

Dissertação apresentada à Pró-Reitoria de Pós-Graduação do Instituto Tecnológico de Aeronáutica, como parte dos requisitos para obtenção do título de Mestre em Ciências no Programa de Pós-Graduação em Pesquisa Operacional, em associação com o ICT-UNIFESP São José dos Campos.

Carla Cristina Doescher Fernandes

**UMA MATH-HEURÍSTICA PARA O PROBLEMA
DE DIMENSIONAMENTO DE LOTES COM
MÁQUINAS PARALELAS E *SETUP*
*CARRY-OVER***

Dissertação aprovada em sua versão final pelos abaixo assinados:

Prof^ª. Dr^ª. Mariá Cristina Vasconcelos Nascimento Rosset
Orientadora

Prof. Dr. Pedro Teixeira Lacava
Pró-Reitor de Pós-Graduação

Campo Montenegro
São José dos Campos - Brasil
2018

Dados Internacionais de Catalogação-na-Publicação (CIP)
Divisão de Informação e Documentação

Fernandes, Carla Cristina Doescher

Uma Math-Heurística para o Problema de Dimensionamento de Lotes com Máquinas Paralelas e *Setup Carry-over* / Carla Cristina Doescher Fernandes.

São José dos Campos, 2018.

69f.

Dissertação de Mestrado – Curso de Pesquisa Operacional. Área de Pesquisa Operacional – Instituto Tecnológico de Aeronáutica e Instituto de Ciência e Tecnologia da Universidade Federal de São Paulo, 2018. Orientadora: Prof^a. Dr^a. Mariá Cristina Vasconcelos Nascimento Rosset.

1. Dimensionamento de lotes. 2. Máquinas Paralelas. 3. Múltiplos Itens. 4. Setup Carry-over. I. Instituto Tecnológico de Aeronáutica. II. Universidade Federal de São Paulo. III. Uma Math-Heurística para o Problema de Dimensionamento de Lotes com Máquinas Paralelas e *Setup Carry-over*.

REFERÊNCIA BIBLIOGRÁFICA

FERNANDES, Carla Cristina Doescher. **Uma Math-Heurística para o Problema de Dimensionamento de Lotes com Máquinas Paralelas e *Setup Carry-over***. 2018. 69f. Dissertação de Mestrado – Instituto Tecnológico de Aeronáutica e Universidade Federal de São Paulo, São José dos Campos.

CESSÃO DE DIREITOS

NOME DA AUTORA: Carla Cristina Doescher Fernandes

TÍTULO DO TRABALHO: Uma Math-Heurística para o Problema de Dimensionamento de Lotes com Máquinas Paralelas e *Setup Carry-over*

TIPO DO TRABALHO/ANO: Dissertação / 2018

É concedida ao Instituto Tecnológico de Aeronáutica e à Universidade Federal de São Paulo permissão para reproduzir cópias desta dissertação e para emprestar ou vender cópias somente para propósitos acadêmicos e científicos. A autora reserva outros direitos de publicação e nenhuma parte desta dissertação pode ser reproduzida sem a autorização da autora.

Carla Cristina Doescher Fernandes

Av. Cesare Monsueto Giulio Lattes, 1201 - Jardim Santa Ines I
12.231-280 – São José dos Campos-SP

**UMA MATH-HEURÍSTICA PARA O PROBLEMA
DE DIMENSIONAMENTO DE LOTES COM
MÁQUINAS PARALELAS E *SETUP*
*CARRY-OVER***

Carla Cristina Doescher Fernandes

Composição e Assinatura da Banca Examinadora:

Prof. Dr. Luiz Leduíno Sales Neto Presidente - UNIFESP

Prof^a. Dr^a. Mariá Cristina Vasconcelos Nascimento Rosset Orientadora - UNIFESP

Prof. Dr. Anibal Tavares de Azevedo UNICAMP

Prof^a. Dr^a. Franklina Maria Bragion de Toledo USP

Aos meus pais.

Agradecimentos

À minha orientadora. Profa Dra Mariá Nascimento, por confiar na minha capacidade, pela infinita disponibilidade e por todo o ensinamento,

Aos meus pais, João e Cleide, pelo incentivo e pelo suporte que sempre me deram,

À minha tia Andréa Leão por me mostrar novos horizontes,

Ao meu irmão João Doescher, pelos conselhos e pelo apoio,

À Desiree, por me auxiliar em muitas tarefas para conclusão desse trabalho,

À CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior pelo suporte financeiro durante o desenvolvimento do trabalho.

*“Não fiz o melhor, mas fiz tudo para que o melhor fosse feito.
Não sou o que deveria ser, mas não sou o que era antes.”*

— MARTIN LUTHER KING

Resumo

Esta Dissertação de Mestrado aborda o Problema de Dimensionamento de Lotes (PDL) com Máquinas Paralelas em que são considerados múltiplos itens e máquinas distintas que produzem os mesmos itens e que possuem restrições de capacidade. Os itens podem ser produzidos em qualquer máquina e no início da produção de cada item incorrem-se tempo e custo de *setup* da máquina utilizada. Além disso, considera-se a possibilidade de aproveitamento de *setup* no final de cada período : *setup carry-over*, para redução de custos e, de forma implícita, para a definição de um planejamento que diminua possíveis impactos ambientais provocados pela preparação de máquinas. Apesar de diversos trabalhos na literatura para o PDL com Máquinas Paralelas, poucos são os métodos de solução eficientes para resolvê-lo considerando *setup carry-over*. Nesse sentido, para resolver esse problema de forma eficiente, nesta Dissertação, realizou-se o seu estudo do ponto de vista de modelagem e de método de solução eficiente. Primeiramente, adaptaram-se três modelagens de programação inteira mista para o PDL com Máquinas Paralelas, referido aqui por PDLMP. Após experimentos, definiram-se dois modelos que foram utilizados para o método de solução proposto baseado em modelagem (math-heurística). A math-heurística baseada na conhecida heurística *relax-and-fix* com restrições *local branching* foi inspirada na meta-heurística *Greedy Randomized Search Procedures* (GRASP). Experimentos, nos quais são comparadas as soluções obtidas pelo *solver* CPLEX e pela math-heurística, demonstraram a eficiência do método proposto.

Abstract

This Master's Dissertation deals with the Lot Sizing Problem (LSP) in Parallel Machines with multiple items are considered and different machines that produce the same items and that have capacity restrictions. The items can be produced in any machine and at the beginning of the production of each item the time and cost of setup of the used machine is incurred. In addition, it is considered the possibility of taking advantage of setup at the end of each period: the setup carry-over, for cost reduction and, implicitly, for the definition of a production planning that reduces possible environmental impacts caused by the preparation of machines. Despite several studies in the literature for LSP in parallel machines, few of them introduce the efficient solution methods to solve it considering the setup carry-over. In this sense, to solve this problem efficiently, in this Dissertation, it is proposed its study from the point of view of modeling and efficient solution method. First, three mixed integer programming models were adapted for the LSP in Parallel Machines. After experiments, two models were defined that were used for the proposed solution method based on modeling (math-heuristics). The math-heuristic based on the well-known heuristic relax-and-fix with local branching constraints was modeled whose structure was inspired by the Greedy Randomized Search Procedures (GRASP). Experiments, in which the solutions obtained by the CPLEX solver and the math-heuristic are compared, demonstrated the efficiency of the proposed method.

Lista de Figuras

FIGURA 3.1 – Exemplo do modelo de caminho mínimo para 3 períodos e 2 máquinas - adaptada de (JANS, 2009)	30
FIGURA 3.2 – Perfil de Desempenho das soluções das instâncias considerando os Modelos 1, 2 e 3.	37
FIGURA 4.1 – Exemplo de iterações do <i>relax-and-fix</i>	43
FIGURA 4.2 – Exemplo: heurística modificada	46
FIGURA 4.3 – Exemplo ilustrando proposta de conjuntos da partição em famílias	48
FIGURA 4.4 – Exemplo: math-heurística proposta	49
FIGURA 4.5 – Exemplos de execuções do método	50
FIGURA 4.6 – Exemplos de execução do método modificado	52
FIGURA 5.1 – Perfil de desempenho das soluções obtidas pela math-heurística proposta comparando com as do CPLEX limitado em 1800 segundos.	55
FIGURA 5.2 – Perfil de desempenho comparando os resultados obtidos pela primeira versão da math-heurística e pelo CPLEX.	57
FIGURA 5.3 – Perfil de Desempenho em Relação ao Tempo de Execução	58
FIGURA 5.4 – Perfil de desempenho dos métodos avaliados, com relação à qualidade da solução obtida.	60
FIGURA 5.5 – Comparação da média das soluções obtidas pela math-heurística modificada para o PDLMP e para o PDLMPSC.	62

Lista de Tabelas

TABELA 3.1 – Quantidade de soluções ótimas encontradas pelo CPLEX, no tempo limite de 1800 segundos.	36
TABELA 5.1 – Médias das soluções obtidas pelo CPLEX e pela math-heurística para o conjunto de 10 instâncias com diferentes quantidades de itens, máquinas e períodos	56
TABELA 5.2 – Médias das soluções obtidas pelo CPLEX e pelas math-heurísticas MH-Modelo 1 e MH-Modelo 2 para o conjunto de 10 instâncias com diferentes quantidades de itens, máquinas e períodos, do tipo AAA.	60
TABELA 5.3 – Médias das soluções obtidas pelo CPLEX e pelas math-heurísticas MH-Modelo 1 e MH-Modelo 2 para o conjunto de 10 instâncias com diferentes quantidades de itens, máquinas e períodos, do tipo NBB.	61
TABELA 5.4 – Médias das soluções obtidas pela math-heurística para o PDLMP (sem <i>setup carry-over</i>) e para o PDLMPSC (com <i>setup carry-over</i>).	62

Lista de Abreviaturas e Siglas

ATSP	<i>Asymmetric Travelling Salesman Problem</i>
GLSP	<i>General Lot Sizing and Scheduling Problem</i>
PCP	Planejamento e Controle da Produção
PDL	Problema de Dimensionamento de Lotes
PDLC	Problema de Dimensionamento de Lotes com restrições de Capacidade
PDLMP	Problema de Dimensionamento de Lotes com Máquinas Paralelas
PDLMPSC	Problema de Dimensionamento de Lotes com Máquinas Paralelas e <i>Setup Carry-over</i>
PIM	<i>Problema de Otimização Inteira ou Mista</i>

Sumário

1	INTRODUÇÃO	14
1.1	Organização	16
2	O PROBLEMA DE DIMENSIONAMENTO DE LOTES	17
2.1	Características de um PDL	17
2.1.1	Demanda	18
2.1.2	Horizonte de Planejamento	18
2.1.3	Número de Produtos ou Itens	19
2.1.4	Número de Níveis	19
2.1.5	Estrutura de <i>setup</i>	20
2.1.6	Estoque e <i>back-orders</i>	20
2.1.7	Restrições de Capacidade	21
2.2	Problemas de Dimensionamento de Lotes em Máquinas Paralelas	21
2.3	Problemas de Dimensionamento de Lotes com <i>Setup Carry-over</i>	24
2.4	Considerações Finais	26
3	MODELOS MATEMÁTICOS	28
3.1	Modelo 1: Tradicional	28
3.2	Modelo 2 - Problema baseado em Caminhos Mínimos	30
3.3	Modelo 3: Problema baseado em Localização de Facilidades	33
3.4	Experimentos Computacionais	34
3.5	Considerações Finais	37
4	MATH-HEURÍSTICA PROPOSTA	38

4.1	Modelos Matemáticos	38
4.1.1	Modelo baseado em Caminhos Mínimos com <i>setup Carry-over</i>	38
4.1.2	Modelo baseado em Localização de Facilidades com <i>setup Carry-over</i>	40
4.1.3	Math - heurísticas	42
4.2	Relax-and-fix	42
4.3	Restrição de Local Branching	44
4.4	Método híbrido	45
4.5	Math-heurística Proposta	47
4.6	Math-heurística Proposta Modificada	51
4.7	Considerações Finais	53
5	EXPERIMENTOS COMPUTACIONAIS	54
5.1	Conjuntos de Instâncias	54
5.2	Experimento I	54
5.3	Experimento II	56
5.3.1	Experimento II.a	57
5.3.2	Experimento II.b	59
5.4	Experimento III	61
6	CONCLUSÕES	63
	REFERÊNCIAS	65

1 Introdução

A grande competitividade presente no meio empresarial acarreta o ensejo de empresas reduzirem seus custos e melhorarem sua produtividade. Para isso, se faz importante um gerenciamento do processo de produção com o intuito de decidir a melhor forma de utilização dos recursos disponíveis. O sistema que coordena esse gerenciamento e denominado planejamento e controle da produção (PCP) e possui três níveis: estratégico, tático e operacional (ANTHONY, 1965).

O planejamento estratégico possui uma estrutura ampla e abrange toda a organização, suas projeções são impostas em longo prazo e seus efeitos e consequências são estendidos para vários anos. Já o planejamento tático propõe variações para o médio prazo e abrange cada unidade da organização. Por fim, o planejamento operacional é projetado em curto prazo, considerando cada tarefa contida nas unidades da organização, preocupando-se com metas específicas (MOREIRA, 1998).

O problema de dimensionamento de lotes faz parte do planejamento tático da produção e estabelece a quantidade a ser produzida (ou o tamanho do lote) de cada item em uma determinada fábrica para cada período de um horizonte de planejamento finito. O objetivo é que esse plano de produção minimize a soma dos custos de preparação, de produção e de estoque, respeitando os recursos disponíveis e atendendo as demandas dos períodos que são previamente determinadas.

O primeiro trabalho a propor uma resolução para o problema de planejamento e controle da produção por meio da programação matemática foi desenvolvido por (WAGNER; WHITIN, 1958). Os autores apresentaram um algoritmo de programação dinâmica, de ordem quadrática sobre o número de períodos, que determina a solução ótima do problema de dimensionamento de lotes com um único item e sem considerar restrições de recursos.

É expressivo o número de estudos de problemas de dimensionamento de lotes (PDLs) nas últimas seis décadas. Uma importante característica que tem motivado os pesquisadores a continuar investigando PDLs é a grande quantidade de variações existentes para o mesmo (NASCIMENTO *et al.*, 2010; KARIMI *et al.*, 2003). Algumas dessas variantes são computacionalmente desafiadoras, que é o caso de problemas de dimensionamento de lotes com múltiplas plantas (NASCIMENTO *et al.*, 2010; CARVALHO; NASCIMENTO, 2016), com

máquinas paralelas (TOLEDO; ARMENTANO, 2006), entre outras.

Neste trabalho, estuda-se o problema de dimensionamento de lotes em que o setor de produção é composto por máquinas paralelas distintas com capacidades limitadas e um tempo de preparação (*setup*) que é imposto no início do processo de produção de cada item em todas as máquinas. Esse problema é formalmente proposto em (SABBAG, 1993) e pode ser encontrado no Brasil em vários setores como, por exemplo, de indústrias de aço laminado, de bebidas, de colchões, entre outras. Adicionalmente, visando a otimização da realização de *setup*, é permitido que a preparação associada a um determinado período seja realizada em um período anterior (*setup carry-over*), seja pela execução antecipada do *setup*, seja pelo aproveitamento do estado da máquina em período anterior (SOX; GAO, 1999).

Para abordar o problema em questão, primeiramente, foram estudados modelos matemáticos para o Problema de Dimensionamento de Lotes com Máquinas Paralelas (PDLMP), com o intuito de identificar o melhor modelo para que o *setup carry-over* fosse inserido e então, elaborar a math-heurística a partir do modelo inferido. Os modelos estudados são apresentados em (TOLEDO, 1998), (JANS, 2009) e em (SILVA, 2013). Toledo (1998) modela o PDLMP como um problema de programação linear inteira mista e, a fim de tornar o problema mais restrito, define um limitante inferior para a produção. Jans (2009) apresenta uma reformulação baseada no Problema de Caminho Mínimo. Silva (2013) estuda uma reformulação baseada no problema de localização de facilidades do modelo clássico para o Problema de Dimensionamento de Lotes com Múltiplas Plantas, que foi adaptado neste trabalho para o tratamento do Problema de Dimensionamento de Lotes com Máquinas Paralelas.

Esses três modelos foram implementados no CPLEX 12.6.3 (ILOG, 2015) para resolver os problemas baseados em instâncias propostas em (TOLEDO, 1998). Os experimentos foram realizados com 180 instâncias da literatura, como o modelo adaptado de (JANS, 2009) se mostrou mais vantajoso pois, com ele, resolveu-se na otimalidade uma quantidade maior de problemas, adicionou-se *setup carry-over* neste modelo. Mas, o modelo baseado no problema de localização de facilidades é um modelo no qual métodos do tipo *relax-and-fix* apresentam bons resultados computacionais para instâncias de médio porte (SILVA, 2013). Considerando que a math-heurística proposta neste trabalho utiliza a heurística *relax-and-fix*, o modelo baseado no problema de localização de facilidades também foi selecionado para a inclusão do *setup carry-over*.

Então, a partir de cada um dos modelos escolhidos implementou-se uma estratégia que combina programação matemática e heurísticas, hoje conhecida como math-heurística, que tem sido bastante explorada na literatura nos últimos anos devido à eficiência na resolução de problemas combinatoriais. As math-heurísticas têm apresentado melhores desempenhos que os demais métodos recentemente propostos para vários problemas, assim

como em (CARVALHO; NASCIMENTO, 2016) e (SILVA, 2013), por exemplo. Neste trabalho, o método proposto por (SILVA, 2013) foi usado como inspiração para o desenvolvimento da math-heurística, que combina a estratégia *relax-and-fix* com restrições de *local-branching*. Essa combinação baseou-se no princípio da meta-heurística *Greedy Randomized Search Procedures* (GRASP) (FEO; RESENDE, 1995), que possui múltiplas iterações, cada uma com duas fases: uma de construção e outra de busca local.

Para verificar a eficiência da math-heurística proposta, também foram utilizadas instâncias de Toledo (1998), as soluções obtidas pelo CPLEX e pela math-heurística foram comparadas. Em uma primeira análise, o CPLEX apresentou resultados melhores, porém, com algumas modificações na estrutura da math-heurística as soluções obtidas por ela, demonstraram-se satisfatórias e melhores do que as soluções obtidas pelo CPLEX, em um tempo máximo de 1800 segundos.

1.1 Organização

O restante deste trabalho está organizado da seguinte maneira: no Capítulo 2 é apresentada uma revisão bibliográfica do problema estudado; no Capítulo 3 são apresentados três diferentes modelos matemáticos da literatura que tratam o PDLMP, que são avaliados por meio da solução de instâncias *benchmark* no pacote de otimização CPLEX 12.6.3 (ILOG, 2015). No Capítulo 4, são exibidos o acréscimo da possibilidade de *setup carry-over* em dois dos três modelos apresentados no Capítulo 3 e a math-heurística desenvolvida a partir desses modelos. No Capítulo 5, são apresentados os resultados dos experimentos computacionais. Por fim, no Capítulo 6, as conclusões e trabalhos futuros são discutidos.

