UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Análise, proposição e solução de modelos para o problema integrado de dimensionamento de lotes e sequenciamento da produção

Willy Alves de Oliveira Soler

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Willy Alves de Oliveira Soler

**Análise, proposição e solução de modelos para o problema
integrado de dimensionamento de lotes e sequenciamento
da produção**

Tese apresentada ao Instituto de Ciências
Matemáticas e de Computação – ICMC-USP,
como parte dos requisitos para obtenção do título
de Doutor em Ciências – Ciências de Computação e
Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e
Matemática Computacional

Orientadora: Profa. Dra. Maristela Oliveira dos Santos

**USP – São Carlos
Janeiro de 2018**

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

S685a Soler, Willy Alves de Oliveira
Análise, proposição e solução de modelos para o
problema integrado de dimensionamento de lotes e
sequenciamento da produção / Willy Alves de Oliveira
Soler; orientadora Maristela Oliveira dos Santos. --
São Carlos, 2017.
131 p.

Tese (Doutorado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional) --
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2017.

1. Planejamento da produção. 2. Programação
matemática inteira mista. 3. Heurística. I. Santos,
Maristela Oliveira dos, orient. II. Título.

Willy Alves de Oliveira Soler

**Analysis, proposition and solution of models for the
simultaneous lot sizing and scheduling problem**

Doctoral dissertation submitted to the Institute of
Mathematics and Computer Sciences – ICMC-USP, in
partial fulfillment of the requirements for the degree of
the Doctorate Program in Computer Science and
Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and
Computational Mathematics

Advisor: Profa. Dra. Maristela Oliveira dos Santos

USP – São Carlos
January 2018

Dedico este trabalho a Deus e à minha família.

AGRADECIMENTOS

Primeiramente, agradeço a Deus pelo dom da vida, pela família e por propiciar a realização deste curso. Por ter iluminado a minha mente, guiando-me em todos os aspectos da vida, inclusive nos resultados que foram alcançados neste trabalho.

Agradeço à minha esposa Francielli e aos meus filhos Arthur e Otávio pela compreensão nos momentos em que fui ausente e por me encorajarem em todas as situações.

Agradeço aos meus pais, Ernestino e Vera, pelo apoio incondicional que foi fundamental para a concretização deste curso.

Agradeço à Professora Dra. Maristela Oliveira dos Santos pela orientação, amizade e compreensão.

Agradeço ainda, aos colegas que colaboraram comigo nessa jornada. Ao Márcio Arantes pelas aulas de programação e agradáveis conversas, ao Junior Ribeiro pela revisão deste trabalho, aos casais Flaviana Moreira e Raimundo e Vanessa Bezerra e Wesley e ao Valdemar Devesse pela hospitalidade, ao Artur Cunha e Aline Leão pelas aulas sobre testes computacionais.

*“O SENHOR é o meu pastor: nada me faltará.
Deitar-me faz em verdes pastos, guia-me mansamente a águas tranquilas.
Refrigera a minha alma; guia-me pelas veredas da justiça, por amor do seu nome.”
(Salmo 23 - Bíblia Sagrada)*

RESUMO

SOLER, W. A. O. **Análise, proposição e solução de modelos para o problema integrado de dimensionamento de lotes e sequenciamento da produção.** 2018. 131 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

Esta tese aborda um problema de dimensionamento e sequenciamento de lotes de produção baseado em uma indústria alimentícia brasileira que opera por meio de diversas linhas de produção heterogêneas. Nesse ambiente produtivo, as linhas de produção compartilham recursos escassos, tais como, trabalhadores e máquinas e devem ser montadas (ativadas) em cada período produtivo, respeitando-se a capacidade disponível de cada recurso necessário para ativação das mesmas. Modelos de programação matemática inteira mista são propostos para representação do problema, bem como diversos métodos heurísticos de solução, compreendendo procedimentos construtivos e de melhoramento baseados na formulação matemática do problema e heurísticas lagrangianas. São propostas heurísticas do tipo *relax-and-fix* explorando diversas partições das variáveis binárias dos modelos e uma heurística baseada na decomposição do modelo para construção de soluções. Procedimentos do tipo *fix-and-optimize* e *matheuristics* do tipo *iterative MIP-based neighbourhood search* são propostos para o melhoramento das soluções iniciais obtidas pelos procedimentos construtivos. Testes computacionais são realizados com instâncias geradas aleatoriamente e mostram que os métodos propostos são capazes de oferecer melhores soluções do que o algoritmo *Branch-and-Cut* de um resolvidor comercial para instâncias de médio e grande porte.

Palavras-chave: Dimensionamento e sequenciamento de lotes, Programação matemática inteira mista, Heurísticas, Relaxação lagrangiana.

ABSTRACT

SOLER, W. A. O. **Analysis, proposition and solution of models for the simultaneous lot sizing and scheduling problem.** 2018. 131 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

This doctoral dissertation addresses the simultaneous lot sizing and scheduling problem in a real world production environment where production lines share scarce production resources. Due to the lack of resources, the production lines cannot operate all simultaneously and they need to be assembled in each period respecting the capacity constraints of the resources. This dissertation presents mixed integer programming models to deal with the problem as well as various heuristic approaches: constructive and improvement procedures based on the mathematical formulation of the problem and lagrangian heuristics. Relax-and-fix heuristics exploring some partitions of the set of binary variables of a model and a decomposition based heuristic are proposed to construct solutions. Fix-and-optimize heuristics and iterative MIP-based neighbourhood search matheuristics are proposed to improvement solutions obtained by constructive procedures. Computational tests are performed with randomly instances and show that the proposed methods can find better solutions than the Branch-and-Cut algorithm of a commercial solver for medium and large size instances.

Keywords: Lot sizing and scheduling problem, Mixed integer programming, Heuristics, Lagrangian relaxation.

LISTA DE ILUSTRAÇÕES

Figura 1 – Esquema de funcionamento de uma heurística <i>relax-and-fix</i>	36
Figura 2 – Esquema de funcionamento de uma heurística lagrangiana	49
Figura 3 – Representação do ambiente industrial estudado	53
Figura 4 – Comparação entre o PDSL em linhas paralelas e o PDSL-LPR	65
Figura 5 – Esquema de funcionamento da heurística FOTO	73
Figura 6 – Um esquema de funcionamento das abordagens lagrangianas propostas . . .	92
Figura 7 – Comparação entre o PDSL e o PDSL-LPR - foram consideradas as instâncias da classe 4 e do grupo padrão	99
Figura 8 – Sensibilidade quanto aos custos para aberturas de linhas	100
Figura 9 – Sensibilidade frente a alterações no número de linhas de produção	100
Figura 10 – Sensibilidade com relação ao número de itens que podem ser produzidos numa mesma linha e período (parâmetro Ψ)	101
Figura 11 – Sensibilidade referente a alterações nas demandas dos clientes	102
Figura 12 – Sensibilidade com relação ao número de itens considerados	102
Figura 13 – Impacto da utilização da regra R1 num algoritmo <i>Branch-and-Bound</i> para resolver o modelo CLSDPRL	106
Figura 14 – <i>Box Plot</i> do desvio médio dos procedimentos de melhoria	114
Figura 15 – Comparação entre os principais métodos de solução.	121

LISTA DE ALGORITMOS

Algoritmo 1 – Heurística D	70
Algoritmo 2 – Heurística FOALE2	74
Algoritmo 3 – <i>Matheuristic</i> INS	76
Algoritmo 4 – Fase de busca local INS	77
Algoritmo 5 – Cálculo do parâmetro γ (passo P7 do método do volume - problema D^{LH})	90
Algoritmo 6 – Cálculo do parâmetro γ (passo P7' do método do volume - problema D^{LD})	91

LISTA DE TABELAS

Tabela 1 – Parâmetros e variáveis para modelo GLSPRL	57
Tabela 3 – Exemplo Ilustrativo - demandas dos clientes	64
Tabela 4 – Parâmetros utilizados nas instâncias do Conjunto 1	95
Tabela 6 – Conjunto 2: descrição das instâncias de teste	97
Tabela 8 – Resultados Computacionais - Classe 2	104
Tabela 10 – Regras de ramificação	108
Tabela 12 – Resultados computacionais: Heurísticas construtivas	111
Tabela 14 – Tempos de execução	113
Tabela 15 – Procedimentos de melhoria	114
Tabela 17 – Heurísticas lagrangianas	118
Tabela 19 – Comparação entre as principais abordagens de solução.	120

SUMÁRIO

1	INTRODUÇÃO	23
1.1	Objetivos	25
1.2	Contribuições e organização do texto	26
2	REVISÃO BIBLIOGRÁFICA	27
2.1	Problema integrado de dimensionamento e sequenciamento de lotes (PDSL)	27
2.2	Abordagens de solução	34
2.2.1	<i>Heurísticas baseadas em modelo</i>	34
2.2.2	<i>Heurísticas Lagrangianas</i>	40
2.2.2.1	<i>Esquemas de relaxação</i>	42
2.2.2.2	<i>O problema dual lagrangiano</i>	43
2.2.2.3	<i>Heurística lagrangiana</i>	48
3	DEFINIÇÃO E MODELAGEM DO PROBLEMA EM ESTUDO . . .	51
3.1	O setor produtivo	51
3.2	O problema de planejamento de produção	52
3.3	Modelagem Matemática	55
3.3.1	<i>Modelo GLSPRL</i>	55
3.3.2	<i>Modelo CCPRL</i>	59
3.3.3	<i>Modelo CCFLPRL</i>	61
3.3.4	<i>Modelo CLSDPRL</i>	62
3.4	Um exemplo ilustrativo	64
4	MÉTODOS HEURÍSTICOS DE SOLUÇÃO	67
4.1	Heurísticas Construtivas	67
4.1.1	<i>Uma heurística baseada em decomposição</i>	67
4.1.2	<i>Heurísticas Relax-and-Fix</i>	70
4.2	Heurísticas Fix-and-Optimize	72
4.3	<i>Matheuristic INS</i>	75
5	RELAXAÇÃO LAGRANGIANA	81
5.1	Esquema de decomposição lagrangiana (LD)	82
5.2	Esquema de relaxação de restrições (LH)	83

5.3	Resolvendo o problema dual lagrangiano	86
5.3.1	<i>Método do subgradiente</i>	86
5.3.2	<i>Algoritmo do volume</i>	88
5.4	Procedimento de factibilização da solução	91
6	RESULTADOS COMPUTACIONAIS	93
6.1	Instâncias	93
6.1.1	<i>Instâncias para análise de sensibilidade</i>	93
6.1.2	<i>Instâncias para análise de desempenho</i>	95
6.2	Ambiente de teste	97
6.3	Análise dos modelos	97
6.3.1	<i>Análise de sensibilidade dos modelos</i>	97
6.3.2	<i>Análise de desempenho dos modelos</i>	103
6.4	Regras de ramificação para acelerar a convergência de algoritmos do tipo <i>Branch-and-Bound</i>	104
6.5	Procedimentos de solução	107
6.5.1	<i>Heurísticas construtivas</i>	108
6.5.2	<i>Heurísticas de melhoria e matheuristics</i>	112
6.5.3	<i>Heurísticas lagrangianas</i>	115
6.5.3.1	<i>Determinação de parâmetros</i>	115
6.5.3.2	<i>Resultados</i>	116
6.6	Análise comparativa dos principais procedimentos de solução	118
7	CONCLUSÕES E PROPOSTAS PARA TRABALHOS FUTUROS .	123
	REFERÊNCIAS	127

INTRODUÇÃO

Problemas de planejamento da produção têm chamado a atenção de diversos pesquisadores, dada a sua importância para a economia global e pela dificuldade enfrentada pelas indústrias na elaboração de bons planejamentos ([Glock, Grosse e Ries \(2014\)](#)). Para [Almada-Lobo *et al.* \(2015\)](#), o planejamento da produção em ambientes industriais visa proporcionar a alocação eficiente dos recursos escassos de produção garantindo o atendimento das necessidades dos clientes.

O planejamento da produção pode envolver a determinação dos tamanhos dos lotes de produtos a serem produzidos ao longo de um dado horizonte de planejamento com o intuito de atender as demandas dos clientes e minimizar os custos incorridos ([Guimarães, Klabjan e Almada-Lobo \(2013\)](#)). Segundo [Copil *et al.* \(2017\)](#), as demandas de clientes por produtos são usualmente associadas a períodos (datas de entrega) e geralmente são resultado de previsões ou de solicitações prévias efetuadas pelos clientes.

Caso ocorram produções excedentes ou a produção das demandas dos produtos sejam realizadas de maneira antecipada, os produtos necessitam ser armazenados em locais apropriados, implicando em custos de estocagem que podem estar relacionados ao controle de temperatura, utilização de espaço físico, entre outros. Além disso, para [Glock, Grosse e Ries \(2014\)](#), os níveis e estrutura de estoque afetam diretamente o atendimento aos clientes com relação à disponibilidade do produto e velocidade de entrega. Dessa forma, um correto gerenciamento do estoque pode auxiliar na redução dos custos e no aumento do nível de satisfação dos clientes.

Em muitos ambientes produtivos, ao interromper a produção de um lote de um dado produto para iniciar a produção de outro item numa mesma linha de produção, fazem-se necessários alguns procedimentos de preparação para viabilizar a produção do lote subsequente, tais como ajustes e limpezas nas máquinas. Estas preparações consomem tempo de produção, que também pode ser traduzido em um custo ([Guimarães, Klabjan e Almada-Lobo \(2014\)](#)).

Normalmente, a produção de grandes lotes tende a minimizar os custos com as prepara-

ções para produção, mas pode aumentar os custos de estocagem, enquanto que a produção de lotes pequenos tende a minimizar os custos de estoque, porém podendo aumentar os custos com as preparações. Assim, torna-se interessante determinar os tamanhos ótimos dos lotes a serem produzidos com o intuito de minimizar a soma dos custos totais envolvidos.

O problema de determinar o quanto se produzir de cada produto em cada período de um horizonte de planejamento garantindo o atendimento das demandas dos clientes a um custo total mínimo é conhecido na literatura como problema de dimensionamento de lotes (PDL).

Em diversos ambientes industriais, os custos e tempos incorridos com as preparações para produção dependem da sequência em que os lotes são produzidos, isto é, variam de acordo com os itens envolvidos. Nesse caso, torna-se interessante determinar, além dos tamanhos dos lotes, a ordem de produção dos itens em cada período produtivo (Almada-Lobo *et al.* (2015)). O problema que consiste em determinar, simultaneamente, o tamanho e o sequenciamento dos lotes de produção minimizando os custos totais envolvidos é conhecido na literatura como problema integrado de dimensionamento e sequenciamento de lotes (PDSL).

Segundo Copil *et al.* (2017), o PDSL tem atraído a atenção de diversos pesquisadores, tanto da academia quanto vinculados a setores produtivos, de modo que diversos modelos matemáticos (Haase (1996), Wolsey (1997), Fleischmann e Meyr (1997) e Haase e Kimms (2000)) e métodos de solução (Almada-Lobo *et al.* (2007), James e Almada-Lobo (2011), Meyr e Mann (2013), Xiao *et al.* (2013), Xiao *et al.* (2015), Toledo *et al.* (2015) e Furlan *et al.* (2015a)) têm sido propostos na literatura para sua representação e solução.

Publicações recentes abordando o PDSL foram motivadas por observações de processos industriais adotados em diferentes ambientes de produção, tais como Ferreira, Morabito e Rangel (2009) que lidaram com indústrias de refrigerantes, Furlan *et al.* (2015b) que observaram indústrias de papel e celulose e Xiao *et al.* (2015) que abordaram indústrias de semicondutores. Em particular, esta tese aborda um problema de dimensionamento e sequenciamento de lotes com preparação para produção dependente da sequência, inspirado no sistema de produção de indústrias alimentícias brasileiras.

No ambiente produtivo analisado, existem diversas linhas de produção que operam de modo paralelo e que compartilham dos mesmos recursos limitados de produção: trabalhadores e máquinas. As linhas de produção, que correspondem a um conjunto de máquinas e estações de trabalho munidas de um fluxo produtivo bem definido, não são fixas, mas devem ser montadas (ativadas) em cada período (dia de trabalho) de acordo com as necessidades de produção. A escassez dos recursos produtivos e as características da demanda impedem que todas as linhas de produção sejam montadas/ativadas simultaneamente, fazendo-se necessário determinar, a cada período, quais linhas de produção serão ativadas.

As linhas de produção são especializadas, de modo que para cada produto existe uma única linha de produção apta a produzi-lo. Assim, a decisão dos itens a serem produzidos em

cada período depende das linhas que foram ativadas. Os itens produzidos são perecíveis, podendo permanecer em estoque por um tempo limitado e fazendo com que os custos de estocagem sejam significantes, uma vez que os itens precisam ser armazenados em locais com temperatura monitorada.

Define-se o problema de dimensionamento e sequenciamento de lotes em linhas paralelas e relacionadas (PDSL-LPR) com base no sistema de produção descrito, consistindo em determinar as linhas de produção a serem ativadas em cada período bem como decidir o quanto se produzir de cada item em cada linha e a sequência de produção, com o objetivo de minimizar os custos de estoque, preparação (dependente da sequência) e de abertura de linhas de produção, atendendo as demandas dos clientes, respeitando as capacidades produtivas e as quantidades disponíveis de cada recurso necessário para a ativação das linhas de produção.

A principal diferença entre o PDSL tradicional em linhas paralelas (considerado por [Almeder e Almada-Lobo \(2011\)](#), [Meyr e Mann \(2013\)](#) e [Xiao *et al.* \(2015\)](#)) e o PDSL-LPR é a necessidade de determinar, a cada período, as linhas de produção a serem ativadas, respeitando as capacidades disponíveis dos recursos necessários para a ativação das mesmas. Num primeiro momento, é natural questionar: por que não se adquirir as quantidades necessárias de recursos de modo a manter todas as linhas de produção ativas durante todos os períodos produtivos? Existem duas respostas para esta questão:

- i) a aquisição de novos recursos requer um investimento infactível, uma vez que a fábrica precisa ser ampliada, diversas máquinas precisam ser adquiridas e trabalhadores precisam ser contratados; e
- ii) se todas as linhas de produção permanecerem ativas durante todo o tempo, então existirão máquinas e trabalhadores ociosos, porque nesse tipo de mercado não existe demanda suficiente para manter todas as linhas trabalhando durante todo o tempo e altas produções podem aumentar significativamente os custos de estoque e o número de itens deteriorados pelo prazo de validade.

