de falta de produção (MFP). A mesma função de avaliação foi também utilizada em duas variantes de heurísticas de busca local e em uma implementação de meta-heurística GRASP. Os resultados obtidos com a aplicação da heurística e da meta-heurística GRASP permitiram avaliar seu desempenho relativo às soluções do *solver*, tanto em termos de qualidade de solução como de tempo computacional. A implementação da meta-heurística GRASP obteve,

em geral, os melhores resultados dentre as heurísticas testadas. Apesar destes resultados terem, em geral, qualidade inferior aos obtidos com o *solver* CPLEX, o tempo computacional e a degradação relativamente pequena da qualidade das soluções em relação ao *solver* mostram que o algoritmo é competitivo e pode ser empregado como alternativa aos pacotes de soluções comerciais, como o GAMS/CPLEX.

Os resultados obtidos com uma instância real fornecida por uma das empresas estudadas indicaram que os métodos heurísticos (assim como o método *branch-and-cut* do CPLEX) são capazes de produzir soluções tão boas ou melhores que as atualmente praticadas pela empresa. Devido aos tempos computacionais relativamente baixos para a aplicação em questão, tais abordagens permitem que sejam feitas várias simulações de programação da produção (explorando diferentes cenários), o que fornece flexibilidade e eficácia aos tomadores de decisão. Além disso, os métodos propostos também facilitam a aplicação de técnicas de horizonte rolante, e permitem aos departamentos de produção e vendas da empresa analisar rapidamente a incorporação de novos pedidos ao longo do horizonte de planejamento.

Cabe mencionar que a instância real foi utilizada também na resolução dos modelos D-FES, CS-FES e P-FES discutidos no capítulo 7. A solução ótima do modelo D-FES foi obtida em apenas 2 minutos, enquanto que o CPLEX requereu 7.200 segundos para a resolução do modelo CS-FES. Com o modelo P-FES, por sua vez, obteve-se um *gap* de 39% em 10.800 segundos. Estes resultados mais uma vez reforçam o fato de que ao serem adicionados custos de excesso, *setup* e tempos de *setup*, os modelos tornam-se difíceis de serem resolvidos otimamente.

No que diz respeito aos experimentos com as versões 7 e 11 do *solver* GAMS/CPLEX na resolução do modelo de minimização de faltas, a diferença de desempenho observada evidenciou uma grande evolução na resolução de problemas de programação inteira mista (MIP). A utilização de 4 núcleos de processamento (*threads*) permitiu a obtenção de soluções com *gaps* e tempos médios de execução consideravelmente menores em relação à utilização de apenas um núcleo. Segundo a Ilog, fabricante do *solver*, a nova versão (12) que está sendo disponibilizada apresenta melhorias adicionais para resolver modelos MIP, o que poderá fazer com que o tempo de obtenção de algumas soluções diminua e até mesmo aumente o número de certificados de otimalidade encontrados. Testes com a versão mais recente de *solvers* de modelos MIP, portanto, são opções que devem ser consideradas na sequência deste trabalho.

Dentre as maiores dificuldades encontradas no desenvolvimento deste estudo, pode-se considerar a confiabilidade dos dados de demanda real. Como foi constatado em visitas realizadas nas empresas, não há um controle acurado das quantidades dos produtos em estoque e em produção. Isso se dá principalmente porque um ou mais produtos podem ser misturados em quantidades diferentes para formar um outro item. O item resultante da mistura é então utilizado para atender a demanda de clientes menos exigentes quanto à qualidade da faixa granulométrica. Esse processo de mistura torna difícil o gerenciamento dos estoques por meio dos sistemas de informação. Devido à isso, apenas uma instância real confiável foi utilizada nos testes e, à partir dela, geradas pouco mais de uma centena de novas instâncias aleatoriamente.

Os resultados com a instância real agradaram a direção e engenheiros da empresa, os quais demonstraram interesse na continuação dos estudos. Neste sentido, se as empresas contarem com uma boa estimativa de custos de falta, excesso e *setup*, os modelos apresentados no capítulo 7 serão mais apropriados e representarão melhor a realidade do setor industrial. O uso de tais modelos poderá trazer um melhor planejamento da produção, embora os testes com a instância real tenham mostrado que são mais difíceis de serem resolvidos.

## 8.2. Perspectivas futuras

Dentre as perspectivas de pesquisa futura cabem destacar:

- A inclusão nos modelos de decisões de mistura de dois ou mais produtos para produção de outro produto. Essa prática ocorre diariamente na empresa, porém, não é gerenciada, ficando a cargo de um funcionário do setor escolher quais produtos (grãos) e quantidades serão misturadas para gerar o item com demanda ainda não atendida. A falta de um padrão de mistura torna o processo dependente do conhecimento do funcionário responsável.
- Uma extensão dos modelos e algoritmos pode ser feita para a consideração de incertezas nos parâmetros de entrada, de forma a serem tratados por técnicas de programação estocástica e otimização robusta.
- Para o tratamento de instâncias difíceis, podem ser utilizadas heurísticas de programação matemática do tipo *relax and fix*.

- Diversos processos de produção de grãos eletrofundidos podem ser formados levando em consideração a demanda e as restrições de produção da empresa, tornando importante o desenvolvimento de um procedimento de geração de colunas para representar tais processos e sua incorporação em um método exato do tipo *branch-and-price* para obtenção de soluções ótimas.
- Apesar dos bons resultados obtidos com as técnicas heurísticas empregadas neste trabalho, o desenvolvimento de implementações de meta-heurísticas alternativas, em particular, de otimização por colônia de formigas (ACO) poderá levar a soluções de melhor qualidade e/ou menor esforço computacional.
- O modelo e as meta-heurísticas empregadas como ferramenta de apoio à tomada de decisão devem ser capazes de gerar vários cenários de produção, permitindo ao tomador de decisão escolher a melhor estratégia a ser seguida utilizando critérios não presentes no objetivo analisado pela ferramenta. Um exemplo é o atendimento diferenciado (melhor nível de serviço) a clientes preferenciais.
- O estudo realizado pode ser estendido para outras áreas de aplicação similares além dos eletrofundidos, tal como minas; na produção de carvão, calcário, brita, areia e outros minérios metálicos e não metálicos.
- A integração do modelo e métodos heurísticos de programação da produção com o ERP (*Enterprise Resource Planning*) da empresa poderá ajudar também no controle dos estoques e, como consequência, fornecer maior confiabilidade nos dados de entrada.
- Melhores estimativas quanto ao consumo de energia para produção e impacto do despejo de resíduos no meio ambiente, podem ser levados em conta pelo modelo e métodos a fim de provocar a menor degradação.
- Estudos sobre os tempos e custos de *setup* devem ser realizados, a fim de tornarem viáveis as trocas de processo durante um período de produção.

## REFERÊNCIAS BIBLIOGRÁFICAS

---

**AARTS, E.H.L.; REIJNHOUDT, M.F., STEHOUWER, H.P., WESSELS, J.** *A novel decomposition approach for on-line lot sizing.* European Journal of Operational Research, v.122, p.339-353, 2000.

**ABC,** Associação Brasileira de Cerâmica - A Indústria de Refratários no Brasil. Disponível em: <[http://www.abceram.org.br/asp/togni.htm?cd\\_noti=201](http://www.abceram.org.br/asp/togni.htm?cd_noti=201)>. Acesso em: 12 maio.2009.

**AKBALIK, A.; PENZ, B.** *Exact methods for single-item capacitated lot sizing problem with alternative machines and piece-wise linear production costs.* International Journal of Production Economics, v.119, n.2, p.367-379, 2009.

**AKSEN, D.** *Loss of customer goodwill in the uncapacitated lot-sizing problem.* Computers and Operations Research, v.34, p.2805-2823, 2007.

**ALFIERI, A.; BRANDIMARTE, P.; ORAZIO, S.** *LP-based heuristics for the capacitated lot-sizing problem: The interaction of model formulation and solution algorithm.* International Journal of Production Research v.40, n.2, p.441-458, 2002.

**ALIDAEI, B.; KOCHENBERGER, G.A.; AMINI, M.M.** *Greedy solutions of selection and ordering problems.* European Journal of Operational Research, v.134, n.1, p.203-215, doi:10.1016/S0377-2217(00)00252-6, 2001.

**ALLEN, T.** *Particle size measurement.* 4 th. Ed. Londres: Chapman and Hall, 806p, 1990.

**ANFACER,** Associação Nacional dos Fabricantes de Cerâmica para Revestimento. Disponível em: <<http://www.anfacer.org.br/>>. Acesso em: 14 maio.2009.

**ARAÚJO, S. A., ARENALES, M. N., CLARK, A. R.** *Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups,* Journal of Heuristics, v.13, n.4, p.337-358, doi: 10.1007/s10732-007-9011-9, 2007.

**ARAÚJO, S. A., ARENALES, M. N., CLARK, A. R.** Dimensionamento de lotes e programação do forno numa fundição de pequeno porte. Gestão & Produção, v.11, n.2, p.165-176, doi: 10.1590/S0104-530X2004000200003, 2004.

**ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H.** Pesquisa Operacional. Rio de Janeiro: Editora Campus/Elsevier, 2007.