2 O Problema de Dimensionamento de Lotes

O problema de dimensionamento de lotes clássico envolve a determinação de quanto produzir em cada um dos períodos de um horizonte de planejamento finito. O objetivo é encontrar um plano de produção de mínimo custo capaz de atender as demandas dos produtos sem violar a capacidade de produção disponível. A literatura destinada ao problema de dimensionamento de lotes é bastante rica e boas revisões podem ser encontradas em (KARIMI *et al.*, 2003; DREXL; KIMMS, 1997; WOLSEY, 1995; KUIK *et al.*, 1994; BAHL *et al.*, 1987; BRAHIMI *et al.*, 2006; JANS; DEGRAEVE, 2008; BUSCHKÜHL *et al.*, 2010). Em (KARIMI *et al.*, 2003), são destacadas algumas características que influenciam a classificação, modelagem e complexidade das decisões do problema de dimensionamento de lotes.

Florian *et al.* (1980) provaram que vários PDLs com um único item e custos de preparação são NP-difíceis. Bitran e Yanasse (1982) mostraram que vários casos de um único item que podem ser resolvidos em tempo polinomial tornam-se NP-difíceis quando um segundo item é introduzido. O problema de encontrar uma solução factível quando se considera o tempo de preparação da máquina é NP-completo, segundo Maes *et al.* (1991). Os autores também mostraram que se os tempos de preparação são nulos, o problema de factibilidade é da classe P. Devido à dificuldade deste problema, poucos métodos exatos foram propostos para sua solução (ARMENTANO *et al.*, 1999; DIABY *et al.*, 1992a; SOUZA; ARMENTANO, 1994). No entanto, várias heurísticas foram introduzidas, das quais podem ser destacadas as desenvolvidas por Toledo e Armentano (2006), Diaby *et al.* (1992b), Trigeiro *et al.* (1989), Lozano *et al.* (1991), Nascimento *et al.* (2010).

2.1 Características de um PDL

Para uma melhor compreensão de um PDL, segue um resumo das principais características deste problema. As terminologias presentes são inspiradas na revisão de (KARIMI *et al.*, 2003).

2.1.1 Demanda

Em um PDL, a demanda corresponde à quantidade necessária de produção no horizonte de planejamento. Este valor é determinado por meio de ordens de compra dos clientes ou valores estimados conforme histórico do processo produtivo. Assim, para cada dupla item-período (i, t) , $1 \leq i \leq n$ e $1 \leq t \leq p$, em que n é a quantidade de itens e p é a quantidade de períodos do horizonte de planejamento, a demanda pode ser do tipo:

- estacionária ou estática: não há alteração em seu valor durante todo o horizonte de planejamento,
- dinâmica: é possível que os valores das demandas sejam diferentes ao longo dos períodos. Esse tipo de demanda é o mais estudado nas últimas décadas e resultados importantes a respeito de variantes de PDLs podem ser encontrados na literatura (WAGNER; WHITIN, 1958).

Independentemente de ser estática ou dinâmica, a demanda pode ser:

- Determinística: quando seu valor é conhecido previamente ao planejamento, e compreende a maior parte dos estudos devido à sua alta aplicabilidade, considerando que ao realizar o planejamento em nível tático, na maioria dos casos, as demandas são antecipadamente definidas. Alguns exemplos de trabalhos envolvendo esse tipo de demanda são encontrados em (TOLEDO, 1998; SABBAG, 1993; JANS, 2009).
- Não determinística ou probabilística: quando os valores de demanda são determinados por uma função de probabilidade. A dificuldade em modelar esse tipo de demanda e o custo computacional envolvido nos algoritmos para resolver problemas com tal tipo de demanda são razões para que ela não tenha sido amplamente estudada na literatura. Entretanto, nos últimos anos, com a evolução dos processadores, alguns estudos interessantes apresentam trabalhos que tratam esse tipo de demanda (PRODUCTION..., 2012).

2.1.2 Horizonte de Planejamento

O horizonte de planejamento, ou o tempo total para o planejamento da produção pode ser definido, de maneira geral, como:

- finito: no caso em que o atendimento da demanda está sujeito a restrições de tempo.
- infinito ou ilimitado: no caso em que não existe restrição de tempo para realizar toda a produção.

Outra característica importante é o tempo relativo a um período de produção, que pode ser de dois tipos:

- *big bucket*: é possível produzir vários itens em um único período, frequente em indústrias alimentícias;
- *small bucket*: a capacidade de tempo disponível é utilizada para produzir apenas um item por período (LASDON; TERJUNG, 1971).

2.1.3 Número de Produtos ou Itens

Considerando a quantidade de itens que podem ser produzidos, o sistema de produção pode fabricar um único item ou múltiplos itens. Em sistemas com único item, admite-se a produção de apenas um produto final, enquanto em sistemas com múltiplos itens é prevista a produção de diversos produtos finais.

Caso exista mais de um item no sistema de produção, eles podem ser:

- dependentes: um item só pode ser produzido depois da fabricação de outro item específico, ou,
- independentes: um item não depende de outro para ser produzido.

2.1.4 Número de Níveis

A dependência e independência nos processos necessários para a transformação da matéria-prima em um produto final influenciam no número de níveis de um sistema de produção. Nesse sentido, um PDL pode ser definido como:

- Único nível (monoestágio): Nesse tipo de PDL, a transformação da matéria-prima em um produto final é direta, isto é, não existem etapas intermediárias ou dependentes entre si para a fabricação do item. Assim, a demanda produzida em cada nível é do tipo independente.
- Multi-nível (múltiplos estágios): Nesse PDL, a fabricação do produto final depende das operações realizadas em cada nível do sistema de produção. Ou seja, a produção de um item depende da disponibilidade de itens produzidos em níveis anteriores. Por isso, nesse caso, a demanda de cada item é do tipo dependente.

2.1.5 Estrutura de *setup*

A fabricação de um item requer tempo e custo de *setup* (preparação) da máquina aonde este produto será produzido. Essas duas características definem as estruturas de *setup* de máquinas em sistemas de produção. A estrutura de preparação pode ser simples ou complexa. Na estrutura de *setup* simples, o tempo e o custo de *setup* não sofrem influência do que ocorre nos períodos anteriores. Na estrutura de *setup* complexa, ambos, tempo e custo de preparação, são dependentes do sequenciamento de tarefas e/ou de decisões tomadas em períodos anteriores. A estrutura complexa possui diversos tipos, alguns citados a seguir:

- *Setup carry-over*: Essa nomenclatura refere-se à preparação da máquina nos finais dos períodos de produção. Nesse caso, se no final de um dado período a máquina está preparada para a produção de um determinado produto, no início do período seguinte não existe a necessidade de realizar uma nova preparação da máquina, se esse for o primeiro item a ser produzido em tal período. A preparação, nesse caso, pode ser “carregada” de um período a outro imediatamente posterior que haja produção. Gopalakrishnan *et al.* (1995) também consideram como *setup carry-over* o adiantamento da preparação da máquina para produção do primeiro item no período imediatamente posterior com produção. Dessa forma, mesmo que não haja a produção de um determinado item em certo período, pode-se aproveitar a capacidade ociosa para adiantar a preparação da máquina para a produção do primeiro item do período posterior.
- *Setup* por similaridade: Itens similares podem ser fabricados por uma mesma máquina, sendo necessário que o *setup* seja realizado apenas na fabricação de um produto similar. Para isso, é importante que os itens similares sejam produzidos em sequência e que o *setup* seja realizado no primeiro item desta sequência (SIMÕES, 2016).
- *Setup* de sequência dependente: Engloba os problemas em que o *setup* é dependente da sequência determinada para a produção dos itens (BARROS; MOCCELLIN, 2004).
- *Setup crossover* ou *Setup Splitting*: É possível que o *setup* comece no fim de um período e termine no início do período seguinte, isto é, o *setup* pode expandir-se sobre dois períodos (PAIVA, 2010).

2.1.6 Estoque e *back-orders*

Em um PDL, pode existir a possibilidade de estoque de um item quando a quantidade de produção dele somado com o que já existe no estoque em um determinado período

ultrapassar a sua demanda neste período, isso pode ocorrer por fatores como:

- Economia de *setup*: quando é mais vantajoso realizar o *setup* e a produção em um período pi e estocar para atender a demanda de algum próximo período pe do que pagar o *setup* e o custo de produção do período pe ;
- Escassez de capacidade: quando a capacidade das máquinas em um determinado período não é o suficiente para a produção de toda a demanda desse período.

Neste caso, a estocagem de produção é permitida e pode ser sujeita a custos, à capacidade do armazém de estocagem, entre outros fatores.

Além disso, em alguns problemas, é admitido que haja atraso (*back-orders* ou *backlogging*) no atendimento de demanda, também sujeito à penalidade nos custos. Para a representação de tal situação, a maioria das modelagens determinam variáveis que representam os atrasos na demanda.

2.1.7 Restrições de Capacidade

Os PDLs podem possuir restrições que limitam os recursos das máquinas. Para levar em conta tal limite, os tempos de produção e de *setup* da máquina são considerados no total de tempo gasto, que deve estar dentro desse limite. Assim, quando existem tais restrições, havendo tempo máximo de uso, o problema é dito capacitado. A ausência dessas restrições implica em um problema sem restrições de capacidade.

Restrições de capacidade têm um significativo impacto na complexidade dos PDLs. Problemas capacitados são, geralmente, mais difíceis de serem solucionados. Isto é, enquanto para uma parte de problemas não capacitados é possível encontrar uma solução factível em tempo polinomial, para uma significativa parte de problemas capacitados, isso não é possível.

2.2 Problemas de Dimensionamento de Lotes em Máquinas Paralelas

O Problema de Dimensionamento de Lotes em Máquinas Paralelas (PDLMP) considera um conjunto de máquinas, que podem ou não produzir todos os itens da fábrica. Os produtos são alocados nessas máquinas, podendo formar uma sequência de produção dos lotes. Em problemas desse tipo, os custos de produção e de troca de preparação dependente da máquina podem ser considerados.

Em um PDLMP, as máquinas podem ser idênticas ou distintas. Em problemas envolvendo máquinas paralelas idênticas, consideram-se idênticos para todas as máquinas os tempos de *setup* por produto, as capacidades e os tempos de produção por produto. Nos PDLMPs considerando máquinas paralelas distintas, os valores desses dados distinguem-se entre as máquinas.

Uma revisão dessa classe de problemas, organizada em ordem cronológica, é apresentada a seguir.

Um pouco mais distante do tema deste estudo, vale mencionar o trabalho de (LASDON; TERJUNG, 1971). Os autores desenvolveram um algoritmo, baseado em técnicas de geração de colunas e no método de delimitação generalizada de Dantzig e Van Slyke, para o problema de dimensionamento de lotes com máquinas paralelas idênticas, sem considerar tempo de preparação de máquina. Em seu estudo, somente um item é produzido por máquina e período. Em experimentos computacionais, compararam-se os resultados obtidos diretamente pela decomposição de Dantzig–Wolfe com a proposta do trabalho e atestou-se que a quantidade de iterações necessárias para que o algoritmo proposto atingisse a otimalidade foi de um meio a um nono em relação à quantidade exigida pela estratégia comparada. Uma observação importante é que o custo computacional de cada iteração em ambos os métodos é basicamente o mesmo.

Mais relacionados ao PDL visado, vale mencionar os trabalhos desenvolvidos em (CARRENO, 1990; SABBAG, 1993; TOLEDO, 1998; BERALDI *et al.*, 2008; JANS, 2009; QUADT; KUHN, 2009; FIOROTTO *et al.*, 2015). Nesses estudos, os autores visaram o problema de dimensionamento de lotes capacitado com máquinas paralelas e múltiplos itens independentes considerando tempo e/ou custo de *setup*.

Carreno (1990) desenvolveu uma heurística para o PDL com máquinas paralelas idênticas no qual leva-se em conta tempo e custo de *setup* das máquinas. Além disso, cada máquina pode produzir apenas um produto por período e o processo de produção de cada item inicia-se apenas quando a quantidade em estoque desse produto é zero. Os resultados de experimentos computacionais mostraram que a heurística proposta solucionou, com menos do que 5% de desvio em relação ao limitante inferior, problemas envolvendo 10 itens e 100 máquinas em baixos tempos computacionais.

Sabbag (1993) tratou do problema de dimensionamento de lotes com máquinas paralelas distintas considerando tempo de *setup* das máquinas e que visa a minimização de custos. Ele desenvolveu uma heurística construtiva que inicia o processo de busca por meio de uma solução infactível, gerada por meio da relaxação das restrições de capacidade ou das variáveis inteiras e busca a factibilidade por meio de transferências da quantidade de produção entre períodos e máquinas. Foram testadas instâncias envolvendo 100 itens, 6 máquinas e 24 períodos que foram resolvidas com *gaps* razoavelmente baixos em um

tempo computacional razoável.

Toledo (1998) estuda o PDL, em que consideraram-se máquinas paralelas não relacionadas na produção de múltiplos itens. Esse problema tem como objetivo minimizar a soma dos custos de produção, *setup* e estoque. Foram propostos dois métodos exatos e uma heurística para resolução deste problema. O primeiro método exato baseia-se em uma relaxação Lagrangiana das capacidades e subsequente transferência de lotes de produção para a factibilização da solução. O segundo método exato foi desenvolvido a partir da reformulação do problema como uma rede generalizada e relaxação linear. A heurística desenvolvida segue a mesma ideia da heurística proposta por Sabbag (1993) em que são implementadas apenas algumas modificações. Neste trabalho, propõem-se experimentos com instâncias artificiais melhor explicadas na Seção 5. Tais instâncias têm 6, 12, 25 ou 50 itens; 6, 12 ou 18 períodos; e 2 ou 4 máquinas. Os resultados mostraram-se satisfatórios com *gaps* em média de 2% e 11% para instâncias com baixo e alto custo de *setup*, respectivamente.

Toledo e Armentano (2006) desenvolveram uma heurística baseada no relaxamento lagrangiano das restrições de capacidade e no método do subgradiente para solucionar o PDLMP. Inicialmente, a heurística é testada em instâncias do problema de uma única máquina e os resultados são comparados com heurísticas da literatura. Para máquinas paralelas e pequenos problemas, o desempenho da heurística testado contra soluções ótimas, e para problemas maiores é comparado com o limite inferior fornecido pelo relaxamento Lagrangiano. Os resultados obtidos mostraram que a heurística proposta obtém melhores resultados.

Beraldi *et al.* (2008) resolveram o PDLMP considerando custos de *setup* dependentes de sequência através da heurística *relax-and-fix* e utilizaram diversas partições do conjunto de variáveis inteiras. Os autores compararam as soluções obtidas com as soluções de duas heurísticas *rolling-horizon*. Os resultados mostraram que as heurísticas *relax-and-fix* são melhores do que as heurísticas de *rolling-horizon*.

Também visando o problema alvo desta dissertação, na literatura existe o trabalho de Jans (2009), porém com máquinas idênticas. Nele, o autor apresenta um modelagem matemática baseada no problema do caminho mínimo e observa que quando diversas máquinas idênticas estão disponíveis, muitas soluções ótimas alternativas podem ser criadas por renumeração das máquinas. Essas soluções alternativas dificultam o processo de *branch-and-bound*. Dessa forma, ele propõe, na modelagem apresentada, novas restrições para quebrar essa simetria de soluções. Os testes computacionais foram realizados utilizando 40 instâncias com 6 produtos, 15 períodos e de 2 até 5 máquinas. Os resultados indicaram que várias das restrições de quebra de simetria melhoraram o tempo de resolução, exceto quando se trata de problemas muito fáceis. Vale ressaltar que as restrições de quebra de simetria não são consideradas na análise aqui realizada, uma vez que só é

factível acrescentá-las em problemas que tratam de máquinas idênticas.

Quadt e Kuhn (2009) estudaram o PDL com máquinas paralelas e *setup carry-over*, entretanto, possibilitando *back-orders* (permissão de atraso para atender a demanda). Os autores propuseram uma formulação matemática considerando o problema como inteiro misto e desenvolveram uma formulação inspirada no denominado “modelo agregado”, cuja ideia principal consiste em evitar variáveis binárias (que guardam um valor binário para cada máquina, produto e período) e usar, ao invés delas, variáveis inteiras que contam a quantidade de máquinas que executam *setup* (como restrições *surrogate*) para um determinado produto em um período específico. Assim, o número de variáveis não contínuas é substancialmente reduzido. Os autores ressaltam que devido a essa característica, a abordagem de resolução apresentada funciona apenas para o caso de máquinas idênticas, não se aplicando para a configuração de máquinas distintas. O problema foi resolvido considerando a formulação proposta e as soluções obtidas, utilizando o CPLEX limitado a 120 segundos, foram ótimas ou próximas das ótimas.

Mais recentemente, Fiorotto *et al.* (2015) resolveram o PDLMP por meio de um método de solução que combina a relaxação Lagrangiana e a decomposição de *Dantzig-Wolfe*. Os experimentos computacionais mostraram que o método híbrido produziu limites inferiores de excelente qualidade.

2.3 Problemas de Dimensionamento de Lotes com *Setup Carry-over*

O *setup carry-over* tem sido utilizado em PDLs capacitados a fim de melhor aproveitar as capacidades das máquinas. Essa consideração economiza o tempo de utilização das máquinas pela não necessidade de realizar *setups* desnecessários através do aproveitamento do estado da máquina em fim de período. Da literatura, vale mencionar os trabalhos pioneiros de (GOPALAKRISHNAN *et al.*, 1995; SOX; GAO, 1999; PORKKA *et al.*, 2003). Esse tipo de consideração é característico de indústrias que não possuem produtos alimentícios perecíveis em seus processos de produção, pois, nesses casos, aproveitar o estado da máquina, de um determinado período para o seguinte, sem nova preparação (eliminação de resíduos) pode influenciar na composição do produto final. Esta seção apresentará a literatura mais recente de PDLs com *setup carry-over*, com o intuito de melhor definir os objetivos descritos neste trabalho.

Almada-Lobo *et al.* (2007) consideram o problema de dimensionamento de lotes capacitado com múltiplos itens no processo de produção de recipientes de vidro: incluindo a produção de vidro (com variáveis contínuas) e a fabricação de recipiente (com variáveis discretas). Nesse problema, o tempo de *setup* depende da sequência, e está relacionado

à preparação do forno que tem capacidade de utilização limitada. Nesse caso, é possível realizar o *setup* do período seguinte no período atual: *setup carry-over*. Duas formulações matemáticas são apresentadas, incorporando todas as características necessárias de *setup carry-over*. Para resolução do problema foi utilizada uma heurística de cinco etapas que se mostrou eficaz em encontrar boas soluções mesmo para instâncias grandes.

Nascimento e Toledo (2008) trataram do problema de dimensionamento de lotes capacitado com um único estágio considerando múltiplas plantas, itens e períodos com *setup carry-over*. As autoras apresentam um modelo matemático e uma heurística GRASP (*Greedy Randomized Adaptive Search Procedure*) com *path relinking* para resolução deste problema. Esse método de solução é uma extensão de uma heurística anteriormente proposta, em que não consideram *setup carry-over*. Os testes computacionais mostraram que a inclusão do *setup carry-over* apresentou uma melhora significativa nas soluções do problema com apenas um pequeno acréscimo no tempo de execução do algoritmo. Sahling *et al.* (2009) consideraram de um problema bastante similar ao de (NASCIMENTO; TOLEDO, 2008) em que múltiplos níveis são considerados, denominado problema de dimensionamento de lotes capacitado multi-nível com *setup carry-over*. A heurística proposta para sua resolução define uma série de subproblemas inteiros mistos. Para cada instância, diversas variáveis binárias são fixadas segundo alguns critérios e apenas um pequeno conjunto de variáveis binárias são otimizadas. Tal heurística é conhecida por *fix-and-optimize* e apresentou boas soluções para o problema em questão.