1.1 Objetivos

O objetivo principal desta tese é desenvolver métodos matemáticos e/ou computacionais para lidar com o PDSL-LPR, possibilitando a confecção de bons planejamentos de produção em tempo computacional aceitável. Para tanto, são traçados os seguintes objetivos específicos:

- i. Estudar trabalhos da literatura que abordaram o PDSL, visando identificar oportunidades de modelagem e solução do PDSL-LPR;
- ii. Obter uma formulação matemática consistente para o problema em estudo, por meio da extensão de abordagens da literatura;

- iii. Desenvolver e propor métodos de solução (heurísticos e/ou exatos) para o PDSL-LPR;
- iv. Realizar testes computacionais a fim de verificar a eficiência dos métodos de solução obtidos.

1.2 Contribuições e organização do texto

As principais contribuições desse trabalho são as seguintes:

- i. Descrição do PDSL-LPR e extensão de quatro modelos da literatura para representação do mesmo;
- ii. Proposição de diversas heurísticas (construtivas, de melhoramento e lagrangianas) para solucionar o PDSL-LPR e que podem ser estendidas para o tradicional PDSL;
- iii. Apresentação de regras de ramificação para acelerar a convergência de algoritmos do tipo *Branch-and-Cut* para a solução do PDSL-LPR.

A seguir, no Capítulo 2, apresenta-se uma revisão da literatura contemplando os principais modelos matemáticos de otimização inteira mista utilizados para representação do PDSL, bem como alguns procedimentos de solução que inspiraram os métodos que serão propostos nesta tese. No Capítulo 3 são propostos quatro modelos matemáticos de otimização linear inteira mista para representação do problema, enquanto que, no Capítulo 4 são propostas quatro heurísticas construtivas e diversas heurísticas de melhoramento baseadas na formulação matemática do problema e no Capítulo 5 são introduzidas duas heurísticas baseadas em relaxação lagrangiana. Os resultados computacionais são apresentados no Capítulo 6 e, por fim, as conclusões e propostas para trabalhos futuros são apontadas no Capítulo 7.

REVISÃO BIBLIOGRÁFICA

Este Capítulo apresenta uma revisão bibliográfica com foco em modelos e métodos de solução que foram propostos na literatura para lidar com o PDSL. Além disso, são apresentados alguns métodos que foram propostos para solucionar outros problemas de otimização linear inteira mista, os quais serão utilizados neste trabalho.

A seguir, na Seção 2.1, descreve-se o PDSL com ênfase nos principais modelos utilizados para a sua representação, enquanto que na Seção 2.2 apresentam-se algumas abordagens de solução, como heurísticas baseadas em modelo e relaxação lagrangiana.

2.1 Problema integrado de dimensionamento e sequenciamento de lotes (PDSL)

Para [Karimi, Ghomi e Wilson \(2003\)](#), decidir corretamente sobre quando e o quanto produzir de cada produto afeta diretamente a performance e a produtividade de sistemas de produção, contribuindo para que as indústrias mantenham-se competitivas no mercado. Nesse sentido, o desenvolvimento de ferramentas capazes de auxiliar as empresas em suas decisões é de fundamental importância.

O problema de determinar o tamanho de lotes de produção a serem produzidos em cada período de um horizonte de planejamento, garantindo o atendimento das demandas dos clientes e objetivando minimizar os custos envolvidos na produção é conhecido na literatura como problema de dimensionamento de lotes (PDL). De acordo com [Glock, Grosse e Ries \(2014\)](#), uma publicação de Ford Whitman Harris (1913) marcou o início do estudo desse problema na literatura e, desde então, o PDL tem recebido atenção de pesquisadores ligados a academia e a setores produtivos devido à sua importância na economia global.

Um problema de dimensionamento de lotes pode ser classificado de acordo com as

características consideradas, tais como número de itens (único ou vários), tipo do horizonte de planejamento (finito ou infinito), número de unidades produtoras (única ou múltiplas), número de estágios de produção (único ou múltiplos estágios), existência ou não de limite de capacidade produtiva, natureza dos itens produzidos (perecíveis ou não) e natureza da estrutura dos custos e tempos de preparação para produção (independente ou dependente da sequência) (Karimi, Ghomi e Wilson (2003)).

Usualmente, ao interromper a produção de um produto para iniciar a produção de outro item, é necessário realizar alguns procedimentos de preparação, como limpeza e ajustes de máquinas. Esses procedimentos consomem tempo de produção que também pode ser traduzido num determinado custo. Quando os tempos e/ou custos decorrentes desses procedimentos de preparação dependem apenas do produto que será produzido, tem-se uma estrutura de preparação dita independente da sequência. Por outro lado, quando os tempos e/ou custos de preparações dependem também do item previamente produzido, tem-se uma estrutura dita *dependente da sequência*.

De acordo com Almada-Lobo *et al.* (2015), em sistemas produtivos com estrutura de preparação dependente da sequência, além de se determinar o quanto produzir de cada item em cada período, é necessário determinar, simultaneamente, a sequência de produção dos lotes dentro de cada período, uma vez que, quando os problemas de dimensionamento e de sequenciamento são resolvidos de maneira desvinculada, soluções subótimas ou ineficazes podem ser obtidas, pois as quantidades de produção determinadas pelo modelo de dimensionamento podem ser muito elevadas, inviabilizando o sequenciamento dos lotes dentro dos limites de capacidade.

O problema que consiste em determinar, simultaneamente, o quanto se produzir de cada produto e a sequência de produção dos lotes em cada período de um horizonte de planejamento é conhecido na literatura como problema de dimensionamento e sequenciamento de lotes (PDSL), o qual tem recebido uma elevada atenção de pesquisadores desde a década de 1990, que propuseram diversos modelos e estratégias de solução.

Copil *et al.* (2017) afirmam que a maioria dos modelos propostos na literatura para o PDSL são extensões de cinco modelos clássicos: *Discrete lot sizing and scheduling problem* (DLSP) proposto em Fleischmann (1990), *Continuous Setup lot sizing problems* (CLSP) proposto em Karmarkar e Schrage (1985), *Proportional lot sizing problem* (PLSP) proposto em Drexel e Haase (1995), *Capacitated Lot Sizing and Scheduling Problem with Sequence-Dependent Setups* (CLSD) proposto em Haase (1996) e o *General Lot Sizing and Scheduling Problem* (GLSP) proposto em Fleischmann e Meyr (1997). O GLSP pode ser visto como uma generalização dos demais, de modo que as características dos outros modelos podem ser obtidas a partir do modelo GLSP por meio da incorporação de restrições e/ou adaptações nos dados de entrada (Copil *et al.* (2017)).

Originalmente, todos esses modelos consideram um sistema produtivo com único estágio, onde diversos itens são produzidos numa única máquina com capacidade limitada, não permitindo

atrasos no atendimento das demandas (*backlogging*) e considerando que cada procedimento de preparação deve ser realizado dentro de um único período do horizonte de planejamento, isto é, não se permite o chamado *setup crossover*.

No modelo DLSP admite-se a produção de um único produto em cada período do horizonte de planejamento e adota-se a hipótese “tudo ou nada”, isto é, em cada período, toda a capacidade produtiva é utilizada na produção de um único item ou não existe produção. Desse modo, as quantidades a serem produzidas de cada produto não são variáveis do problema, sendo necessário apenas associar os itens aos períodos produtivos. O modelo CSLP difere do DLSP por não adotar a hipótese “tudo ou nada”. Já o modelo PLSP admite uma única troca de produto em cada período, de modo que, o primeiro item a ser produzido é o último item produzido no período anterior e uma troca de produto pode ser realizada a qualquer momento de um período. Dessa forma, tem-se que $z_{DLSP}^* \geq z_{CSLP}^* \geq z_{PLSP}^*$, onde z_{PLSP}^* , z_{CSLP}^* e z_{DLSP}^* são os valores ótimos obtidos pelos modelos PLSP, CLSP e DLSP, respectivamente, ao resolver uma mesma instância (Drexler e Haase (1995)).

Os modelos CLSD e GLSP são mais gerais do que os modelos DLPS, CSLP e PLSP, pois permitem a produção de diversos itens dentro de um mesmo período. Esses modelos diferem entre si basicamente pela estratégia utilizada na modelagem do sequenciamento da produção. No CLSD, o sequenciamento dos lotes é feito com o auxílio de variáveis e restrições oriundas de modelos para o problema do caixeiro viajante (*travelling salesman problem*), enquanto que no modelo GLSP, cada período do horizonte de planejamento é subdividido em microperíodos de tamanhos variáveis, onde um único item pode ser produzido. Assim, no GLSP, ao determinar os itens que serão produzidos em cada microperíodo, obtém-se, diretamente, o sequenciamento da produção.

Suponha que o horizonte de planejamento tenha sido dividido em S microperíodos e seja P_t o conjunto formado pelos microperíodos (consecutivos) que compõem o período t . Mais especificamente, será considerado $P_t = \{f_t, \dots, l_t\}$, onde f_t e l_t são o primeiro e o último microperíodo associado ao período t , respectivamente.

Considere os seguintes parâmetros gerais para definição de uma instância do PDSL.

T : número de períodos do horizonte de planejamento (indexados por t);

J : número de produtos (indexados por i, j);

d_{jt} : demanda do produto j durante o período t ;

sc_{ij} : custo de preparação para produção gerado pela troca de produção do item i para o item j ;

st_{ij} : tempo de preparação para produção gerado pela troca de produção do item i para o item j ;

C_t : capacidade de produção (em tempo) do período t ;

a_j : tempo consumido para produção de uma unidade do item j ;

h_j : custo unitário para manutenção do item j em estoque;

m_j : quantidade mínima a ser produzida caso a linha de produção seja configurada para a produção do item j .

Para definir o modelo GLSP, utilizam-se as seguintes variáveis de decisão:

x_{js} : quantidade a ser produzida do item j durante o microperíodo s ;

y_{js} : 1, se a linha de produção está configurada para produzir o item j durante o microperíodo s e 0, caso contrário;

z_{ijs} : 1, se existe troca de produção, do produto i para o produto j , durante o microperíodo s e 0, caso contrário;

I_{jt} : quantidade em estoque do item j ao final do período t .

O modelo GLSP, considerando tempos e custos de preparação dependentes da sequência, é caracterizado pela função objetivo (2.1) e pelas restrições (2.2) - (2.11).

$$\text{Min} \quad \sum_j \sum_t h_j I_{jt} + \sum_i \sum_j \sum_s sc_{ij} z_{ijs} \quad (2.1)$$

$$\text{s.a.} \quad I_{j,t-1} + \sum_{s \in P_t} x_{js} = d_{jt} + I_{jt}, \quad \forall j, t; \quad (2.2)$$

$$\sum_j \sum_{s \in P_t} x_{js} + \sum_i \sum_j \sum_{s \in P_t} st_{ij} z_{ijs} \leq C_t, \quad \forall t; \quad (2.3)$$

$$x_{js} \leq \frac{C_t}{a_j} y_{js}, \quad \forall j, s; \quad (2.4)$$

$$x_{js} \geq m_j (y_{js} - y_{j,s-1}), \quad \forall j, s; \quad (2.5)$$

$$\sum_j y_{js} = 1, \quad \forall s; \quad (2.6)$$

$$z_{ijs} \geq y_{i,s-1} + y_{js} - 1, \quad \forall i, j, s; \quad (2.7)$$

$$I_{jt} \geq 0, \quad \forall j, t; \quad (2.8)$$

$$z_{ijs} \geq 0, \quad \forall i, j, s; \quad (2.9)$$

$$x_{js} \geq 0, \quad \forall j, s; \quad (2.10)$$

$$y_{js} \in \{0, 1\}, \quad \forall j, s. \quad (2.11)$$

A função objetivo (2.1) representa a soma dos custos de estoque com os custos de preparação para produção. As restrições (2.2) são responsáveis pelo balanceamento de estoque e, em conjunto com as restrições (2.8) e (2.10), garantem que as demandas serão completamente atendidas sem atrasos. As restrições (2.3) garantem que os limites de capacidade da linha de produção serão respeitados em cada período do horizonte de planejamento e as restrições (2.4)

garantem que cada item j só pode ser produzido em s se a linha de produção estiver preparada para j em s . As restrições (2.5) introduzem um lote mínimo de produção, isto é, caso a linha seja preparada para a produção de um item j num dado microperíodo s , então, pelos menos m_j unidades deverão ser produzidas. As restrições (2.6) garantem que a linha de produção está sempre apta para a produção de algum produto, enquanto que, as trocas entre produtos na linha de produção são capturadas pelas restrições (2.7). Finalmente, as restrições (2.8) a (2.11) definem os domínios das variáveis de decisão.

Fleischmann e Meyr (1997) destacam que o modelo GLSP possui uma redundância indesejável. Se uma sequência de microperíodos dentro de um mesmo período é designada à produção de um mesmo item, então a quantidade total produzida pode ser arbitrariamente distribuída entre esses microperíodos, produzindo diferentes soluções com o mesmo sequenciamento e, consequentemente, com o mesmo valor para a função objetivo. As restrições (2.12) e (2.13) foram propostas em Fleischmann e Meyr (1997) a fim de evitar a ocorrência dessa redundância. Essas restrições também são capazes de acelerar o desempenho computacional do modelo ao utilizarmos um algoritmo *Branch-and-Bound* ou *Branch-and-Cut* para solucioná-lo.

$$\sum_i \sum_{j \neq i} z_{ij,s-1} \geq \sum_i \sum_{j \neq i} z_{ijs}, \quad \forall t, s = f_t + 2, \dots, l_t; \quad (2.12)$$

$$x_{js} \leq \frac{C_t}{a_j} \left(2 - \sum_i z_{ij,s-1} - z_{jjs} \right), \quad \forall j, t, s = f_t + 1, \dots, l_t. \quad (2.13)$$

As restrições (2.12) forçam que os micro períodos não produtivos sejam alocados no final de cada período, enquanto que as restrições (2.13) permitem que apenas o primeiro micro período associado a cada lote de produção alocado dentro de um mesmo período seja produtivo.

Em Clark e Clark (2000) foi proposta uma reformulação do modelo GLSP com o objetivo de obter melhores limitantes duais para o problema via resolução da relaxação linear do modelo. A reformulação de Clark e Clark (2000) é caracterizada por suprimir as variáveis de estado de preparação (y_{js}). Note que a linha de produção estará preparada para o item j num dado microperíodo s , $y_{js} = 1$ se, e somente se, uma troca de produção é realizada saindo de um produto qualquer (podendo ser, inclusive, o produto j) e chegando no item j . Assim, tem-se a equação (2.14).

$$y_{js} = \sum_i z_{ijs}, \quad \forall j, s; \quad (2.14)$$

O modelo de Clark e Clark (2000) pode ser obtido substituindo-se as variáveis $y_{js}, \forall j, s$ do modelo GLSP de acordo com a equação (2.14) e declarando as variáveis de mudança de produção $z_{ijs}, \forall i, j, s$ como binárias. Assim, as restrições (2.4) a (2.7) e (2.9) são substituídas pelas restrições (2.15) a (2.19), respectivamente.

$$x_{js} \leq \frac{C_t}{a_j} \sum_i z_{ijs}, \quad \forall j, s; \quad (2.15)$$

$$x_{js} \geq m_j \sum_{i, i \neq j} z_{ijs}, \quad \forall j, s; \quad (2.16)$$

$$\sum_i \sum_j z_{ijs} = 1, \quad \forall s; \quad (2.17)$$

$$\sum_i z_{ij, s-1} = \sum_i z_{jis}, \quad \forall j, s; \quad (2.18)$$

$$z_{ijs} \in \{0, 1\}, \quad \forall i, j, s. \quad (2.19)$$

Desse modo, o modelo de [Clark e Clark \(2000\)](#) consiste em minimizar a função objetivo (2.1), sujeito as restrições (2.2), (2.3), (2.15) a (2.19), (2.8) e (2.10).

O modelo CLSD, na forma proposta em [Haase \(1996\)](#), utiliza as restrições MTZ, propostas em [Miller, Tucker e Zemlin \(1960\)](#) para a eliminação de subciclos no problema do caixeiro viajante, para a realização do sequenciamento da produção. As variáveis utilizadas no modelo CLSD são as seguintes:

x_{jt} : quantidade produzida do item j durante o período t ;

y_{jt} : 1, se o item j é o primeiro item a ser produzido no período t e 0, caso contrário;

z_{ijt} : 1, se existe troca de produção do item i para o item j durante o período t e 0, caso contrário;

V_{jt} : variável auxiliar utilizada nas restrições para eliminação de subciclos.

O modelo CLSD é definido pela função objetivo (2.20) e pelas restrições (2.21) a (2.30).

$$\text{Min} \quad \sum_j \sum_t h_j I_{jt} + \sum_i \sum_j \sum_t sc_{ij} z_{ijt} \quad (2.20)$$

$$\text{s.a.} \quad I_{j,t-1} + x_{jt} = d_{jt} + I_{jt}, \quad \forall j, t; \quad (2.21)$$

$$\sum_j a_j x_{jt} + \sum_i \sum_j st_{ij} z_{ijt} \leq C_t, \quad \forall t; \quad (2.22)$$

$$x_{jt} \leq \frac{C_t}{a_j} \left(y_{jt} + \sum_i z_{ijt} \right), \quad \forall j, t; \quad (2.23)$$

$$x_{js} \geq m_j \left(y_{jt} + \sum_i z_{ijt} \right), \quad \forall j, t; \quad (2.24)$$

$$\sum_j y_{jt} = 1, \quad \forall t; \quad (2.25)$$

$$y_{jt} + \sum_i z_{ijt} = \sum_i z_{jit} + y_{j,t+1}, \quad \forall i, j, t; \quad (2.26)$$

$$V_{jt} \geq V_{it} + 1 - J(1 - z_{ijt}), \quad \forall i, j, t; \quad (2.27)$$

$$x_{jt}, I_{jt}, V_{jt} \geq 0, \quad \forall j, t; \quad (2.28)$$

$$y_{jt} \in \{0, 1\}, \quad \forall j, t; \quad (2.29)$$

$$z_{ijt} \in \{0, 1\}, \quad \forall i, j, t. \quad (2.30)$$

A função objetivo (2.20) representa a soma dos custos de estoque e preparação a serem minimizados, enquanto que as restrições (2.21) são responsáveis pelo balanceamento de estoque e as restrições (2.22) referem-se aos limites de capacidade de produção em cada período. As restrições (2.23) garantem que um item j só poderá ser produzido no período t se a linha estiver preparada para a sua produção no mesmo período. As restrições (2.24) introduzem as quantidades mínimas a serem produzidas, enquanto que as restrições (2.25) garantem que um único item será o primeiro item produzido e as restrições (2.26) são restrições de fluxo oriundas do problema do caixeiro viajante. As restrições (2.27) são as restrições MTZ (introduzidas em Miller, Tucker e Zemlin (1960)) para eliminação de subciclos e as restrições (2.28) a (2.30) definem os domínios das variáveis de decisão.