**ARMENTANO, V. A.; FRANÇA, P.M.; TOLEDO, F.M.B.** *A network flow model for the capacitated lot-sizing problem.* Omega International Journal of Management Science v.27, n.2, p.275-284, 1999.

**ASKIN, R.; STANDRIDGE, C.** *Modelling and analysis of manufacturing systems.* New York: Wiley & Sons, 1993.

**BASNET, C.; LEUNG, J.M.Y.** *Inventory lot-sizing with supplier selection.* Computers and Operations Research, v.32, p.1-14, 2005.

**BELTRÁN, J. L.; KRASS, D.** *Dynamic lot sizing with returning items and disposals.* IIE Transactions, v.34, p.437-448, 2002.

**BELVAUX, G.; WOLSEY, L.** *Modelling practical lot-sizing problems as mixed-integer programs.* Management Science, v.47, n.7, p.993-1007, 2001.

**BELVAUX, G.; WOLSEY, L.** *A specialized branch-and-cut system for lot-sizing problems.* Management Science, v.46, n.5, p.724-738, 2000.

**BERRETTA, R.; FRANÇA, P. M.; ARMENTANO, V. A.** *Metaheuristic Approaches for the Multilevel Resource-Constrained Lot-Sizing Problem with Setup and Lead Times.* Asia Pacific Journal Of Operational Research, Singapura, v.22, n.2, p.261-286, 2005.

**BERTRAND, J. W. M.; FRANSOO, J. C.** *Modelling and simulation – operations management research methodologies using quantitative modeling.* International Journal of operations & Production Management, v.22, n.3, p.241-264, 2002.

**BITRAN, G. R.; MATSUO, H.** *Approximation formulations for the single-product capacitated lot size problem.* Operations Research v.34, n.1, p.63-74, 1986.

**BITRAN, G. R.; YANASSE, H.H.** *Computational complexity of the capacitated lot size problem.* Management Science 28, 10, 1174-1186, 1982.

**BRAHIMI, N.; DAUZERE-PERES, S; NAJID, N. M.** *Capacitated Multi-Item Lot-Sizing Problems with Time Windows.* Operations Research, v.54, n.5, p.951-967, doi: [doi: 10.1287/opre.1060.0325](https://doi.org/10.1287/opre.1060.0325), 2006a.

**BRAHIMI, N.; DAUZERE-PERES, S; NAJID, N. M.; NORDLI, A.** *Single item lot sizing problems.* European Journal of Operational Research, v.168, n.1, p.1-16, doi:[10.1016/j.ejor.2004.01.054](https://doi.org/10.1016/j.ejor.2004.01.054), 2006b.

**BRANDIMARTE, A.; ALFIERI, A.; LEVI, R.** *LP-Based heuristics for the capacitated lot sizing problem.* CIRP Annals Manufacturing Technology, v.47, n.1, p.423-426, doi:[10.1016/S0007-8506\(07\)62866-2](https://doi.org/10.1016/S0007-8506(07)62866-2), 1998.

**BROOKE, A.; KENDRICK, D.; MEERAUS, A.** *GAMS: a user's guide (release 2.25).* San Francisco: The Scientific Press, 1992.

**BRUGGEMANN, W.; JAHNKE, H.** *The discrete lot-sizing and scheduling problem: Complexity and modification for batch availability.* European Journal of Operational Research, v.124, n.2, p.511-528, doi:[10.1016/S0377-2217\(99\)00190-3](https://doi.org/10.1016/S0377-2217(99)00190-3), 2000.

**BUSCHKÜHL, L.; SAHLING, F.; HELBER, S.; TEMPELMEIER, H.** *Dynamic capacitated lot-sizing problems: a classification and review of solutions approaches*, OR Spectrum, v.32, p.231-261, doi: 10.1007/s00291-008-0150-7, 2010.

**CALLISTER, W. D.** *Materials Science and Engineering: An Introduction*. 7<sup>th</sup> ed. John Wiley & Sons, 2006.

**CARRIJO, E. C.** Processamento paralelo aplicado em análise não linear de cascas. Tese. São Carlos, SP, Brasil: Engenharia de Estruturas, Escola de Engenharia de São Carlos, 2001.

**CHANG, D.S.; CHYR, F.C.; YANG, F.C.** *Incorporating a database approach into large-scale multi-level lot sizing problem*. Computers & Mathematics with Applications, v.60, n.9, p.2536-2547, doi:10.1016/j.camwa.2010.08.037, 2010.

**CHF, Chemical Heritage Foundation** – *Eletrochemistry and eletrochemical industries*. Disponível em: <<http://www.chemheritage.org/classroom/chemach/electrochem/index.html>>. Acesso em: 12 maio.2009.

**CLARK, A.; MORABITO, R.; TOSO, E.** *Production setup-sequencing and lot-sizing at an animal nutrition plant through ATSP subtour elimination and patching*. Journal of Scheduling, v.13, n.2, p.111-121, doi: 10.1007/s10951-009-0135-7, 2010.

**CLARK, A. R.; CLARK, S. J.** *Rolling-horizon lot-sizing when set-up times are sequence-dependent*. International Journal of Production Research, v.38, n.10, p.2287-2307, 2000.

**COLIN, E. C.** Pesquisa Operacional: 170 aplicações em estratégia, finanças, logística, produção, marketing e vendas. Editora LTC, 2007.

**CONTADOR, J. C.; CONTADOR, J. L.** Programação e controle da produção para a indústria intermitente. In: CONTADOR, J. C. (Coordenador): Gestão de operações. Editora Edgard Blücher, 1997.

**CRUZ, C.R.V.** Refratários para equipamentos siderúrgicos. Editora Édile, 1982.

**DASTIDAR, S.G.; NAGI, R.** *Scheduling injection molding operations with multiple resource constraints and sequence dependent setup times and costs*. Computers & Operations Research, v.32, n.11, p.2987-3005, doi:10.1016/j.cor.2004.04.012, 2005.

**DESSOUKY, M.M.; KIJOWSKI, B.A.; VERMA, S.** *Simultaneous batching and scheduling for chemical processing with earliness and tardiness penalties*. Production and Operations Management, v.8, n.4, p.433-444, 1999.

**DOGAN, M.E.; GROSSMANN, I.E.** *A decomposition method for the simultaneous planning and scheduling of single-stage continuous multiproduct plants*. Industrial & Engineering Chemistry Research, v.45, n.1, p.299-315, doi:10.1021/ie050778z, 2006.

**DREXL, A.; KIMMS, A.** *Lot sizing and scheduling: survey and extensions*. European Journal of Operational Research, v.99, n.2, p.221-235, doi:10.1016/S0377-2217(97)00030-1, 1997.

**DREXL, A.; HAASE, K.** *Sequential-Analysis Based Randomized-Regret-Methods for Lot-Sizing and Scheduling*, Operational Research Society, v. 47, n.2, p. 251-265, 1996.



**DREXL, A.; HAASE, K.** *Proportional lot sizing and scheduling*. International Journal of Production Economics, v.40, n.1, p.73-87, [doi:10.1016/0925-5273\(95\)00040-U](https://doi.org/10.1016/0925-5273(95)00040-U), 1995.

**ELFUSA.** Elfusa eletrosusão. Disponível em: <http://www.elfusa.com.br/br/index.htm>. Acesso em: 18 ago. 2011.

**FEDERGRUEN, A.; MEISSNER, J.; TZUR, M.** *Progressive interval heuristics for multi-item capacitated lot-sizing problems*. Management Science. v.55, n.3, p.490-502, [doi:10.1287/opre.1070.0392](https://doi.org/10.1287/opre.1070.0392), 2007.

**FENG, H.; CHENG, H.** *Solving mixed integer programming production planning problems with setups by shadow price information*. Computers & Operations Research 25, 12, 1027-1042, 1998.

**FEO, T. A.; RESENDE, M.G.** *A probabilistic heuristic for a computationally difficult set covering problem*. Operations Research Letters, v.8, p.67-71, 1989.

**FERREIRA, D.; MORABITO, R.; RANGEL, S.** *“Relax and fix heuristics to solve one-stage one-machine lot-scheduling models for small-scale soft drink plants”*, Computers & Operations Research, v.37, n.4, 684-691, [doi:10.1016/j.cor.2009.06.007](https://doi.org/10.1016/j.cor.2009.06.007), 2010.

**FERREIRA, D.; MORABITO, R.; RANGEL, S.** *“Solution approaches for the soft drink integrated production lot sizing and scheduling problem”*, European Journal of Operational Research, v.196, p.697-706, 2009.

**FERREIRA, D.** Abordagens para o problema integrado de dimensionamento e sequenciamento de lotes da produção de bebidas. Tese. São Carlos, SP, Brasil: Engenharia de Produção, Universidade Federal de São Carlos, 2006.

**FLEISCHMANN, B.; MEYR, H.** *The General Lotsizing and Scheduling Problem*, OR Spectrum. v.19, n.1, p.11-21, [doi: 10.1007/BF01539800](https://doi.org/10.1007/BF01539800), 1997.