Chen (2015) também utilizou a heurística *fix-and-optimize*, mas para resolver o problema de dimensionamento de lotes capacitado monoestágio com múltiplos itens considerando *setup carry-over*, *setup splitting* e *backlogging* (possibilidade de atraso para atender a demanda). Este foi um dos primeiros trabalhos da literatura que considera, no problema de dimensionamento de lotes, o *setup carry-over* e o *setup splitting* simultaneamente. Os resultados da heurística proposta pelos autores apresentaram diferenças médias (entre a solução ótima e a solução dada pela heurística) de 6% para os casos sem *backlogging* e de 8% considerando *backlogging*.

A consideração de *backlogging* também é encontrada em (PAIVA; TOLEDO, 2010), no problema de dimensionamento de lotes com restrições de capacidade (PDLC). Neste trabalho, o autor propôs um método *local branching* para solução do PDLC com três características adicionais: 1) *backlogging*; 2) *setup carry-over*; 3) *setup crossover*. Avaliou-se a influência destas características na melhoria do custo dos planos de produção. Os resultados obtidos mostraram que as soluções encontradas pelo método *local branching* proposto são melhores com relação à qualidade do que às soluções obtidas pelo *software* comercial CPLEX, versão 11.0, limitado em 250 segundos. Com esse estudo, os autores sugeriram que considerar *backlogging*, *carry-over* e *setup crossover* pode proporcionar economia significativa durante o planejamento da produção.

Toso *et al.* (2008) estudaram a combinação de abordagens GLSP (*General Lot Sizing and Scheduling Problem*) e ATSP (*Asymmetric Travelling Salesman Problem*) para o problema de dimensionamento e sequenciamento de lotes na indústria de nutrição animal. Inicialmente, os autores consideraram cada período do horizonte de planejamento de maneira independente (admitindo-se limpeza da linha no final de cada período) e, depois, consideraram todos os períodos de maneira dependente, incluindo variáveis para guardar a sequência entre os períodos (*carry-over*). Para a resolução do problema, foram utilizados dois métodos baseados na eliminação de subrotas e *patching*. A abordagem proposta foi comparada com abordagens *relax-and-fix* para o modelo GLSP estudadas em trabalhos anteriores. Considerando todos os experimentos realizados, foi constatado que a melhor estratégia para abordar o problema é o modelo ATSP - considerando *carry-over*, resolvido com o método combinado de eliminação de sub-rotas e *patching*.

Caserta *et al.* (2010) resolveram o problema de dimensionamento de lotes multi-nível com *setup carry-over*. Para isso, os autores propuseram uma meta-heurística para resolver iterativamente uma menor quantidade de subproblemas resultantes do problema original. O algoritmo utilizado explora porções adjacentes do espaço de busca. Para realização dos testes foram utilizadas instâncias contidas em uma biblioteca padrão e os resultados apresentados mostraram a robustez e a eficácia do esquema proposto.

Belo-Filho *et al.* (2014) propuseram dois modelos para o PDL capacitado com *backlogging*, *setup carry-over* e *setup cross-over*. O primeiro possui características semelhantes à outros modelos da literatura, enquanto o segundo considera uma variável de configuração desagregada, que rastreia os tempos de início e de conclusão do *setup*. Esta abordagem permitiu uma formulação mais compacta do problema em questão. Os resultados computacionais mostram que os modelos propostos superaram as formulações da literatura.

Mais recentemente, Cunha *et al.* (2015) propuseram amplificações para o problema de dimensionamento de lotes multi-nível, considerando: custos de estoque, de *setup* e de *overtime*; tempo de *setup*; *setup carry-over*; custo de aquisição e custo de estoque das matérias-primas visando obter planos produtivos financeiramente mais eficientes do que os obtidos com os modelos tradicionais. Os experimentos computacionais, utilizando o CPLEX 12.6, mostraram que esse modelo é capaz de encontrar soluções de melhor qualidade quando comparadas com as obtidas pelo modelo exclusivamente de dimensionamento de lotes, com custos de aquisição e estoque de matérias-primas computados *a posteriori*.

2.4 Considerações Finais

Este capítulo apresentou as principais características de um PDL e uma revisão dos trabalhos da literatura que abordam estudos de PDLs, em particular, dos problemas relaci-

onados ao problema alvo desta dissertação: o PDLMP e o Problema de Dimensionamento de Lotes com *Setup Carry-over* (PDLSC). Neste trabalho, o problema tratado denomina-se Problema de Máquinas Paralelas com *Setup Carry-over* (PDLMPSC) e consideram-se um único nível de produção, múltiplos itens, demanda estacionária e determinística, máquinas capacitadas e a possibilidade de produção de vários itens em um único período (*big bucket*).

Por existir poucos trabalhos que tratam máquinas paralelas e *setup carry-over* no mesmo modelo, a revisão dos PDLs com essas características foi feita separadamente. O ponto inicial de estudo para este projeto foi a revisão de modelos para o PDLMP, a fim de encontrar uma modelagem que gere bons resultados para então incluir *setup carry-over*. Esse estudo é apresentado no capítulo seguinte.

3 Modelos Matemáticos

A fim de comparar a eficiência de três diferentes modelagens para o PDLMP sem *setup carry-over*, nesta dissertação, os modelos apresentados por Toledo (1998), Jans (2009) e Silva (2013) são adaptados e os resolvidos através do *solver* CPLEX. Supondo que o problema trata de n itens, p períodos e m máquinas. Sejam os conjuntos $A = \{1, \dots, n\}$; $B = \{1, \dots, m\}$ e $C = \{1, \dots, p\}$. Para todos os modelos, consideram-se os seguintes parâmetros:

s_{ijt} : custo de preparação para produção do item i na máquina j no período t , $\forall i \in A, j \in B, t \in C$

c_{ijt} : custo de produção do item i na máquina j no período t , $\forall i \in A, j \in B, t \in C$

h_{it} : custo unitário de estoque do item i no período t , $\forall i \in A, t \in C$

d_{it} : demanda do item i no período t , $\forall i \in A, t \in C$

f_{ij} : tempo de preparação para produção do item i na máquina j , $\forall i \in A, j \in B$

b_{ij} : tempo de produção do item i na máquina j , $\forall i \in A, j \in B$

C_{jt} : capacidade da máquina j no período t , $\forall j \in B, t \in C$.

3.1 Modelo 1: Tradicional

O PDLMP baseia-se em planejar a produção de n itens para uma quantidade p de períodos em um ambiente de produção com m máquinas. Toledo (1998) apresenta, para o PDLMP, um modelo de programação linear inteira mista e com o intuito de tornar o problema mais restrito, define um limitante inferior para a produção.

As variáveis do problema são:

x_{ijt} : quantidade do item i produzida na máquina j no período t ,

I_{it} : estoque do item i no período t ,

y_{ijt} : variável binária que assume valor 1 se o item i é produzido na máquina j no

período t , e 0 caso contrário.

Essa formulação é apresentada a seguir.

$$\min \sum_{t=1}^p \sum_{j=1}^m \sum_{i=1}^n (c_{ijt}x_{ijt} + s_{ijt}y_{ijt}) + \sum_{t=1}^p \sum_{i=1}^n h_{it}I_{it} \quad (3.1)$$

sujeito a

$$\sum_{j=1}^m x_{ijt} + I_{i,t-1} - I_{i,t} = d_{it} \quad \forall i \in A, t \in C \quad (3.2)$$

$$\sum_{i=1}^n (b_{ij}x_{ijt} + f_{ij}y_{ijt}) \leq C_{jt} \quad \forall j \in B, t \in C \quad (3.3)$$

$$b_{ij}x_{ijt} \leq \theta_{ijt}y_{ijt} \quad \forall i \in A, j \in B, t \in C \quad (3.4)$$

$$I_{i0} = 0 \quad \forall i \in A \quad (3.5)$$

$$x_{ijt} \geq 0, y_{ijt} \in \{0, 1\}, I_{it} \geq 0 \quad \forall i \in A, j \in B, t \in C \quad (3.6)$$

em que:

$$\theta_{ijt} = \min \left\{ C_{jt} - f_{ij}, \sum_{k=t}^p b_{ij}d_{ik} \right\}$$

A função objetivo expressa por (3.1) representa a minimização dos custos de produção, *setup* e estoque. As restrições (3.2) tratam do balanço de estoque. A garantia de que as capacidades das máquinas não serão violadas é dada pelo conjunto de restrições (3.3).

As restrições (3.4) certificam a incidência de custos e tempos de preparação quando há produção. Esse conjunto de restrições utiliza o limitante superior θ_{ijt} , para a produção do item i na máquina j no período t (x_{ijt}), que deixa o problema mais restrito. É possível observar que, pelo modelo apresentado, a solução ótima apresentará o estoque final nulo, pois considerando uma solução que apresenta estoque final maior do que zero para algum produto i ($I_{ip} > 0$), conclui-se que, nessa solução, $\sum_{t=1}^p x_{ijt} > \sum_{t=1}^p d_{ij}$, e portanto, existe uma outra solução, de menor custo, tal que $\sum_{t=1}^p x_{ijt} = \sum_{t=1}^p d_{ij}$ (ZANGWILL, 1966). Dado que os estoques finais são não nulos, tem-se que x_{ijt} será, no máximo, igual a $\sum_{t=1}^p d_{ij}$.

Os estoques iniciais são nulos, conforme as restrições (3.5). A quantidade de produção e estoque não negativos são garantidos pelas restrições (3.6).

3.2 Modelo 2 - Problema baseado em Caminhos Mínimos

Eppen e Martin (1987) reformularam o modelo tradicional para o PDL como um problema de caminhos mínimos. Jans (2009) adaptou essa abordagem para considerar no modelo máquinas paralelas idênticas. Nesta dissertação, o modelo de Jans (2009) foi minimamente modificado para que fossem consideradas máquinas paralelas distintas ao invés de máquinas paralelas idênticas, isso foi feito através da adição de um índice que representa cada máquina nos custos, tempos de *setup* e capacidades.

Neste modelo considera-se a variável z_{vijtl} que representa a fração do plano de produção do item i na máquina j , em que a produção no período t satisfaz, total ou parcialmente, as demandas do item i do período t até o período l . Considere um problema com três períodos e duas máquinas. A Figura 3.1 ilustra esse exemplo.

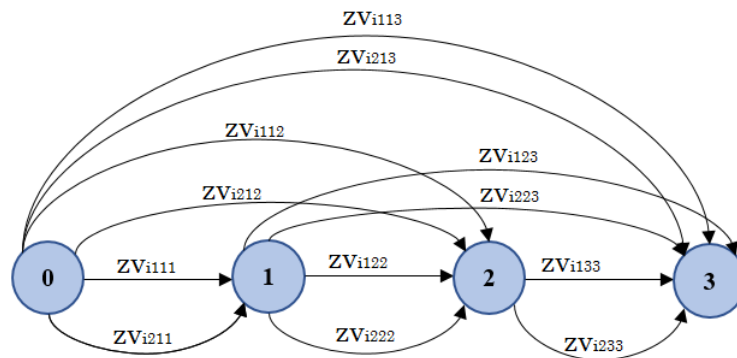


FIGURA 3.1 – Exemplo do modelo de caminho mínimo para 3 períodos e 2 máquinas - adaptada de (JANS, 2009)

No exemplo da Figura 3.1 cada nó representa um período e z_{vi1tl} indica a fração do produto i produzida na máquina 1 no período t para atender a demanda do período l , enquanto z_{vi2tl} indica a fração do produto i produzida na máquina 2 no período t para atender a demanda do período l , $l \geq t$.

Nessa formulação, além das variáveis z_{vijtl} têm-se:

w_{it} : fração do plano de estoque inicial do item i em que a demanda é atendida para os primeiros t períodos,

sd_{itl} : soma das demandas do item i do período t até o período l ,

fc_i : custo de uma unidade do estoque inicial para o item i ,

y_{ijt} : variável binária que assume valor 1 se o item i é produzido na máquina j no período t , e 0 caso contrário,

cv_{ijtl} : custo total de produção e estoque para produzir o item i , na máquina j no

período t atendendo a demanda dos períodos t até l , dado pela Equação (3.7),

ci_{it} : custo total para que o estoque inicial do item i atenda a demanda dos períodos 1 até o período t , dado pela Equação (3.8). O custo de produção do item i na máquina j , em que a produção no período t satisfaz, total ou parcialmente, as demandas do item i do período t até o período l é dado pela Equação (3.7).

$$cv_{ijtl} = c_{ijt}sd_{itl} + \sum_{s=t+1}^l \sum_{u=t}^{s-1} h_{iu}d_{is}, \quad (3.7)$$

Já o custo total para que o estoque inicial do item i atenda a demanda dos períodos 1 até o período t , dado pela Equação (3.8).

$$ci_{it} = fc_i sd_{i1t} + \sum_{s=2}^t \sum_{u=1}^{s-1} h_{iu}d_{is}. \quad (3.8)$$

Segue o modelo:

$$\min \sum_{t=1}^p \sum_{j=1}^m \sum_{i=1}^n s_{ijt}y_{ijt} + \sum_{i=1}^n \sum_{j=1}^m \sum_{t=1}^p \sum_{l=t}^p cv_{ijtl}z_{vijtl} + \sum_{i=1}^n \sum_{t=1}^p ci_{it}w_{it} \quad (3.9)$$

sujeito a

$$1 = \sum_{l=1}^p \sum_{j=1}^m z_{vijl} + \sum_{l=1}^p w_{il} \quad \forall i \in A \quad (3.10)$$

$$w_{i,t-1} + \sum_{s=1}^{t-1} \sum_{j=1}^m z_{v_{i,j,s,t-1}} = \sum_{l=t}^p \sum_{j=1}^m z_{vijtl} \quad \forall i \in A, t \in C; (t \geq 2) \quad (3.11)$$

$$\sum_{t=1}^p \sum_{j=1}^m z_{vijtp} = 1 \quad \forall i \in A \quad (3.12)$$

$$\sum_{l=t}^p z_{vijtl} \leq y_{ijt} \quad \forall i \in A, j \in B, t \in C \quad (3.13)$$

$$\sum_{i=1}^n f_{ij}y_{ijt} + \sum_{i=1}^n \sum_{l=t}^p b_{ij}sd_{itl}z_{vijtl} \leq C_{jt} \quad \forall j \in B, t \in C \quad (3.14)$$

$$z_{vijtl} \geq 0; y_{ijt} \in \{0, 1\}; w_{it} \geq 0 \quad \forall i \in A, j \in B, t \in C, l \in C; (l \geq t) \quad (3.15)$$

Pelas definições de z_{vijtl} e w_{it} , temos que:

$$x_{ijt} = \sum_{l=t}^p sd_{itl} z v_{ijtl}$$

$$I_{i0} = \sum_{l=1}^p sd_{i1l} w_{il}$$

A função objetivo expressa por (3.9) minimiza a soma dos custos de estoque inicial, preparação, produção e estocagem. As restrições (3.10), (3.11) e (3.12) representam restrições de fluxo para a rede de caminho mínimo, o conjunto de restrições (3.10) e (3.12) definem que para cada produto, uma unidade de fluxo é enviada à rede, impondo que a demanda de cada produto deve ser atendida sem atrasos, as restrições (3.11) certificam que a quantidade disponível de um determinado produto em um dado período deve descender de um período anterior juntamente com o estoque inicial e a produção daquele período. As restrições (3.13) forçam o *setup* para cada item que será produzido e o conjunto de restrições (3.14) limita a soma dos tempos de *setup* e produção pela capacidade disponível em cada período e em cada máquina.

É importante ressaltar que os outros dois modelos apresentados neste trabalho não consideram estoque inicial. Por isso, no modelo descrito nesta seção a variável f_{c_i} recebe um valor alto de modo que não há vantagem em atribuir estoque inicial de qualquer produto considerado.

Nesta reformulação considera-se estoque inicial por dois motivos:

- os problemas serão sempre factíveis, pois caso não exista possibilidade de produção no primeiro período, o modelo atribuirá estoque inicial para atender a demanda deste período;
- uma reformulação utilizando o Problema de Caminho Mínimo sem a consideração de estoque inicial não modelaria corretamente instâncias em que a demanda no primeiro período é zero. A razão é que, nesse caso, os somatórios $\sum_{j=1}^m z v_{ij11}$, para cada produto i , que representam as proporções das demandas do período 1 atendidas no período 1, deveriam equivaler a 1, a fim de atender as restrições (3.9) - sem consideração de proporções do estoque inicial w_{it} . Levando-se em conta o conjunto de restrições (3.11), isso implicaria numa ativação de *setup*, mesmo que nada fosse produzido. Esse problema é resolvido pelo estoque inicial: se a demanda no primeiro período for zero, será considerada uma proporção de estoque inicial neste período (w_{i1}) e isso não implicará na ativação de custo de *setup*. Além disso, como o custo deste estoque inicial é zero, pois não há demanda no primeiro período (o custo é multiplicado pela demanda), também não existirá custo de estoque.

3.3 Modelo 3: Problema baseado em Localização de Facilidades

Wu e Shi (2011) observaram que a modelagem para o PDL proposta por Bilde e Krarup (1977) resulta em limitantes de boa qualidade quando obtidos pela relaxação linear. Essa formulação é baseada no problema de localização de facilidades e é um modelo no qual métodos do tipo *relax-and-fix* comumente obtêm boas soluções para instâncias de médio porte. Nesse sentido, Silva (2013) adaptou o modelo proposto por Bilde e Krarup (1977) para o PDL em múltiplas plantas. A ideia central desse modelo consiste em desagregar as variáveis de produção em relação às plantas e aos períodos no quais os produtos são produzidos e entregues nas devidas plantas.

No presente trabalho, o modelo proposto em (SILVA, 2013) foi ajustado para tratar o PDLMP. Portanto, a desagregação foi realizada em relação aos períodos. Assim, as variáveis de produção (x_{ijt}) que representam a quantidade produzida do item i , na máquina j , no período t passam a ser da forma:

x_{ijtu} : quantidade do item i produzido na máquina j no período t para atender a demanda do período u .

A desagregação das variáveis de produção descarta a necessidade de variáveis de estoque, pois a produção é agora direcionada a um período específico. Portanto, sabe-se implicitamente por quanto tempo os produtos serão estocados (entre os períodos t e u).

Dadas as mudanças ocasionadas pela desagregação, os custos de produção em associação às variáveis devem englobar os custos de produção e estoque dos produtos, esses custos são obtidos da seguinte forma:

$$\bar{c}_{ijtu} = \begin{cases} 0, & \text{se } u < t \\ c_{ijt} + \sum_{v=t}^u h_{iv}, & \text{caso contrário.} \end{cases}$$

Segue o modelo proposto.

$$\min \sum_{i=1}^n \sum_{j=1}^m \sum_{t=1}^p \sum_{u=t}^p \bar{c}_{ijtu} x_{ijtu} + \sum_{i=1}^n \sum_{j=1}^m \sum_{t=1}^p s_{ijt} y_{ijt} \quad (3.16)$$

sujeito a

$$\sum_{j=1}^m \sum_{t=1}^u x_{ijtu} = d_{iu} \quad \forall i \in A, u \in C \quad (3.17)$$

$$\sum_{i=1}^n (f_{ij} y_{ijt} + \sum_{u=t}^p b_{ij} x_{ijtu}) \leq C_{jt} \quad \forall j \in B, t \in C \quad (3.18)$$

$$x_{ijtu} \leq \min \left\{ d_{iu}; \left\lfloor \frac{C_{jt} - f_{ij}}{b_{ij}} \right\rfloor \right\} y_{ijt} \quad \forall i \in A, j \in B, u \in C, t \in C; (u > t) \quad (3.19)$$

$$x_{ijtu} \geq 0, y_{ijt} \in \{0, 1\} \quad \forall i \in A, j \in B, t \in C, u \in C; (u > t) \quad (3.20)$$

A função objetivo expressa por (3.16) representa a minimização dos custos de produção, *setup* e estoque. As restrições (3.17) impõem o atendimento da demanda de cada item. A garantia de que as capacidades das máquinas não serão violadas é dada pelo conjunto de restrições (3.18).