Outras restrições podem ser utilizadas para eliminação de subciclos no lugar das restrições MTZ (2.27), tais como as propostas em Almada-Lobo *et al.* (2007). Uma comparação entre as performances do modelo CLSD utilizando diferentes restrições existentes na literatura para eliminação de subciclos pode ser encontrada em Guimarães, Klabjan e Almada-Lobo (2014).

Problemas reais oriundos de diversos setores produtivos têm sido modelados com base nas estruturas dos modelos CLSD e GLSP. Por exemplo, modelos baseados no GLSP foram utilizados para lidar com problemas em indústrias de metais (Araujo, Arenales e Clark (2008)), de refrigerantes (Ferreira, Morabito e Rangel (2009)) e de papel e celulose (Furlan *et al.* (2015a)), enquanto que extensões do modelo CLSD foram propostas para lidar com problemas de indústrias de recipientes de vidro (Almada-Lobo *et al.* (2007)), de iogurtes (Kopanos, Puigjaner e Maravelias (2010)) e de semicondutores (Xiao *et al.* (2015)).

De acordo com Copil *et al.* (2017), o modelo GLSP pode ser mais facilmente estendido para lidar com características de problemas reais, porém o modelo CLSD tem apresentado melhores performances computacionais quando resolvido por algoritmos do tipo *Branch-and-Bound* e/ou *Branch-and-Cut* de resolvedores comerciais. Testes computacionais realizados em Almeder e Almada-Lobo (2011) e em Guimarães, Klabjan e Almada-Lobo (2014) mostraram que o modelo CLSD é capaz de oferecer melhores limitantes.

Copil *et al.* (2017) afirmam que, em diversas aplicações práticas, recursos secundários escassos, podendo ser ferramentas, máquinas ou trabalhadores com funções específicas, precisam ter a sua utilização sincronizada entre diversas linhas produção a fim de possibilitar a fabricação de produtos. Essa sincronização de recursos foi considerada, por exemplo, em Almeder e Almada-Lobo (2011), onde variáveis e restrições foram incluídas nos modelos GLSP e CLSD a fim de capturar os tempos de início e fim de utilização de cada recurso em cada linha de produção.

No problema considerado nesta tese, trabalhadores e diversas máquinas representam recursos secundários que precisam ser utilizados para a ativação de diversas linhas de produção. Porém, diferentemente do que foi considerado em Almeder e Almada-Lobo (2011), esses recursos são alocados nas linhas de produção apenas no início de cada período, permanecendo vinculados às respectivas linhas até o fim do mesmo. Dessa forma, no problema aqui abordado,

não existe a necessidade de capturar os tempos de início e fim de utilização de cada recurso. Assim, variáveis binárias foram incluídas nos modelos CLSD e GLSP apenas para indicar se cada linha de produção é ou não ativada em cada período produtivo. Restrições oriundas do problema da mochila multidimensional foram utilizadas para garantir que as quantidades utilizadas de cada recurso em cada período do horizonte não excedam as quantidades disponíveis.

2.2 Abordagens de solução

O PDSL com capacidade limitada pertence à classe de problemas NP-difícil (James e Almada-Lobo (2011)), de modo que instâncias de grande porte não podem ser satisfatoriamente resolvidas via métodos exatos em tempo computacional aceitável. Assim, Copil *et al.* (2017) entendem que uma alternativa para lidar essas instâncias (algumas vezes, baseadas em dados reais) é a utilização de métodos heurísticos de solução. Em particular, procedimentos que combinam heurísticas com algoritmos exatos têm sido propostos na literatura motivados pelo surgimento de resolvidores comerciais de problemas de otimização linear inteira mista (MIP resolvidores) de alto desempenho, que conseguem solucionar instâncias de pequeno porte em baixo tempo computacional. Normalmente, esses procedimentos utilizam abordagens heurísticas para decompor o problema original em diversos subproblemas menores que são iterativamente solucionados utilizando MIP resolvidores.

Nesta Seção são apresentadas abordagens de solução que foram propostas na literatura combinando métodos exatos com heurísticas e/ou meta-heurísticas para solucionar problemas de programação inteira mista. A seguir, a Seção 2.2.1 descreve as heurísticas baseadas em modelos e, por fim, a Seção 2.2.2 aborda heurísticas baseadas em relaxação lagrangiana.

2.2.1 Heurísticas baseadas em modelo

De acordo com Sel e Bilgen (2014), as heurísticas baseadas na formulação matemática do problema (MIP heurísticas) mais utilizadas na literatura para solucionar problemas de planejamento da produção são as heurísticas *relax-and-fix* (RF) e *fix-and-optimize* (FO). A RF foi proposta em Dillenberger *et al.* (1994) e generalizada em Escudero e Salmeron (2005) com o objetivo de lidar com um problema de sequenciamento de produção. O objetivo da heurística RF é construir, iterativamente, uma solução factível para o problema e tem sido bastante utilizada como procedimento para obtenção de soluções iniciais visando alimentar procedimentos de melhoria. Exemplos de utilização da heurística RF podem ser encontrados em Ferreira, Morabito e Rangel (2009), James e Almada-Lobo (2011), Sel e Bilgen (2014) e Toledo *et al.* (2015).

Sejam P um modelo para um problema de otimização e V o conjunto formado por todas as variáveis inteiras (inclusive as binárias) do modelo P . Na sua forma mais simples, a RF opera por meio de uma partição finita $V^1, V^2, \dots, V^r$ do conjunto V , formada por conjuntos dois a dois disjuntos, cuja união é igual a V . O número r de subconjuntos da partição adotada define o

número de iterações da heurística, de modo que a cada iteração $i \in \{1, \dots, r\}$, um subproblema (SP^i) é resolvido onde as variáveis do conjunto V^i são consideradas como inteiras/binárias, as variáveis dos conjuntos V^j com $j < i$ são fixadas nos seus valores atuais (obtidos nas iterações prévias) e as variáveis dos conjuntos V^j com $j > i$ são linearmente relaxadas.

Se algum subproblema SP^i não possuir solução factível, então a heurística RF com a partição de variáveis adotada não é capaz de fornecer uma solução e o procedimento deve ser interrompido. Uma vez encontrada uma solução factível, o valor da função objetivo associado à solução encontrada é igual ao valor da função objetivo na última iteração do método.

Normalmente, são adotadas decomposições do conjunto das variáveis binárias do problema onde cada subproblema SP^i é suficientemente pequeno para ser solucionado via algoritmos do tipo *Branch-and-Cut* de MIP resolvidores em baixo tempo computacional.

De modo geral, a fim de se obter uma heurística RF basta definir uma decomposição do conjunto V e não, necessariamente, uma partição. Ou seja, os conjuntos $V^1, \dots, V^r$ não precisam ser dois a dois disjuntos. Quando isso ocorre, diz-se que o procedimento possui sobreposição. Nesse caso, nem todas as variáveis otimizadas numa iteração são fixadas na próxima iteração.

Para formalizar um procedimento com sobreposição, tome uma decomposição $V^1, \dots, V^r$ do conjunto V com $V^i \cap V^{i+1} \neq \emptyset, i \in \{1, \dots, r-1\}$. Mais especificamente, considere que cada conjunto V^i é dado por $V^i = F^i \cup L^i$, onde $F^i \cap L^i = \emptyset$ e $L^i \subset F^{i+1}$. Nesse caso, na iteração $i+1$ são fixados os valores das variáveis do conjunto $\bigcup_{j=1}^i F^j$, enquanto que as variáveis do conjunto $V^{i+1} = F^{i+1} \cup L^{i+1}$ são liberadas para otimização e as variáveis do conjunto $\bigcup_{j=i+2}^r V^j \setminus L^{i+1}$ são linearmente relaxadas. Note que as variáveis do conjunto L^i são liberadas para otimização tanto na iteração i como na iteração $i+1$ e terão o seu valor fixado na iteração $i+2$.

Numa heurística RF, se cada conjunto V^i possuir um número muito reduzido de variáveis, a solução encontrada pode não fornecer um bom valor para a função objetivo, enquanto que se os conjuntos V^i possuírem muitas variáveis, o tempo consumido pode ser bastante elevado. Assim, a decomposição adotada no conjunto das variáveis é de fundamental importância para o bom desempenho do método.

Inicialmente, [Dillenberger et al. \(1994\)](#) propuseram a heurística RF para solução de um problema de dimensionamento e sequenciamento da produção em linhas paralelas utilizando uma partição do conjunto das variáveis por períodos. Partições por períodos continuam sendo constantemente utilizadas na literatura (por exemplo, em [James e Almada-Lobo \(2011\)](#) e em [Sel e Bilgen \(2014\)](#)). Partições por máquinas (ou por estágios de produção) têm sido utilizadas em problemas com múltiplas máquinas (ou múltiplos estágios) e partições por produtos também têm sido largamente exploradas. Exemplos explorando essas partições podem ser encontrados em [Ferreira, Morabito e Rangel \(2009\)](#).

A Figura 1 apresenta o esquema de funcionamento das heurísticas RF, com e sem sobreposição de variáveis.

Figura 1 – Esquema de funcionamento de uma heurística *relax-and-fix*

A FO foi proposta em Pochet e Wolsey (2006) e trata-se de um procedimento de melhoria, isto é, que parte de uma solução factível inicial e tenta encontrar uma solução melhor de maneira iterativa. Mais especificamente, dados o problema (P) ((2.31) a (2.34)) e uma solução factível inicial $(\bar{x}, \bar{y})$ com valor associado $\bar{z}$, a cada iteração it , seleciona-se um subconjunto de índices V^{it} das variáveis binárias do modelo (x) e resolve-se o subproblema (SP^{it}), onde apenas as variáveis binárias com índices em V^{it} são (re)otimizadas, enquanto que as demais variáveis tem o seu valor fixado no valor incumbente. Sejam z^{it} e (x^{it}, y^{it}) , o valor e a solução encontrados para subproblema (SP^{it}), respectivamente. Se $z^{it} < \bar{z}$, então a solução e o valor incumbentes são atualizados, isto é, $(\bar{x}, \bar{y}) \leftarrow (x^{it}, y^{it})$ e $\bar{z} \leftarrow z^{it}$. Esses passos são repetidos até que algum critério de parada seja satisfeito.

$$\text{P: Min } f(x, y) \quad (2.31)$$

$$\text{s.a. } Ax + Cy = b \quad (2.32)$$

$$y \geq 0 \quad (2.33)$$

$$x \in \{0, 1\} \quad (2.34)$$

$$SP^{it} : \text{Min } (2.31) \quad (2.35)$$

$$\text{s.a. } (2.32) \text{ a } (2.34) \quad (2.36)$$

$$x_i = \bar{x}_i, \forall i \notin V^{it} \quad (2.37)$$

A escolha dos subconjuntos de variáveis que serão reotimizadas a cada iteração pode ser realizada de maneira determinística (como em Sel e Bilgen (2014)) ou de maneira aleatória

(como em [Chen \(2015\)](#)), não havendo a obrigatoriedade de se operar com uma partição disjunta do conjunto das variáveis. Quando essa escolha é feita de maneira aleatória, obtém-se um procedimento estocástico.

Num procedimento determinístico é possível prever um número máximo de iterações a serem realizadas, enquanto que, num procedimento estocástico o critério de parada mais usual é o atingimento do tempo limite de execução ou de um dado número de iterações sem melhoria no valor da função objetivo.

Normalmente, é desejável que durante um procedimento do tipo FO, todas as variáveis binárias sejam reotimizadas em alguma iteração, isto é, que $\bigcup_{it} V^{it} = V$, onde V é conjunto de todas as variáveis binárias do problema.

De acordo com [Caserta e Voß \(2009\)](#), meta-heurísticas são estratégias munidas de dispositivos para escapar de mínimos locais, que guiam outras heurísticas a fim de resolver um dado problema. Para [Sörensen e Glover \(2013\)](#), as meta-heurísticas são, em essência, procedimentos heurísticos, pois não é possível garantir que o procedimento converge para a solução ótima do problema tratado. Algumas meta-heurísticas são extensões de estratégias de busca local, como a *simulated annealing* (SA) que permite com determinada probabilidade, a realização de movimentos de uma solução para outra com “pior” valor para a função objetivo; *tabu search* (TS) que utiliza informações sobre o histórico de busca para guiar uma busca local; e *variable neighbourhood search* (VNS) que opera com diversas estruturas de vizinhança baseando-se no fato de que um ótimo local com relação a uma estrutura de vizinhança pode não ser um ótimo local com relação a outra estrutura ([Lazic \(2010\)](#)). Existem ainda, outras meta-heurísticas que não operam com busca em vizinhança, como os algoritmos genéticos.

Meta-heurísticas têm sido amplamente utilizadas na solução de diversos problemas de otimização como forma de obtenção de boas soluções em tempo computacional viável ([Meyr \(2002\)](#), [Shignemoto e Armentano \(2010\)](#), [Santos, Massago e Almada-Lobo \(2010\)](#), [Seeanner, Almada-Lobo e Meyr \(2013\)](#), [Furlan et al. \(2015a\)](#) e [Sifaleras, Konstantaras e Mladenović \(2015\)](#)), mostrando-se, algumas vezes, ser uma alternativa superior à utilização de métodos exatos para solução de grandes instâncias de problemas da classe NP-difícil ([Sörensen e Glover \(2013\)](#)).

Por outro lado, os recentes avanços computacionais motivaram a utilização de métodos exatos para solução de problemas práticos de otimização. Porém, a solução de instâncias de grande porte de problemas NP-difícil por intermédio desses métodos ainda é impraticável. Dessa forma, tem-se observado na literatura um crescente interesse por métodos que combinam princípios de meta-heurísticas com técnicas de programação matemática. Esse procedimentos são conhecidos como *matheuristics* e baseiam-se no fato de que pequenas instâncias de problemas difíceis podem ser rapidamente resolvidas utilizando técnicas de programação matemática. Assim, meta-heurísticas podem ser utilizadas, por exemplo, como uma estratégia para decompo-

sição do problema original num conjunto de pequenos problemas. Também é possível utilizar meta-heurísticas para acelerar a convergência de métodos exatos (Sörensen e Glover (2013)).

Em James e Almada-Lobo (2011) a *matheuristic iterative MIP-based neighbourhood search* (INS) foi proposta com o objetivo de encontrar soluções para problemas de dimensionamento e sequenciamento de lotes com tempos e custos de preparação para produção dependentes da sequência, considerando múltiplas linhas de produção não relacionadas. Os procedimentos desenvolvidos basearam-se no modelo CLSD.

Assim como na FO, a *matheuristic* INS parte de uma solução factível inicial e decompõe o problema original em diversos subproblemas menores onde apenas algumas variáveis binárias são reotimizadas. Esses subproblemas são iterativamente resolvidos via técnicas de programação matemática e as variáveis a serem reotimizadas a cada iteração são selecionadas de acordo regras pré-estabelecidas conhecidas como estruturas de vizinhanças.

A INS também se assemelha à meta-heurística *variable neighbourhood search* (VNS) (proposta em Mladenović e Hansen (1997)), pois se utiliza de diversas estruturas de vizinhança ($V_1, \dots, V_n$) e cada iteração é composta por duas fases: i) fase de busca local; e ii) fase de mudança de estrutura de vizinhança.

De acordo com a proposta realizada em James e Almada-Lobo (2011) para solução do PDSL, as estruturas de vizinhança utilizadas na INS são regras que permitem determinar, a cada iteração, as variáveis binárias do problema que serão reotimizadas. Em James e Almada-Lobo (2011) foram propostas duas estruturas de vizinhança, sendo a primeira baseada na decomposição do problema via períodos e a segunda via produtos. Na primeira estrutura proposta, um dado número $r \geq 1$ de períodos produtivos consecutivos $t_1, \dots, t_r$, são aleatoriamente selecionados a cada iteração e todas as variáveis binárias referentes a esses períodos são liberadas para a reotimização. A probabilidade inicial de cada período produtivo ser selecionado é a mesma. Porém, essa probabilidade diminui conforme a frequência em que cada período foi selecionado e quão recente foi a sua última seleção. Quanto maior o número de vezes que um dado período t foi selecionado durante a fase de busca local incumbente e quanto mais recente foi a última ocorrência, menor a probabilidade do período ser novamente selecionado. Na segunda estrutura de vizinhança, seleciona-se a cada iteração $s \geq 1$ produtos $(i_1, \dots, i_s)$ e todas as variáveis binárias contendo algum dos índices i_β ($\beta \in \{1, \dots, s\}$) são liberadas para reotimização.

A INS, na forma proposta em James e Almada-Lobo (2011), opera com maior ênfase na primeira estrutura de vizinhança (baseada em períodos). Quando todas as combinações de r períodos são testadas ou quando não se observa melhoria no valor incumbente para a função objetivo, passa-se para a próxima estrutura de vizinhança. Uma vez que uma solução melhor é encontrada, o esquema retorna à primeira estrutura de vizinhança.

Nesta tese, variações da INS considerando combinações de cinco estruturas de vizinhança e de quatro esquemas de interrupção da fase de busca local serão propostas como método de

solução para o PDSL-LPR.

Em [Guimarães, Klabjan e Almada-Lobo \(2013\)](#) foi proposta uma *matheuristic* para tratar o PDSL consistindo na utilização de princípios de geração de colunas (decomposição de Dantzig-Wolfe) dentro de heurísticas construtivas e de melhoramento baseadas na formulação matemática do problema. Um modelo orientado na sequência de produção foi proposto para representação do problema utilizando-se um conjunto de sequências de produção pré-selecionadas. Assim, variáveis binárias (W_{st}) são utilizadas a fim de indicar se uma sequência s é utilizada ($W_{st} = 1$) ou não ($W_{st} = 0$) durante o período t . Nesse tipo de modelagem, o número de possíveis sequências de produção cresce exponencialmente com relação ao número de produtos. Desse modo, a consideração de todas as possíveis sequências é impraticável para instâncias de médio e grande porte. Assim, técnicas baseadas em geração de colunas são utilizadas para incorporar ao modelo, de maneira dinâmica, as sequências de produção mais promissoras. Porém [Guimarães, Klabjan e Almada-Lobo \(2013\)](#) argumentam que a utilização do procedimento de geração de colunas de maneira completa pode comprometer a eficiência do modelo e, portanto, esse procedimento foi utilizado dentro de heurísticas *relax-and-fix* e *fix-and-optimize*.