**FLEISCHMANN, B.** *The discreet lot sizing and scheduling problem with sequence-dependent setup-costs*, European Journal of Operational Research. v.75, n.2, p.395-404, [doi:10.1016/0377-2217\(94\)90083-3](https://doi.org/10.1016/0377-2217(94)90083-3), 1994.

**FLEISCHMANN, B.** *The discrete lot-sizing and scheduling problem*. European Journal of Operational Research. v.44, p.337-348, 1990.

**FLESZAR, K.; HINDI, K.S.** *Solving the resource-constrained project problem by a variable neighbourhood scheduling search*. European Journal of Operational Research 155, 2, 402-413, 2004.

**GAAFAR, L.K.; CHOUYEIKI, M.H.** A neural network model for solving the lot-sizing problem. Omega, v.28, p.175-184, 2000.

**GAO, L., ALTAY, N., ROBINSON, E.P.** *A comparative study of modeling and solution approaches for the coordinated lot-size problem with dynamic demand*. Mathematical and Computer Modelling, v.47, n.11-12, p.1254-1263, [doi:10.1016/j.mcm.2007.02.035](https://doi.org/10.1016/j.mcm.2007.02.035), 2008.

**GERSHWIN, S.** *Manufacturing systems engineering*. Prentice Hall, New Jersey, 1994.

**GICQUEL, C.; MIEGEVILLE, N.; MINOUX, M.; DALLERY, Y.** *Discrete lot sizing and scheduling using product decomposition into attributes*. Computers & Operations Research, v.36, n.9, p.2690-2698, doi:10.1016/j.cor.2008.11.017, 2009.

**GOPALAKRISHNAN, M.; DING, K.; BOURJOURJOLLY, J. M.; MOHAN, S.** *A tabu-search heuristic for the capacitated lot-sizing problem with set-up carryover*. Management Science, v.47, n.6, p.851-863, 2001.

**GRAVES, S.C.; RINNOOY, A.H.G.; ZIPKIN, P.H.** *Logistics of production and inventory*. Handbook in operations research and management science, Amsterdam: North-Holland, 1993.

**GUPTA, D.; MAGNUSSON, T.** *The capacitated lot-sizing and scheduling problem with sequence-dependent setup costs and setup times*. Computers & Operations Research, v.32, n.4, p.727-747, 2005.

**HAASE, K.; KIMMS, A.** *Lot sizing and scheduling with sequence-dependent setup costs and times and efficient rescheduling opportunities*. International Journal of Production Economics v.66, n.2, p.159-169, 2000.

**HAX, A.; CANDEA, D.** *Production and inventory management*. New Jersey: Prentice-Hall; Englewood Cliffs, 1984.

**HEUVEL, W.; WAGELMANS, A. P. M.** *Four equivalent lot-sizing models*. Operations Research Letters, v.36, p.465-470, 2008.

**HEUVEL, W.; WAGELMANS, A. P. M.** *A comparison of methods for lot-sizing in a rolling horizon environment*. Operations Research Letters, v.33, p.486-496, 2005.

**HIND, K. S.** *Solving the single-item capacitated dynamic lot sizing problem with startup and reservation costs by tabu search*. Computers and Industrial Engineering, v.28, n.4, p.701-707, 1995.

**HOLLAND, J. H.** *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, 1975.

**HSU, W.** *On the general feasibility test of scheduling lot sizes for several products on one machine*. Management Science, v.29, p.93-105, 1983.

**ILOG.** *High – performance software for mathematical programming and optimization*. <<http://www.ilog.com/products/cplex/>>. Acesso em: 05 maio.2009.

**JANS, R.; DEGRAEVE, Z.** *Modeling industrial lot sizing problems, a review*. International Journal of Production Research, v.46, n.6, p.1619-1643, 2008.

**JANS, R.; DEGRAEVE, Z.** *Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches*. European Journal of Operational Research, v.177, n.3, p.1855-1875, doi:10.1016/j.ejor.2005.12.008, 2007.

**JARUPHONGSA, W.; LEE, C. Y.** *Dynamic lot-sizing problem with demand time windows and container-based transportation cost*. Optimization Letters, v.2, n.1, p.39-51, doi:10.1007/s11590-006-0042-1, 2007.

**JOHNSON, L. A.; MONTGOMERY, D.C.** *Operations research in production planning, scheduling and inventory control*. New York: Wiley, 1974.

**JORDAN, C.; DREXL, A.** *Discrete lotsizing and scheduling by batch sequencing*. *Management Science*, v.44, n.5, p.698-713, doi:[10.1287/mnsc.44.5.698](https://doi.org/10.1287/mnsc.44.5.698), 1998.

**KARIMI, B.; GHOMI, S.M.T.F; WILSON, J.M.** *The capacitated lot sizing problem: a review of models and algorithms*. *Omega International Journal of Management Science* v.31, n.5, p.365-378, doi:[10.1016/S0305-0483\(03\)00059-8](https://doi.org/10.1016/S0305-0483(03)00059-8), 2003.

**KARMARKAR, U. S.; KEKRE, S.; KEKRE, S.** *The dynamic lot sizing problem with startup and reservation costs*. *Operations Research* v.35, n.3, p.389-398, 1987.

**KIMMS, A.** *A genetic algorithm for multi-level, multi-machine lot sizing and scheduling*. *Computers & Operations Research* v.26, p.829-848, 1999.

**KIMMS, A.** *Multi-level, single-machine lot sizing and scheduling (with initial inventory)*. *European Journal of Operational Research* v.89, n.1, p.86-99, 1996a.

**KIMMS, A.** *Competitive methods for multi-level lot sizing and scheduling: Tabu search and randomized regrets*. *International Journal of Production Research* v.34, n.8, p.2279-2298, 1996b.

**KOCLAR, A.** *The general lot sizing and scheduling problem with sequence dependent changeovers*. Tese. Ankara, Turkey: Department of Industrial Engineering, Middle East Technical University, 2005.

**LAMBRECHT, M. R.; VANDERVEKEN, H.** *"A heuristic procedures for the single operation multiitem loading problem"*, *AIIE Transactions*, v.11, n.4, p.319-326, 1979.

**LASDON, L. S, TERJUNG, R.C.** *Na eficiente algorithm for multi-item scheduling*. *Operations Research*, v.19, n.4, p.946-969, 1971.

**LAWRENCE, H.V.V.** *Elements of materials science and engineering*. 5<sup>th</sup> ed. Longman Higher Education, 1985.

**LEE, C. Y.; ÇETINKAYA, S.; WAGELMANS, A.P.M.** *A dynamic lot-sizing model with demand time windows*, *Management Science*, v.47, n. 10, p.1384-1395, 2001.

**LIN, C.K.Y.; WONG, C.L.; YEUNG, Y.C.** *Heuristic approaches for a scheduling problem in the plastic molding department of an audio company*. *Journal of Heuristics*, v.8, n.5, p.515-540, doi:[10.1023/A:1016588608032](https://doi.org/10.1023/A:1016588608032), 2002.

**LOBO, B.A.; KLABJAN, D.; CARRAVILLA, M.A.; OLIVEIRA, J.F.** *Single machine multi-product capacitated lot sizing with sequence-dependent setups*. *International Journal of Production Research*, v.45, n.20, p.4873-4894, 2007.

**LOZANO, S.; LARRANETA, J.; ONIEVA, L.** *Primal-dual approach to the single level capacitated lot-sizing problem*. *European Journal of Operational Research*, v.51, n. , p.354-366, 1991.

**LUCHE, J.R.D.; MORABITO, R.; PUREZA, V.** *Combining process selection and lot sizing models for the production scheduling of electrofused grains.* Asia-Pacific Journal of Operations Research, v.26, n. 3, p. 421-443, doi: [10.1142/S0217595909002286](https://doi.org/10.1142/S0217595909002286), 2009.

**MAES, J.; MCCLAIN, J.O.; WASSENHOVE, L.N.** *Multilevel capacitated lot sizing complexity and LP-based heuristics.* European Journal of Operational Research 53, 2, 131-148, 1991.

**MAES, J.; VAN WASSENHOVE, L.N.** *Multi-item single-level capacitated dynamic lot-sizing heuristics: a general review,* Journal of the Operational Research Society, v.39, n. , p.991-1004, 1988.

**MAES, J.; VAN WASSENHOVE, L.N.** *Multi-item single-level capacitated dynamic lot-sizing heuristic: a computational comparison (Part I: Static case),* IIE Transactions, v.18, n. , p.114-123, 1986.

**MANNE, A.S.** *Programming of economic lot sizes.* Management Science. v.4, p.115-135, 1958.

**MATEUS, G.R.; RESENDE, M.G.C.; SILVA, R. M. A.** GRASP: Procedimentos de busca gulosos, aleatórios, e adaptativos. Submetido ao Manual de Computação Evolutiva e Meta-heurística, Editora UFMG, Outubro de 2009.

**MATTA, R.; GUIGNARD, M.** *Production scheduling with sequence independent changeover cost.* Pennsylvania: The Wharton School; University of Pennsylvania, 1989.