As restrições (3.19) garantem a ativação de *setup* ($y_{ijt} = 1$), levando a capacidade C_{jt} em consideração, quando a máquina j é utilizada para produzir o item i no período t . A quantidade de produção e estoque não negativos são garantidos pelas restrições (3.20).

3.4 Experimentos Computacionais

Os três modelos apresentados foram implementados utilizando o CPLEX 12.6.3 (ILOG, 2015) e testados utilizando instâncias propostas em (TOLEDO, 1998). Todos os testes foram realizados em um microcomputador sistema operacional Ubuntu 16.04. Utilizaram-se 10 instâncias para cada configuração (p, m, n) de número de períodos ($p = 6, 12, 18$) e de (máquinas, itens), $(m, n) = (2, 6), (4, 12)$ e $(6, 50)$. Essas instâncias estão divididas em dois grupos:

- Capacidade apertada (A), custo de preparação alto (A) e tempo de preparação alto (A) - classe AAA;
- Capacidade normal (N), custo de preparação baixo (B) e tempo de preparação baixo (B) - classe NBB.

Os parâmetros foram gerados em intervalos $[a, b]$ com distribuição uniforme e denotado $U[a, b]$:

- custo de produção (c_{ijt}) $\sim U[1,5;2,5]$
- custo de *setup* (s_{ij}) $\sim U[5;95]$

- custo de estoque (h_{ci}) $\sim U[0;2;0,4]$
- tempo de produção (b_{ij}) $\sim U[1;5]$
- tempo de preparação (f_{ij}) $\sim U[10;50]$
- demanda (d_{it}) $\sim U[0;180]$

Com o intuito de gerar instâncias com altos custos de *setup*, os custos de *setup* seguindo a distribuição normal com os parâmetros acima foram multiplicados por 10. Da mesma forma, para criar exemplos com tempos de preparação altos multiplicaram-se por 1,5 os limites do intervalo de tempo de preparação dados.

A capacidade em cada máquina e período foi obtida através da divisão da demanda de cada item e cada período entre as máquinas. A média foi calculada sobre o número de máquinas e períodos, o que resulta na seguinte expressão:

$$Cap = \frac{\sum_{t=1}^p \sum_{j=1}^m \sum_{i=1}^n \left(\frac{d_{it}}{m} b_{ij} + f_{ij} \right)}{mp}$$

Para ajustar Cap de acordo com o número de máquinas conforme a quantidade de máquinas consideradas, utilizou-se a seguinte expressão:

$$C_{jt} = \left(1, 4 - \frac{m}{10} \right) Cap$$

O valor da capacidade das instâncias com capacidade apertada é obtido multiplicando Cap por 0,9. O valor da capacidade das instâncias com capacidade normal é obtido multiplicando-se Cap por 1.

A Tabela 3.1 mostra a quantidade de instâncias das classes AAA e NBB, para cada configuração de períodos, máquinas e itens (p, m, n) , para as quais o CPLEX provou a otimalidade em tempo igual ou inferior ao tempo máximo imposto de 1800 segundos. Nesta tabela, é possível avaliar a dificuldade em se obter soluções para os problemas dependendo de suas características.

Observa-se que o CPLEX encontrou soluções ótimas para todas as instâncias com 6 períodos considerando 6 e 12 produtos. Para instâncias com 12 períodos uma quantidade razoável de soluções ótimas não foi encontrada. Poucas soluções foram encontradas para instâncias de 18 períodos quando se considerou mais do que 6 itens.

Verifica-se ainda que para os problemas com capacidade normal, com custos e tempos de preparação baixos (NBB), o CPLEX encontrou soluções ótimas, em até 1800 segundos, para todas as 10 instâncias de diferentes configurações (p, m, n) em 55% para o Modelo

TABELA 3.1 – Quantidade de soluções ótimas encontradas pelo CPLEX, no tempo limite de 1800 segundos.

			Modelo 1		Modelo 2		Modelo 3	
p	m	n	AAA	NBB	AAA	NBB	AAA	NBB
6	2	6	10	10	10	10	10	10
	4	12	10	10	10	10	10	10
	6	50	5	6	3	7	2	7
12	2	6	5	10	7	10	6	10
	4	12	5	10	6	10	2	10
	6	50	0	3	2	3	1	2
18	2	6	6	10	7	10	7	8
	4	12	0	3	0	2	1	2
	6	50	0	2	0	1	0	1
Total			41	64	45	63	39	60

1 e para o Modelo 2 e em 44% para o Modelo 3. Já para instâncias com capacidade apertada, custos e tempos de preparação alto (AAA), a porcentagem de configurações em que o *solver* provou a otimalidade para todas as instâncias cai para 22% nos três modelos.

Outra forma utilizada para comparar os resultados foi por meio do gráfico de perfil de desempenho de Dolan e Moré (2002). Para isso, considera-se um conjunto I de n_{ip} instâncias e um conjunto F de n_{sf} algoritmos de solução. A comparação do desempenho de um determinado algoritmo $f \in F$ na resolução de um problema $g \in I$ em relação ao desempenho do melhor algoritmo, é dada pela razão de desempenho:

$$r_{gf} = \frac{t_{gf}}{\min\{t_{gf} : f \in F\}} \quad (3.21)$$

Na Equação (3.21), t_{gf} é o tempo computacional gasto para a resolução de g utilizando-se f . O menor t_{gf} define a melhor performance obtida pelo algoritmo. Assim, de acordo com a Equação (3.21) o melhor valor para o r_{gf} é 1.

A Figura 3.2 ilustra os perfis de desempenho na resolução de problemas através das três formulações estudadas. Para a criação deste gráfico, utilizaram-se as soluções obtidas pelo *solver* CPLEX para o Modelo 1, Modelo 2 e Modelo 3 nas 180 instâncias consideradas. O eixo das ordenadas é dado pelo número de problemas que a execução do CPLEX de cada modelo atingiu uma razão de desempenho igual, ou melhor, a um coeficiente τ (eixo das abcissas).

Assim, o gráfico indica a probabilidade de s ter uma razão de desempenho dentro de um fator τ . Em linguagem matemática : $P(\log_2(r_{ps}) \leq \tau < s < n_s)$.

Pela Figura 3.2 pode-se concluir que o CPLEX utilizando o Modelo 2 conseguiu encontrar melhores soluções na maior proporção de instâncias dentro de um dado fator τ do

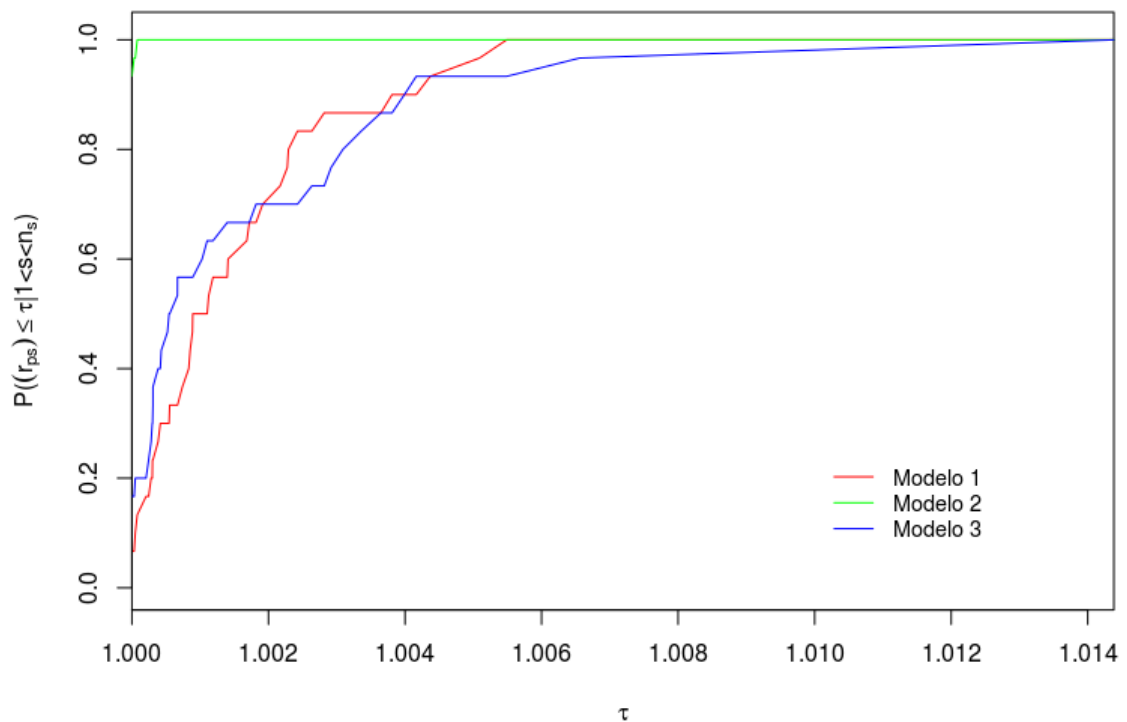


FIGURA 3.2 – Perfil de Desempenho das soluções das instâncias considerando os Modelos 1, 2 e 3.

desempenho do CPLEX utilizando o Modelo 1 e o Modelo 3. Enquanto que os Modelos 1 e 3 mostraram-se competitivos, já que para alguns valores de τ o Modelo 1 apresentou melhor desempenho, enquanto para outros valores de τ o Modelo 3 demonstrou maior eficiência.

3.5 Considerações Finais

Neste capítulo foram estudadas três diferentes modelagens da literatura para o PDLMP, cada uma delas foi implementada no pacote de otimização CPLEX 12.3.6 (ILOG, 2015). Os resultados obtidos foram comparados e o Modelo 2 apresentou melhor desempenho, portanto foi incluído nesse modelo o *setup carry-over*. Mas, a utilização do Modelo 3 em métodos do tipo *relax-and-fix* apresentam bons resultados computacionais para instâncias de médio porte (SILVA, 2013) e portanto, como a math-heurística desenvolvida neste trabalho baseia-se na heurística *relax-and-fix*, esse modelo também foi utilizado para o desenvolvimento do método. No próximo capítulo são apresentadas essas duas modelagens com adição do *setup carry-over*, bem como a math-heurística desenvolvida para solucionar o PDLMPSC.

4 Math-heurística Proposta

Este capítulo discorre sobre os métodos de solução desenvolvidos para a solução do PDL em máquinas paralelas com *setup carry-over*.

4.1 Modelos Matemáticos

Conforme mencionado no Capítulo 3, dois modelos foram escolhidos para o acréscimo do *setup carry-over*, modelo baseado em caminhos mínimos por apresentar melhores resultados nos testes realizados no capítulo anterior e o modelo baseado em localização de facilidades apresentado por Silva (2013), por ser um modelo no qual métodos do tipo *relax-and-fix* apresentam bons resultados computacionais para instâncias de médio porte (SILVA, 2013). Supondo que o problema trata de n itens, p períodos e m máquinas. Sejam os conjuntos $A = \{1, \dots, n\}$; $B = \{1, \dots, m\}$ e $C = \{1, \dots, p\}$. Então, os parâmetros são:

s_{ijt} : custo de preparação para produção do item i na máquina j no período t , $\forall i \in A, j \in B, t \in C$

c_{ijt} : custo de produção do item i na máquina j no período t , $\forall i \in A, j \in B, t \in C$

h_{it} : custo unitário de estoque do item i no período t , $\forall i \in A, t \in C$

d_{it} : demanda do item i no período t , $\forall i \in A, t \in C$

f_{ij} : tempo de preparação para produção do item i na máquina j , $\forall i \in A, j \in B$

b_{ij} : tempo de produção do item i na máquina j , $\forall i \in A, j \in B$

C_{jt} : capacidade da máquina j no período t , $\forall j \in B, t \in C$.

4.1.1 Modelo baseado em Caminhos Mínimos com *setup Carry-over*

Variáveis:

zv_{ijtl} : a fração do plano de produção do item i na máquina j , em que a produção no período t satisfaz, total ou parcialmente, a demanda do período t até o período l ,

w_{it} : fração do plano de estoque inicial do item i em que a demanda é atendida para os primeiros t períodos,

sd_{itl} : soma das demandas do item i do período t até o período l ,

fc_i : custo de uma unidade do estoque inicial para o item i ,

cv_{ijtl} : custo total de produção e estoque para produzir o item i , na máquina j no período t atendendo a demanda dos períodos t até l , dado pela Equação (3.7).

ci_{it} : custo total para que o estoque inicial do item i atenda a demanda dos períodos 1 até o período t , dado pela Equação (3.8),

y_{ijt} : variável binária que assume valor 1 se o item i é produzido na máquina j no período t , e 0 caso contrário,

z_{ijt} : variável binária que assume valor 1 e o estado de *setup* do produto i na máquina j é transportado do período t para $t + 1$, se 0 caso contrário, sendo $t < p$.

$$\min \sum_{t=1}^p \sum_{j=1}^m \sum_{i=1}^n s_{ijt} y_{ijt} + \sum_{i=1}^n \sum_{j=1}^m \sum_{t=1}^p \sum_{l=t}^p cv_{ijtl} zv_{ijtl} + \sum_{i=1}^n \sum_{t=1}^p ci_{it} w_{it} \quad (4.1)$$

sujeito a

(3.10) - (3.12)

$$\sum_{l=t}^p zv_{ijtl} \leq N(y_{ijt} + z_{ijt}) \quad \forall i \in A, j \in B, t \in C \quad (4.2)$$

$$\sum_{i=1}^n z_{ijt} = 1 \quad \forall j \in B, t \in C \quad (4.3)$$

$$z_{ijt} - y_{ijt} - z_{i,j,t-1} \leq 0 \quad \forall i \in A, j \in B, t \in C; (t < p) \quad (4.4)$$

$$z_{ijt} + z_{i,j,t-1} - y_{ijt} + y_{kjt} \leq 2 \quad \forall i \in A, k \in A, j \in B, t \in C; (i \neq k), (t < p) \quad (4.5)$$

$$zv_{ijtl} \geq 0; y_{ijt}, z_{ijt} \in \{0, 1\}; w_{it} \geq 0 \quad \forall i \in A, j \in B, t \in C, l \in C; (l > t) \quad (4.6)$$

Sendo $N \geq \sum_{i=1}^n \sum_{t=1}^p d_{it}$.

A função objetivo e os conjuntos de restrições (3.10) até (3.12) são os mesmos apresentados na Seção 3.2 do capítulo anterior. As restrições (4.2) foram modificadas para que o *setup* (y_{ijt}) seja ativado quando houver produção e ele não for aproveitado do período

anterior e para que, caso ocorra produção e o *setup* for aproveitado, a variável de *setup carry over* (z_{ijt}) seja ativada.

As restrições (4.3) até (4.5) garantem a interpretação correta do *setup carry-over*: as condições (4.3) implicam que cada máquina só pode transportar o *setup* para apenas um produto, as restrições (4.4) indicam que uma máquina precisa ter o estado de *setup* correto para transportá-lo para o próximo período. Ou seja, se uma máquina j transporta um *setup* para o produto i do período t para o $t+1$ ($z_{ijt} = 1$) a máquina deve ter feito um *setup* para este produto no período t ($y_{ijt} = 1$) ou o *setup* foi transferido do período anterior ($z_{i,j,t-1} = 1$). Isso implica uma conservação do estado de instalação durante tempos inativos e períodos de inatividade. As restrições (4.5) lidam com a seguinte situação: se a máquina j fizer *setup carry-over* para o item i do período $t-1$ até t e também de t para $t+1$ ($z_{i,j,t-1} = z_{ijt} = 1$) e, no período t , um *setup* é realizado para outro produto k ($y_{kjt} = 1$), então deve-se redefinir a máquina para i no período t ($y_{ijt} = 1$).

4.1.2 Modelo baseado em Localização de Facilidades com *setup Carry-over*

x_{ijtu} : quantidade do item i produzida na máquina j no período t para atender o período u , $u \geq t$,

y_{ijt} : variável binária que assume valor 1 se o item i é produzido na máquina j no período t , e 0 caso contrário,

z_{ijt} : variável binária que assume valor 1 e o estado de *setup* do produto i na máquina j é transportado do período t para $t+1$, e 0 caso contrário, sendo $t < p$.

$$\min \sum_{i=1}^n \sum_{j=1}^m \sum_{t=1}^p \sum_{u=t}^p \bar{c}_{ijtu} x_{ijtu} + \sum_{i=1}^n \sum_{j=1}^m \sum_{t=1}^p s_{ijt} y_{ijt} \quad (4.7)$$

sujeito a

(3.17) - (3.18)

$$x_{ijtu} \leq \min \left\{ d_{iu}; \left\lfloor \frac{C_{jt} - f_{ij}}{b_{ij}} \right\rfloor \right\} (y_{ijt} + z_{ijt}) \quad \forall i \in A, j \in B, t \in C, u \in C \quad (4.8)$$

$$\sum_{i=1}^n z_{ijt} = 1 \quad \forall j \in B, t \in C \quad (4.9)$$

$$z_{ijt} - y_{ijt} - z_{i,j,t-1} \leq 0 \quad \forall i \in A, j \in B, t \in C; (t < p) \quad (4.10)$$

$$z_{ijt} + z_{i,j,t-1} - y_{ijt} + y_{kjt} \leq 2 \quad \forall i \in A, k \in A, j \in B, t \in C; (i \neq k), (t < p) \quad (4.11)$$

$$x_{ijtu} \geq 0, y_{ijt} \in \{0, 1\}, z_{ijt} \in \{0, 1\} \quad \forall i \in A, j \in B, t \in C, u \in C; (u > t) \quad (4.12)$$

Sendo,

$$\bar{c}_{ijtu} = \begin{cases} 0, & \text{se } u < t \\ c_{ijt} + \sum_{v=t}^u h_{iv}, & \text{caso contrário.} \end{cases}$$

A função objetivo e os conjuntos de restrições (3.17) e (3.18) são os mesmos apresentados na Seção 3.3 do capítulo anterior.

As restrições (4.8) garantem a ativação do *setup* ou do *setup carry-over* quando existe produção.

As restrições (4.9) até (4.11) garantem a interpretação correta do *setup carry-over*: as condições (4.9) implicam que cada máquina só pode transportar o *setup* para apenas um produto, as restrições (4.10) indicam que uma máquina precisa ter o estado de *setup* correto para transportá-lo para o próximo período. Ou seja, se uma máquina j transporta um *setup* para o produto i do período t para o $t + 1$ ($z_{ijt} = 1$) a máquina deve ter feito um *setup* para este produto no período t ($y_{ijt} = 1$) ou o *setup* foi transferido do período anterior ($z_{i,j,t-1} = 1$). Isso implica uma conservação do estado de instalação durante tempos inativos e períodos de inatividade. As restrições (4.11) lidam com a seguinte situação: se a máquina j fizer *setup carry-over* para o item i do período $t - 1$ até t e também de t para $t + 1$ ($z_{i,j,t-1} = z_{ijt} = 1$) e, no período t , um *setup* é realizado para outro produto k ($y_{kjt} = 1$), então deve-se redefinir a máquina para i no período t ($y_{ijt} = 1$).