Para construir uma solução inicial, a cada iteração r , um modelo híbrido foi considerado, onde para os períodos produtivos t com $t \leq r$ se utiliza um modelo orientado na sequência de produção e para os demais períodos, utiliza-se o modelo CLSD. Em cada iteração, fixa-se as sequências produtivas obtidas nos períodos prévios ($t < r$) e utiliza-se o procedimento de geração de colunas para determinar uma sequência a ser utilizada no período incumbente $t = r$, considerando-se para os períodos posteriores ($t > r$) a relaxação linear do modelo CLSD. Ao final do procedimento, obtém-se uma solução factível inicial para o modelo orientado na sequência de produção. No procedimento de melhoria, adota-se o modelo orientado na sequência para todos os períodos produtivos e a cada iteração algumas das variáveis inteiras do modelo são liberadas para reotimização, enquanto que, as demais são fixadas no valor incumbente. Antes de se resolver cada subproblema, um procedimento heurístico é realizado a fim de gerar novas sequências de produção para serem incorporadas ao modelo.

Em [Chen \(2015\)](#), a meta-heurística VNS foi combinada com a heurística *fix-and-optimize* para lidar com um problema de dimensionamento de lotes múlti-estágio. Primeiramente, níveis de inter-relação entre as variáveis binárias foram definidos de acordo com as restrições do problema. Esses níveis de inter-relação foram utilizados na definição das estruturas de vizinhança utilizadas no procedimento VNS. A fase de busca local do procedimento VNS proposto consiste na aplicação da heurística *fix-and-optimize*, onde as variáveis a serem reotimizadas são determinadas de acordo com a estrutura de vizinhança incumbente. Resultados computacionais, mostraram que o procedimento proposto em [Chen \(2015\)](#) foi capaz de encontrar melhores soluções do que heurísticas do tipo *fix-and-optimize* utilizando-se as tradicionais decomposições do conjunto das variáveis binárias do problema.

2.2.2 Heurísticas Lagrangianas

Heurísticas baseadas em relaxação lagrangiana têm sido propostas na literatura para lidar com diversos problemas de planejamento da produção (Toledo e Armentano (2006), Xiao *et al.* (2015), Carvalho e Nascimento (2016)). Essas heurísticas oferecem tanto uma solução factível como um limitante dual, possibilitando estimar a qualidade da solução incumbente.

Uma heurística lagrangiana consistindo na dualização das restrições de capacidade foi proposta em Toledo e Armentano (2006) para lidar com um problema de dimensionamento de lotes em linhas paralelas. Nessa aplicação, o subproblema resultante foi decomposto por itens e foi resolvido por um algoritmo de programação dinâmica. A relaxação das restrições de capacidade em problemas de dimensionamento de lotes também foi explorada em Absi *et al.* (2014), Wolosewicz, Dauzère-Pérès e Aggoune (2015) e em Brahimi *et al.* (2015), que considerou janelas de tempo de produção.

Em Xiao *et al.* (2015) um esquema combinando a estratégia de decomposição lagrangiana com a meta-heurística *simulated annealing* foi proposto para resolver um PDSL em linhas paralelas.

A técnica de relaxação lagrangiana foi introduzida na década de 1970 por Held e Karp (1970). Para Fisher (2004) esse procedimento caracteriza-se por considerar que alguns problemas “difíceis” de serem resolvidos podem ser vistos como uma série de problemas “fáceis” que são complicados pela incorporação de um conjunto de restrições. A relaxação lagrangiana consiste então, em dualizar esse conjunto de restrições “complicadoras”, obtendo um subproblema, dito lagrangiano, onde essas restrições são consideradas apenas na função objetivo mediante uma penalização para soluções infactíveis.

Sejam A uma matriz em $\Re^{m \times n}$, C uma matriz em $\Re^{q \times n}$, b um vetor em $\Re^m$ e d um vetor em $\Re^q$. Considere o problema de otimização linear inteiro misto (P) que consiste em minimizar a função objetivo (2.38), sujeito as restrições (2.39), (2.40) e (2.41). A variável de decisão x possui n componentes e o conjunto B contém as restrições de sinal e de integralidade de x , isto é, $B = \Re_+^n$, ou $B = \Re_+^{n-p} \times \{0, 1\}^p$, ou $B = \Re_+^{n-p} \times Z^p$, onde p é o número de componentes binárias e/ou inteiras do vetor x .

$$(P): \quad z = \min \quad f(x) \quad (2.38)$$

$$\text{s.a.} \quad Ax \leq b, \quad (2.39)$$

$$Cx \leq d, \quad (2.40)$$

$$x \in B. \quad (2.41)$$

Suponha que as restrições (2.39) sejam “complicadoras”, ou seja, que o problema (P) na ausência dessas restrições pode ser “facilmente” resolvido. Seja $\lambda \in \Re_+^m$ um vetor de penalizações, conhecido como vetor das variáveis duais ou de multiplicadores de Lagrange. O

subproblema lagrangiano (LR_λ) referente ao problema (P) com o vetor de multiplicadores $\lambda \in \Re_+^m$ associado às restrições (2.39) é definido da seguinte forma:

$$(LR_\lambda) : z_\lambda = \text{Min}\{f(x) + \lambda(Ax - b) | Cx \leq d, x \in B\}.$$

A Definição 2.1 formaliza o conceito de relaxação para um problema de otimização envolvendo a minimização da função objetivo, conforme apresentado em Wolsey (1998) e em Guignard (2003). A definição de relaxação para problemas de maximização pode ser obtida de maneira análoga.

Definição 2.1. Um problema (RP): $w = \min\{g(x) | x \in W\}$ é uma relaxação do problema (P): $v = \min\{f(x) | x \in V\}$, com a mesma variável de decisão x , se e somente se

- i) a região factível do problema P está contida na região factível do problema RP, isto é, $V \subset W$; e
- ii) para todo $x \in V$, tem-se $g(x) \leq f(x)$.

Segue diretamente da Definição 2.1 que o valor ótimo do problema relaxado (RP) é um limitante inferior para o valor ótimo do problema (P), isto é, $w \leq v$. De fato, seja x^* a solução ótima do problema (P), assim $v = f(x^*)$ e $x^* \in V$. Como $V \subset W$, então $x^* \in W$, donde segue que $w \leq g(x^*)$. Porém, pelo item (ii) da Definição 2.1, como $x^* \in V$, tem-se que $w \leq g(x^*) \leq f(x^*) = v$.

É possível verificar que para qualquer $\lambda \in \Re_+^m$, o problema (LR_λ) é uma relaxação para o problema (P) (Wolsey (1998) e Guignard (2003)). De fato, a região factível do problema (P), $\{x \in \Re^n | Ax \leq b, Cx \leq d, x \in B\}$, está contida na região factível do problema (LR_λ) , dada por $\{x \in \Re^n | Cx \leq d, x \in B\}$. Além disso, dado qualquer $\lambda \geq 0$ e $x \in \{Ax \leq b, Cx \leq d, x \in B\}$, tem-se que $\lambda(Ax - b) \leq 0$, donde $f(x) + \lambda(Ax - b) \leq f(x)$. Portanto, $z_\lambda \leq z, \forall \lambda \in \Re_+^m$, isto é, a solução ótima do problema (LR_λ) é um limitante inferior (dual) para o problema (P).

O problema que consiste em encontrar o “melhor” (maior) limitante inferior possível para o problema (P) é chamado problema dual lagrangiano e pode ser definido conforme a Definição 2.2 (Guignard (2003)).

Definição 2.2. O problema dual lagrangiano obtido ao dualizar as restrições (2.39) do problema (P) é dado por

$$(LD) : w = \max_{\lambda \geq 0} \{z_\lambda\} = \max_{\lambda \geq 0} \{\min\{f(x) + \lambda(Ax - b) | Cx \leq d, x \in B\}\}.$$

Seja $x(\lambda)$ a solução ótima do problema (LR_λ) . O Teorema 2.3 permite identificar se $x(\lambda)$ é uma solução ótima para o problema original (P) (Wolsey (1998)).

Teorema 2.3. Sejam $\lambda \in \mathfrak{R}_+^m$ e $x(\lambda)$ a solução ótima de (LR_λ) . Se $x(\lambda)$ é uma solução factível para o problema original (P), isto é, $Ax(\lambda) \leq b$ e se $(Ax(\lambda))_i = b_i$ quando $\lambda_i > 0$, então $x(\lambda)$ é uma solução ótima do problema (P).

Demonstração. De fato, como $x(\lambda)$ é factível para o problema original (P), então tem-se que $z \leq f(x(\lambda))$ (I). Por outro lado, como (LR_λ) é uma relaxação de (P), então $z \geq z_\lambda = f(x(\lambda)) + \lambda[Ax(\lambda) - b]$ (II). Por fim, como $(Ax(\lambda))_i = b_i$ quando $\lambda_i > 0$, então $\lambda[Ax(\lambda) - b] = 0$, donde tem-se por (II) que $z \geq z_\lambda = f(x(\lambda)) + \lambda[Ax(\lambda) - b] = f(x(\lambda))$ (III). Assim, por (I) e (III), tem-se que $z = f(x(\lambda))$. Portanto, $x(\lambda)$ é uma solução ótima de (P). $\square$

2.2.2.1 Esquemas de relaxação

Dado um modelo de otimização, inúmeros procedimentos baseados em relaxação lagrangiana podem ser obtidos de acordo com as restrições escolhidas para serem relaxadas. Em [Guignard \(2003\)](#), são apresentados esquemas de relaxação que têm sido propostos na literatura de acordo com as características dos problemas tratados. Assim, para [Guignard \(2003\)](#) procedimentos lagrangianos podem ser propostos, por exemplo, nos seguintes casos.

1. Quando é possível isolar um subproblema que possui uma estrutura especial por meio da dualização de todas as demais restrições. Normalmente, utiliza-se essa abordagem quando existe um algoritmo especializado como, por exemplo, um algoritmo de programação dinâmica para solução dos subproblemas. Para [Guignard \(2003\)](#), essa abordagem é bastante comum na literatura;
2. Quando existem dois (ou mais) subproblemas com estruturas especiais que são interligados por um conjunto de variáveis comuns. Nesse caso, as variáveis comuns podem ser renomeadas em parte das restrições do problema e restrições redundantes são incorporadas e dualizadas a fim de que o subproblema lagrangiano possa ser decomposto em dois (ou mais) subproblemas com estruturas especiais.

Mais especificamente, considere que os conjuntos de restrições $\{Ax \leq b, x \in B\}$ e $\{Cx \leq d, x \in B\}$ caracterizem duas estruturas especiais e tome o problema de otimização (M), definido por (2.42). Note que os problemas (M) e (N) (dado por (2.43)) são equivalentes.

$$(M): \quad \min_x \{f(x) | Ax \leq b, Cx \leq d, x \in B\} \quad (2.42)$$

$$(N): \quad \min_x \{f(x) | Ax \leq b, Cy \leq d, x \in B, y \in B, y = x\}. \quad (2.43)$$

Dualizando as restrições $y = x$ no problema (N), utilizando o vetor de multiplicadores λ , obtém-se o subproblema lagrangiano (2.44), que pode ser decomposto como em (2.45).

$$z_\lambda = \min_{x,y} \{f(x) + \lambda(y - x) | Ax \leq b, Cy \leq d, x \in B, y \in B\} \quad (2.44)$$

$$z_\lambda = \min_x \{f(x) - \lambda x | Ax \leq b, x \in B\} + \min_y \{\lambda y | Cy \leq d, y \in B\}. \quad (2.45)$$

Esse esquema é chamado de decomposição lagrangiana e um exemplo de sua utilização para o PDSL pode ser encontrado em [Xiao et al. \(2015\)](#).

3. Quando existem dois (ou mais) subproblemas com estruturas especiais que são interligados por um conjunto de restrições comuns. Nesse caso, as restrições que interligam as duas estruturas especiais podem ser dualizadas e o problema resultante poderá ser decomposto isolando-se as estruturas especiais do problema original.

Considere, agora, o problema $\min_{x,y} \{cx + \bar{c}y | Ax \leq b, Cy \leq d, Ex + Fy \leq g, x \in B^1, y \in B^2\}$, onde os conjuntos B^1 e B^2 contêm as restrições de sinal e de integralidade das componentes das variáveis x e y , respectivamente. Dualizando as restrições $Ex + Fy \leq g$ utilizando o vetor de penalizações λ , obtém-se o subproblema lagrangiano (2.46) que pode ser decomposto conforme (2.47).

$$z_\lambda = \min_{x,y} \{cx + \bar{c}y + \lambda(Ex + Fy - g) | Ax \leq b, Cy \leq d, x \in B^1, y \in B^2\} \quad (2.46)$$

$$z_\lambda = \min_x \{cx + \lambda(Ex - g) | Ax \leq b, x \in B^1\} + \min_y \{\bar{c}y + \lambda Fy | Cy \leq d, y \in B^2\}. \quad (2.47)$$

Os casos 2 e 3 serão aplicados ao problema estudado nesta tese com o objetivo de obter um subproblema lagrangiano que pode ser decomposto num problema de dimensionamento de lotes com decisões sobre abertura de linhas e num problema de sequenciamento em linhas paralelas.

2.2.2.2 O problema dual lagrangiano

Dado um problema de otimização e um esquema de relaxação lagrangiana, o problema dual lagrangiano consiste em encontrar o melhor limitante dual possível por meio do esquema de relaxação proposto. Diversas abordagens de solução têm sido propostas na literatura para solução desse problema, tais como, o método do subgradiente, o algoritmo do volume, planos de cortes, geração de colunas, lagrangiano aumentado e os *bundle methods* ([Guignard \(2003\)](#)).

Considere o esquema de relaxação lagrangiana caracterizado pela dualização da restrição (2.39) do problema (P) (definido por (2.38), (2.39), (2.40) e (2.41)). O problema dual lagrangiano (LD) (Definição 2.2) consiste em maximizar a função lagrangiana z_λ (definida no problema LR_λ) no domínio $\{\lambda \geq 0\}$.

Os métodos existentes para solução do problema dual lagrangiano, normalmente, levam em consideração as propriedades da função lagrangiana z_λ . As propriedades mais imediatas são dadas pelos Teoremas 2.4 e 2.6.

Teorema 2.4. A função lagrangiana z_λ é uma função côncava de λ .

Demonstração. De fato, sejam $\lambda^1, \lambda^2 \in \Re_+^m$ dois vetores de multiplicadores de Lagrange e $\alpha \in [0, 1]$ um escalar. Seja, ainda, $\bar{x}$ a solução ótima do problema $LR(\alpha\lambda^1 + (1 - \alpha)\lambda^2)$, isto é,

$$z_{(\alpha\lambda^1 + (1-\alpha)\lambda^2)} = f(\bar{x}) + (\alpha\lambda^1 + (1 - \alpha)\lambda^2)(A\bar{x} - b).$$

Naturalmente, tem-se que:

$$\text{i) } z_{\lambda^1} \leq f(\bar{x}) + \lambda^1(A\bar{x} - b); \text{ e}$$

$$\text{ii) } z_{\lambda^2} \leq f(\bar{x}) + \lambda^2(A\bar{x} - b).$$

Assim, como $\alpha \geq 0$ e $1 - \alpha \geq 0$ então, multiplicando (i) por α , (ii) por $1 - \alpha$ e somando as duas expressões obtém-se que

$$\alpha z_{\lambda^1} + (1 - \alpha) z_{\lambda^2} \leq \alpha f(\bar{x}) + \alpha \lambda^1(A\bar{x} - b) + (1 - \alpha) f(\bar{x}) + (1 - \alpha) \lambda^2(A\bar{x} - b).$$

Donde segue que:

$$\alpha z_{\lambda^1} + (1 - \alpha) z_{\lambda^2} \leq f(\bar{x}) + (\alpha \lambda^1 + (1 - \alpha) \lambda^2)(A\bar{x} - b) = z_{(\alpha \lambda^1 + (1 - \alpha) \lambda^2)}.$$

Portanto, z_λ é uma função côncava de λ . □

Definição 2.5. A envoltória convexa de um conjunto $X \subseteq \Re^n$, denotada por $\text{Co}(X)$, é definida como $\text{Co}(X) = \{x | x = \sum_{t=1}^T \alpha_t x^t, \sum_{t=1}^T \alpha_t = 1, \alpha_t \geq 0, \forall t\}$, onde $\{x^1, \dots, x^T\}$ é qualquer subconjunto finito de X .

É possível mostrar que, para todo $X \subseteq \Re^n$, $\text{Co}(X)$ é um poliedro e que todos os pontos extremos de $\text{Co}(X)$ estão em X (Wolsey (1998)).

Teorema 2.6. Seja $X = \{x | Cx \leq d, x \in B\}$. Se $\text{Co}(X)$ é um poliedro limitado, isto é, $\text{Co}(X) = \text{Co}\{x^1, x^2, \dots, x^T\}$ onde $\{x^1, x^2, \dots, x^T\}$ é um conjunto finito de pontos extremos, então a função lagrangiana z_λ é uma função linear por partes.

Demonstração. De fato, se $\text{Co}(X) = \text{Co}\{x^1, x^2, \dots, x^T\}$, segue que a função lagrangiana pode ser escrita como

$$z_\lambda = \min_x \{f(x) + \lambda(Ax - b) | x \in X\} = \min_{t=1, \dots, T} \{f(x^t) + \lambda(Ax^t - b)\}. \quad (2.48)$$

Logo, z_λ é linear por partes. □

Assim, pelos Teoremas 2.4 e 2.6, a função lagrangiana z_λ é côncava e linear por partes (quando $\text{Co}(X)$ é limitado). Além disso, de acordo com Guignard (2003), z_λ é uma função contínua que pode ser não diferenciável em alguns pontos do seu domínio.