**MELO, R.A.; WOLSEY, L.** *Uncapacitated two-level lot-sizing.* Operations Research Letters, v.38, p.241-245, 2010.

**MENEZES, A.A.; CLARK, A.; LOBO, B. A.** *Capacitated lot-sizing and scheduling with sequence-dependent, period-overlapping and non-triangular setups.* Journal of Scheduling, doi: [10.1007/s10951-010-0197-6](https://doi.org/10.1007/s10951-010-0197-6), 2010.

**MERCE, C.; FONTAN, G.** *MIP-based heuristics for capacitated lotsizing problems.* International Journal of Production Economics, v.85, n.1, p.97-111, [doi:10.1016/S0925-5273\(03\)00090-2](https://doi.org/10.1016/S0925-5273(03)00090-2), 2003.

**MEYR, H.** *Simultaneous lot sizing and scheduling on parallel machines.* European Journal of Operational Research, v.139, n. , p.277-292, 2002.

**MEYR, H.** *Simultaneous lot sizing and scheduling by combining local search whit dual reoptimization.* European Journal of Operational Research v.139, n. , p.311-326, 2000.

**MILLER, A.J.; NEMHAUSER, G.L.; SAVELSBERGH, M.W.** *Solving multi-item capacitated lot-sizing problems with setup times by branch-and-cut.* Technical Report, v.39, Université Catholique de Louvain - Center for Operations Research and Economics, 2000.

**MOLINA, F.; SANTOS, M. O.; TOLEDO, F. M. B.; ARAUJO, S. A.** *An approach using lagrangean/surrogate relaxation for lot-sizing with transportation costs.* Pesquisa Operacional, v.29, p.269-288, 2009.

**MORABITO, R.; PUREZA, V.** Metodologia de pesquisa em engenharia de produção e gestão de operações. Capítulo 8 – Modelagem e simulação. Campus, 2010.

**MORTENSEN, A.** *Concise encyclopedia of composite materials*. 2ed. Elsevier, 2006. Disponível em <[http://books.google.com.br/books?id=zs\\_lGeGsuaAC&printsec=frontcover](http://books.google.com.br/books?id=zs_lGeGsuaAC&printsec=frontcover)>. Acesso em: 15 jul.2009.

**MUNHOZ, J. R.; MORABITO, R.** Um modelo baseado em programação linear e programação de metas para análise de um sistema de produção e distribuição de suco concentrado congelado de laranja. *Gestão e Produção (UFSCar)*, v.8, n.2, p.139-159, 2001.

**NAHMIA, S.** *Production and operation analysis*. 5<sup>th</sup> Edition, McGraw-Hill Irwin, New York, 2005.

**PAN, Z.; TANG, J.; LIU, O.** *Capacitated dynamic lot sizing problems in closed-loop supply chain*. *European Journal of Operational Research*, v.198, n.3, p.810-821, doi:10.1016/j.ejor.2008.10.018, 2009.

**PARVEEN, S.; HAQUE, A. F. M.** *A heuristic solution of multi-item single level capacitated dynamic lot-sizing problem*. *Journal of Mechanical Engineering*, v.38, p.1-7, doi:10.3329/jme.v38i0.893, 2007.

**PERALTA, M.** Peneiramento industrial - apontamentos de aula. Disponível em <<http://www.ebah.com.br/content/ABAAAAlhkAL/peneiramento-industrial-apontes-aula-parte-1>>. Acesso em: 15 out.2011.

**PILEGGI, R.; ORTEGA, F.; MORABITO, R.; VENDRASCIO, S. PANDOLFELLI, V.** Desenvolvimento e aplicação de um software que automatiza o processo de combinação de matérias-primas na obtenção de produtos cerâmicos. *Ceramica*, v.44, n.289, p.189-195, 1998.

**POLTRONIERE, S. C.; POLDI, K. C.; TOLEDO, F. M. B.; ARENALES, M. N.** *A coupling cutting stock-lot sizing problem in the paper industry*. *Annals of Operations Research*, v.157, p.91-104, 2008.

**QUADT, D., KUHN, H.** *Capacitated lot-sizing with extensions: a review*. *4OR – A Quarterly Journal of Operations Research*, v.6, n.1, p.61-83, 2008.

**ROBINSON, P., NARAYANAN, A., SAHIN, F.** *Coordinated deterministic demand lot-sizing problem: A review of models and algorithms*. *Omega*, v.37, n.1, p.3-15, doi:10.1016/j.omega.2006.11.004, 2009.

**SAINT-GOBAIN.** Compagnie de Saint-Gobain. Disponível em: <<http://www.sgabrasivos.com.br/portugues/abrasivos/default.asp>>. Acesso em: 12 jun.2011.

**SALOMON, M.; SOLOMON, M.; VAN WASSENHOVE, L.N.; DUMAS, Y.** *Solving the discrete lot sizing and scheduling problem with sequence dependent set-up costs and set-up times using the travelling salesman problem with time windows*. *European Journal of Operational Research*, v.100, n.3, p.494-513, 1997.

**SARGUT, F. Z.; ROMELIJN, H. E.** *Lot-sizing with non-stationary cumulative capacities*. *Operations Research Letters*, v.35, p.549-557, 2007.

**SCHRAGE, L.** “*The multiproduct lot scheduling problem*”, em: **DEMPSTER, M. A. H.; LENSTRA, J. K.; RINNOOY, A. H. G.** Deterministic and stochastic scheduling, Nato Advanced Study Institutes Series, Dordrecht, 1984.

**SEAE.** Secretaria de acompanhamento econômico. Disponível em: <<http://www.seae.fazenda.gov.br/>>. Acesso em: 18 ago. 2011.

**SLACK, N.; CHAMBERS, S.; JOHNSTON, R.** Administração da produção. 3.ed. Atlas, 2009.

**SMITH, W. F.** *Principles of materials science and engineering*. 3.th. McGraw-Hill, 1996.

**STAGGEMEIER, A. T., CLARK, A. R.** *A survey of lot-sizing and scheduling models*. 33° SBPO, 2001.

**SUNG, C.; MARAVELIAS, C.T.** *A mixed-integer programming formulation for the capacitated lot-sizing problem*. Computers and Chemical Engineering, v.32, p.244-259, 2008.

**SUPITHAK, W.; LIMAN, D.S.; MONTES, E.J.** *Lot-sizing and scheduling problem with earliness tardiness and setup penalties*. Computers and Industrial Engineering, v.58, n.3, p.363-372, [doi:10.1016/j.cie.2008.10.005](https://doi.org/10.1016/j.cie.2008.10.005), 2010.

**SURIE, C.; STADTLER, H.** *The capacitated lot-sizing problem with linked lot sizes*. Management Science, v.49, n.8, p.1039-1054. 2003.

**TAHA, H. A.** Pesquisa operacional: uma visão geral. 8.ed. Pearson Prentice Hall, 2008.

**TOLEDO, C. F.; FRANÇA, P. M.; MORABITO, R.; KIMMS, A.** *Multi-population genetic algorithm to solve the synchronized and integrated two-level lot-sizing and scheduling problem*, International Journal of Production Research, v.47, n.11, p.3097-3119, [doi:10.1080/00207540701675833](https://doi.org/10.1080/00207540701675833), 2009.

**TOLEDO, C. F.; FRANÇA, P. M.; MORABITO, R.; KIMMS, A.** Um Modelo de Otimização para o Problema Integrado de Dimensionamento de Lotes e Programação da Produção em Fábrica de Refrigerantes. Pesquisa Operacional, v. 27, p. 155-186, 2007.

**TOLEDO, F. M. B.; ARMENTANO, V. A.** *A lagrangian-based heuristic for the capacitated lot sizing problem in parallel machines*. European Journal of Operational Research, v.175, p.1070-1083, 2006.

**TONAKI, V. S.; TOLEDO, F. M. B.** *An approach for solving the lot-sizing problem of a market-driven foundry*. Journal of the Operational Research Society, v.61, p.108-114, 2010.

**TOSO, E.; MORABITO, R.; CLARK, A.,** “*Lot-sizing and sequencing optimisation at an animal-feed plant*”, Computers & Industrial Engineering, v.57, n.3, p.813-821, [doi:10.1016/j.cie.2009.02.011](https://doi.org/10.1016/j.cie.2009.02.011), 2009.

**TOSO, E.** Dimensionamento e sequenciamento de lotes de produção na indústria de suplementos para nutrição animal. Tese. São Carlos, SP, Brasil: Engenharia de Produção, Universidade Federal de São Carlos, 2008.

**TREIBACHER.** Treibacher Schleifmittel Brasil. Disponível em: <<http://www.treibacher.com.br>>. Acesso em: 18 ago.2011.

**WAGNER, H. M.; WHITIN, T. M.** *Dynamic Version of the Economic Lot Size Model.* Management Science, v.5, n.1, p. 89-96, 1958.

**WATERER, H.** *Lot sizing with inventory gains.* Operations Research Letters, v.35, n.6, p.759-766, doi:10.1016/j.orl.2007.01.005, 2007.

**WILLIAMS, P.** *Model building in mathematical programming.* New York: Wiley & Sons, 1993.

**WINSTON, W.** *Operations Research: Applications and algorithms.* PWS-Kent, Boston, 1991.

**WOLSEY, L. A.** *Lot-sizing with production and delivery time windows.* Social Science Research Network, 2005. Disponível em: <[http://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=760686](http://papers.ssrn.com/sol3/papers.cfm?abstract_id=760686)>. Acesso em: 24 jan.2010.