As quantidades de produção e estoque não negativas são garantidas pelas restrições (4.12).

4.1.3 Math - heurísticas

As math-heurísticas são métodos heurísticos desenvolvidos por meio da combinação de métodos heurísticos e técnicas de programação matemática. Ultimamente, essa combinação tem sido utilizada na resolução de algumas variações do PDLs, demonstrando eficiência e obtendo bons resultados. Além de Caserta *et al.* (2010) mencionada na seção 2.3, há outras referências que apresentam soluções eficientes do PDL por meio de math-heurísticas.

Caserta e Voß (2013) resolveram o PDL capacitado com múltiplos itens independentes e considerando *setups* por meio de uma math-heurística baseada na decomposição de Dantzig-Wolfe. Segundo os autores, esse artigo foi o pioneiro em empregar essa técnica dentro de um quadro meta-heurístico. Os resultados apresentados permitem concluir que a abordagem proposta é promissora, tanto em termos de qualidade da solução quanto de tempo de execução, e deixa várias opções de estudo em outros tipos de problemas.

Mais recentemente, Carvalho e Nascimento (2016) consideraram PDL em múltiplas plantas. As autoras utilizaram a heurística Lagrangiana com uma estratégia baseada na meta-heurística *path-relinking* para encontrar limitantes e soluções factíveis de boa qualidade para o problema tratado. A relaxação Lagrangiana foi aplicada nas restrições de capacidade do problema e o método do subgradiente foi utilizado para determinar iterativamente valores adequados aos multiplicadores de Lagrange. Os resultados de experimentos com instâncias da literatura comprovaram a eficiência do método proposto. Silva (2013) também tratou do o PDL em múltiplas plantas e utilizou uma math-heurística que combina a abordagem *relax-and-fix* com a restrição *local branching*. Os testes computacionais mostraram que o método proposto apresenta boas soluções factíveis. O math-heurística apresentada no presente trabalho foi inspirada no método desenvolvido em (SILVA, 2013). O método proposto nesta dissertação, particiona o conjunto de variáveis inteiras do PDLMPSC pelo número de períodos (cada conjunto de variáveis possui períodos distintos), e busca uma solução factível através da heurística *relax-and-fix* adicionando a restrição *local branching*, em que as partições consideradas em cada iteração da heurística *relax-and-fix* são fixadas aleatoriamente. Nesse sentido, as próximas seções focam em revisões de trabalhos relacionados à desenvolvida nesta Dissertação.

4.2 Relax-and-fix

A heurística *relax-and-fix* (WOLSEY, 1998) utiliza uma abordagem de solução baseada em métodos exatos, em que o conjunto contendo todas as variáveis inteiras de um problema de otimização inteira mista é particionado em W conjuntos disjuntos P_s , $s = 1, \dots, W$.

O número W de conjuntos representa o número de iterações da heurística. Em uma determinada iteração it , apenas as variáveis do conjunto P_{it} são definidas como inteiras, as demais variáveis inteiras são relaxadas e/ou fixadas e o problema resultante é então resolvido. Caso esse problema seja inviável, ou seja, não é possível encontrar uma solução factível com as variáveis dos conjuntos P_s , $s = 1, \dots, (it - 1)$ fixas nos valores atuais, o método é interrompido. Se o problema for factível, as variáveis do conjunto P_{it} são fixadas em seus valores correntes e o processo se repete incrementando-se o valor da iteração.

A cada iteração, portanto, as variáveis inteiras são divididas em três grupos:

1. Variáveis inteiras fixas (F),
2. Variáveis inteiras não fixas e relaxadas linearmente (R),
3. Variáveis inteiras não fixas e não relaxadas (I).

Período	1	2	3	4	5	6	7
Partição	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇
Iteração 1	I	R	R	R	R	R	R
Iteração 2	F	I	R	R	R	R	R
Iteração 3	F	F	I	R	R	R	R
Iteração 4	F	F	F	I	R	R	R
Iteração 5	F	F	F	F	I	R	R
Iteração 6	F	F	F	F	F	I	R
Iteração 7	F	F	F	F	F	F	I
Solução	F	F	F	F	F	F	F

FIGURA 4.1 – Exemplo de iterações do *relax-and-fix*.

Na Figura 4.1, os retângulos azuis com a letra **R** representam as variáveis relaxadas linearmente no problema da iteração indicada, os retângulos rosas com a letra **I** representam as variáveis que serão consideradas inteiras no subproblema da iteração correspondente, esta figura exemplifica a utilização da heurística *relax-and-fix* para o tratamento do PDL. Neste exemplo, as variáveis inteiras são particionadas de acordo com os períodos relacionados à elas, o PDL considera sete períodos e cada retângulo representa um conjunto de partições P_k , $k = 1, \dots, 7$. Na iteração k , as variáveis do conjunto P_k são definidas como inteiras (I), caso $k > 1$ então as variáveis dos conjuntos anteriores (P_1 a P_{k-1}) são fixadas (F) e caso $k < 7$ as variáveis dos conjuntos P_{k+1} a P_7 são relaxadas (R). Neste exemplo foi possível obter uma solução factível e, por isso, o método foi executado em todas as 7 iterações.

A abordagem de solução *relax-and-fix* tem sido amplamente utilizada de forma pura ou híbrida na resolução de diversas classes de problemas, nos quais as variáveis inteiras são geralmente particionadas por meio dos períodos, isto é, variáveis referentes ao mesmo

período pertencem à mesma partição. No tratamento de PDLs, além do trabalho de (BERALDI *et al.*, 2008) mencionado na Seção 2.2, existem outras referências nas quais diferentes variações dessa abordagem são aplicadas.

Ferreira *et al.* (2009) desenvolveram um modelo de programação de número inteiro misto que integra decisões de dimensionamento de lotes de produção e agendamento de plantas de bebidas com custos e horários de instalação dependentes de sequência, este modelo considera que o processo industrial produz garrafas de refrigerante em diferentes sabores e tamanhos, e é realizado em duas fases de produção: preparação líquida (estágio I) e engarrafamento (estágio II). A heurística *relax-and-fix* foi utilizada para resolver o problema usando o modelo proposto pelos autores e os testes foram realizados com base em dados reais de uma empresa brasileira de refrigerantes de pequena escala. Os resultados mostram que as soluções são capazes de produzir melhores soluções que as usadas pela empresa.

Mohammadi *et al.* (2010) apresentaram um modelo para o PDL capacitado, multi-nível e *setup* dependente de sequência. Dado que a resolução exata do problema não foi obtida em um tempo computacional aceitável para vinte instâncias de diferentes tamanhos (considerando quantidade de produtos (n), máquinas (m) e períodos (p)) nas quais (n, m, p) foram selecionados randomicamente entre $\{3, 4, 5, \dots, 15\}$, os autores desenvolveram heurísticas *relax-and-fix* para resolvê-lo e observaram em seus testes computacionais que o *gap* da solução heurística foi, no máximo, 6%.

Toledo *et al.* (2015) propuseram uma heurística que combina construção e melhoria de ideias heurísticas para duas classes de PDLs multi-nível: o PDL capacitado com *back-orders* e o PDL de recipientes de vidro em dois estágios. A heurística *relax-and-fix* foi utilizada para obtenção de uma solução inicial que foi melhorada por outra matheurística, a *fix-and-optimize*. Os resultados computacionais mostraram que essa abordagem heurística combinada é muito eficiente e competitiva, superando os métodos de referência para a maioria das instâncias testadas.

4.3 Restrição de Local Branching

A restrição de *local branching* (FISCHETTI; LODI, 2003) baseia-se na busca local associada à ideia de separação (*branching*) do método *branch & bound*. A partir de uma solução inicial, não necessariamente ótima, de um problema de otimização inteira ou mista (PIM), esta restrição busca outra solução factível de melhor qualidade na vizinhança desta solução inicial. A vizinhança é dada por todas as soluções vizinhas que contenham um máximo imposto de variáveis inteiras com valores distintos aos valores dados pela solução inicial.

Considerando uma solução inicial s_i e um PIM com variáveis binárias, sejam:

- Y^0 : conjunto contendo todas variáveis binárias nulas em s_i ,
- Y^1 : conjunto contendo todas variáveis binárias unitárias em s_i ,
- C_{LB} : limite das variáveis binárias que podem ter seu valor alterado.

No PDLMP as variáveis binárias que indicam a ativação do *setup* são responsáveis por torná-lo um PIM. Neste problema, a restrição de *local branching* é dada por:

$$\sum_{y_{ijt} \in Y^0} y_{ijt} + \sum_{y_{ijt} \in Y^1} (1 - y_{ijt}) \leq C_{LB} \quad (4.13)$$

O primeiro somatório contabiliza as variáveis binárias que terão seus valores alterados de um para zero, enquanto o segundo contabiliza as variáveis binárias que terão seus valores alterados de zero para um. Assim, o valor de C_{LB} indica a quantidade de variáveis binárias que podem ter seu valor alterado. E, quanto maior o valor de C_{LB} maior a vizinhança considerada.

Assim como Silva (2013), Paiva e Toledo (2010) utilizam restrições *local branching* na busca de soluções factíveis para o PDL capacitado, o autor propôs um método *local branching* para solução do PDLC com três características adicionais: *backlogging*; *setup carry-over* e *setup crossover* (continuidade de preparação de um período para o seguinte, o mesmo que *setup splitting*). Os resultados obtidos mostraram que as soluções encontradas pelo método *local branching* proposto são melhores em qualidade do que as soluções obtidas pelo *software* comercial CPLEX, versão 11.0, limitado em 250 segundos.

4.4 Método híbrido

Silva (2013) utilizou a heurística *relax-and-fix* na busca de soluções para o PDL em múltiplas plantas, o autor particionou o conjunto de variáveis binárias por períodos e observou que a utilização dessa heurística de forma pura possui grandes chances de resultar em soluções infactíveis, pois decisões equivocadas no início do horizonte de planejamento podem inviabilizar o atendimento da demanda nos períodos finais. Foi observado que possibilitando a alteração de valores de uma parte de variáveis inteiras das iterações anteriores o método fica mais flexível, facilitando a obtenção de soluções factíveis. Porém, tal permissão resulta em subproblemas mais difíceis de resolver. Silva (2013) propõe a flexibilização das variáveis de um partição adicionando a restrição (4.13) em cada iteração

do método *relax-and-fix*, conforme mostra a Heurística 1.

Heurística 1: Algoritmo proposto por (SILVA, 2013)

Passo 1: Particione as variáveis binárias do problema por período.

Passo 2: Relaxe todas as variáveis binárias linearmente.

Passo 3: Inicialize os conjuntos Y^0 and Y^1 como vazios e $k = 1$.

Passo 4: Selecione o conjunto P_k de variáveis relaxadas e considere-as inteiras.

Passo 5: Se $k > 1$, então adicione a restrição (4.13) ao problema e resolva-o.

Passo 6: Se nenhuma solução viável for encontrada, retorne erro e pare a execução.

Passo 7: Atualize os conjuntos Y^0 e Y^1 com a solução obtida na iteração k , e incremente em 1 o valor da iteração k .

Passo 8: Se k for menor que o número de períodos do problema, que é o número de conjuntos da partição, retorne ao Passo 4. Senão, retorne a solução corrente.

O autor constatou que quando ocorrem mudanças nas variáveis inteiras em uma determinada iteração do método proposto, elas ocorrem geralmente nos períodos imediatamente anteriores ao período tratado nessa iteração. Por isso, ele propôs uma alteração na Heurística 1 para diminuir o tempo de execução de cada subproblema.

Silva (2013) observou empiricamente que as mudanças das variáveis inteiras (permitidas pela restrição *local branching*) em cada iteração geralmente ocorreram nas partições tratadas em iterações imediatamente anteriores à iteração atual. Quando se fixam as partições mais distantes (tratadas nas primeiras iterações) o espaço de busca dos últimos subproblemas é reduzido. A Figura 4.2 ilustra esquematicamente a ideia desta heurística modificada.

Período	1	2	3	4	5	6	7
Partição	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇
Iteração 1	I	R	R	R	R	R	R
Iteração 2	PF	I	R	R	R	R	R
Iteração 3	PF	PF	I	R	R	R	R
Iteração 4	PF	PF	PF	I	R	R	R
Iteração 5	F	PF	PF	PF	I	R	R
Iteração 6	F	F	PF	PF	PF	I	R
Iteração 7	F	F	F	PF	PF	PF	I
Solução	F	F	F	F	F	F	F

FIGURA 4.2 – Exemplo: heurística modificada

Na Figura 4.2, os retângulos azuis com a letra **R** representam as variáveis relaxadas linearmente no problema da iteração indicada, os retângulos rosas com a letra **I** representam as variáveis que serão consideradas inteiras no subproblema da iteração correspondente, os retângulos verdes contendo a letra **PF** representam as variáveis inteiras que podem

ter seus valores alterados no problema da iteração indicada devido as restrições de *local branching* e por fim, os retângulos contendo a letra **F** representam as variáveis fixadas com valores obtidos em soluções de problemas definidos em iterações anteriores. Portanto, as variáveis que podem sofrer alterações são aquelas dos conjuntos da partição que estão indicadas pelas letras **I** e **PF**.

Nesta variante proposta as variáveis inteiras são permanentemente fixadas em um limite de iterações após terem sido tratadas como inteiras.

Heurística 2: Algoritmo Modificado proposto por (SILVA, 2013)

Passo 1: Particione as variáveis binárias do problema por período.

Passo 2: Relaxe todas as variáveis binárias linearmente.

Passo 3: Inicialize os conjuntos Y^0 and Y^1 como vazios e $k = 1$.

Passo 4: Selecione o conjunto P_k de variáveis relaxadas e considere-as inteiras.

Passo 5: Se $k > 1$, então adicione a restrição (4.13) ao problema e resolva-o.

Passo 6: Se nenhuma solução viável for encontrada, retorne erro e pare a execução.

Passo 7: Atualize os conjuntos Y^0 e Y^1 com a solução obtida na iteração k , e atualize $k = k + 1$.

Passo 8: Se $k > \beta$, fixe permanentemente as variáveis do conjunto $P_{k-\beta}$.

Passo 9: Se k for menor do que o número de períodos, que corresponde ao número de conjuntos da partição, retorne ao Passo 4. Senão, retorne a solução corrente.

A Figura 4.2 ilustra uma situação em que $\beta = 3$. A Heurística 2 apresenta o pseudo-código desta variação do método.

Silva (2013) utilizou as math-heurísticas dadas pelas Heurísticas 1 e 2 para resolver o PDL com múltiplas plantas e concluiu que esses métodos conseguem obter soluções factíveis de boa qualidade para as instâncias utilizadas. Oliveira *et al.* (2014) utilizaram uma estratégia similar, que combina a heurística *relax-and-fix* com restrições *local branching*, para tratar o problema de reposições vazias nas empresas de aluguel de carros. Para satisfazer os requisitos de reservas, essas empresas são obrigadas a transferir carros entre estações de aluguel, suportando custos significativos. Nos testes executados foram utilizados exemplos reais e o método obteve uma melhoria de 33% quando comparado às práticas atuais da empresa.

4.5 Math-heurística Proposta

A partir da math-heurística proposta por (SILVA, 2013), para esta Dissertação, desenvolveu-se uma math-heurística que também combina a abordagem *relax-and-fix* com a restrição *local branching*. Porém, aqui são propostas algumas alterações que buscam

diversificar o espaço de busca a cada passo da math-heurística.

Nesta math-heurística, as variáveis binárias também são particionadas por período, no entanto, a ordem de fixação das partições é aleatória. Primeiramente é gerado um número aleatório α tal que, $1 \leq \alpha \leq p$. A partir daí, são criadas $\frac{p}{\alpha}$ famílias de conjuntos contendo α conjuntos de partições cada uma. Caso p não seja divisível por α , a última família terá uma quantidade de conjuntos menor do que α .

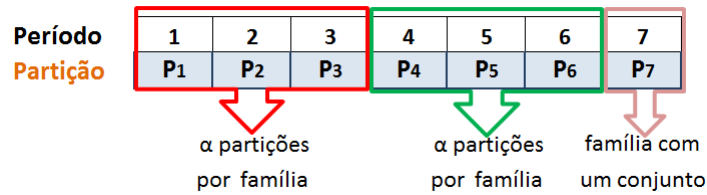


FIGURA 4.3 – Exemplo ilustrando proposta de conjuntos da partição em famílias

A Figura 4.3 ilustra a ideia da divisão de conjuntos de partições quando $\alpha = 3$ e $p = 7$. Essa divisão resulta em 3 famílias de conjuntos, sendo duas delas com α conjuntos e uma com um conjunto, pois $7 \bmod 3$ é 1.

O método proposto trabalha com blocos de iterações, relacionados às famílias de conjuntos, que delimitam sistematicamente os subproblemas a serem resolvidos. O número do bloco da iteração corresponde ao índice da família da partição. O domínio das variáveis pertencentes às famílias não investigadas será sempre os reais. Variáveis já investigadas em blocos anteriores ao corrente terão valores fixos de acordo com soluções encontradas anteriormente, quando as variáveis foram investigadas. Variáveis sendo investigadas, ou seja, do bloco de iterações corrente, poderão ser: relaxadas, inteiras ou pré-fixadas. Na primeira iteração, é gerado um número aleatório N_1 , tal que $1 \leq N_1 \leq \alpha$, e então as variáveis contidas no conjunto de partição referente ao período N_1 , ou seja, P_{N_1} , são definidas inteiras no subproblema. Toda vez que um número é sorteado, portanto, o conjunto de variáveis ao qual aquele número sorteado se relaciona terá seu domínio definido como inteiro. Como essa é a primeira iteração, todas as demais são relaxadas. Tal subproblema é então resolvido. Na segunda iteração (caso $\alpha \neq 1$), é gerado um número aleatório N_2 , tal que $1 \leq N_2 \leq \alpha$ e $N_2 \neq N_1$, e as variáveis contidas no conjunto P_{N_2} são tratadas como inteiras para o próximo subproblema a ser resolvido enquanto as variáveis do conjunto P_{N_1} são parcialmente fixadas com valores obtidos na solução da iteração anterior. Assim como em (SILVA, 2013), a diferença entre conjuntos parcialmente fixos e fixos é que nos conjuntos parcialmente fixos é permitido a alteração de valores, no limite da restrição de *local branching*, das variáveis inteiras, enquanto que nos conjuntos fixos essa alteração não é permitida, impondo que as variáveis inteiras sejam fixadas nos valores obtidos em iterações anteriores. Todas as demais variáveis serão relaxadas linearmente. O processo se repete até que o número da iterações seja igual a α , indicando que todos os conjuntos

da família foram investigados.

Acabado o primeiro bloco de iterações, o segundo se inicia. Nesse sentido, na iteração $\alpha + 1$ (caso $\alpha \neq p$) é gerado um número aleatório $N_{\alpha+1}$, tal que $\alpha + 1 \leq N_{\alpha+1} \leq 2\alpha$, se $2\alpha < p$ ou $\alpha + 1 \leq N_{\alpha+1} \leq p$ se $2\alpha > p$, e as variáveis contidas na partição referente ao período $N_{\alpha+1}$ são tratadas como inteiras. Durante todo esse bloco de iterações, as variáveis dos conjuntos avaliados no primeiro bloco são fixadas com valores encontrados em tal etapa.