- O método do subgradiente:

O método do subgradiente tem sido, sem dúvida, o mais utilizado para solução do problema dual lagrangiano (LD) obtido via problemas de otimização inteira. Para Fisher (2004) isso se deve, em parte, a facilidade de implementação do método e, em parte, por ter funcionado bem em muitos problemas práticos. Exemplos de utilização desse método podem ser encontrados,

por exemplo, em Toledo e Armentano (2006), Brahimi e Dauzère-Pérès (2015), Wolosewicz, Dauzère-Pérès e Aggoune (2015), Xiao *et al.* (2015) e Carvalho e Nascimento (2016).

O método do subgradiente foi proposto em Held e Karp (1970) para lidar com o problema do caixeiro viajante e sua validação matemática foi apresentada em Held, Wolfe e Crowder (1974). Esse método baseia-se no fato de que a função lagrangiana, apesar de não diferenciável, é subdiferenciável (no sentido da Definição 2.7) em todo o domínio (Guignard (2003)).

Definição 2.7. Um vetor $y \in \Re^m, y \neq 0$ é um subgradiente de uma função côncava $f(x)$ no ponto $x^0 \in \Re^m$ se para todo $x \in \Re^m$, tem-se

$$f(x) - f(x^0) \leq y \cdot (x - x^0).$$

O conjunto de todos os subgradientes da função $f(x)$ no ponto x^0 é chamado subdiferencial de f em x^0 e é denotado por $\partial f(x^0)$.

A obtenção de um subgradiente para a função lagrangiana z_λ é trivial, de acordo com o Teorema 2.8.

Teorema 2.8. Sejam $\lambda^0 \in \Re_+^m$ e $x(\lambda^0)$ a solução do subproblema lagrangiano (LR_{λ^0}) . Então, o vetor $y = Ax(\lambda^0) - b$ é um subgradiente para a função lagrangiana z_λ em λ^0 .

Demonstração. De fato, seja $\lambda \in \Re_+^m$ um vetor qualquer. Note que $z_\lambda \leq f(x) + \lambda(Ax - b), \forall x \in X$. Em particular, tem-se que $z_\lambda \leq f(x(\lambda^0)) + \lambda(Ax(\lambda^0) - b)$ (I). Por outro lado, note que $z_{\lambda^0} = f(x(\lambda^0)) + \lambda^0(Ax(\lambda^0) - b)$, donde segue que $f(x(\lambda^0)) = z_{\lambda^0} - \lambda^0(Ax(\lambda^0) - b)$ (II). Substituindo (II) em (I), tem-se que

$$\begin{aligned} z_\lambda &\leq f(x(\lambda^0)) + \lambda(Ax(\lambda^0) - b) = z_{\lambda^0} - \lambda^0(Ax(\lambda^0) - b) + \lambda(Ax(\lambda^0) - b) \\ &= z_{\lambda^0} + (\lambda - \lambda^0)(Ax(\lambda^0) - b). \end{aligned}$$

Portanto, $Ax(\lambda^0) - b$ é um subgradiente para a função lagrangiana z_λ . □

Para Fisher (2004), o método do subgradiente é uma adaptação do método do gradiente projetado trocando o gradiente pelo subgradiente. Assim, a cada iteração, um passo é dado ao longo da direção do vetor subgradiente, mantendo a factibilidade do vetor de multiplicadores via projeção na região de factibilidade ($\lambda \geq 0$). Dado um vetor inicial de multiplicadores de Lagrange λ^0 , gera-se uma sequência $\{\lambda^k\}$ da seguinte forma

$$\lambda_i^{k+1} = \max\{0, \lambda_i^k + s_k(Ax(\lambda^k) - b)_i\}, \quad i = 1, \dots, m, \quad (2.49)$$

onde s_k é um escalar positivo que representa o tamanho do passo que será dado na direção do vetor subgradiente $(Ax(\lambda^k) - b)$, $\lambda_i^k, \lambda_i^{k+1}$ representam a i -ésima coordenada do vetor λ nas iterações k e $k+1$, respectivamente e $(Ax(\lambda^k) - b)_i$ representa a i -ésima coordenada do vetor subgradiente $Ax(\lambda^k) - b$.

O tamanho do passo s_k é de fundamental importância para o bom funcionamento do método. De acordo com [Fisher \(2004\)](#), a convergência teórica do método do subgradiente pode ser garantida quando $s_k \rightarrow 0$ e $\sum_{i=0}^k s_i \rightarrow \infty$. Em [Held, Wolfe e Crowder \(1974\)](#), propõe-se a atualização do tamanho do passo de acordo com a equação (2.50), onde σ_k é um parâmetro no intervalo $[0,2]$, z^* é o valor ótimo do problema (P) e z_{λ^k} é o valor da função lagrangiana na iteração k .

$$s_k = \sigma_k \frac{z^* - z_{\lambda^k}}{\|Ax^k - b\|^2} \quad (2.50)$$

A utilização da equação (2.50) é, muitas vezes, inviável uma vez que o valor z^* é desconhecido. Porém, em diversos casos, uma solução factível pode ser obtida por meio de um procedimento heurístico de factibilização aplicado na solução incumbente x^k . Nesses casos, utiliza-se, normalmente, o valor $\bar{z}^k$ de uma solução factível no lugar do valor ótimo do problema z^* ([Fisher \(2004\)](#)). Assim, o tamanho do passo é atualizado de acordo com a equação (2.51).

$$s_k = \sigma_k \frac{\bar{z}^k - z_{\lambda^k}}{\|Ax^k - b\|^2} \quad (2.51)$$

Outra estratégia para atualização do tamanho do passo (s_k) pode ser encontrada em [Bazaraa e Sherali \(1981\)](#).

- O algoritmo do volume:

O algoritmo do Volume, proposto por [Barahona e Anbil \(2000\)](#), é uma extensão do método do gradiente que estima, a cada iteração, o valor de uma solução primal do problema, tentando manter o baixo custo computacional do método do subgradiente.

De acordo com [Silva \(2015\)](#), o limitante dual é obtido de maneira análoga ao método do subgradiente, enquanto que uma aproximação da solução primal é obtida estimando o volume formado pelas faces ativas relativas às restrições dualizadas. O Algoritmo do Volume, conforme proposto em [Barahona e Anbil \(2000\)](#), consiste nos passos P0 a P6 apresentados a seguir, onde a cada iteração t , z^t é o valor do limitante dual incumbente.

P0: Inicie com um vetor inicial de multiplicadores $\bar{\lambda}$ e resolva $LR(\bar{\lambda})$ obtendo a solução ótima $\bar{x}$ e o valor ótimo $\bar{z}$;

P1: $x^0 \leftarrow \bar{x}$, $z^0 \leftarrow \bar{z}$, $t \leftarrow 1$;

P2: $v^t \leftarrow A\bar{x} - b$, $\lambda^t \leftarrow \bar{\lambda} + sv^t$, onde s é o tamanho do passo;

P3: Resolva $LR(\lambda^t)$ obtendo a solução ótima x^t e o valor ótimo z^t ;

P4: $\bar{x} \leftarrow \alpha x^t + (1 - \alpha)\bar{x}$, onde α é um escalar entre 0 e 1;

P5: Se $z^t > \bar{z}$, então $\bar{\lambda} \leftarrow \lambda^t$ e $\bar{z} \leftarrow z^t$;

P6: $t \leftarrow t + 1$ e volte ao passo P2.

Os passos P0 e P1 correspondem à inicialização do método e são executados uma única vez. Os passos P2 a P4 compõe a chamada iteração menor (*minor iteration*), enquanto que, o passo P5 constitui a iteração maior (*major iteration*), de acordo com a nomenclatura utilizada em Barahona e Anbil (2000).

Ainda de acordo com Barahona e Anbil (2000), é possível obter uma sequência de iterações menores sem que haja uma melhoria no valor do limitante incumbente e portanto, sem que a condição do passo P5 seja satisfeita. Durante essas iterações, procura-se mover numa vizinhança de $(\bar{z}, \bar{\lambda})$ na tentativa de estimar os volumes abaixo das faces que estão ativas nessa vizinhança (Barahona e Anbil (2000)).

O algoritmo do volume atualiza, a cada iteração, todas as componentes da direção de busca (dada pelo vetor v), enquanto que o algoritmo do subgradiente atualiza apenas as variáveis duais referentes às restrições não satisfeitas na igualdade. Além disso, o método do volume possui um critério de parada mais eficiente do que o subgradiente. Em Barahona e Anbil (2000), propõe-se que o procedimento seja interrompido quando $\|v\|$ e $|c\bar{x} - \bar{z}|$ são ambos inferiores a um dado limite, onde $c\bar{x}$ é a função objetivo do problema original avaliada no vetor $\bar{x}$, $\bar{z}$ é o melhor limitante dual encontrado durante o procedimento, v é a direção de busca e $\bar{x}$ é uma estimativa para uma solução primal do problema original.

O parâmetro α utilizado no passo P4 é utilizado para obtenção de uma combinação convexa entre a direção atual e a direção anterior de busca e é de fundamental importância para o desempenho do método. Sejam $\bar{v} = A\bar{x} - b$ e $v^t = Ax^t - b$. Tome, ainda, um limitante superior $\alpha_{max} > 0$ para o parâmetro α e α_{opt} , a solução ótima do problema definido em (2.52).

$$\alpha_{opt} = \min_{\alpha} \|\alpha v^t + (1 - \alpha)\bar{v}\|. \quad (2.52)$$

Define-se $\alpha = \frac{\alpha_{max}}{10}$ quando $\alpha_{opt} < 0$ e $\alpha = \min\{\alpha_{opt}, \alpha_{max}\}$, caso contrário. Também sugere-se a utilização de $\alpha_{max} = 0, 1$, porém esse valor foi empiricamente determinado tomando como base um pequeno conjunto de problemas de otimização linear inteira.

O tamanho do passo s utilizado no passo P2 é determinado de maneira análoga ao método do subgradiente, segundo a equação (2.53), onde UB é o valor de um limitante primal (valor de uma solução factível) e σ é um parâmetro no intervalo $(0, 2)$.

$$s = \sigma \frac{UB - \bar{z}}{\|v\|^2}. \quad (2.53)$$

Cada iteração do algoritmo do volume é classificada como uma iteração vermelha, amarela ou verde e o parâmetro σ é atualizado de acordo com o tipo de iteração realizada. Uma iteração é dita vermelha quando $z^t < \bar{z}$. Se uma dada iteração t não é vermelha, isto é, $z^t \geq \bar{z}$, então define-se o parâmetro $d = v^t \cdot (Ax^t - b) = (A\bar{x} - b) \cdot (Ax^t - b)$. Se $d < 0$, diz-se que a iteração t é amarela e se $d \geq 0$, diz-se que a iteração t é verde.

A cada iteração verde, o valor de σ deve ser aumentado e Barahona e Anbil (2000) sugerem que esse parâmetro seja multiplicado pelo escalar 1,1. Sugerem, ainda, que a cada

20 iterações vermelhas consecutivas, o parâmetro σ seja multiplicado por 0,66. Quando uma iteração é classificada como amarela, têm-se que a adoção de um valor elevado para σ pode resultar numa redução de z_λ . Nesse caso, mantém-se constante o valor de σ .

O algoritmo do volume foi testado para solução dos modelos clássicos para os problemas de particionamento, recobrimento, corte máximo em grafos e localização de facilidades em Barahona e Anbil (2000). Os resultados computacionais obtidos foram satisfatórios, porém os autores ressaltam que problemas com estrutura mais geral não foram estudados e que estes problemas podem gerar maiores dificuldades para utilização do algoritmo do volume. Essa dificuldade foi observada, por exemplo, em Silva (2015) quando o algoritmo do volume foi utilizado para resolver o problema dual gerado por um esquema de relaxação lagrangiana aplicado ao PDSL. Testes computacionais mostraram que o algoritmo do subgradiente obteve melhores limitantes para a maioria das instâncias do que o algoritmo do volume com tempo máximo de execução fixado tanto em 5 minutos, como em 1 hora.

Nesta tese, o algoritmo do volume foi implementado para solucionar os problemas duais lagrangianos referentes a dois esquemas de relaxação lagrangiana, propostos no Capítulo 5, que foram utilizados para lidar com o problema estudado. Os resultados obtidos através da utilização do algoritmo do volume são comparados com os resultados obtidos pelo método do subgradiente.

2.2.2.3 Heurística lagrangiana

A obtenção de uma solução factível durante as iterações lagrangianas é bastante interessante, independentemente do método utilizado para solução do problema dual lagrangiano. O algoritmo do volume fornece apenas uma estimativa para uma solução primal que, por muitas vezes, não é satisfatória. O conhecimento de uma solução primal, a cada iteração, pode aprimorar a atualização do tamanho do passo a ser dado na direção de busca (conforme as equações 2.51 e 2.53). Dessa forma, diversos procedimentos lagrangianos munidos de um esquema heurístico que consegue determinar, a cada iteração, uma solução factível para o problema original a partir da solução dual incumbente x^k têm sido desenvolvidos na literatura. Esses procedimentos são conhecidos como heurísticas lagrangianas. A Figura 2 apresenta um esquema básico de funcionamento de uma heurística lagrangiana.

A atualização das variáveis duais apresentada na Figura 2 pode ser feita tendo como base qualquer método para solução do problema (LD). Normalmente, os critérios de parada utilizados são, além de verificadas as condições necessárias e suficientes do Teorema 3.1, o atingimento de um limite máximo de tempo de execução e/ou de iterações.

Em Toledo e Armentano (2006) foram dualizadas as restrições de capacidade do problema de dimensionamento de lotes em máquinas paralelas. Assim, a solução do problema dual, quando infactível, viola os limites de capacidade de alguma máquina em algum período. Um procedimento de factibilização foi proposto consistindo em mover parte da produção alocada nas máquinas e períodos em que os limites de capacidade foram excedidos para outras máquinas e/ou

Figura 2 – Esquema de funcionamento de uma heurística lagrangiana

períodos com capacidade disponível. Já em [Xiao *et al.* \(2015\)](#) que propuseram uma decomposição lagrangiana para lidar com um PDSL em múltiplas máquinas, a solução ótima do subproblema lagrangiano poderia não cumprir toda a demanda requerida de alguns produtos. Assim, soluções factíveis foram obtidas por meio da utilização de um algoritmo de inserção para alocação, em alguma máquina e período, da produção dos itens que não tiveram a sua demanda completamente atendida.

[Carvalho e Nascimento \(2016\)](#) abordaram o problema de dimensionamento de lotes em múltiplas plantas produtoras propondo um esquema lagrangiano caracterizado pela dualização das restrições de capacidade produtiva das plantas. Soluções factíveis foram obtidas partindo da solução ótima do subproblema lagrangiano por meio da transferência de produção de itens entre as diversas plantas produtoras e períodos do horizonte.

No Capítulo 5 propõe-se uma abordagem baseada em relaxação de restrições e uma decomposição lagrangiana, ambas visando a obtenção de subproblemas lagrangianos que possam ser decompostos em um problema de dimensionamento de lotes e em um problema de sequenciamento.

DEFINIÇÃO E MODELAGEM DO PROBLEMA EM ESTUDO

Apresenta-se neste Capítulo a definição e modelagem do problema de planejamento de produção em estudo. O problema foi inspirado no sistema produtivo de uma indústria de carnes que possui características ainda não abordadas pelos modelos existentes na literatura. Na Seção 3.1 faz-se uma breve descrição do setor produtivo considerado; na Seção 3.2 é apresentada a descrição do problema de planejamento de produção estudado e na Seção 3.3 propõe-se modelos de programação matemática inteira mista para representação do problema.

3.1 O setor produtivo

De acordo com o Relatório Anual da Associação Brasileira das Indústrias Exportadoras de Carnes - ABEIC (2016), a produção de carnes é uma atividade significativa para economia nacional, representando cerca 6,30% do produto interno bruto - PIB do Brasil. Além disso, o Brasil é o 2º maior produtor mundial de carnes, responsável por cerca de 14% da produção mundial, atrás apenas dos Estados Unidos da América. O Brasil também é considerado um dos principais exportadores de carne do mundo. Das 9,56 milhões de toneladas de carne produzida no Brasil em 2015, cerca 1,88 milhões de toneladas (19,62% da produção) foram destinadas para a exportação, o que representou, em receita, 3% de tudo o que o Brasil exportou no ano de 2015 e contribuiu para o fechamento positivo da balança comercial daquele ano.

Além de ofertar carne *in natura* para o mercado internacional, de acordo com o site *Beef Point*, o Brasil é, ao lado da Argentina, um dos principais exportadores de carnes industrializadas, e os principais importadores dessas carnes são a União Europeia (que consome cerca de 50% da produção sul-americana) e os Estados Unidos da América (que consomem cerca de 30% da produção sul-americana).

Nesse setor produtivo, o catálogo de produtos é normalmente extenso, incluindo carnes

de diversos tipos de animais (bovinos, suínos e aves), com diversos tipos de cortes, podendo ser temperadas ou não, além dos produtos “embutidos” (diversos tipos de salsichas, presuntos, salames, mortadelas e linguças). Os itens produzidos são perecíveis, apresentando diferentes prazos de validade. Por exemplo, carnes *in natura* quando congeladas podem permanecer estocadas por vários meses, enquanto que, carnes temperadas não podem ser congeladas e seu prazo de validade é de apenas alguns dias.

Os produtos devem ser armazenados em ambientes com temperatura rigorosamente monitorada. A manutenção de baixas temperaturas consome grande quantidade de energia elétrica, fazendo com que os custos de estocagem sejam relevantes. A seguir, na Seção 3.2, descreve-se detalhadamente um problema de planejamento de produção enfrentado por indústrias produtoras de carnes.

3.2 O problema de planejamento de produção

Nesta Seção, descreve-se um problema de planejamento de produção inspirado numa indústria brasileira de carnes embaladas. Nesse ambiente, diversas linhas de produção compartilham recursos produtivos, tais como trabalhadores e máquinas, e devido à escassez desses recursos e características do problema, essas linhas de produção não podem operar todas simultaneamente. As linhas de produção são especializadas, de modo que, para cada item uma única linha é apta para produzi-lo.