**WOLSEY, L. A.** *Solving multi-item lot-sizing problems with an mip solver using classification and reformulation.* Management Science, v.48, n.12, p.1587-1602, 2002.

**WOLSEY, L. A.** *Progress with single-item lot-sizing.* European Journal of Operational Research, v.86, n.3, p.395-401, 1995.

**YAMAN, H.** *Polyhedral analysis for two-item uncapacitated lot-sizing problem with one-way substitution.* Discrete Applied Mathematics, v.157, p.3133-3151, 2009.

**YOSHIZAWA, Y; HIRAO, K.; KANZAKI, S.** *Fabrication of low cost fine-grained alumina powders by seeding for high performance sintered bodies.* Journal of the European Ceramic Society, v.24, n. , p.325-330, 2004.

**ZHU, X.; WILHELM, W. E.** *Scheduling and lot sizing with sequence dependent setup: a literature review.* IIE Transactions, v.38, n.11, p.987-1007, 2006.

## APÊNDICE I

**/\* MODELO MFP IMPLEMENTADO NO GAMS \*/**

SETS

T    periodos    /1\*10/  
 J    processos   /1\*10/  
 I    produtos  
 / EK8A\_16, EK8A\_20, EK8A\_24, EK8A\_30, EK8A\_36, EK8A\_46, EK8A\_54, EK8A\_60,  
 EK8A\_80, EK8A\_100/;

ALIAS (T,TL);

TABLE A(I,J)    tonelada por dia produzida do item i utilizando o processo j

	1	2	3	4	5	6	7	8	9	10
EK8A_16	0	1000	2000	0	500	0	0	0	1000	700
EK8A_20	0	1000	500	0	500	0	0	0	1000	700
EK8A_24	0	0	500	0	300	0	0	0	200	700
EK8A_30	0	0	500	0	300	0	0	600	200	0
EK8A_36	0	0	500	0	0	0	0	700	200	0
EK8A_46	0	0	0	1500	0	0	0	700	200	0
EK8A_54	0	0	0	800	0	0	0	700	200	0
EK8A_60	1500	0	0	1500	2000	3000	3000	0	1000	0
EK8A_80	1000	1500	500	1500	1500	2000	2500	0	1000	0
EK8A_100	300	300	0	500	300	700	800	0	1000	0;

TABLE D(I,T)    demanda do produto i no periodo t

	1	2	3	4	5	6	7	8	9	10
EK8A_16	0	0	0	0	0	0	0	0	0	0
EK8A_20	0	0	0	0	0	0	0	0	0	0
EK8A_24	0	0	1000	0	0	0	0	0	0	0
EK8A_30	0	0	0	0	0	0	0	0	0	0
EK8A_36	0	0	0	0	0	0	0	0	0	0
EK8A_46	0	0	0	0	0	0	0	0	0	0
EK8A_54	0	0	0	0	0	0	0	0	0	0
EK8A_60	0	0	0	2000	2000	0	0	0	0	0
EK8A_80	0	0	0	0	0	0	0	0	0	0
EK8A_100	0	0	0	0	0	0	0	0	1000	0;

VARIABLES

X(J,T)    indica se o processo j eh utilizado no periodo t  
 F(I,T)    falta de producao do produto i no periodo t  
 Z         falta dos produtos;

BINARY VARIABLE Y ;

POSITIVE VARIABLE F;

EQUATIONS

FALTA                    funcao objetivo  
 DEMANDA(I,T)         satisfaz a demanda do produto i no periodo t  
 PROPORCAO(T)         satisfaz utilizacao dos processos;

FALTA ..                Z =E= SUM((I,T), F(I,T)) ;

DEMANDA(I,T) ..        SUM((J,TL)\$ (ORD(TL) le ORD(T)), A(I,J)\*X(J,TL)) + F(I,T) =G=  
 SUM(TL\$(ORD(TL) le ORD(T)),D(I,TL));

PROPORCAO(T) ..        SUM(J, X(J,T)) =L= 1 ;

MODEL MFP /ALL/ ;

SOLVE MFP USING MIP MINIMIZING Z ;

DISPLAY X.L, F.L;



## APÊNDICE II

**/\* MODELO D-FES IMPLEMENTADO NO GAMS \*/**

option OPTCR = 0.000, ITERLIM = 10000000, RESLIM = 10800;

SETS

```

    T   periodos   /1*8/
    J   processos  /1*8/
    I   produtos
/ EK8A_16,
  EK8A_20,
  EK8A_24,
  EK8A_30,
  EK8A_36,
  EK8A_46,
  EK8A_54,
  EK8A_60,
  EK8A_80,
  EK8A_100/;

```

ALIAS (T,TL);

TABLE G(I,T) custo unitario de falta do item i no periodo t

	1	2	3	4	5	6	7	8
EK8A_16	1.5	2	1.5	2	2	1.5	1.5	2
EK8A_20	2	1.5	2	1.5	2	2	1.5	2
EK8A_24	1.5	2	2	1.5	2	2	2	1.5
EK8A_30	2	2	2	2	1.5	2	2	1.5
EK8A_36	2	1.5	1.5	2	2	2	1	1.5
EK8A_60	2	2	2	1	2	1.5	1	2
EK8A_80	2	2	1.5	2	1	2	2	2
EK8A_100	2	1.5	2	2	2	2	2	1.5;

TABLE H(I,T) custo unitario de estocagem do item i no periodo t

	1	2	3	4	5	6	7	8
EK8A_16	1	1	1	1	1	1	1	1
EK8A_20	1	1	1	1	1	1	1	1
EK8A_24	1	1	1	1	1	1	1	1
EK8A_30	1	1	1	1	1	1	1	1
EK8A_36	1	1	1	1	1	1	1	1
EK8A_60	1	1	1	1	1	1	1	1
EK8A_80	1	1	1	1	1	1	1	1
EK8A_100	1	1	1	1	1	1	1	1;

TABLE A(I,J) tonelada por dia produzida do item i utilizando o processo j

	1	2	3	4	5	6	7	8
EK8A_16	200	0	50	0	0	0	100	70
EK8A_20	50	0	50	0	0	0	100	70
EK8A_24	50	0	30	0	0	0	20	70
EK8A_30	50	0	30	0	0	60	20	0
EK8A_36	50	0	0	0	0	70	20	0
EK8A_60	0	150	200	300	300	0	100	0
EK8A_80	50	150	150	200	250	0	100	0
EK8A_100	0	50	30	70	80	0	100	0;

TABLE D(I,T) demanda original do produto i no periodo t

	1	2	3	4	5	6	7	8
EK8A_16	0	0	0	0	40	0	60	50
EK8A_20	0	0	0	0	40	60	0	100
EK8A_24	100	0	0	0	0	50	0	200
EK8A_30	0	0	30	70	0	0	100	0
EK8A_36	0	0	30	0	0	100	80	0
EK8A_60	0	100	200	0	0	0	150	0
EK8A_80	0	0	0	50	0	100	0	200
EK8A_100	0	0	0	0	0	0	300	0;

VARIABLES

X(J,T) indica se o processo p e utilizado no periodo t  
 E(I,T) excesso de producao do produto i no periodo t  
 F(I,T) falta de producao do produto i no periodo t  
 Y(J,T) setup ocorreu para o processo J no periodo T  
 Z funcao objetivo;

BINARY VARIABLE X ;

POSITIVE VARIABLE E, F, Y;

EQUATIONS

FALTA .. funcao objetivo  
 DEMANDA(I,T) satisfaz a demanda do produto i no periodo t  
 CAPACIDADE(T) capacidade de producao  
 SETUP\_Y(J,T) se a linha estah preparada para o processo J no periodo T;

FALTA .. Z =e= SUM((I,T), H(I,T)\*E(I,T)) + SUM((I,T), G(I,T)\*F(I,T)) + SUM((J, T), y(J,T));

DEMANDA(I,T) .. E(I,T-1) - F(I,T-1) + SUM(J, A(I,J)\*X(J,T)) - E(I,T) + F(I,T) =e= D(I,T);

CAPACIDADE(T) .. SUM(J, X(J,T)) =l= 1 ;

SETUP\_Y(J,T) .. Y(J,T) =g= X(J,T) - X(J,T-1);

MODEL DLSP\_FES\_BACKLOG /ALL/ ;

SOLVE DLSP\_FES\_BACKLOG USING MIP MINIMIZING Z ;

DISPLAY E.L, F.L, Y.L, X.L, Z.L;

### APÊNDICE III

**/\* MODELO CS-FES IMPLEMENTADO NO GAMS \*/**

option OPTCR = 0.000, ITERLIM = 10000000, RESLIM = 10800;

SETS

```

    T   periodos   /1*8/
    J   processos  /1*8/
    I   produtos
/ EK8A_16,
  EK8A_20,
  EK8A_24,
  EK8A_30,
  EK8A_36,
  EK8A_46,
  EK8A_54,
  EK8A_60,
  EK8A_80,
  EK8A_100/;
```

ALIAS (T,TL);