O processo se repete até a geração do número N_p e o tratamento do conjunto de variáveis referente ao período N_p .

Um exemplo ilustrativo das iterações do método proposto é apresentado na Figura 4.4.

Período	1	2	3	4	5	6	7
Partição	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇
Iteração 1	I	R	R	R	R	R	R
Iteração 2	PF	R	I	R	R	R	R
Iteração 3	PF	I	PF	R	R	R	R
Iteração 4	F	F	F	R	I	R	R
Iteração 5	F	F	F	I	PF	R	R
Iteração 6	F	F	F	PF	PF	I	R
Iteração 7	F	F	F	F	F	F	I
Solução	F	F	F	F	F	F	F

FIGURA 4.4 – Exemplo: math-heurística proposta

A Figura 4.4 representa uma execução do método com $\alpha = 3$, $p = 7$, $N_1 = 1$, $N_2 = 3$, $N_3 = 2$, $N_4 = 5$, $N_5 = 4$, $N_6 = 6$, $N_7 = 7$. Primeiramente, N_1 foi gerado ($1 \leq N_1 \leq 3$) e o domínio das variáveis do conjunto referente ao período, P_1 foi tratado como inteiro (I), enquanto as demais variáveis foram relaxadas (R); depois foi gerado o número N_2 ($1 \leq N_2 \leq 3$ e $N_2 \neq N_1$) e as variáveis do conjunto referente ao período N_2 foram tratadas como inteiras (I), as variáveis dos conjuntos P_{N_3}, \dots, P_{N_7} foram relaxadas (R) e as variáveis do conjunto P_{N_1} foram parcialmente fixas (PF), podendo alterar uma quantidade C_{LB} (limite da restrição *local branching*) de variáveis. E assim o processo segue até a geração do número N_α (N_3). Então na iteração 4 ($\alpha + 1$) todas variáveis dos primeiros conjuntos são fixadas (F) com valores obtidos na solução obtida na última iteração, e o processo se repete para a segunda família de α conjuntos. Nesse caso, são gerados valores aleatórios para N_4 , N_5 e N_6 e as variáveis são parcialmente fixadas logo após serem tratadas como inteiras dentro desse conjunto. O conjunto de variáveis referente ao período 7 é o último a ser fixado.

Considerando o fator aleatório do método, é possível afirmar que em diversas execuções soluções distintas podem ser obtidas. Por isso, na math-heurística proposta é definido um número M de execuções do método. E então, é retornada a melhor solução obtida nessas

M execuções. A Figura 4.5 ilustra um exemplo com 3 execuções do método. Nesse caso a solução retornada será s , tal que $s = \min\{\text{Solução 1}, \text{Solução 2}, \text{Solução 3}\}$

Execução 1							
Período	1	2	3	4	5	6	7
Iteração 1	I	R	R	R	R	R	R
Iteração 2	PF	R	I	R	R	R	R
Iteração 3	PF	I	PF	R	R	R	R
Iteração 4	F	F	F	R	I	R	R
Iteração 5	F	F	F	I	PF	R	R
Iteração 6	F	F	F	PF	PF	I	R
Iteração 7	F	F	F	F	F	F	I
Solução 1	F	F	F	F	F	F	F

Execução 2							
Período	1	2	3	4	5	6	7
Iteração 1	I	R	R	R	R	R	R
Iteração 2	PF	I	R	R	R	R	R
Iteração 3	PF	PF	R	I	R	R	R
Iteração 4	PF	PF	I	PF	R	R	R
Iteração 5	F	F	F	F	R	R	I
Iteração 6	F	F	F	F	I	R	PF
Iteração 7	F	F	F	F	PF	I	PF
Solução 2	F	F	F	F	F	F	F

Execução 3							
Período	1	2	3	4	5	6	7
Iteração 1	R	I	R	R	R	R	R
Iteração 2	I	PF	R	R	R	R	R
Iteração 3	PF	PF	R	R	I	R	R
Iteração 4	PF	PF	I	R	PF	R	R
Iteração 5	PF	PF	PF	I	PF	R	R
Iteração 6	F	F	F	F	F	R	I
Iteração 7	F	F	F	F	F	I	FL
Solução 3	F	F	F	F	F	F	F

FIGURA 4.5 – Exemplos de execuções do método

No exemplo da Figura 4.5, na execução 1, $\alpha_1 = 3$; na execução 2, $\alpha_2 = 4$; e na execução 3, $\alpha_3 = 5$.

Um pseudocódigo da math-heurística proposta é apresentado na Heurística 3. Considere a restrição (4.13), sendo y_{ijt} definidas na Seção 4.1. Considere também a seguinte restrição, na qual z_{ijt} são definidas na seção 4.1, e Z^0 e Z^1 são os conjuntos de variáveis z_{ijt} que são 0 e 1, respectivamente, na solução da iteração anterior.

$$\sum_{z_{ijt} \in Z^0} z_{ijt} + \sum_{z_{ijt} \in Z^1} (1 - z_{ijt}) \leq C_{LB} \tag{4.14}$$

Heurística 3: Math-heurística proposta

Passo 1: Defina a quantidade de iterações M e $mi = 1$.

Passo 2 Selecione randomicamente um número α , tal que $1 \leq \alpha \leq p$, em que p é o número de períodos. Faça $st = 1$.

Passo 3 Particione as variáveis binárias do problema por período.

Passo 4: Relaxe linearmente todas as variáveis binárias.

Passo 5: Inicialize os conjuntos Y^0 , Y^1 , Z^0 e Z^1 como vazios e $ca = 1$.

Passo 6: Se α for menor do que ca , defina $st = mi$, se $2\alpha \geq p$, defina $\alpha = p$; se $2\alpha \leq p$, defina $\alpha = 2\alpha$.

Passo 7: Selecione aleatoriamente o número N_{ca} , tal que $st \leq N_{ca} \leq \alpha$ e se ca é maior que 1, então faça $N_{ca} \neq N_l$, $l = 1, \dots, ca - 1$.

Passo 8: Selecione o conjunto $P_{N_{ca}}$ de variáveis e considere-as inteiras.

Passo 9: Se $ca > 1$, então adicione as restrições (4.13) e (4.14) ao problema e resolva-o.

Passo 10: Se nenhuma solução viável for encontrada, retorne erro e pare a execução.

Passo 11: Atualize os conjuntos Y^0 , Y^1 , Z^0 e Z^1 com os valores da solução obtida na iteração ca , e faça $ca = ca + 1$.

Passo 12: Se mi for igual a α , fixe permanentemente as variáveis pertencentes aos conjuntos $P_{N_1} \dots P_{N_\alpha}$ com os valores encontrados na solução da última iteração.

Passo 13: Se ca for menor que p , retorne ao Passo 6. Senão, defina a solução corrente como s_i .

Passo 14: Se mi é menor do que M , faça $i = i + 1$ e retorne ao Passo 2. Senão, faça Solução = $\min\{s_1, \dots, s_M\}$ e retorne Solução.

No próximo capítulo serão apresentados os resultados obtidos resolvendo um conjunto de instâncias com essa math-heurística. Conforme será demonstrado, ela obtém bons resultados em um tempo razoavelmente baixo, mas, as soluções obtidas pelo *solver* CPLEX limitado em 1800 segundos foram melhores. Por isso, algumas modificações, apresentadas na próxima seção, foram implementadas.

4.6 Math-heurística Proposta Modificada

A alteração proposta, a fim de tratar as limitações identificadas na math-heurística inicialmente proposta, visa permitir que as variáveis inteiras sejam modificadas em uma quantidade maior que $\alpha - 1$ conjuntos, isto é, a quantidade de conjuntos nos quais as variáveis inteiras são parcialmente fixadas (considerando a restrição de *local branching*) é dada por β , tal que $\beta = \alpha + \gamma$. Além disso, $2 \leq \alpha \leq \frac{p}{2}$.

Portanto, as variáveis dos últimos β conjuntos considerados inteiros são parcialmente fixadas e as variáveis de um dado conjunto são fixadas com o valor da última solução depois de β iterações que as variáveis deste conjunto foram definidas inteiras.

Período	1	2	3	4	5	6	7	8	9
Iteração 1	I	R	R	R	R	R	R	R	R
Iteração 2	PF	R	I	R	R	R	R	R	R
Iteração 3	PF	I	PF	R	R	R	R	R	R
Iteração 4	PF	PF	PF	R	I	R	R	R	R
Iteração 5	PF	PF	PF	I	PF	R	R	R	R
Iteração 6	PF	PF	PF	PF	PF	I	R	R	R
Iteração 7	PF	PF	PF	PF	PF	PF	R	I	R
Iteração 8	F	PF	PF	PF	PF	PF	I	PF	R
Iteração 9	F	PF	F	PF	PF	PF	PF	PF	I
Solução	F	F	F	F	F	F	F	F	F

FIGURA 4.6 – Exemplos de execução do método modificado

A Figura 4.6 representa uma execução deste método sendo $\alpha = 3$, $\gamma = 3$ e, portanto, $\beta = 6$, $p = 9$, $N_1 = 1$, $N_2 = 3$, $N_3 = 2$, $N_4 = 5$, $N_5 = 4$, $N_6 = 6$, $N_7 = 8$, $N_8 = 7$ e $N_9 = 9$. Primeiramente, N_1 foi determinado aleatoriamente ($1 \leq N_1 \leq 3$) e o domínio das variáveis do conjunto referente ao período N_1 foi tratado como inteiro (I), as demais variáveis (R); o número N_2 é escolhido aleatoriamente ($1 \leq N_2 \leq 3$ e $N_2 \neq N_1$) e as variáveis do conjunto P_{N_2} são definidas inteiras (I), enquanto as variáveis dos conjuntos P_{N_3} a P_{N_7} foram relaxadas (R) e as do conjunto P_{N_1} são parcialmente fixadas (PF), ou seja, podem alterar uma quantidade C_{LB} (limite da restrição *local branching*) de variáveis dentre as PF. E assim o processo segue até a geração do número N_α (N_3), ou seja, até a iteração 3. Então na iteração 4, já no segundo bloco de iterações, são escolhidos valores aleatórios para N_4 , N_5 e N_6 e as variáveis são parcialmente fixadas logo após terem sido tratadas como inteiras neste bloco de iterações, e as variáveis anteriormente tratadas como inteiras continuam parcialmente fixas até a iteração 8, quando as variáveis do conjunto referente ao período 1 são totalmente fixadas (F), pois foram executadas $\beta = 6$ iterações desde que as variáveis deste conjunto foram consideradas inteiras.

Além disso, nesta modificação, ao invés de pré-estabelecer um número M de iterações, e selecionar a melhor solução encontrada nas M iterações, o método é executado até que se obtenha uma solução com $GAP \leq 0,05$, ou então, após 1800 segundos desde o início de sua execução. O GAP é calculado da seguinte maneira:

$$GAP = \frac{|SR - SI|}{|SI|}$$

sendo

SI : solução obtida pela math-heurística;

SR : solução relaxada linearmente.

4.7 Considerações Finais

Neste capítulo foram apresentadas duas modelagens utilizadas para tratar o PDLMPSC e a math-heurística desenvolvida para sua resolução. Além disso, com o intuito de melhorar os resultados obtidos, foram propostas modificações na math-heurística. No capítulo seguinte, são definidas as instâncias utilizadas para testes e os resultados computacionais obtidos.

5 Experimentos Computacionais

Este capítulo apresenta os experimentos computacionais com análise dos resultados obtidos para o PDLMPSC. Em um primeiro experimento, atesta-se a superioridade da math-heurística proposta em relação ao CPLEX limitado em 1800 segundos por instância para o problema sem a consideração do *setup carry-over*, ou seja, para o PDLMP.

Em um segundo experimento, ambas as versões da math-heurística desenvolvidas nesta Dissertação são contrastadas e comparadas aos resultados do CPLEX limitado a 1800 segundos para a solução de cada problema.

Por fim, os resultados obtidos para o PDLMP e o PDLMPSC são avaliados, atestando a economia obtida ao adicionar *setup carry-over* nessas instâncias que foram proposta na literatura. Essa avaliação visa atestar que o PDLMP com essa estrutura de preparação garante resultados vantajosos e possibilita uma considerável economia ao setor produtivo.

5.1 Conjuntos de Instâncias

As instâncias utilizadas para os experimentos deste capítulo seguem a mesma definição de parâmetros das instâncias do Capítulo 3, apresentados na Seção 3.4.

5.2 Experimento I

Por apresentar melhores resultados, o modelo para o PDLMP apresentado na Seção 3.2, baseado em caminhos mínimos, foi implementado utilizando a math-heurística modificada proposta e testada utilizando as 180 instâncias mencionadas na seção anterior. Foram empregados os seguintes valores de parâmetros para a math-heurística, obtidos após testes preliminares:

- Limite da restrição *local-branching*: $p \times n$, onde p representa o número de períodos da instância e n a quantidade de produtos;

- CPLEX limitado a 150 segundos para a resolução de cada subproblema gerado pelo método;
- Tempo máximo de execução do método: 1800 segundos;
- α é um valor aleatório entre 1 e $p/2$ e $\gamma = 5$;
- $GAP = 0,05$.

Os resultados obtidos pela math-heurística modificada e pelo CPLEX v. 12.6 limitado em 1800 segundos, ambos utilizando o modelo baseado em caminhos mínimos de (JANS, 2009) foram comparados utilizando gráfico de perfil de desempenho de Dolan e Moré (2002), descrito na Seção 3.4.

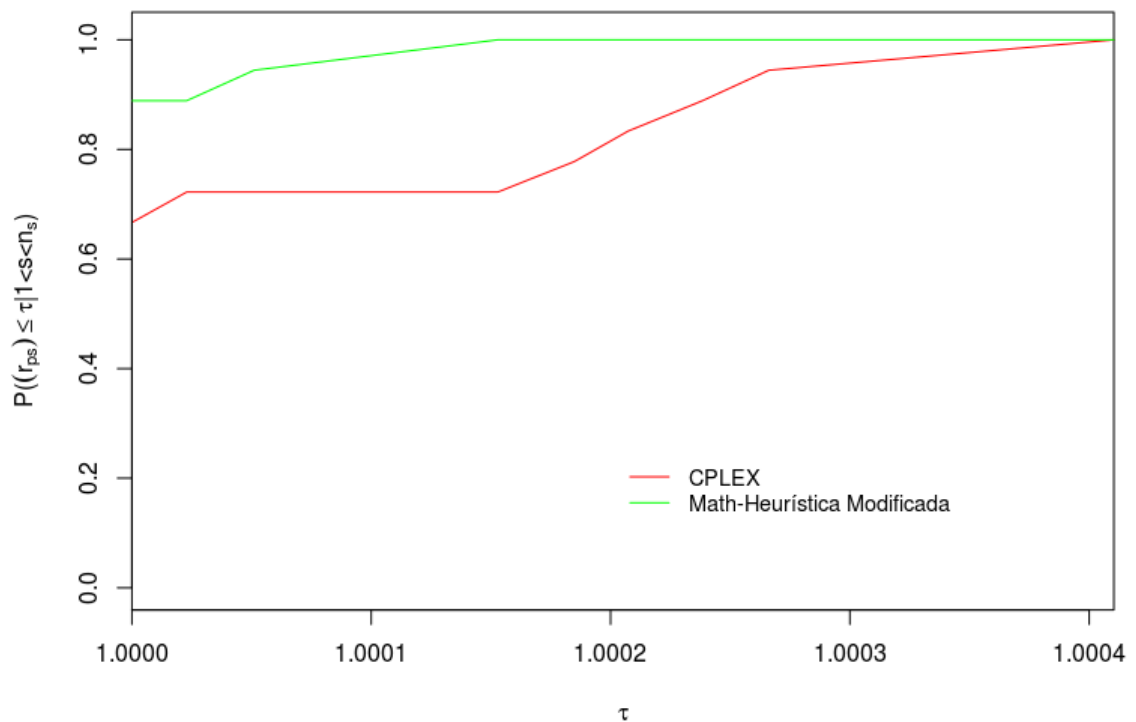


FIGURA 5.1 – Perfil de desempenho das soluções obtidas pela math-heurística proposta comparando com as do CPLEX limitado em 1800 segundos.

A Figura 5.1 mostra os perfis de desempenho na resolução das instâncias por meio da math-heurística modificada e dos resultados obtidos pelo CPLEX. O eixo das ordenadas é dado pelo número de problemas que a execução do CPLEX e da math-heurística atingiu uma razão de desempenho igual ou melhor que um coeficiente τ (eixo das abcissas). Conforme pode-se observar, a math-heurística modificada apresentou um melhor desempenho, pois conseguiu encontrar melhores soluções na maior proporção de instâncias dentro de

um dado fator τ do desempenho do CPLEX. Entretanto, vale a pena apontar que a vantagem da math-heurística foi por um fator bem baixo, pois para o eixo-y atingir 100%, o τ necessário foi de 1,0004.

Além do gráfico de desempenho, a Tabela 5.1 apresenta a média para o conjunto de 10 instâncias, totalizando 18 conjuntos, dos resultados obtidos por esses métodos.

TABELA 5.1 – Médias das soluções obtidas pelo CPLEX e pela math-heurística para o conjunto de 10 instâncias com diferentes quantidades de itens, máquinas e períodos

n	m	p	AAA		NBB	
			CPLEX	<i>Math-Heurística</i>	CPLEX	<i>Math-Heurística</i>
6	2	6	27.380,49	27.228,51	16.242,26	16.159,86
		12	26.719,50	26.517,94	14.916,33	14.844,93
		18	43.830,91	43.485,51	23.064,10	23.018,56
12	4	6	22.278,69	21.831,04	13.533,90	13.436,57
		12	41.819,18	41.480,20	26.919,65	26.789,89
		18	61.834,69	61.416,13	40.436,40	40.335,75
50	6	6	78.061,10	77.418,65	51.840,07	51.683,64
		12	150.151,29	149.548,64	103.710,11	103.477,77
		18	221.487,60	220.887,26	155.230,36	155.098,02

Pela Tabela 5.1 é possível observar que as médias das soluções obtidas pela math-heurística nas duas classes de instâncias e em todas as configurações de itens, máquinas e períodos foram menores do que as médias das soluções obtidas pelo CPLEX. Além disso, percebe-se que, tanto para a classe AAA quanto para a classe NBB, a diferença das médias é maior para configurações que contém maiores quantidades de itens máquinas e períodos. Isso ocorre, provavelmente, porque para a maioria dessas instâncias o CPLEX não encontra soluções ótimas em 1800 segundos e, os *gaps* das soluções obtidas pela math-heurística são menores do que os *gaps* das soluções obtidas pelo CPLEX para essas instâncias.

5.3 Experimento II

Esta seção apresenta os resultados atingidos pela primeira versão da math-heurística (Experimento II.a) e pela versão modificada, apresentando os resultados atingidos com dois modelos (Experimento II.b).

5.3.1 Experimento II.a

Neste experimento, considerou-se o modelo para o PDLMPSC apresentado na Seção 3.2 empregando o CPLEX 12.6.3 (ILOG, 2015) para a solução dos problemas das instâncias propostas em (TOLEDO, 1998). O conjunto de 180 instâncias explicadas neste capítulo foram modeladas e resolvidas pelo CPLEX limitado em 1800 segundos. Essas instâncias também foram solucionadas utilizando a primeira versão da math-heurística proposta utilizando os seguintes valores de parâmetros, obtidos após testes preliminares:

- Limite da restrição *local-branching*: $\frac{p}{2}$;
- CPLEX limitado a 80 segundos para a resolução de cada subproblema gerado pelo método;
- Quantidade de iterações $M = 2$.