Como os recursos de produção necessários para manter as linhas em atividade são compartilhados entre as diferentes linhas, faz-se necessário montar (abrir, ativar) as linhas de produção em cada período produtivo, que no problema corresponde a um dia. Assim, a cada período, os gestores necessitam decidir quais linhas de produção serão montadas respeitando a quantidade disponível dos recursos produtivos. A escolha das linhas que serão montadas num determinado período impacta no conjunto de itens que podem ser produzidos nesse período. A montagem das linhas de produção requer a realização de alguns procedimentos, tais como, a criação de novas estações (temporárias) de trabalho e alterações na disposição física de equipamentos, de modo a ser desejável que as linhas de produção sejam montadas apenas no início de cada período e permaneçam abertas até o fim do período.

Ao fim de cada período, uma limpeza deve ser efetuada em todas as máquinas e estações de trabalho. Nesse momento, o estado de preparação das linhas é perdido, pois as máquinas voltam todas em sua configuração original e não estão aptas para a produção de nenhum produto. Além disso, a preparação para produção do primeiro item em cada linha e período pode ser feita juntamente com a montagem das linhas sem aumento significativo no tempo de montagem.

Na Figura 3, apresenta-se um esquema do ambiente industrial baseado numa indústria alimentícia do interior do estado de São Paulo. Nesse caso, o horizonte de planejamento é composto por três períodos ($t = 1$, $t = 2$ e $t = 3$) e existem seis linhas de produção ($L1, \dots, L6$)

compartilhando três recursos produtivos (R1, R2 e R3). Os recursos podem ser máquinas, ferramentas ou trabalhadores. Por exemplo, R1 pode representar trabalhadores e R2 e R3 podem representar diferentes tipos de máquinas. As linhas ativadas estão representadas com fundo azul na Figura 3, enquanto que as não ativadas estão com fundo branco. Assim, por exemplo, no período $t = 1$, as linhas de produção L1, L2 e L5 foram montadas e, portanto, estão em funcionamento, enquanto que as linhas L3, L4 e L6 não estão operando. As setas indicam movimentações entre os recursos produtivos (R1, R2 e R3) ao longo das linhas entre os diferentes períodos.

A Figura 3 também apresenta uma tabela contendo as quantidades disponíveis (D), consumida (C) e ociosa (O) de cada recurso em cada período. A quantidade de recurso consumida é dada pela soma dos recursos utilizados em cada linha aberta, enquanto que a quantidade de recurso ocioso é dada pela quantidade disponível menos a utilizada. Por exemplo, a linha L1 requer 2 unidades do recurso R1, 3 unidades do recurso R2 e uma unidade de R3 para ser ativada. É possível ver que, em cada período, não se pode montar (abrir) outras linhas além das que já estão em funcionamento pois alguns recursos não estão disponíveis.

Figura 3 – Representação do ambiente industrial estudado

Assim como em [Meyr \(2000\)](#), considera-se nesse trabalho que uma linha de produção consiste em um fluxo automático entre algumas estações de trabalho e/ou máquinas. Então cada linha de produção pode ser considerada como uma única máquina.

No problema, considera-se que a demanda é determinística, dinâmica e que deve ser completamente atendida dentro do horizonte de planejamento. O atraso no atendimento da demanda (*backlogging*) é permitido, porém penalizado na função objetivo. Essa suposição é realística em ambientes industriais onde sucessivos atrasos no atendimento das demandas podem gerar perda de clientes. Os itens produzidos são perecíveis, isto é, podem permanecer em estoque por um limitado espaço de tempo (*shelf-life*) e necessitam ser armazenados em ambientes com temperatura monitorada para não serem deteriorados, acarretando em altos custos de estocagem.

No ambiente produtivo considerado, a troca na produção de itens numa mesma linha implica na realização de alguns procedimentos de preparação. Por exemplo, considere três itens que devem produzidos numa mesma linha de produção: bife fino temperado (p_1), bife grosso para churrasco (p_2) e bife fino sem tempero (p_3). Os procedimentos de preparação decorrentes

da troca do item (p_1) para (p_2) consistem na limpeza da linha e num ajuste na máquina de corte, enquanto que para troca de (p_1) para (p_3) é necessário apenas a limpeza da linha e de (p_3) para (p_1) são necessários a limpeza da linha e preparação de tempero. Essas preparações consomem um determinado tempo de produção que também pode ser traduzido num determinado custo. Como trocas envolvendo diferentes itens implicam em diferentes procedimentos, os custos e tempos de preparação são dependentes da sequência, tornando-se necessário determinar o sequenciamento da produção em cada período e em cada linha de produção.

Assume-se que os recursos produtivos correspondem a máquinas e trabalhadores. Considere K recursos (indexados por k) onde $k = 1$ denota trabalhadores e $k = 2, \dots, K$ denotam diferentes tipos de máquinas. Considera-se L linhas de produção compartilhando esses K recursos produtivos e são conhecidas as quantidades necessárias de cada recurso para montagem de cada linha de produção. O tempo de produção de cada item não pode ser reduzido ao adicionar recursos extras nas linhas de produção, ou seja, se alocarmos numa dada linha uma quantidade de um dado recurso maior do que a quantidade necessária para a ativação na linha, a taxa de produtividade não será ampliada.

Nesse problema, visa-se determinar:

- i) quais linhas de produção ativar em cada período, respeitando os limites de capacidade de cada recurso necessário para abertura das linhas;
- ii) os itens que serão produzidos em cada linha e período, bem como os tamanhos dos respectivos lotes; e
- iii) o sequenciamento da produção em cada linha e período,

com o objetivo de minimizar os custos de estoque, atraso no atendimento das demandas, preparação para produção e montagem das linhas. A esse novo problema dá-se o nome de problema de dimensionamento e sequenciamento de lotes em linhas paralelas e relacionadas - PDSL-LPR.

Em resumo, o PDSL-LPR é caracterizado pelas seguintes suposições:

1. As linhas de produção compartilham recursos e devido à escassez desses recursos, não podem operar todas simultaneamente;
2. As linhas de produção são montadas apenas no início de cada período produtivo do horizonte de planejamento e devem permanecer montadas até o fim do mesmo;
3. Para cada produto existe uma única linha de produção apta a produzi-lo;
4. Não considera-se, a priori, preservação do estado de preparação entre períodos. Essa suposição é realística para o ambiente industrial abordado, uma vez que ao término de cada período é necessário limpar todas as máquinas para evitar contaminação e, portanto, o estado de preparação é perdido;

5. Não são computados custos e tempos de preparação para o primeiro item produzido em cada uma das linhas de produção ativas em cada período. Esta suposição provém do fato de que os procedimentos de preparação para o primeiro item podem ser realizados juntamente com os procedimentos de montagem das linhas de produção;
6. Demanda determinística e dinâmica que deve ser completamente atendida durante o horizonte de planejamento, podendo haver atrasos (*backlogging*) que são penalizados na função objetivo;
7. Os itens são perecíveis e possuem diferentes períodos de validade (*shelf-life*);
8. Existem tempos e custos de preparação para produção dependentes da sequência;
9. Os procedimentos de preparação são completamente realizados dentro de um único período produtivo.

3.3 Modelagem Matemática

Nesta Seção, apresenta-se quatro modelos de programação matemática inteira mista para representação do PDSL-LPR. Os modelos propostos são extensões de modelos da literatura que têm sido frequentemente utilizados como base para representação de problemas reais (Almeder e Almada-Lobo (2011), (FURLAN *et al.*, 2015b)).

Primeiramente, o modelo GLSPPL (proposto Meyr (2002)) é estendido para o PDSL-LPR originando o modelo GLSPPL. Em seguida, o modelo GLSPPL é reformulado de acordo com Clark e Clark (2000) (modelo CCPRL), onde as variáveis de estado de preparação são suprimidas e também de acordo com a formulação via localização de facilidades para problemas de dimensionamento de lotes introduzida em Eppen e Martin (1987) (modelo CCFLPRL). Por fim, propõe-se o modelo CLSDPRL que consiste numa extensão do modelo CLSD proposto em Haase (1996).

3.3.1 Modelo GLSPPL

O modelo GLSPPL (Meyr (2002)) foi estendido para representar o PDSL-LPR (GLSPPL). Os parâmetros e variáveis utilizados na formulação do modelo GLSPPL são definidos na Tabela 1 e o modelo consiste em minimizar a função objetivo (3.1) sujeito as restrições (3.2) a (3.16).

• GLSPPRL:

$$\text{Min} \quad \sum_{j,t} (h_j I_{jt} + b_j B_{jt}) + \sum_{l,i,j,s} sc_{lij} z_{lijs} + \sum_{l,t} ac_l \delta_{lt} \quad (3.1)$$

$$\text{s.a.} \quad I_{jt} - B_{jt} = I_{j,t-1} - B_{j,t-1} + \sum_{l,s \in S_{lt}} x_{ljs} - d_{jt}, \quad \forall t, j \quad (3.2)$$

$$\sum_{j \in P_l, s \in S_{lt}} a_{lj} x_{ljs} + \sum_{i,j \in P_l, s \in S_{lt}} st_{lij} z_{lijs} \leq C_{lt}, \quad \forall l, t \quad (3.3)$$

$$x_{ljs} \leq \frac{C_{lt}}{a_{lj}} y_{ljs}, \quad \forall l, s, j \in P_l \quad (3.4)$$

$$x_{ljs} \geq m_{lj} (y_{ljs} - y_{lj,s-1}), \quad \forall l, t, j \in P_l, s \in S_{lt} \setminus \{f_{lt}\} \quad (3.5)$$

$$x_{ljf_{lt}} \geq m_{lj} y_{ljf_{lt}}, \quad \forall l, j \in P_l, t \quad (3.6)$$

$$\sum_{j \in P_l} y_{ljs} = \delta_{lt}, \quad \forall l, t, s \in S_{lt} \quad (3.7)$$

$$\sum_l r_{kl} \delta_{lt} \leq R_{kt}, \quad \forall k, t \quad (3.8)$$

$$I_{jt} \leq \sum_{\alpha=t+1}^{t+sl_j} d_{j\alpha}, \quad \forall j, t \quad (3.9)$$

$$z_{lijs} \geq y_{li,s-1} + y_{ljs} - 1, \quad \forall l, t, i, j \in P_l, s \in S_{lt} \setminus \{f_{lt}\} \quad (3.10)$$

$$B_{jT} = 0, \quad \forall j \quad (3.11)$$

$$I_{jt} \geq 0, B_{jt} \geq 0, \quad \forall j, t \quad (3.12)$$

$$x_{ljs} \geq 0, \quad \forall l, j, s \quad (3.13)$$

$$0 \leq z_{lijs} \leq 1, \quad \forall l, i, j, s \quad (3.14)$$

$$0 \leq \delta_{lt} \leq 1, \quad \forall l, t \quad (3.15)$$

$$y_{ljs} \in \{0, 1\}, \quad \forall l, j, s. \quad (3.16)$$

A função objetivo (3.1) representa a soma dos custos de estocagem, atrasos nos atendimentos das demandas, preparação dependente da sequência e montagem das linhas de produção. As restrições (3.2), (3.11), (3.12) e (3.13) garantem que as demandas serão completamente atendidas, podendo haver atrasos, durante o horizonte de planejamento, enquanto que as restrições de capacidade são dadas por (3.3). As restrições (3.4) garantem que os itens só poderão ser produzidos numa dada linha se esta estiver preparada para a produção desse item.

As restrições (3.5) e (3.6) referem-se aos lotes mínimos de produção e foram introduzidas para evitar trocas de estados de preparação sem que hajam trocas de itens produzidos. Essas restrições são necessárias quando a matriz de tempo/custo de preparação não satisfaz a desigualdade triangular (Fleischmann e Meyr (1997)). Mais especificamente, as restrições (3.5) garantem que, se a linha for preparada num dado microperíodo (exceto o primeiro de cada período) para um item diferente do item que foi produzido no microperíodo anterior, então uma quantidade mínima (m_{lj}) deve ser produzida. Já as restrições (3.6) garantem a produção de uma quantidade mínima para o primeiro item produzido em cada linha e período. Essas restrições

Tabela 1 – Parâmetros e variáveis para modelo GLSPRL

Parâmetros	
T	Número de períodos
L	Número de linhas de produção
J	Número de itens
K	Número de recursos
t	$\in \{1, \dots, T\}$ Índice para os períodos
l	$\in \{1, \dots, L\}$ Índice para as linhas de produção
i, j	$\in \{1, \dots, J\}$ Índices para os itens
k	$\in \{1, \dots, K\}$ Índice para os recursos
d_{jt}	Demanda do item j no período t
C_{lt}	Capacidade de produção (tempo) da linha l no período t
a_{lj}	Tempo consumido para produção de uma unidade do item j na linha l
m_{lj}	Lote mínimo de produção do item j na linha l
h_j	Custo unitário de estoque do item j
b_j	Custo por unidade atrasada do item j
sc_{lij}	Custo de troca de produtos (i para j) na linha l
st_{lij}	Tempo de troca de produtos (i para j) na linha l
r_{kl}	Quantidade do recurso k necessária para montagem da linha l
R_{kt}	Quantidade disponível do recurso k no período t
sl_j	Tempo no qual o item j pode permanecer em estoque
P_l	Conjunto dos itens que podem ser produzidos na linha l
S_{lt}	Conjunto dos micro-períodos da linha l no período t
S'_l	Número total de microperíodos na linha l (indexado por s)
f_{lt}	Primeiro microperíodo da linha l no período t
u_{lt}	Último microperíodo da linha l no período t
ac_l	Custo de montagem da linha l
Variáveis	
I_{jt}	Quantidade em estoque do item j no fim do período t
B_{jt}	Quantidade atrasada do item j no fim do período t
δ_{lt}	$= 1$, se a linha l será montada no período t
x_{ljs}	Quantidade produzida do item j , na linha l e no microperíodo s
y_{ljs}	$= 1$, se a linha l é preparada para a produção do item j durante o microperíodo s
z_{lijs}	$= 1$, se existe troca de produção na linha l , de i para j em s

foram introduzidas em virtude da não consideração da preservação do estado de preparação das linhas.

As restrições (3.7) garantem que as linhas montadas estão sempre aptas para a produção de algum item, enquanto que as linhas não ativadas não podem produzir. As restrições (3.8) referem-se aos limites de capacidade dos recursos necessários para montagem das linhas de produção. As restrições (3.9) foram baseadas em [Seyedhosseini e Ghoreyshi \(2014\)](#) para garantir que os produtos não sejam deteriorados pelo prazo de validade. Elas garantem que as quantidades armazenadas de cada item j em cada período não excedem as demandas pelo item j para os próximos sl_j períodos consecutivos, onde sl_j é o prazo máximo que o item j pode permanecer em estoque sem que seja deteriorado. As restrições (3.10) capturam as trocas de produtos em

cada linha de produção e em cada período. As restrições (3.12) a (3.16) definem os domínios das variáveis de decisão.

Note que as variáveis binárias δ_{lt} podem ser tratadas como contínuas, uma vez que as restrições (3.7) combinadas com (3.15) e (3.16) garantem que essas variáveis assumirão apenas valores binários.

Em outras aplicações, as linhas de produção podem permanecer ativadas em períodos adjacentes, de modo a ser interessante considerar a preservação do estado de preparação das linhas entre esses períodos. As restrições (3.17) permitem considerar a preservação do estado de preparação entre períodos (*setup carryover*).

$$z_{lijf_{lt}} \geq y_{liu_{l,t-1}} + y_{ljf_{lt}} + \delta_{lt-1} + \delta_{lt} - 3, \forall l, i \neq j, t. \quad (3.17)$$

Note que, se $\delta_{lt-1} = 0$ ou $\delta_{lt} = 0$, então as restrições (3.17) tornam-se redundantes e $z_{lijf_{lt}} = 0$ na solução ótima, uma vez que valores positivos para essas variáveis são penalizados na função objetivo. Porém, se $\delta_{lt-1} = \delta_{lt} = 1$, então as restrições (3.17) possuem a mesma função das restrições (3.10), garantindo que $z_{lijf_{lt}} = 1$ quando $y_{liu_{l,t-1}} = y_{ljf_{lt}} = 1$.

Ao propor o modelo GLSP, [Fleischmann e Meyr \(1997\)](#) chamam a atenção para o fato de que, se uma sequência de microperíodos dentro de um mesmo período é designada à produção de um mesmo item, então a quantidade total produzida pode ser arbitrariamente distribuída ao longo desses microperíodos sem alteração no sequenciamento e no valor da função objetivo. Nesse sentido, as restrições (3.18) e (3.19) são incluídas no modelo GLSPRL para evitar a existência dessas múltiplas soluções.

$$\sum_{i \in P_l} \sum_{j \in P_l \setminus \{i\}} z_{lijs-1} \geq \sum_{i \in P_l} \sum_{j \in P_l \setminus \{i\}} z_{lijs} \quad \forall l, t, s = f_{lt} + 2, \dots, u_{lt}; \quad (3.18)$$

$$x_{ljs} \leq \frac{C_{lt}}{a_{lj}} \left(2 - \sum_{i \in P_l} z_{lijs-1} - z_{ljjs} \right) \quad \forall l, j, t, s = f_{lt}, \dots, u_{lt}. \quad (3.19)$$

As restrições (3.18) alocam os microperíodos vazios após os microperíodos com produção, enquanto que as restrições (3.19) garantem que: se o item j foi produzido no microperíodo $s - 1$ e no micro período s a linha continuará preparada para produção do mesmo item j , então a produção em s deverá ser nula.

Testes computacionais realizados em [Guimarães, Klabjan e Almada-Lobo \(2014\)](#) mostraram que o modelo GLSP apresenta performance computacional inferior a outras formulações quando se utiliza um algoritmo do tipo *Branch-and-Cut* como método de solução. Em particular, o modelo GLSP apresenta um número elevado de variáveis e de restrições em virtude da estratégia de divisão dos períodos em diversos microperíodos para modelar o sequenciamento dos lotes, fazendo com que o modelo consuma grande espaço da memória do computador e aumente o número de variáveis a serem ramificadas na árvore de busca de algoritmos do tipo *Branch-and-Bound*. Assim, outras formulações matemáticas para o PDSL-LPR serão propostas neste trabalho.