TABLE G(I,T) custo unitario de falta do item i no periodo t

	1	2	3	4	5	6	7	8
EK8A_16	1.5	2	1.5	2	2	1.5	1.5	2
EK8A_20	2	1.5	2	1.5	2	2	1.5	2
EK8A_24	1.5	2	2	1.5	2	2	2	1.5
EK8A_30	2	2	2	2	1.5	2	2	1.5
EK8A_36	2	1.5	1.5	2	2	2	1	1.5
EK8A_60	2	2	2	1	2	1.5	1	2
EK8A_80	2	2	1.5	2	1	2	2	2
EK8A_100	2	1.5	2	2	2	2	2	1.5;

TABLE H(I,T) custo unitario de estocagem do item i no periodo t

	1	2	3	4	5	6	7	8
EK8A_16	1	1	1	1	1	1	1	1
EK8A_20	1	1	1	1	1	1	1	1
EK8A_24	1	1	1	1	1	1	1	1
EK8A_30	1	1	1	1	1	1	1	1
EK8A_36	1	1	1	1	1	1	1	1
EK8A_60	1	1	1	1	1	1	1	1
EK8A_80	1	1	1	1	1	1	1	1
EK8A_100	1	1	1	1	1	1	1	1;

TABLE A(I,J) tonelada por dia produzida do item i utilizando o processo j

	1	2	3	4	5	6	7	8
EK8A_16	200	0	50	0	0	0	100	70
EK8A_20	50	0	50	0	0	0	100	70
EK8A_24	50	0	30	0	0	0	20	70
EK8A_30	50	0	30	0	0	60	20	0
EK8A_36	50	0	0	0	0	70	20	0
EK8A_60	0	150	200	300	300	0	100	0
EK8A_80	50	150	150	200	250	0	100	0
EK8A_100	0	50	30	70	80	0	100	0;

TABLE D(I,T) demanda original do produto i no periodo t

	1	2	3	4	5	6	7	8
EK8A_16	0	0	0	0	40	0	60	50
EK8A_20	0	0	0	0	40	60	0	100
EK8A_24	100	0	0	0	0	50	0	200
EK8A_30	0	0	30	70	0	0	100	0
EK8A_36	0	0	30	0	0	100	80	0
EK8A_60	0	100	200	0	0	0	150	0
EK8A_80	0	0	0	50	0	100	0	200
EK8A_100	0	0	0	0	0	0	300	0;

**/\* DEMANDA II: PODENDO SER UTILIZADA NO LUGAR DA DEMANDA ORIGINAL \*/**

TABLE D(I,T) demanda II do produto i no periodo t

	1	2	3	4	5	6	7	8
EK8A_16	0	0	0	0	0	0	60	70
EK8A_20	0	0	0	0	0	0	60	70
EK8A_24	0	0	0	0	0	0	60	70
EK8A_30	50	0	0	0	0	0	0	0
EK8A_36	60	0	0	0	0	0	0	0
EK8A_60	0	150	0	0	140	140	0	0
EK8A_80	0	150	0	0	140	140	0	0
EK8A_100	0	50	0	0	40	40	0	0;

VARIABLES

X(J,T) indica se o processo j pode ser utilizado no periodo t  
E(I,T) excesso de producao do produto i no periodo t  
F(I,T) falta de producao do produto i no periodo t  
Y(J,T) setup ocorreu para o processo j no periodo t  
Q(J,T) proporcao do tempo que o processo j foi utilizado no periodo t  
Z funcao objetivo;

BINARY VARIABLE X ;

POSITIVE VARIABLE E, F, Y, Q;

EQUATIONS

FALTA funcao objetivo  
DEMANDA(I,T) satisfaz a demanda do produto i no periodo t 5.18  
CAPACIDADE(J,T) capacidade de producao 5.19  
UM\_J\_POR\_DIA(t) restricao 5.14  
SETUP\_Y(J,T) setup para o processo j no periodo t 5.15;

FALTA .. Z =e= SUM((I,T), H(I,T)\*E(I,T) + G(I,T)\*F(I,T)) +  
SUM((J, T), y(J,T));

DEMANDA(I,T) .. E(I,T-1) - F(I,T-1) + SUM(J, A(I,J)\*Q(J,T)) - E(I,T) +  
F(I,T) =e= D(I,T);

CAPACIDADE(J,T) .. Q(J,T) =l= X(J,T);

UM\_J\_POR\_DIA(T) .. SUM(J, X(J,T)) =l= 1;

SETUP\_Y(J,T) .. Y(J,T) =g= X(J,T) - X(J,T-1);

MODEL CSLP\_FES\_BACKLOG /ALL/ ;

SOLVE CSLP\_FES\_BACKLOG USING MIP MINIMIZING Z ;  
DISPLAY E.L, F.L, Y.L, X.L, Q.L, Z.L;



TABLE A(I,J)	tonelada por dia produzida do item i utilizando o processo j							
	1	2	3	4	5	6	7	8
EK8A_16	200	0	50	0	0	0	100	70
EK8A_20	50	0	50	0	0	0	100	70
EK8A_24	50	0	30	0	0	0	20	70
EK8A_30	50	0	30	0	0	60	20	0
EK8A_36	50	0	0	0	0	70	20	0
EK8A_60	0	150	200	300	300	0	100	0
EK8A_80	50	150	150	200	250	0	100	0
EK8A_100	0	50	30	70	80	0	100	0;

TABLE D(I,T)	demanda original do produto i no periodo t							
	1	2	3	4	5	6	7	8
EK8A_16	0	0	0	0	40	0	60	50
EK8A_20	0	0	0	0	40	60	0	100
EK8A_24	100	0	0	0	0	50	0	200
EK8A_30	0	0	30	70	0	0	100	0
EK8A_36	0	0	30	0	0	100	80	0
EK8A_60	0	100	200	0	0	0	150	0
EK8A_80	0	0	0	50	0	100	0	200
EK8A_100	0	0	0	0	0	0	300	0;

#### VARIABLES

X(J,T) indica se a linha estah preparada para o processo j no periodo t  
 E(I,T) excesso de producao do produto i no periodo t  
 F(I,T) falta de producao do produto i no periodo t  
 Y(J,T) setup ocorreu para o processo j no periodo t  
 Q(J,T) proporcao do tempo que o processo j foi utilizado no periodo t  
 Z funcao objetivo;

BINARY VARIABLE X ;

POSITIVE VARIABLE E, F, Y, Q;

#### EQUATIONS

FALTA funcao objetivo  
 DEMANDA(I,T) satisfaz a demanda do produto i no periodo t 5.18  
 CAPACIDADE(T) capacidade de producao 5.19  
 UTILIZACAO\_Q(J,T)  
 UM\_J\_POR\_DIA(t) restricao 5.14  
 SETUP\_Y(J,T) setup para o processo j no periodo t 5.15;

FALTA .. Z =e= SUM((I,T), H(I,T)\*E(I,T) + G(I,T)\*F(I,T)) + SUM((J, T), y(J,T));

DEMANDA(I,T) .. E(I,T-1) - F(I,T-1) + SUM(J, A(I,J)\*Q(J,T)) - E(I,T) + F(I,T) =e= D(I,T);

CAPACIDADE(T) .. SUM(J, ST(J)\*Y(J,T)+Q(J,T)) =l= 1;

UTILIZACAO\_Q(J,T) .. Q(J,T) =l= X(J,T-1) + X(J,T);

UM\_J\_POR\_DIA(T) .. SUM(J, X(J,T)) =l= 1;

SETUP\_Y(J,T) .. Y(J,T) =g= X(J,T) - X(J,T-1);

MODEL PLSP\_FES\_BACKLOG /ALL/ ;

SOLVE PLSP\_FES\_BACKLOG USING MIP MINIMIZING Z ;

DISPLAY E.L, F.L, Y.L, X.L, Q.L, Z.L;

## APÊNDICE V

**/\* MODELO G-FES IMPLEMENTADO NO GAMS \*/**

option OPTCR = 0.000, ITERLIM = 10000000, RESLIM = 10800;

SETS

T periodos /1\*8/  
 J processos /1\*8/  
 P maximo de sub-periodos /1\*100/  
 I produtos  
 / EK8A\_16, EK8A\_20, EK8A\_24, EK8A\_30, EK8A\_36,  
 EK8A\_46, EK8A\_54, EK8A\_60, EK8A\_80, EK8A\_100/;

ALIAS (T, TL);

PARAMETERS N(T) numero maximo de lotes no periodo t

/1 3, 2 2, 3 4, 4 2, 5 3, 6 3, 7 4, 8 5/;

PARAMETERS ST(J) tempo de setup do processo j

/1 0.1, 2 0.11, 3 0.12, 4 0.13, 5 0.14, 6 0.15, 7 0.16, 8 0.17/;

TABLE G(I,T) custo unitario de falta do item i no periodo t

	1	2	3	4	5	6	7	8
EK8A_16	1.5	2	1.5	2	2	1.5	1.5	2
EK8A_20	2	1.5	2	1.5	2	2	1.5	2
EK8A_24	1.5	2	2	1.5	2	2	2	1.5
EK8A_30	2	2	2	2	1.5	2	2	1.5
EK8A_36	2	1.5	1.5	2	2	2	1	1.5
EK8A_60	2	2	2	1	2	1.5	1	2
EK8A_80	2	2	1.5	2	1	2	2	2
EK8A_100	2	1.5	2	2	2	2	2	1.5;