A Figura 5.2 apresenta os perfis de desempenho de (DOLAN; MOREÉ, 2002) comparando a qualidade das soluções obtidas pelo método desenvolvido e pelo CPLEX.

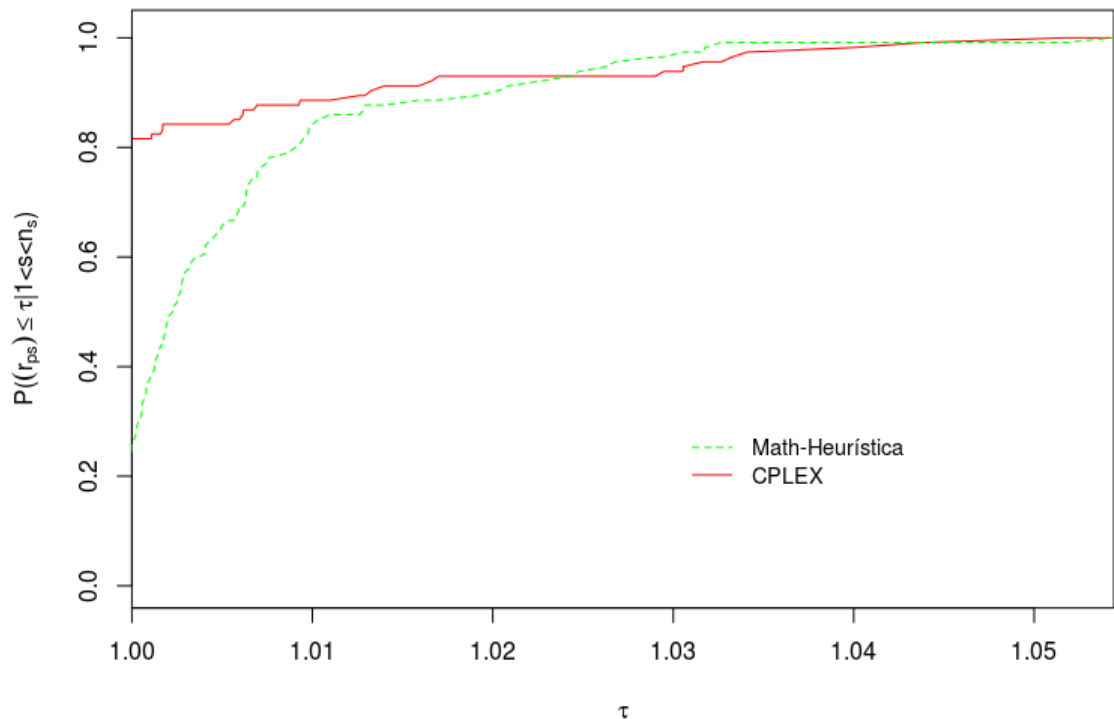


FIGURA 5.2 – Perfil de desempenho comparando os resultados obtidos pela primeira versão da math-heurística e pelo CPLEX.

Observando a Figura 5.2 pode-se afirmar que o CPLEX conseguiu resolver uma maior proporção de instâncias, dentro de um dado fator τ do desempenho da math-heurística, mas essa proporção é pequena, apenas no intervalo $[1, 0; 1, 02]$ do eixo das abcissas (τ). Pode-se dizer que os desempenhos de ambos os métodos (CPLEX e math-heurística), em relação às soluções obtidas são parecidos, apesar do CPLEX possuir o desempenho um pouco melhor.

A Figura 5.3 apresenta o perfil de desempenho avaliando-se o tempo de solução necessário pelo método desenvolvido e pelo CPLEX.

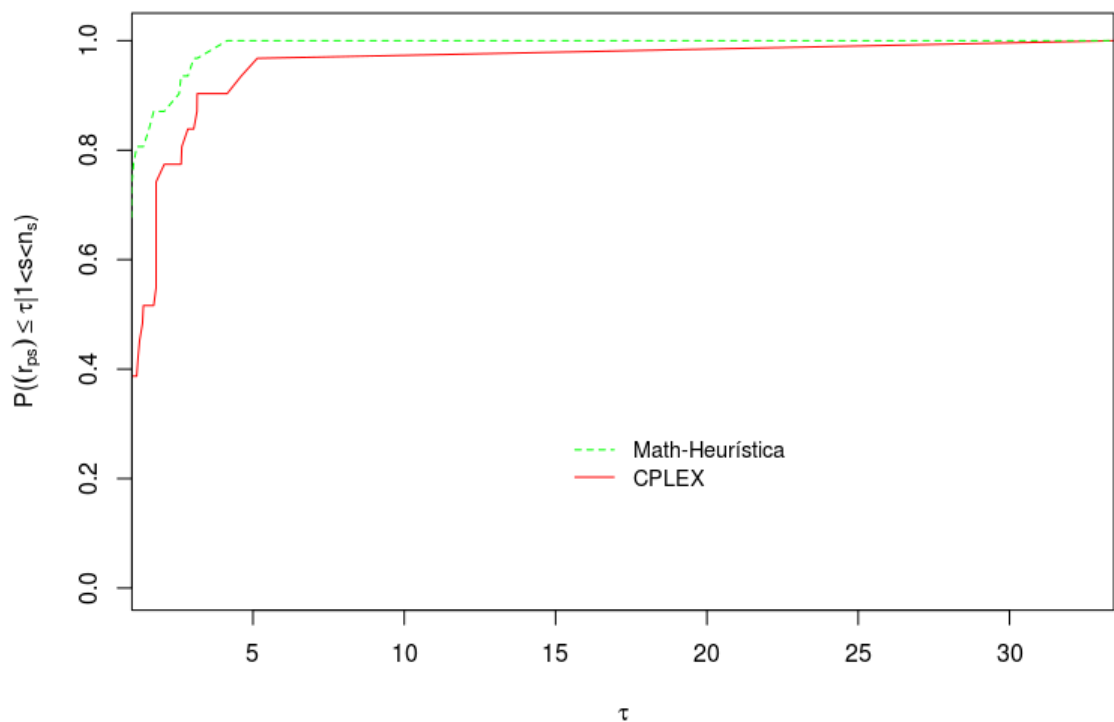


FIGURA 5.3 – Perfil de Desempenho em Relação ao Tempo de Execução

Pela Figura 5.3 pode-se concluir que a math-heurística conseguiu resolver sempre a maior proporção de instâncias dentro de um dado fator τ do desempenho em relação ao tempo do CPLEX.

Vale a pena mencionar que se aumentando o tempo ou o número de iterações da math-heurística desenvolvida, não houve evolução na qualidade das soluções obtidas pela math-heurística. Dessa forma, estudaram-se outras formas de melhoria, que originou a versão modificada dela, já discutida no capítulo anterior. Os experimentos com tal versão da math-heurística são apresentados na seção seguinte.

5.3.2 Experimento II.b

A fim de testar a eficiência da versão modificada da math-heurística, ela foi implementada considerando o modelo baseado em caminhos mínimos e o modelo de localização de facilidades para o PDLMPSC.

Neste experimento, consideram-se dois modelos para avaliação do desempenho da math-heurística modificada proposta. O modelo baseado em caminhos mínimos foi escolhido para compor o método proposto, pois apresentou melhores resultados em testes realizados no Capítulo 3. O modelo baseado em localização de facilidades também foi avaliado, pois é um modelo no qual métodos do tipo *relax-and-fix* apresentam bons resultados computacionais para instâncias de médio porte (SILVA, 2013).

Utilizaram-se as 180 instâncias mencionadas neste capítulo para avaliar o desempenho dos métodos. Foram utilizados os seguintes valores de parâmetros, obtidos após testes preliminares:

- Limite da restrição *local-branching*: $p \times n$, em que p representa o número de períodos da instância e n a quantidade de produtos;
- CPLEX limitado a 150 segundos para a resolução de cada subproblema gerado pelo método;
- Tempo máximo de execução do método: 1800 segundos;
- α é um valor aleatório entre 1 e $p/2$ e $\gamma = 5$;
- $GAP = 0,05$;
- Tempo limite total para execução: 1800 segundos.

Os resultados obtidos pela math-heurística modificada considerando ambos os modelos, distinguidos na legenda por MH-Modelo 1 (na versão usando o modelo de caminhos mínimos) e MH-Modelo 2 (na versão usando o modelo de localização de facilidades) e pelo CPLEX também considerando ambos os modelos, rotulando CPLEX-Modelo 1 (na versão usando o modelo de caminhos mínimos e CPLEX-Modelo 2 (na versão usando o modelo de localização de facilidades) foram comparados utilizando gráfico de perfil de desempenho de Dolan e Moré (2002), apresentada na Figura 5.4.

Pela Figura pode-se concluir que a math-heurística utilizando os Modelos 1 e 2 conseguiu resolver sempre a maior proporção de instâncias dentro de um dado fator τ do desempenho em relação ao tempo do CPLEX utilizando esses modelos. Além disso, é possível observar que a probabilidade da math-heurística encontrar soluções melhores que o CPLEX utilizando o Modelo 1 é maior do que quando utiliza-se o Modelo 2.

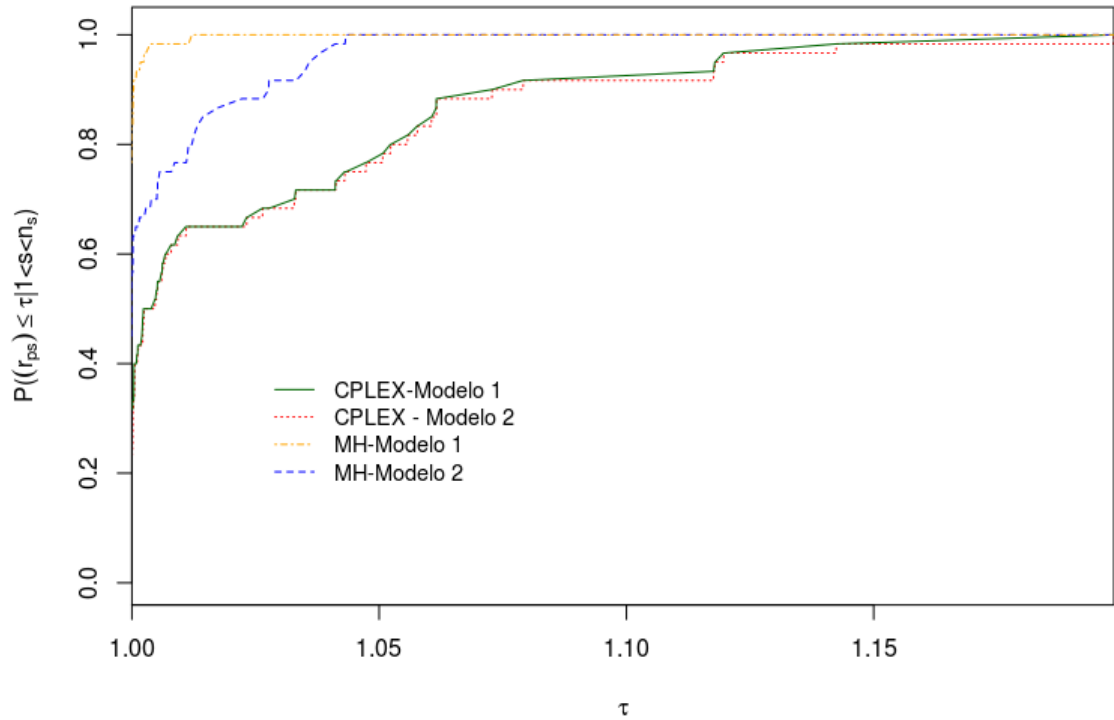


FIGURA 5.4 – Perfil de desempenho dos métodos avaliados, com relação à qualidade da solução obtida.

Além do gráfico, as Tabelas 5.2 e 5.3 apresentam as soluções médias por conjunto de 10 instâncias obtidas pelo CPLEX e pela math-heurística utilizando o Modelo 1 (caminhos mínimos) e Modelo 2 (localização de facilidades) para os conjuntos de instâncias AAA e NBB, respectivamente.

TABELA 5.2 – Médias das soluções obtidas pelo CPLEX e pelas math-heurísticas MH-Modelo 1 e MH-Modelo 2 para o conjunto de 10 instâncias com diferentes quantidades de itens, máquinas e períodos, do tipo AAA.

n	m	p	CPLEX-Modelo 1	MH-Modelo 1	MH-Modelo 2
6	2	6	25.281,20	25.275,66	25.276,86
		12	21.813,76	21.806,32	21.807,21
		18	33.948,24	33.941,66	33.944,01
12	4	6	19.468,59	19.453,02	19.454,13
		12	36.847,18	36.832,20	36.834,84
		18	56.115,74	56.101,17	56.102,30
50	6	6	74.888,26	74.867,69	74.870,82
		12	148.347,70	148.326,25	148.329,03
		18	218.433,59	218.412,62	218.413,94

TABELA 5.3 – Médias das soluções obtidas pelo CPLEX e pelas math-heurísticas MH-Modelo 1 e MH-Modelo 2 para o conjunto de 10 instâncias com diferentes quantidades de itens, máquinas e períodos, do tipo NBB.

n	m	p	CPLEX-Modelo 1	MH-Modelo 1	MH-Modelo 2
6	2	6	15.024,61	15.018,85	15.020,07
		12	14.676,55	14.673,67	14.675,20
		18	19.312,65	19.306,00	19.307,31
12	4	6	12.258,22	12.254,66	12.255,68
		12	25.059,06	25.053,50	25.056,52
		18	40.337,79	40.331,14	40.332,36
50	6	6	58.573,17	58.570,71	58.571,74
		12	102.159,57	102.153,12	102.154,13
		18	152.952,22	152.948,65	152.950,68

Analisando as Tabelas 5.2 e 5.3 percebe-se que as médias obtidas pelas math-heurísticas são sempre menores do que as obtidas pelo CPLEX. Vale mencionar que dentro do limite de 1800 segundos, usando o Modelo 1 ou 2, o CPLEX atingiu os mesmos resultados. As diferenças entre as soluções obtidas pelas math-heurísticas e o CPLEX na classe NBB são, no geral, menores do que as diferenças obtidas pela classe AAA. Isso provavelmente decorre da menor dificuldade de resolução dos problemas da classe NBB e, por isso, o CPLEX encontrou soluções boas em 1800 segundos.

5.4 Experimento III

Os resultados obtidos pela math-heurística modificada baseado em caminhos mínimos (Modelo 1) para o PDLMP e para o PDLMPSC foram comparados a fim de averiguar a influência da inclusão do *setup carry-over* no PDLMP. Para isso, calculou-se a média de todas as soluções obtidas para o PDLMP e de todas as soluções obtidas para o PDLMPSC, a comparação dos valores obtidos é dada pela Figura 5.5.

Através do gráfico apresentado é possível constatar que o PDLMP com o *setup carry-over* garante uma economia média de 10% levando-se em consideração as instâncias testadas.

A Tabela 5.4 apresenta a economia gerada pela inclusão do *setup carry-over* nas duas classes de instâncias estudadas, por grupo de instâncias.

É possível notar pelos resultados que a melhor economia foi nas instâncias com 2 máquinas quando a quantidade de períodos considerada foi maior ou igual a 12. Além disso, observam-se os maiores *GAPs* nas soluções obtidas pela classe AAA. Isso provavelmente

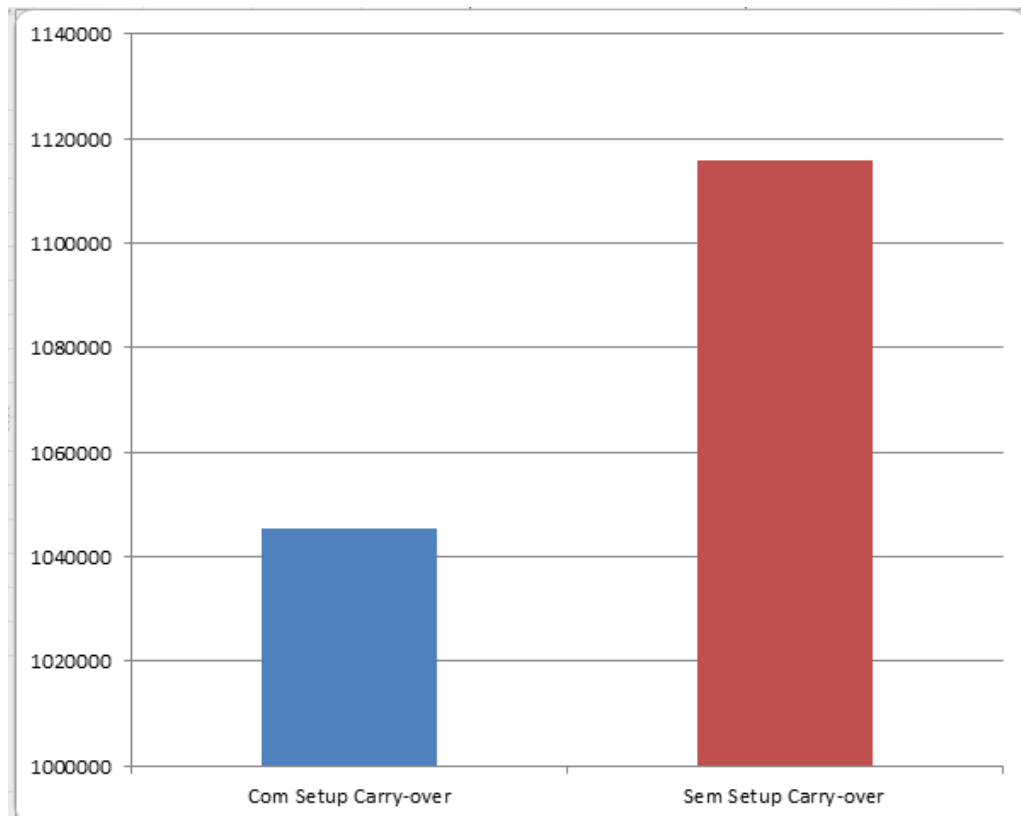


FIGURA 5.5 – Comparação da média das soluções obtidas pela math-heurística modificada para o PDLMP e para o PDLMPSC.

TABELA 5.4 – Médias das soluções obtidas pela math-heurística para o PDLMP (sem *setup carry-over*) e para o PDLMPSC (com *setup carry-over*).

n	m	p	AAA			NBB		
			PDLMPSC	PDLMP	GAP	PDLMPSC	PDLMP	GAP
6	2	6	25.275,66	27.228,51	8%	15.018,85	16.242,26	8%
		12	21.806,32	26.517,94	22%	14.673,67	14.916,33	2%
		18	33.941,66	43.485,51	28%	19.306,00	23.064,10	19%
12	4	6	19.453,02	21.831,04	12%	12.254,66	13.533,90	10%
		12	36.832,20	41.480,20	13%	25.053,50	26.919,65	7%
		18	56.101,17	61.416,13	9%	40.331,14	40.436,40	2%
50	6	6	74.867,62	77.418,65	3%	51.570,71	58.840,07	14%
		12	148.326,25	149.548,64	2%	102.153,12	103.710,11	2%
		18	218.412,62	220.887,26	1%	152.948,65	155.230,36	1%

ocorre porque as capacidades das máquinas das instâncias nessas classes de problemas são escassas e o *setup carry-over* proporciona a economia dessas capacidades.