3.3.2 Modelo CCPRL

O modelo CCPRL consiste numa reformulação do modelo GLSPPRL baseada em [Clark e Clark \(2000\)](#). Nessa reformulação, as variáveis de estado de preparação (y_{ljs}) são suprimidas e as variáveis de trocas de produtos (z_{lijs}) tornam-se binárias. A reformulação proposta em [Clark e Clark \(2000\)](#) baseia-se no fato de que uma linha de produção está preparada para produção de um item j num dado microperíodo s se uma troca de produtos foi realizada, partindo de um item qualquer com destino ao produto j durante o micro período s , o que implica a equação (3.20).

$$y_{ljs} = \sum_{i \in P_l} z_{lijs}, \forall l, j \in P_l, s. \quad (3.20)$$

Desse modo, o modelo CCPRL pode ser obtido a partir do modelo GLSPPRL, substituindo as variáveis y_{ljs} de acordo com (3.20). Assim, as restrições (3.4) a (3.7), (3.10) e (3.14) do modelo GLSPPRL são substituídas pelas restrições (3.21) a (3.26), respectivamente.

$$x_{ljs} \leq \frac{K_{lt}}{a_{lj}} \sum_{i \in P_l} z_{lijs}, \quad \forall l, s, j \in P_l \quad (3.21)$$

$$x_{ljs} \geq m_{lj} \sum_{i \in P_l, i \neq j} z_{lijs}, \quad \forall l, t, j \in P_l, s \in S_{lt} \setminus \{f_{lt}\} \quad (3.22)$$

$$x_{ljf_{lt}} \geq m_{lj} \sum_i z_{lijf_{lt}}, \quad \forall l, j \in P_l, t \quad (3.23)$$

$$\sum_{i \in P_l} \sum_{j \in P_l} z_{lijs} = \delta_{lt}, \quad \forall l, t, s \in S_{lt} \quad (3.24)$$

$$\sum_i z_{lij, s-1} = \sum_i z_{ljis}, \quad \forall l, t, i \in P_l, j \in P_l \quad (3.25)$$

$$z_{lijs} \in \{0, 1\}, \quad \forall l, t, i \in P_l, j \in P_l \quad (3.26)$$

As restrições (3.21) garantem que os itens podem ser produzidos numa determinada linha apenas se esta estiver preparada para a produção desse item, enquanto que as restrições (3.22) e (3.23) definem os lotes mínimos de produção. As restrições (3.24) garantem que só ocorrerão trocas de produtos (e consequentemente, produção) em linhas de produção abertas, enquanto que as restrições (3.25) são restrições de fluxo que garantem o sequenciamento da produção. As restrições (3.26) definem o domínio das variáveis de troca de produtos z_{lijt} . As restrições (3.18) e (3.19) continuam válidas para o modelo CCPRL.

Portanto, o modelo CCPRL consiste em minimizar a função objetivo (3.1), sujeito as restrições (3.2), (3.3), (3.21), (3.22), (3.23), (3.24), (3.8), (3.9), (3.25), (3.11), (3.12), (3.13), (3.26) e (3.15).

Suponha, agora, que os poliedros F_1 e F_2 são duas formulações para uma mesma região factível. De acordo com [Wolsey \(1998\)](#) a formulação F_1 é melhor do que a formulação F_2 se $F_1 \subset F_2$. Se F_1 é melhor do que F_2 , então a relaxação linear do modelo baseado na formulação F_1 fornece um limitante dual melhor ou igual a relaxação linear do modelo baseado na formulação F_2 .

Em Guimarães, Klabjan e Almada-Lobo (2014), prova-se que o modelo baseado em Clark e Clark (2000) para o PDSL é uma melhor formulação do que o modelo GLSP. Esse resultado é estendido para o PDSL-LPR por meio do Teorema 3.1 que afirma que a formulação CCPRL é melhor do que a formulação GLSPRL.

Teorema 3.1. A formulação CCPRL é melhor do que a formulação GLSPRL.

Demonstração. Sejam P_1 e P_2 os poliedros referentes aos modelos CCPRL e GLSPRL, respectivamente. Assim, $P_1 = \{(x, z, I, B, \delta) | (3.2), (3.3), (3.8), (3.9), (3.11) \text{ a } (3.15), (3.21) \text{ a } (3.25), 0 \leq z_{ljs} \leq 1, \forall l, i, j, s\}$ e $P_2 = \{(x, y, z, I, B, \delta) | (3.2) \text{ a } (3.15), 0 \leq y_{ljs} \leq 1, \forall l, j, s\}$.

Seja $(x, z, I, B, \delta) \in P_1$ e considere as variáveis $0 \leq y_{ljs} \leq 1$ definidas conforme (3.20). É preciso mostrar que $(x, y, z, I, B, \delta) \in P_2$. Note-se que:

i) Por (3.21) e (3.20) segue imediatamente (3.4);

ii) Observando que, dados quaisquer $l, j \in P_l$ e $s \in S_{lt} \setminus \{f_{lt}\}$, segue que

$$\sum_{i, i \neq j} z_{lijs} = \sum_i z_{lijs} - z_{ljjs} = y_{ljs} - z_{ljjs}$$

e

$$z_{ljjs} \leq \sum_i z_{ljis} = \sum_i z_{lij, s-1} = y_{l, j, s-1}.$$

Portanto, de (3.22) tem-se que, dados quaisquer $l, j \in P_l$ e $s \in S_{lt} \setminus \{f_{lt}\}$, obtém-se

$$x_{ljs} \geq m_{lj} \sum_{i, i \neq j} z_{lijs} = m_{lj}(y_{ljs} - z_{ljjs}) \geq m_{lj}(y_{ljs} - y_{l, j, s-1}),$$

donde segue (3.5).

iii) Por (3.23) e (3.20) obtém-se diretamente (3.6).

iv) Somando membro a membro todas as restrições (3.20) e utilizando (3.24), tem-se que

$$\sum_{j \in P_l} y_{ljs} = \sum_{j \in P_l} \sum_{i \in P_l} z_{lijs} = \delta_{lt}.$$

Logo, (3.7) é válida.

v) Aplicando (3.20) e (3.25), pode-se concluir que, dados $l, t, i, j \in P_l$ e $s \in S_{lt} \setminus \{f_{lt}\}$, segue que

$$y_{li, s-1} = \sum_{k \in P_l} z_{lki, s-1} = \sum_{k \in P_l} z_{lik s}$$

Novamente por (3.20), vem

$$y_{ljs} = \sum_{k \in P_l} z_{lkjs}.$$

Portanto,

$$\begin{aligned}
y_{li,s-1} + y_{ljs} - 1 &= \sum_{k \in P_l} z_{lik} + \sum_{k \in P_l} z_{lkjs} - 1 \\
&= z_{lijs} + \sum_{k \in P_l, k \neq j} z_{lik} + \sum_{k \in P_l} z_{lkjs} - 1 \\
&\leq z_{lijs} + \sum_{i \in P_l} \sum_{j \in P_l} z_{lijs} - 1 \\
&\leq z_{lijs} + \delta_{lt} - 1 \\
&\leq z_{lijs}.
\end{aligned}$$

Assim, (3.10) é válida.

Logo, $(x, y, z, I, B, \delta) \in P_2$ e então $P_1 \subset P_2$.

Por outro lado, tem-se ainda, que $P_2 \not\subset P_1$. De fato, por (3.10), (3.14) e $0 \leq y_{ljs} \leq 1, \forall l, j, s$, uma solução com $y_{li,s-1} = 0, y_{ljs} = 0$ e $z_{lijs} = \frac{1}{2}$ para algum l, i, j e s pertence a P_2 . Porém, por (3.20) e pelo fato de que $z_{lijs} \geq 0, \forall l, i, j, s$, essa solução não pertence a P_1 . Portanto, a formulação CCPRL é melhor do que a formulação GLSPRL. $\square$

3.3.3 Modelo CCFLPRL

Partindo do modelo CCPRL, uma reformulação para o PDSL-LPR é proposta baseada no problema de localização de facilidades (CCFLPRL). Nessa reformulação, a cada período é decidida a quantidade de cada item que será produzida para atender as demandas de cada um dos demais períodos do horizonte de planejamento. A reformulação via localização de facilidades para problemas de dimensionamento de lotes foi introduzida em Eppen e Martin (1987). Testes computacionais realizados em Guimarães, Klabjan e Almada-Lobo (2014) mostraram que essa reformulação pode melhorar o desempenho computacional de algoritmos do tipo *Branch-and-Cut*.

No modelo CCFLPRL, as variáveis x_{ljs} são substituídas por x_{ljst} para indicar a quantidade do item j produzida na linha l durante o microperíodo s para atender a demanda do período t . Os parâmetros λ_{ls} são introduzidos para indicar o período no qual o microperíodo s da linha l está contido e o parâmetros hc_{ljst} , definidos por (3.27), representam os custo de estoque e/ou atraso gerado pela produção do item j na linha l durante o microperíodo s para atendimento da demanda do período t .

$$hc_{ljst} = \begin{cases} h_j(t - \lambda_{ls}), & \text{se } \lambda_{ls} \leq t, \\ b_j(\lambda_{ls} - t), & \text{se } \lambda_{ls} > t. \end{cases} \quad (3.27)$$

O modelo CCFLPRL consiste em minimizar a função objetivo (3.28), sujeito as restrições (3.8), (3.15), (3.24), (3.25), (3.26) e (3.29) a (3.34). Essas restrições são oriundas da adaptação feita nas variáveis de produção (x_{ljst}).

• CCFLPRL:

$$\text{Min} \quad \sum_{l,j,s,t} h_{l,jst} x_{l,jst} + \sum_{l,i,j,s} s_{cli} z_{lijs} + \sum_{l,t} a_{cl} \delta_{lt} \quad (3.28)$$

$$\text{s. a.} \quad (3.8), (3.15), (3.24), (3.25), (3.26),$$

$$\sum_{l,s \geq f_{l \max\{1,t-s\}_j}} x_{l,jst} = d_{jt}, \quad \forall t, j \quad (3.29)$$

$$\sum_{j \in P_l, s \in S_{lt}, t' \geq 1} a_{lj} x_{l,jst'} + \sum_{i,j \in P_l, s \in S_{lt}} s_{li} z_{lijs} \leq C_{lt}, \quad \forall l, t \quad (3.30)$$

$$\sum_{t' \geq 1} x_{l,jst'} \leq \frac{C_{lt}}{a_{lj}} \sum_{i \in P_l} z_{lijs}, \quad \forall l, s, j \in P_l \quad (3.31)$$

$$\sum_{t' \geq 1} x_{l,jst'} \geq m_{lj} \sum_{i \in P_l, i \neq j} z_{lijs}, \quad \forall l, t, j \in P_l, s \in S_{lt} \setminus \{f_{lt}\} \quad (3.32)$$

$$\sum_{t' \geq 1} x_{l,jf_{lt}t'} \geq m_{lj} z_{ljjf_{lt}}, \quad \forall l, j \in P_l, t \quad (3.33)$$

$$x_{l,jst} \geq 0, \quad \forall l, j, s. \quad (3.34)$$

Incluí-se as restrições (3.18) e (3.35) a fim de que os microperíodos não produtivos sejam alocados após os microperíodos com produção dentro de cada período para cada linha de produção.

$$x_{l,jst'} \leq \frac{C_{lt}}{a_{lj}} \left(2 - \sum_{i \in P_l} z_{lij,s-1} - z_{ljj,s} \right) \quad \forall l, j, t', t, s = f_{lt}, \dots, u_{lt}. \quad (3.35)$$

3.3.4 Modelo CLSDPRL

O modelo CLSD (Haase (1996)) também foi estendido para lidar com o PDSL-LPR (CLSDPRL). Nesse modelo, os períodos não são subdivididos em microperíodos, mas restrições e variáveis são introduzidas a fim de se obter a ordem de produção dos itens em cada linha de produção.

Seja Ψ_{lt} um parâmetro para indicar o número máximo de itens que podem ser produzidos na linha l durante o período t . Assim, de acordo com a notação utilizada no modelo GLSPRL, tem-se $\Psi_{lt} = u_{lt} - f_{lt} + 1$. Considere as seguintes variáveis de decisão:

$x_{l,jt}$: quantidade produzida do item j na linha l durante o período t ;

I_{jt} : quantidade em estoque do item j ao final do período t ;

B_{jt} : quantidade de demanda atrasada do item j ao final do período t ;

$V_{l,jt}$: ordem do item j na sequência de produção da linha l durante o período t ;

$y_{l,jt}$: 1, se o item j é o primeiro item a ser produzido na linha l durante o período t e 0, caso contrário;

$z_{lij t}$: 1, se existe troca de produção do item i para o item j na linha l durante o período t e 0, caso contrário;

δ_{lt} : 1, se a linha l é ativada no período t e 0, caso contrário.

O modelo CLSDPRL consiste em minimizar a função objetivo (3.36) sujeito as restrições (3.37) a (3.53).

• CLSDPRL:

$$\text{Min} \sum_{t,j} (h_j I_{jt} + b_j B_{jt}) + \sum_{l,t,i,j} sc_{lij} z_{lij t} + \sum_{l,t} oc_l \delta_{lt} \quad (3.36)$$

$$\text{s.a.} \quad I_{j,t-1} - B_{j,t-1} + \sum_{l \in L_j} x_{lj t} = d_{jt} + I_{jt} - B_{jt}, \quad \forall j, t \quad (3.37)$$

$$\sum_{j \in P_l} a_{lj} x_{lj t} + \sum_{i,j \in P_l} st_{lij} z_{lij t} \leq C_{lt}, \quad \forall l, t \quad (3.38)$$

$$x_{lj t} \leq \frac{C_{lt}}{a_{lj}} \left(y_{lj t} + \sum_{i \in P_l \setminus \{j\}} z_{lij t} \right), \quad \forall l, j, t \quad (3.39)$$

$$\sum_{j \in P_l} y_{lj t} = \delta_{lt}, \quad \forall l, t \quad (3.40)$$

$$y_{lj t} + \sum_{i \in P_l \setminus \{j\}} z_{lij t} \geq \sum_{i \in P_l \setminus \{j\}} z_{lji t}, \quad \forall l, j, t \quad (3.41)$$

$$x_{lj t} \geq m_{lj} \left(y_{lj t} + \sum_{i \in P_l \setminus \{j\}} z_{lij t} \right), \quad \forall l, j, t \quad (3.42)$$

$$\sum_{i,j \in P_l, i \neq j} z_{lij t} \leq (\Psi_{lt} - 1) \delta_{lt}, \quad \forall l, t \quad (3.43)$$

$$y_{lj t} + \sum_{i \in P_l \setminus \{j\}} z_{lij t} \leq \delta_{lt}, \quad \forall l, j, t \quad (3.44)$$

$$V_{lj t} \geq V_{lit} + 1 - \Psi_{lt}(1 - z_{lij t}), \quad \forall l, t, i, j, i \neq j \quad (3.45)$$

$$\sum_l r_{kl} \delta_{lt} \leq R_{kt}, \quad \forall k, t \quad (3.46)$$

$$I_{jt} \leq \sum_{\alpha=t+1}^{t+sl_j} d_{j\alpha}, \quad \forall j, t \quad (3.47)$$

$$B_{jT} = 0, \quad \forall j \quad (3.48)$$

$$I_{jt} \geq 0, B_{jt} \geq 0, \quad \forall j, t \quad (3.49)$$

$$x_{lj t}, V_{lj t} \geq 0, \quad \forall l, j, t \quad (3.50)$$

$$y_{lj t} \in \{0, 1\}, \quad \forall l, j, t \quad (3.51)$$

$$z_{lij t} \in \{0, 1\}, \quad \forall l, i, j, t \quad (3.52)$$

$$\delta_{lt} \in \{0, 1\}, \quad \forall l, t. \quad (3.53)$$

A função objetivo (3.36) e as restrições (3.37) a (3.42) são análogas aos demais modelos, assim como (3.46), (3.47), (3.48) e (3.49). As restrições (3.43) definem o número máximo

$(\Psi_{lt} - 1)$ de trocas de itens que podem ser realizadas no período t e na linha l . As restrições (3.44) garantem que os itens só podem ser produzidos em linhas ativadas, enquanto que as restrições (3.45) são as restrições de Miller-Tucker-Zemlin (MTZ) propostas para eliminação de subciclos no problema do Caixeiro Viajante.

3.4 Um exemplo ilustrativo

Considere um cenário composto de duas linhas de produção (L1 e L2) que compartilham um único recurso produtivo (R1), quatro itens (P1, ..., P4) e três períodos. Suponha que os itens P1 e P2 são produzidos exclusivamente na linha L1 e os demais itens são produzidos exclusivamente na linha L2. É necessária a utilização de uma unidade do recurso R1 para montagem de cada linha de produção, porém só existe uma unidade disponível de R1.

Considere o tempo de produção unitária de cada item em cada linha (a_{jl}) e os lotes mínimos de produção (m_{jl}) como sendo iguais a 1 para todos os itens e todas as linhas. Os custos de preparação para produção (sc_{lij}) escolhidos aleatoriamente em $U[20, 40]$ e o tempos de preparação (st_{lij}) escolhidos aleatoriamente em $U[10, 20]$. Os custos unitários de estoque (h_j) e de atraso no atendimento das demandas (b_j) foram fixados em 5 e 50, respectivamente, para todos os itens. A capacidade produtiva foi fixada em $C_{lt} = 130, \forall l, t$ e não se considera perecibilidade dos produtos e nem custo para abertura da linhas de produção. As demandas são dadas na Tabela 3.

Tabela 3 – Exemplo Ilustrativo - demandas dos clientes

Item/Período	1	2	3
P1	5	8	4
P2	8	9	2
P3	10	5	5
P4	5	3	7

A Figura 4 compara a solução ótima do exemplo ilustrativo considerando a limitação do recurso R1 (PDSL-LPR) e sem considerar essa limitação (PDSL). As linhas indicam a sincronização da utilização do recurso R1 nos diferentes períodos. É possível notar que no PDSL-LPR apenas uma linha de produção pode ser montada em cada período, de modo que, apenas dois itens podem ser produzidos em cada período, enquanto que no tradicional LSP todas as linhas de produção permanecem ativas e então, todos os itens podem ser produzidos num mesmo período. Assim, a quantidade de produtos entregues com atraso e os níveis de estoque são substancialmente mais elevados no PDSL-LPR do que no tradicional LSP.

Figura 4 – Comparação entre o PDSL em linhas paralelas e o PDSL-LPR

MÉTODOS HEURÍSTICOS DE SOLUÇÃO

O PDSL pertence à classe de problemas NP-difícil (James e Almada-Lobo (2011)), de modo que a solução de instâncias de médio/grande porte via algoritmos exatos pode ser computacional inviável. A análise de desempenho dos modelos propostos nesta tese, apresentada na Seção 6.3.2 do Capítulo 6, mostra a dificuldade encontrada por um resolvedor comercial de alto desempenho na obtenção de boas soluções em tempo computacional aceitável, indicando a necessidade de obtenção de procedimentos heurísticos para lidar com o problema em estudo.