TABLE H(I,T) custo unitario de estocagem do item i no periodo t

	1	2	3	4	5	6	7	8
EK8A_16	1	1	1	1	1	1	1	1
EK8A_20	1	1	1	1	1	1	1	1
EK8A_24	1	1	1	1	1	1	1	1
EK8A_30	1	1	1	1	1	1	1	1
EK8A_36	1	1	1	1	1	1	1	1
EK8A_60	1	1	1	1	1	1	1	1
EK8A_80	1	1	1	1	1	1	1	1
EK8A_100	1	1	1	1	1	1	1	1;

TABLE A(I,J) tonelada por dia produzida do item i utilizando o processo j

	1	2	3	4	5	6	7	8
EK8A_16	200	0	50	0	0	0	100	70
EK8A_20	50	0	50	0	0	0	100	70
EK8A_24	50	0	30	0	0	0	20	70
EK8A_30	50	0	30	0	0	60	20	0
EK8A_36	50	0	0	0	0	70	20	0
EK8A_60	0	150	200	300	300	0	100	0
EK8A_80	50	150	150	200	250	0	100	0
EK8A_100	0	50	30	70	80	0	100	0;

TABLE D(I,T) demanda original do produto i no periodo t

	1	2	3	4	5	6	7	8
EK8A_16	0	0	0	0	40	0	60	50
EK8A_20	0	0	0	0	40	60	0	100
EK8A_24	100	0	0	0	0	50	0	200
EK8A_30	0	0	30	70	0	0	100	0
EK8A_36	0	0	30	0	0	100	80	0
EK8A_60	0	100	200	0	0	0	150	0
EK8A_80	0	0	0	50	0	100	0	200
EK8A_100	0	0	0	0	0	0	300	0;

VARIABLES

X(J,P) indica se a linha estah preparada para o processo j no periodo t  
 E(I,T) excesso de producao do produto i no periodo t  
 F(I,T) falta de producao do produto i no periodo t  
 Y(J,P) setup ocorreu para o processo j no sub-periodo p  
 Q(J,P) proporcao do tempo que o processo j foi utilizado no sub-periodo p  
 FIRST(T)  
 LAST(T)  
 Z funcao objetivo;

BINARY VARIABLE Y ;

POSITIVE VARIABLE E, F, X, Q;

INTEGER VARIABLE FIRST, LAST;

EQUATIONS

FALTA funcao objetivo  
 PRIMEIRO(T) primeira posicao do periodo t  
 ULTIMO(T) ultima posicao do periodo t  
 DEMANDA(I,T) satisfaz a demanda do produto i no periodo t (5.20)  
 UTILIZACAO\_Q(J,P) tempo utilizando o processo J no sib periodo P (5.21)  
 CAPACIDADE(T) capacidade de producao (5.22)  
 UM\_J\_POR\_SUB(P) restricao de um processo por subperiodo (5.23)  
 SETUP\_Y(J,P) setup para o processo j no subperiodo p (5.24);

FALTA .. Z =e= SUM((I,T), H(I,T)\*E(I,T) + G(I,T)\*F(I,T)) +  
 SUM((J, P), X(J,P));

PRIMEIRO(T) .. FIRST(T) =e= 1 + SUM(TL \$(ORD(TL) le ORD(T)-1), N(TL));

ULTIMO(T) .. LAST(T) =e= FIRST(T) + N(T) -1;

DEMANDA(I,T) .. E(I,T-1) - F(I,T-1) + SUM((J,P)\$(ORD(P) ge  
 FIRST(T) AND (FIRST(T) le LAST(T))), A(I,J)\*Q(J,P)) -  
 E(I,T) + F(I,T) =e= D(I,T);

UTILIZACAO\_Q(J,P) .. Q(J,P) =l= Y(J,P-1) + Y(J,P);

CAPACIDADE(T) .. SUM((J,P)\$(ORD(P) ge FIRST(T) AND (FIRST(T) le  
 LAST(T))), ST(J)\*X(J,P)+Q(J,P)) =l= 1;

UM\_J\_POR\_SUB(P) .. SUM(J, Y(J,P)) =l= 1;

SETUP\_Y(J,P) .. X(J,P) =g= Y(J,P) - Y(J,P-1);

MODEL DLSP\_FES\_BACKLOG /ALL/ ;

SOLVE DLSP\_FES\_BACKLOG MINIMIZING Z USING MIP;

DISPLAY E.L, F.L, Y.L, X.L, Q.L, Z.L;



## APÊNDICE VI

### Experimentos com parâmetros modificados

As marcantes melhorias de desempenho observadas com a versão 11 do CPLEX estimularam a realização de testes com este *solver*, desta vez, alterando parâmetros associados aos métodos de solução disponíveis e à utilização de estratégias de pré-processamento. Os parâmetros *default* (padrão) que controlam o algoritmo *branch-and-cut* utilizado no *solver* são fornecidos como uma proposta de resolução geral; o ajuste fino desses parâmetros, por outro lado, pode gerar ganho em tempo e/ou qualidade da solução em problemas com estruturas particulares.

Algumas etapas do algoritmo B&C incluem estratégias de pré-processamento, relaxações, heurísticas e planos de corte. No caso do parâmetro que controla a utilização de heurísticas na solução dos modelos, o *default* do CPLEX é a opção automática, ou seja, critérios internos do CPLEX são testados e decidem se as heurísticas serão utilizadas. O CPLEX pode aplicar as heurísticas tanto para determinar uma solução inicial para o modelo (heurística de arredondamento), quanto na resolução dos subproblemas da árvore de busca (FERREIRA, 2006).

Os experimentos aqui descritos seguem a proposta descrita em Toso (2008), na qual é inicialmente alterado o parâmetro que controla as opções de seleção de nós (*mipemphasis*, *nodeselect*, *bbinterval*). Outro experimento consiste em desligar as estratégias de pré-processamento (*presolve* = 0). Para avaliar a influência dos planos de corte na solução do modelo, primeiramente todos os planos são desligados (*mipcuts*=-1). Em seguida, são realizados experimentos ligando os planos de corte um a um (*mipcuts*=-1 *cliquecuts*=1, *mipcuts*=-1 *covercuts*=1, etc). Por fim, os planos de cortes são desligados um a um (*cliquecuts*=-1, *covercuts*=-1, etc).

Na Tabela 1 são apresentados os resultados obtidos com as instâncias 2, 4 e 7 do conjunto T5 (5×19 períodos). Estas instâncias foram selecionadas uma vez que não foi possível provar a otimalidade da solução apresentada com a utilização dos parâmetros *default* do CPLEX 11 no tempo máximo de 10.800 segundos; este mesmo tempo máximo foi utilizado para executar as instâncias com parâmetros modificados.

Como pode ser observado, a utilização dos parâmetros modificados não provocou melhorias na qualidade da melhor solução factível de nenhuma das três instâncias em relação

aos parâmetros *default*. Para as instâncias 2 e 7 foram verificadas pequenas reduções do *gap* médio obtido com os parâmetros *default*, as quais se mostram mais concentradas quando os planos de cortes são desligados um a um (com *fraccuts=-1*, *gubcuts=-1*, *impliedcut=-1*, *mircuts=-1*, por exemplo).

Tabela 1 - Resultados com variação do número de períodos: resolução de três instâncias do conjunto T5 com CPLEX 11 sob diferentes combinações de parâmetros modificados.