6 Conclusões

A redução de custos e uma maior produtividade são dois grandes alvos de empresas que buscam tornarem-se competitivas em suas áreas. Para que esses alvos sejam atingidos, é importante que ocorra o planejamento e controle da produção. Nesta Dissertação de Mestrado, estudou-se o problema de dimensionamento de lotes, que faz parte do planejamento da produção, e é composto por máquinas paralelas distintas com capacidades limitadas e um tempo de preparação (*setup*) que é imposto no início do processo de produção de cada item em todas as máquinas. Adicionalmente, visando a otimização na realização de *setup*, é permitido que a preparação associada a um determinado período seja realizada em um período anterior (*setup carry-over*), seja pela execução antecipada do *setup*, seja pelo aproveitamento do estado da máquina em período anterior.

Esse problema, denominado Problema de Dimensionamento de Lotes com *Setup Carry-Over* (PDLMPSC) sem sequenciamento, até o momento e no limite do conhecimento da autora, não foi explorado na literatura. Tendo em vista que a consideração do *setup carry-over* está cada vez mais frequente devido à evolução de pacotes de otimização e, em particular, por possibilitar grandes economias para variantes do PDL com várias máquinas, esse estudo mostra-se bastante relevante para o setor produtivo.

Nesse sentido, essa Dissertação focou principalmente no aspecto computacional do problema, para o desenvolvimento de uma estratégia eficiente para resolver o PDLMPSC.

Em um primeiro momento, foram estudados três modelos para o Problema de Dimensionamento de Lotes com Máquinas Paralelas (PDLMP) e a partir dos resultados obtidos, dois modelos foram escolhidos para o acréscimo do *setup carry-over* e posterior desenvolvimento do método proposto para a solução do PDLMPSC.

A math-heurística desenvolvida é baseada na conhecida heurística *relax-and-fix* com restrições *local branching* e foi inspirada na meta-heurística *Greedy Randomized Search Procedures* (GRASP), que possui múltiplas iterações, cada uma com duas fases: uma de construção e outra de busca local.

Em uma versão preliminar a math-heurística demonstrou potencial na resolução do problema. Mas, com o intuito de melhorar ainda mais os resultados obtidos, foram propostas melhorias no método desenvolvido.

Devido à falta de estudos desse problema na literatura, a análise comparativa se limitou ao CPLEX com tempo de execução fixo. Entretanto, vale ressaltar que a heurística escolhida foi em função de estudos recentes do PDLMP. Por meio dos resultados obtidos, assegura-se a superioridade do método frente à literatura do PDLMP.

Como trabalhos futuros, entende-se incluir as restrições de sequenciamento a fim de atestar a eficiência dessa math-heurística que mostrou-se de grande potencial para a resolução de PDLs.

Referências

- ALMADA-LOBO, B.; KLABJAN, D.; CARRAVILLA, M. A.; OLIVEIRA, J. F. Single machine multi-product capacitated lot sizing with sequence-dependent setups. **International Journal of Production Research**, Taylor & Francis, v. 45, n. 20, p. 4873–4894, 2007.
- ANTHONY, R. N. **Planning and Control Systems: A Framework for Analysis**. Harvard University, Graduate School of Business Administration: Division of Research, 1965.
- ARMENTANO, V. A.; FRANÇA, P. M.; TOLEDO, F. M. de. Multilevel capacitated lotsizing complexity and lp-based heuristics. **European Journal of Operational Research**, v. 53, n. 2, p. 131–148, 1999.
- BAHL, H. C.; RITZMAN, L. P.; GUPTA, J. N. Or practice—determining lot sizes and resource requirements: A review. **Operations Research**, INFORMS, v. 35, n. 3, p. 329–345, 1987.
- BARROS, A. d.; MOCCELLIN, J. V. Análise da flutuação do gargalo em flow shop permutacional com tempos de setup assimétricos e dependentes da seqüência. **Gestão da Produção. São Carlos**, SciELO Brasil, v. 11, n. 1, 2004.
- BELO-FILHO, M. A.; TOLEDO, F. M.; ALMADA-LOBO, B. Models for capacitated lot-sizing problem with backlogging, setup carryover and crossover. **Journal of the Operational Research Society**, v. 65, n. 11, p. 1735–1747, Nov 2014. ISSN 1476-9360.
- BERALDI, P.; GHIANI, G.; GRIECO, A.; GUERRIERO, E. Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs. **Computers & Operations Research**, v. 35, n. 11, p. 3644 – 3656, 2008. ISSN 0305-0548.
- BILDE, O.; KRARUP, J. Sharp lower bounds and efficient algorithms for the simple plant location problem. In: **Annals of Discrete Mathematics**. [S.l.]: Elsevier, 1977. v. 1, p. 79–97.
- BITRAN, G. R.; YANASSE, H. H. Computational complexity of the capacitated lot size problem. **Management Science**, INFORMS, v. 28, n. 10, p. 1174–1186, 1982.

- BRAHIMI, N.; DAUZERE-PERES, S.; NAJID, N. M.; NORDLI, A. Single item lot sizing problems. **European Journal of Operational Research**, Elsevier, v. 168, n. 1, p. 1–16, 2006.
- BUSCHKÜHL, L.; SAHLING, F.; HELBER, S.; TEMPELMEIER, H. Dynamic capacitated lot-sizing problems: a classification and review of solution approaches. **Or Spectrum**, Springer, v. 32, n. 2, p. 231–261, 2010.
- CARRENO, J. J. Economic lot scheduling for multiple products on parallel identical processors. **Management Science**, INFORMS, v. 36, n. 3, p. 348–358, 1990.
- CARVALHO, D. M.; NASCIMENTO, M. C. Lagrangian heuristics for the capacitated multi-plant lot sizing problem with multiple periods and items. **Computers & Operations Research**, Elsevier, v. 71, p. 137–148, 2016.
- CASERTA, M.; RAMIREZ, A.; VOSS, S. A math-heuristic for the multi-level capacitated lot sizing problem with carryover. In: SPRINGER. **European Conference on the Applications of Evolutionary Computation**. [S.l.], 2010. p. 462–471.
- CASERTA, M.; VOSS, S. A math-heuristic dantzig-wolfe algorithm for capacitated lot sizing. **Annals of Mathematics and Artificial Intelligence**, v. 69, n. 2, p. 207–224, Oct 2013. ISSN 1573-7470.
- CHEN, C. **Fix-and-optimize heuristic and mp-based approaches for capacitated lot sizing problem with setup carry over, setup splitting and backlogging**. Tese (Doutorado) — Arizona State University, EUA, 2015.
- CUNHA, A. L.; SANTOS, M. O.; MORABITO, R.; PÓVOA, A. B. Um modelo para o problema de dimensionamento de lotes com aquisição de matérias-primas. **Anais do Simpósio Brasileiro de Pesquisa Operacional**, v. 47, p. 423 – 450, 12 2015.
- DIABY, M.; BAHL, H.; KARWAN, M.; ZIONTS, S. Capacitated lot-sizing and scheduling by Lagrangean relaxation. **European Journal of Operational Research**, Elsevier, v. 59, n. 3, p. 444–458, 1992.
- DIABY, M.; BAHL, H. C.; KARWAN, M. H.; ZIONTS, S. A Lagrangean relaxation approach for very-large-scale capacitated lot-sizing. **Management Science**, INFORMS, v. 38, n. 9, p. 1329–1340, 1992.
- DOLAN, E. D.; MORÉ, J. J. Benchmarking optimization software with performance profiles. **Mathematical programming**, Springer, v. 91, n. 2, p. 201–213, 2002.
- DREXL, A.; KIMMS, A. Lot sizing and scheduling - survey and extensions. **European Journal of Operational Research**, Elsevier, v. 99, n. 2, p. 221–235, 1997.
- EPPEN, G. D.; MARTIN, R. K. Solving multi-item capacitated lot-sizing problems using variable redefinition. **Operations Research**, INFORMS, v. 35, n. 6, p. 832–848, 1987.
- FEO, T. A.; RESENDE, M. G. Greedy randomized adaptive search procedures. **Journal of global optimization**, Springer, v. 6, n. 2, p. 109–133, 1995.

- 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, n. 2, p. 697 – 706, 2009. ISSN 0377-2217.
- FIOROTTO, D. J.; ARAUJO, S. A. de; JANS, R. Hybrid methods for lot sizing on parallel machines. **Computers & Operations Research**, Elsevier, v. 63, p. 136–148, 2015.
- FISCHETTI, M.; LODI, A. Local branching. **Mathematical Programming**, v. 98, n. 1, p. 23–47, Sep 2003.
- FLORIAN, M.; LENSTRA, J. K.; KAN, A. R. Deterministic production planning: Algorithms and complexity. **Management science**, INFORMS, v. 26, n. 7, p. 669–679, 1980.
- GOPALAKRISHNAN, M.; MILLER, D. M.; SCHMIDT, C. P. A framework for modelling setup carryover in the capacitated lot sizing problem. **International Journal of Production Research**, Taylor & Francis, v. 33, n. 7, p. 1973–1988, 1995.
- ILOG. **CPLEX 12.6 Reference Manual**. [S.l.], 2015. V. 12.6.
- JANS, R. Solving lot-sizing problems on parallel identical machines using symmetry-breaking constraints. **INFORMS Journal on Computing**, INFORMS, v. 21, n. 1, p. 123–136, 2009.
- JANS, R.; DEGRAEVE, Z. Modeling industrial lot sizing problems: a review. **International Journal of Production Research**, Taylor & Francis, v. 46, n. 6, p. 1619–1643, 2008.
- KARIMI, B.; GHOMI, S. F.; WILSON, J. The capacitated lot sizing problem: a review of models and algorithms. **Omega**, Elsevier, v. 31, n. 5, p. 365–378, 2003.
- KUIK, R.; SALOMON, M.; WASSENHOVE, L. N. V. Batching decisions: structure and models. **European journal of operational research**, Elsevier, v. 75, n. 2, p. 243–263, 1994.
- LASDON, L. S.; TERJUNG, R. An efficient algorithm for multi-item scheduling. **Operations research**, INFORMS, v. 19, n. 4, p. 946–969, 1971.
- LOZANO, S.; LARRANETA, J.; ONIEVA, L. Primal-dual approach to the single level capacitated lot-sizing problem. **European Journal of Operational Research**, Elsevier, v. 51, n. 3, p. 354–366, 1991.
- MAES, J.; MCCLAIN, J. O.; WASSENHOVE, L. N. V. Multilevel capacitated lotsizing complexity and LP-based heuristics. **European Journal of Operational Research**, Elsevier, v. 53, n. 2, p. 131–148, 1991.
- MOHAMMADI, M.; GHOMI, S. M. T. F.; KARIMI, B.; TORABI, S. A. Rolling-horizon and fix-and-relax heuristics for the multi-product multi-level capacitated lotsizing problem with sequence-dependent setups. **Journal of Intelligent Manufacturing**, v. 21, n. 4, p. 501–510, Aug 2010. ISSN 1572-8145.
- MOREIRA, D. A. Produção e operações. **São Paulo: Pioneira**, 1998.

- NASCIMENTO, M. C.; RESENDE, M. G.; TOLEDO, F. M. GRASP heuristic with path-relinking for the multi-plant capacitated lot sizing problem. **European Journal of Operational Research**, Elsevier, v. 200, n. 3, p. 747–754, 2010.
- NASCIMENTO, M. C. V.; TOLEDO, F. M. B. A hybrid heuristic for the multi-plant capacitated lot sizing problem with setup carry-over. **Journal of the Brazilian Computer Society**, scielo, v. 14, p. 7 – 15, 12 2008. ISSN 0104-6500.
- OLIVEIRA, B. B.; CARRAVILLA, M. A.; OLIVEIRA, J. F.; TOLEDO, F. M. A relax-and-fix-based algorithm for the vehicle-reservation assignment problem in a car rental company. **European Journal of Operational Research**, v. 237, n. 2, p. 729 – 737, 2014. ISSN 0377-2217.
- PAIVA, R. A. de; TOLEDO, F. M. B. Local branching aplicado ao problema de dimensionamento de lotes. **Anais do Simpósio Brasileiro de Pesquisa Operacional**, v. 42, p. 2386 – 2393, 2010.
- PAIVA, R. Andrade de. **Local branching aplicado ao problema de dimensionamento de lotes**. Dissertação (Mestrado) — Universidade de São Paulo, 2010.
- PORKKA, P.; VEPSALAINEN, A. P. J.; KUULA, M. Multiperiod production planning carrying over set-up time. **International Journal of Production Research**, Taylor & Francis, v. 41, n. 6, p. 1133–1148, 2003.
- PRODUCTION planning in furniture settings via robust optimization. **Computers & Operations Research**, v. 39, n. 2, p. 139 – 150, 2012. ISSN 0305-0548.
- QUADT, D.; KUHN, H. Capacitated lot-sizing and scheduling with parallel machines, back-orders, and setup carry-over. **Naval Research Logistics (NRL)**, Wiley Online Library, v. 56, n. 4, p. 366–384, 2009.
- SABBAG, Z. **Planejamento da produção em máquinas paralelas sob restrições de capacidade**. Tese (Dissertação) — Universidade Estadual de Campinas, 1993.
- SAHLING, F.; BUSCHKUHLE, L.; TEMPELMEIER, H.; HELBER, S. Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. **Computers & Operations Research**, v. 36, n. 9, p. 2546 – 2553, 2009. ISSN 0305-0548.
- SILVA, D. H. **Métodos híbridos para o problema de dimensionamento de lotes com múltiplas plantas**. Tese (Doutorado) — Universidade de São Paulo, 2013.
- SIMÕES, W. L. Abordagem metaheurística híbrida para a otimização de sequenciamento de produção em flow shop permutacional com tempos de setup dependentes da sequência. Universidade do Vale do Rio dos Sinos, 2016.
- SOUZA, K. X. S. D.; ARMENTANO, V. A. Multi-item capacitated lot-sizing by a cross decomposition based algorithm. **Annals of Operations Research**, Springer, v. 50, n. 1, p. 557–574, 1994.
- SOX, C. R.; GAO, Y. The capacitated lot sizing problem with setup carry-over. **IIE Transactions**, Taylor & Francis, v. 31, n. 2, p. 173–181, 1999.

- TOLEDO, C. F. M.; ARANTES, M. da S.; HOSSOMI, M. Y. B.; FRANÇA, P. M.; AKARTUNALI, K. A relax-and-fix with fix-and-optimize heuristic applied to multi-level lot-sizing problems. **Journal of Heuristics**, v. 21, n. 5, p. 687–717, Oct 2015. ISSN 1572-9397.
- TOLEDO, F. M. B. **Dimensionamento de lotes em máquinas paralelas**. Tese (Doutorado) — Universidade Estadual de Campinas, 1998.
- 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**, Elsevier, v. 175, n. 2, p. 1070–1083, 2006.
- TOSO, E. A. V.; MORABITO, R.; CLARK, A. Combinação de abordagens glsp e atsp para o problema de dimensionamento e sequenciamento de lotes de produção de suplementos para nutrição animal. **Pesquisa Operacional**, scielo, v. 28, p. 423 – 450, 12 2008. ISSN 0101-7438.
- TRIGEIRO, W. W.; THOMAS, L. J.; MCCLAIN, J. O. Capacitated lot sizing with setup times. **Management science**, INFORMS, v. 35, n. 3, p. 353–366, 1989.
- WAGNER, H. M.; WHITIN, T. M. Dynamic version of the economic lot size model. **Management Science**, v. 5, n. 1, p. 89–96, 1958.
- WOLSEY, L. A. Progress with single-item lot-sizing. **European Journal of Operational Research**, Elsevier, v. 86, n. 3, p. 395–401, 1995.
- WOLSEY, L. A. **Integer programming**. [S.l.]: Wiley, 1998.
- WU, T.; SHI, L. Mathematical models for capacitated multi-level production planning problems with linked lot sizes. **International Journal of Production Research**, Taylor & Francis, v. 49, n. 20, p. 6227–6247, 2011.
- ZANGWILL, W. I. A deterministic multi-period production scheduling model with backlogging. **Management Science**, INFORMS, v. 13, n. 1, p. 105–119, 1966.

FOLHA DE REGISTRO DO DOCUMENTO

1. CLASSIFICAÇÃO/TIPO DM	2. DATA 09 de fevereiro de 2018	3. REGISTRO NÚMERO	4. NÚMERO DE PÁGINAS 69
5. TÍTULO E SUBTÍTULO: Uma Math-Heurística para o Problema de Dimensionamento de Lotes com Máquinas Paralelas e <i>Setup Carry-over</i>			
6. AUTORA(ES): Carla Cristina Doescher Fernandes			
7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES): Instituto Tecnológico de Aeronáutica – ITA / Universidade Federal de São Paulo – UNIFESP			
8. PALAVRAS-CHAVE SUGERIDAS PELA AUTORA: Dimensionamento de lotes; Máquinas Paralelas; Múltiplos Itens; Setup Carry-over.			
9. PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO: Dimensionamento de lotes; Máquinas Paralelas; Múltiplos Itens; Setup Carry-over.			
10. APRESENTAÇÃO:		<input checked="" type="checkbox"/> Nacional <input type="checkbox"/> Internacional	
ITA/UNIFESP, São José dos Campos. Curso de Mestrado. Programa de Pós-Graduação em Pesquisa Operacional. Orientador: Prof ^ª . Dr ^ª . Mariá Cristina Vasconcelos Nascimento Rosset. Defesa em 09/02/2018.			
11. RESUMO: <p>Esta Dissertação de Mestrado aborda o Problema de Dimensionamento de Lotes (PDL) com Máquinas Paralelas em que são considerados múltiplos itens e máquinas distintas que produzem os mesmos itens e que possuem restrições de capacidade. Os itens podem ser produzidos em qualquer máquina e no início da produção de cada item incorrem-se tempo e custo de <i>setup</i> da máquina utilizada. Além disso, considera-se a possibilidade de aproveitamento de <i>setup</i> no final de cada período : <i>setup carry-over</i>, para redução de custos e, de forma implícita, para a definição de um planejamento que diminua possíveis impactos ambientais provocados pela preparação de máquinas. Apesar de diversos trabalhos na literatura para o PDL com Máquinas Paralelas, poucos são os métodos de solução eficientes para resolvê-lo considerando <i>setup carry-over</i>. Nesse sentido, para resolver esse problema de forma eficiente, nesta Dissertação, realizou-se o seu estudo do ponto de vista de modelagem e de método de solução eficiente. Primeiramente, adaptaram-se três modelagens de programação inteira mista para o PDL com Máquinas Paralelas, referido aqui por PDLMP. Após experimentos, definiram-se dois modelos que foram utilizados para o método de solução proposto baseado em modelagem (math-heurística). A math-heurística baseada na conhecida heurística <i>relax-and-fix</i> com restrições <i>local branching</i> foi inspirada na meta-heurística <i>Greedy Randomized Search Procedures</i> (GRASP). Experimentos, nos quais são comparadas as soluções obtidas pelo <i>solver</i> CPLEX e pela math-heurística, demonstraram a eficiência do método proposto.</p>			
12. GRAU DE SIGILO: <input checked="" type="checkbox"/> OSTENSIVO <input type="checkbox"/> RESERVADO <input type="checkbox"/> SECRETO			