Parâmetros	Instância 2		Instância 4		Instância 7	
	z	Gap	z	Gap	z	Gap
<i>default</i>	99038	0,71%	125559	0,29%	8336	27,49%
<i>mipemphasis=0</i>	99038	<b>0,69%</b>	125559	0,29%	8336	27,55%
<i>mipemphasis=2</i>	101182	3,20%	125559	0,29%	8336	<b>26,76%</b>
<i>mipemphasis=4</i>	100605	2,41%	125559	0,29%	8336	<b>21,32%</b>
<i>nodeselect=1</i>	99270	0,95%	125559	0,29%	8336	28,82%
<i>nodeselect=3</i>	99441	7,00%	125559	0,29%	8428	34,71%
<i>nodeselect=2, bbinterval=3</i>	99038	0,71%	125559	0,29%	8336	31,07%
<i>nodeselect=2, bbinterval=10</i>	99038	0,79%	125559	0,29%	8336	33,44%
<i>presolve=0</i>	115319	18,16%	125559	0,29%	10837	47,90%
<i>mipcuts=-1</i>	104135	7,98%	125559	0,29%	11366	57,43%
<i>mipcuts=1</i>	99840	1,52%	125559	0,29%	8336	<b>22,26%</b>
<i>mipcuts=2</i>	106676	9,53%	125559	0,29%	8336	28,68%
<i>mipcuts=-1, cliquecuts=1</i>	114708	21,52%	125559	0,29%	12390	60,96%
<i>mipcuts=-1, cliquecuts=2</i>	114804	21,58%	125559	0,29%	12390	60,96%
<i>mipcuts=-1, covercuts=1</i>	114804	21,58%	125559	0,29%	12390	60,96%
<i>mipcuts=-1, covercuts=2</i>	114630	21,46%	125559	0,29%	13036	63,03%
<i>mipcuts=-1, disjcuts=1</i>	114708	21,52%	125559	0,29%	11822	59,09%
<i>mipcuts=-1, disjcuts=2</i>	114708	21,52%	125559	0,29%	12390	60,96%
<i>mipcuts=-1, disjcuts=3</i>	103793	14,53%	125559	0,29%	20720	80,39%
<i>mipcuts=-1, flowcovers=1</i>	100063	2,07%	125559	0,29%	8393	27,97%
<i>mipcuts=-1, flowcovers=2</i>	105526	9,41%	125559	0,29%	8879	34,84%
<i>mipcuts=-1, flowpaths=1</i>	114708	21,52%	125559	0,29%	11822	59,09%
<i>mipcuts=-1, flowpaths=2</i>	115260	22,05%	125559	0,29%	12390	61,43%
<i>mipcuts=-1, fraccuts=1</i>	99650	12,26%	125559	0,29%	10604	52,98%
<i>mipcuts=-1, fraccuts=2</i>	99840	1,90%	125559	0,29%	15709	71,75%
<i>mipcuts=-1, gubcuts=1</i>	101721	5,68%	125559	0,29%	11301	57,19%
<i>mipcuts=-1, gubcuts=2</i>	100353	4,21%	125559	0,29%	8879	34,84%
<i>mipcuts=-1, impliedcuts=1</i>	122322	26,38%	125559	0,29%	12390	60,96%
<i>mipcuts=-1, impliedcuts=2</i>	114804	21,58%	125559	0,29%	10964	56,04%
<i>mipcuts-1, mircuts=1</i>	99441	1,51%	125559	0,29%	8336	<b>26,76%</b>
<i>mipcuts-1, mircuts=2</i>	99038	0,73%	125559	0,29%	8336	<b>23,61%</b>
<i>cliquecuts=-1</i>	100799	4,13%	125559	0,29%	16022	71,65%
<i>covercuts=-1</i>	99038	<b>0,70%</b>	125559	0,29%	8336	27,74%
<i>disjcuts=-1</i>	99038	0,71%	125559	0,29%	8336	27,74%
<i>flowcuts=-1</i>	99441	1,50%	125559	0,29%	8336	<b>20,59%</b>
<i>flowpathcuts=-1</i>	99038	<b>0,70%</b>	125559	0,29%	8336	27,54%
<i>fraccuts=-1</i>	99038	<b>0,65%</b>	125559	0,29%	8336	<b>25,73%</b>
<i>gubcuts=-1</i>	99038	<b>0,68%</b>	125559	0,29%	8336	<b>27,25%</b>
<i>impliedcuts=-1</i>	99038	<b>0,69%</b>	125559	0,29%	8336	<b>27,21%</b>
<i>mircuts=-1</i>	99038	<b>0,66%</b>	125559	0,29%	8336	<b>24,45%</b>

# ÍNDICE REMISSIVO DE REFERÊNCIAS

---

## A

Aarts *et al.* (2000), 50  
ABC (2009), 16, 17, 20, 23, 24  
Akbalik & Penz (2009), 50  
Aksen (2007), 49  
Alfieri *et al.* (2002), 50  
Alidaee *et al.* (2001), 50  
Allen (1990), 25  
Anfacar (2001), 16  
Araujo *et al.* (2004), 50  
Araujo *et al.* (2007), 39  
Arenales *et al.* (2007), 42  
Armentano *et al.* (1999), 44  
Askin e Standridge (1993), 42

## B

Basnet & Leung (2005), 49  
Beltrán & Krass (2002), 46  
Belvaux & Wolsey (2001), 55, 57  
Berretta *et al.* (2005), 50  
Bertrand & Fransoo (2002), 14  
Bitran & Matsuo (1986), 44  
Bitran & Yanasse (1982), 50  
Brahimi *et al.* (2006a), 47  
Brahimi *et al.* (2006b), 49  
Brandimarte *et al.* (1998), 50  
Brooke *et al.* (1992), 77  
Brüggerman & Jahnke (2000), 53  
Buschkühl *et al.* (2010), 50

## C

Callister (2006), 18  
Carrizo (2001), 93  
Chang (2010), 50  
CHF (2009), 22, 23  
Clark & Clark (2000), 40, 44  
Clark *et al.* (2010), 57  
Colin (2007), 42  
Contador & Contador (1997), 41  
Cruz (1982), 17

## D

Dastidar & Nagi (2005), 50  
Dessouky *et al.* (1999), 53  
Dogan & Grossmann (2006), 57  
Drexler & Haase (1995), 55

Drexler & Haase (1996), 55  
Drexler & Kimms (1997), 44, 51, 56

## E

Elfusa (2011), 22, 32, 34

## F

Federgruen *et al.* (2007), 50  
Feng & Cheng (1998), 44  
Feo & Resende (1989), 72  
Ferreira (2006), 137  
Ferreira *et al.* (2009), 44, 57  
Ferreira *et al.* (2010), 57  
Fleischmann & Meyr (1997), 56  
Fleischmann (1990), 53  
Fleischmann (1994), 53  
Fleszar & Hindi (2004), 44

## G

Gaafar & Choueiki (2000), 50  
Gao *et al.* (2008), 44  
Gershwin (1994), 42  
Gicquel *et al.* (2009), 53  
Gopalakrishnan *et al.* (2001), 50  
Graves *et al.* (1993), 42  
Gupta & Magnusson (2005), 54

## H

Haase & Kimms (2000), 44  
Hax & Candeia (1984), 42  
Heuvel & Wagelmans (2005), 49  
Heuvel & Wagelmans (2008), 44  
Hindi (1995), 54  
Holland (1975), 72  
HSU (1983), 13

## I

Ilog (2009), 77

## J

Jans & Degraeve (2007), 50  
Jans & Degraeve (2008), 44  
Jaruphongsas & Lee (2007), 47  
Johnson & Montgomery (1974), 41, 42  
Jordan & Drexler (1998), 53

**K**

Karimi *et al.* (2003), 44, 49, 60  
 Karmarkar *et al.* (1987), 44  
 Kimms (1996a), 44  
 Kimms (1996b), 55  
 Kimms (1999), 55  
 Koclar (2005), 57

**L**

Lambrecht & Vanderveken (1979), 50  
 Lasdon & Terjung (1971), 52  
 Lawrence (1985), 19  
 Lee *et al.* (2001), 47  
 Lin *et al.* (2002), 50  
 Lobo *et al.* (2007), 50  
 Lozano *et al.* (1991), 50  
 Luche *et al.* (2009), 13, 14, 40, 59, 60, 77, 78

**M**

Maes & Van Wassenhove (1986), 50  
 Maes & Van Wassenhove (1988), 50  
 Maes *et al.* (1991), 50  
 Manne (1958), 49  
 Mateus *et al.* (2009), 50  
 Matta & Guignard (1989), 44  
 Melo & Wolsey (2010), 49  
 Menezes *et al.* (2010), 57  
 Merce & Fontan (2003), 50  
 Meyr (2000), 44  
 Meyr (2002), 50  
 Miller *et al.* (2000), 50  
 Molina *et al.* (2009), 44  
 Morabito & Pureza (2010), 14  
 Mortensen (2006), 7, 19  
 Munhoz & Morabito (2001), 112

**N**

Nahmias (1995), 41

**P**

Pan *et al.* (2009), 46  
 Parveen & Haque (2007), 50  
 Peralta (2011), 36, 125  
 Pileggi *et al.* (1998), 25  
 Poltroniere *et al.* (2008), 44

**Q**

Quadt & Kuhn (2008), 44

**R**

Robinson *et al.* (2009), 44

**S**

Saint-Gobain (2011), 19  
 Salomon *et al.* (1997), 52  
 Sargut & Romeijn (2007), 47  
 Schrage (1984), 51, 52  
 SEAE (2011), 7, 20, 21, 28  
 Slack *et al.* (2009), 39  
 Smith (1996), 18, 20  
 Staggemeier & Clark (2001), 44  
 Sung & Maravelias (2008), 57  
 Supithak *et al.* (2010), 53  
 Surie & Stadler (2003), 44

**T**

Taha (2008), 42  
 Toledo *et al.* (2006), 50  
 Toledo *et al.* (2007), 44  
 Toledo *et al.* (2009), 44  
 Tonaki & Toledo (2010), 44  
 Toso (2008), 42, 137  
 Toso *et al.* (2009), 57  
 Treibacher (2011) 21, 24, 25, 26, 30, 32, 33

**W**

Wagner & Whitin (1958), 49  
 Waterer (2007), 50  
 Williams (1993), 42  
 Winston (1991), 42  
 Wolsey (1995), 49  
 Wolsey (2002), 53  
 Wolsey (2005), 47

**Y**

Yaman (2009), 49  
 Yoshizawa *et al.* (2004), 20

**Z**

Zhu & Wilhelm (2006), 44, 46