Diversos procedimentos são propostos a fim de obter soluções de boa qualidade em tempo computacional aceitável para o PDSL-LPR. A análise de desempenho dos modelos propostos (Seção 6.3.2 do Capítulo 6) aponta que o modelo CLSDPRL é o mais promissor. Desse modo, o CLSDPRL foi utilizado como modelo base para o desenvolvimento das heurísticas que serão propostas. Como grande parte dos procedimentos heurísticos propostos na literatura baseiam-se em busca em vizinhança partindo de uma solução factível conhecida, este capítulo se inicia com o desenvolvimento de procedimentos para construção de soluções em baixo tempo computacional (Seção 4.1). Em seguida, na Seção 4.2 são propostas heurísticas de melhoramento baseadas na formulação matemática do problema e na Seção 4.3 propõem-se procedimentos combinando estratégias de meta-heurísticas com programação matemática.

4.1 Heurísticas Construtivas

4.1.1 Uma heurística baseada em decomposição

No problema considerado, as linhas de produção são especializadas, de modo que, para cada item existe uma única linha de produção apta a produzi-lo. Assumindo esta característica no modelo CLSDPRL, apenas as restrições (3.46) são responsáveis por relacionar as diferentes linhas de produção. Portanto, se $\bar{\delta}$ é um padrão factível de abertura de linhas, isto é, uma matriz $L \times T$ de valores binários satisfazendo as restrições (3.46) então, ao fixar os valores das variáveis

$\delta_{lt}, \forall l, t$ nos valores $\bar{\delta}_{lt}$ no modelo CLSDPRL, é obtido um problema que pode ser decomposto em L subproblemas com uma única linha de produção.

Assim, é proposta a heurística D nesta Seção, que consiste na utilização de um modelo (AL definido por (4.18) a (4.24)) para determinar um padrão factível de abertura de linhas $\bar{\delta}$ e, em seguida, fixar as variáveis δ_{lt} do modelo CLSDPRL nos valores $\bar{\delta}_{lt}$, gerando um problema ($SP(\bar{\delta})$) que pode ser decomposto em L subproblemas ($SP^l(\bar{\delta}), l = 1, \dots, L$) menores.

Note que, se $\bar{\delta}_{lt} = 0$, então não existe produção na linha l durante o período t , de modo que $x_{ljt} = y_{ljt} = 0, \forall l, j, t$ e $z_{lijt} = 0, \forall l, i, j, t$. Dessa forma, no período t do subproblema $SP^l(\bar{\delta})$ não existem variáveis binárias a serem otimizadas, donde segue que o número de períodos com variáveis a serem otimizadas em cada subproblema $SP^l(\bar{\delta})$ pode ser menor do que T .

Se $\bar{\delta}_{lt}$ é um padrão factível de aberturas de linhas, os subproblemas $SP^l(\bar{\delta})$ consistem em minimizar a função objetivo (4.1) sujeito as restrições (4.2) a (4.16).

• $SP^l(\bar{\delta})$:

$$f^l = \text{Min} \quad \sum_{t,j} (h_j I_{jt} + b_j B_{jt}) + \sum_{t,i,j} sc_{lij} z_{ij} + \sum_t ac_l \bar{\delta}_{lt} \quad (4.1)$$

$$\text{s.a.} \quad I_{j,t-1} - B_{j,t-1} + x_{jt} = d_{jt} + I_{jt} - B_{jt}, \quad \forall j, t \quad (4.2)$$

$$\sum_j a_{lj} x_{jt} + \sum_{i,j} st_{lij} z_{ij} \leq C_{lt}, \quad \forall t \quad (4.3)$$

$$x_{jt} \leq C_{lt} \left(y_{jt} + \sum_i z_{ij} \right), \quad \forall j, t \quad (4.4)$$

$$\sum_j y_{jt} = \bar{\delta}_{lt}, \quad \forall t \quad (4.5)$$

$$y_{jt} + \sum_i z_{ij} \geq \sum_i z_{jit}, \quad \forall j, t \quad (4.6)$$

$$x_{jt} \geq m_{lj} \left(\sum_i z_{ij} + y_{jt} \right), \quad \forall j, t \quad (4.7)$$

$$\sum_{i,j} z_{ij} \leq (\Psi_{lt} - 1) \bar{\delta}_{lt}, \quad \forall t \quad (4.8)$$

$$y_{jt} + \sum_i z_{ij} \leq \bar{\delta}_{lt}, \quad \forall j, t \quad (4.9)$$

$$V_{jt} \geq V_{it} + 1 - \Psi_{lt}(1 - z_{ij}), \quad \forall i, j, t \quad (4.10)$$

$$I_{jt} \leq \sum_{\alpha=t+1}^{t+sl_j} d_{j\alpha}, \quad \forall j, t \quad (4.11)$$

$$B_{jT} = 0, \quad \forall j \quad (4.12)$$

$$I_{jt}, B_{jt} \geq 0, \quad \forall j, t \quad (4.13)$$

$$x_{jt}, V_{jt} \geq 0, \quad \forall j, t \quad (4.14)$$

$$y_{jt} \in \{0, 1\}, \quad \forall j, t \quad (4.15)$$

$$z_{ij} \in \{0, 1\}, \quad \forall i, j, t. \quad (4.16)$$

Considere os parâmetros $\bar{s}^l$ e $\bar{st}^l$ como sendo o menor tempo de validade dentre os itens produzidos na linha l e a média aritmética entre os tempos de preparação para produção dos produtos da linha l , respectivamente. Considere ainda, para cada linha de produção l , os $T + 1 - \bar{s}^l$ intervalos compostos por $\bar{s}^l$ períodos consecutivos do horizonte de planejamento, a saber, $\{1, \dots, \bar{s}^l\}, \{2, \dots, \bar{s}^l + 1\}, \{3, \dots, \bar{s}^l + 2\}, \dots, \{T - \bar{s}^l, \dots, T - 1\}, \{T - \bar{s}^l + 1, \dots, T\}$.

A ideia central do modelo *AL* é garantir que cada linha l seja ativada em cada intervalo do tipo $\{t', \dots, t' + \bar{s}^l\}$ com $t' = 1, \dots, T - \bar{s}^l + 1$, um número de vezes que possibilite o atendimento de toda a demanda associada à linha l dentro do próprio intervalo. Para tanto, estima-se o número de vezes no qual a linha l deve ser ativada durante o intervalo $\{t', \dots, t' + \bar{s}^l - 1\}$ ($AE_{lt'}$), conforme a equação (4.17).

$$AE_{lt'} = \left\lceil \frac{\sum_{j \in P_l} \sum_{\alpha=t'}^{t'+\bar{s}^l-1} a_{lj} d_{j\alpha} + \Theta \bar{s}^l \frac{|P_l|}{T} \bar{s}^l}{\frac{1}{\bar{s}^l} \sum_{\alpha=t'}^{t'+\bar{s}^l-1} C_{l\alpha}} \right\rceil. \quad (4.17)$$

O numerador da equação (4.17) representa a soma entre: o tempo consumido na produção de toda a demanda dos itens do conjunto P_l nos períodos $t', \dots, t' + \bar{s}^l - 1$, e uma estimativa do tempo consumido com as preparações para produção durante os mesmos períodos. O parâmetro $\Theta \geq 0$ controla o tempo estimado para as preparações durante cada intervalo, enquanto que o denominador representa a média aritmética entre as capacidades produtivas da linha l no intervalo $t', \dots, t' + \bar{s}^l - 1$.

Considere as variáveis binárias w_{ljt} , onde $w_{ljt} = 1$ se o item j é produzido na linha l durante o período t e $w_{ljt} = 0$, caso contrário. O modelo *AL* consiste em minimizar a soma dos custos de estoque, atrasos e montagem das linhas (4.18) sujeito as restrições (3.37), (3.46), (3.47) e (4.19) a (4.24).

• *AL* :

$$\text{Min} \quad \sum_{t,j} (h_j I_{jt} + b_j B_{jt}) + \sum_{l,t} oc_l \delta_{lt} \quad (4.18)$$

$$\text{s.a.} \quad (3.37), (3.46), (3.47)$$

$$m_{lj} w_{ljt} \leq x_{ljt} \leq \frac{C_{lt}}{a_{lj}} w_{ljt}, \quad \forall l, j, t \quad (4.19)$$

$$\sum_j w_{ljt} \leq \Psi_{lt} \delta_{lt}, \quad \forall l, t \quad (4.20)$$

$$\sum_{t=m}^{m+\bar{s}^l-1} \delta_{lt} \geq AE_{lm}, \quad \forall l, m = 1, \dots, T - \bar{s}^l + 1 \quad (4.21)$$

$$x_{ljt} \geq 0, \quad \forall l, j, t \quad (4.22)$$

$$w_{ljt} \in \{0, 1\}, \quad \forall l, j, t \quad (4.23)$$

$$\delta_{lt} \in \{0, 1\}, \quad \forall l, t. \quad (4.24)$$

As restrições (4.19) exigem um lote mínimo de produção e garantem que os itens só

podem ser produzidos em linhas preparadas, enquanto que as restrições (4.20) definem o número máximo de itens que podem ser produzidos em cada linha e período e garantem que só existe produção em linhas montadas. As restrições (4.21) referem-se ao número mínimo de vezes que cada linha deve ser montada em cada intervalo de $\overline{sl}^l$ períodos consecutivos.

O modelo *AL* considera apenas a parte de dimensionamento de lotes do problema e uma previsão do tempo gasto com as preparações é considerada para a estimar o número de vezes que cada linha de produção deve ser ativada. Desse modo, o modelo *AL* pode ser mais facilmente resolvido do que modelo *CLSDPRL*, mas encontrar a solução ótima pode ser inviável para instâncias de grande porte. Contudo, basta encontrar uma solução factível do modelo *AL* para obter um padrão factível de abertura de linhas.

Um pseudocódigo para a heurística *D* é dado no Algoritmo 1, onde os parâmetros em negrito representam vetores ou matrizes de parâmetros, por exemplo, **d** é a matriz $J \times T$ contendo os valores das demandas de todos os itens em todos os períodos ($d_{jt}, \forall j, t$).

Algoritmo 1 – Heurística D

- 1: **procedimento** $D(\Theta, L, T, J, \mathbf{d}, \overline{\mathbf{st}}, \overline{\mathbf{sl}}, \mathbf{a}, \mathbf{C}, AL, SP)$
 - 2: $f \leftarrow 0$;
 - 3: Calcule $AE_{lm}, \forall l, m$ de acordo com a equação (4.17);
 - 4: Determine $\bar{\delta}$, uma solução factível (ótima ou não) para o modelo *AL*;
 - 5: **para todo** l **faça**
 - 6: Resolva o modelo $SP^l(\bar{\delta})$, obtendo uma solução $(y^*, z^*, x^*, I^*, B^*)$ com valor f^l ;
 - 7: $y_{ljt} \leftarrow y_{jt}^*, \forall j, t$;
 - 8: $z_{lijt} \leftarrow z_{ijt}^*, \forall i, j, t$;
 - 9: $x_{ljt} \leftarrow x_{jt}^*, \forall j, t$;
 - 10: $I_{jt} \leftarrow I_{jt}^*, \forall j, t$;
 - 11: $B_{jt} \leftarrow B_{jt}^*, \forall j, t$;
 - 12: $f \leftarrow f + f^l$;
 - 13: **fim para**
 - 14: $\delta_{lt} \leftarrow \bar{\delta}_{lt}, \forall l, t$;
 - 15: **retorna** $(\mathbf{y}, \mathbf{z}, \mathbf{x}, \mathbf{I}, \mathbf{B}, \delta, f)$
 - 16: **fim procedimento**
-

4.1.2 Heurísticas Relax-and-Fix

Heurísticas do tipo *relax-and-fix* (RF) têm sido frequentemente utilizadas na solução de problemas de planejamento de produção oriundos de diversas aplicações industriais (Baldo *et al.* (2014) e Sel e Bilgen (2014)). Neste trabalho são propostas três heurísticas do tipo *relax-and-fix*, explorando diferentes partições das variáveis binárias do problema estudado.

Uma primeira heurística (RF1) consiste na adoção da tradicional partição do conjunto das variáveis binárias por períodos. Trata-se de um procedimento composto por T iterações, onde na iteração $t \in \{1, \dots, T\}$ as variáveis binárias referentes ao período incumbente são otimizadas,

enquanto que as variáveis referentes aos períodos anteriores são fixadas no valor incumbente e as variáveis referentes aos períodos posteriores são linearmente relaxadas. Assim, a heurística RF1 é fortemente baseada nos trabalhos de Dillenberger *et al.* (1994) e Escudero e Salmeron (2005).

A segunda heurística (RF2) explora também uma partição baseada em períodos. A cada iteração, as variáveis referentes a dois períodos produtivos são otimizadas. Mais especificamente, na iteração α , as variáveis binárias y_{ljt} , z_{lijt} e δ_{lt} com $t \in \{\alpha, T - \alpha + 1\}$ são consideradas como binárias e são otimizadas, enquanto que as variáveis com $t < \alpha$ e com $t > T - \alpha + 1$ são fixadas no valor incumbente e as variáveis com $\alpha < t < T - \alpha + 1$ são linearmente relaxadas.

Por fim, é proposta a heurística RFH do tipo *relax-and-fix*, que utiliza uma estratégia de decomposição do problema similar à estratégia utilizada na Heurística D. Nessa heurística são primeiramente determinadas as linhas de produção que serão ativadas em cada período do horizonte de planejamento e, em seguida, são tomadas as decisões de dimensionamento e de sequenciamento de lotes, lembrando que, se as variáveis δ_{lt} são fixadas no modelo CLSDPRL, então o problema resultante pode ser decomposto por linhas de produção, desde que para cada item exista uma única linha apta a produzi-lo.

A fim de introduzir essa heurística, apresenta-se a reformulação CLSDPRL^w do modelo CLSDPRL ((3.36)-(3.53)), onde as variáveis binárias que indicam se um item é produzido num determinado período são explicitadas. Considere, então, as variáveis binárias w_{ljt} , onde $w_{ljt} = 1$ se o item j é produzido na linha l durante o período t e $w_{ljt} = 0$, caso contrário. Assim, tem-se que:

$$w_{ljt} = y_{ljt} + \sum_i z_{ijlt}, \forall l, j, t. \quad (4.25)$$

O modelo CLSDPRL^w pode ser obtido a partir do modelo CLSDPRL, introduzindo as restrições (4.25) e (4.27) e substituindo (3.39) por (4.26), onde:

$$x_{ljt} \leq C_{lt} w_{ljt}, \quad \forall l, j, t \quad (4.26)$$

$$w_{ljt} \in \{0, 1\}, \quad \forall l, j, t. \quad (4.27)$$

O procedimento RFH é definido do seguinte modo. Um subproblema (S_1) é resolvido na primeira iteração onde as variáveis $\delta_{lt}, \forall l, t$ e $w_{ljt}, \forall l, j, t$ do modelo CLSDPRL^w são consideradas como binárias e as demais variáveis, $y_{ljt}, \forall l, j, t$ e $z_{lijt}, \forall l, i, j, t$, são linearmente relaxadas. Na segunda iteração, soluciona-se um subproblema S_2 onde as variáveis $\delta_{lt}, \forall l, t$ são fixadas nos valores incumbentes (obtidos no subproblema S_1) e as variáveis $w_{ljt}, \forall l, j, t$, $y_{ljt}, \forall l, j, t$ e $z_{lijt}, \forall l, i, j, t$ são consideradas como binárias. Como na segunda iteração, as variáveis $\delta_{lt}, \forall l, t$ estão fixadas então, o subproblema S_2 pode ser decomposto por linhas de produção.

O subproblema S_1 , consiste em minimizar (3.36), sujeito as restrições (3.37), (3.38), (3.40) a (3.50), (4.25), (4.26), (4.27), (3.53), $0 \leq y_{ljt} \leq 1, \forall l, j, t$, e $0 \leq z_{lijt} \leq 1, \forall l, i, j, t$.

Sejam $\bar{\delta}_{lt}, l = 1, \dots, L, t = 1, \dots, T$, as soluções de S_1 referentes às variáveis $\delta_{lt}, l = 1, \dots, L, t = 1, \dots, T$. Então, S_2 pode ser decomposto em L problemas S_2^l , com $l \in \{1, \dots, L\}$.

Para cada l , o subproblema S_2^l é dado por:

$$\begin{aligned}
 \bullet SP_2^l: \quad & f^l = \text{Min} \quad \sum_{t,j} (h_j I_{jt} + b_j B_{jt}) + \sum_{t,i,j} sc_{lij} z_{ijt} + \sum_t ac_l \bar{\delta}_{lt} \\
 \text{s.a.} \quad & I_{j,t-1} - B_{j,t-1} + x_{jt} = d_{jt} + I_{jt} - B_{jt}, \quad \forall j, t \\
 & \sum_j a_{lj} x_{jt} + \sum_{i,j} st_{lij} z_{ijt} \leq C_{lt}, \quad \forall t \\
 & x_{jt} \leq C_{lt} w_{jt}, \quad \forall j, t \\
 & w_{jt} = y_{jt} + \sum_j z_{ijt}, \quad \forall j, t \\
 & \sum_j y_{jt} = \bar{\delta}_{lt}, \quad \forall t \\
 & y_{jt} + \sum_i z_{ijt} \geq \sum_i z_{jit}, \quad \forall j, t \\
 & x_{jt} \geq m_{lj} \left(\sum_i z_{ijt} + y_{jt} \right), \quad \forall j, t \\
 & \sum_{i,j} z_{ijt} \leq (\Psi_{lt} - 1) \bar{\delta}_{lt}, \quad \forall t \\
 & y_{jt} + \sum_i z_{ijt} \leq \bar{\delta}_{lt}, \quad \forall j, t \\
 & V_{jt} \geq V_{it} + 1 - \Psi_{lt} (1 - z_{ijt}), \quad \forall i, j, t \\
 & I_{jt} \leq \sum_{\alpha=t+1}^{t+sl_j} d_{j\alpha}, \quad \forall j, t \\
 & B_{jT} = 0, \quad \forall j \\
 & I_{jt}, B_{jt} \geq 0, \quad \forall j, t \\
 & x_{jt}, V_{jt} \geq 0, \quad \forall j, t \\
 & w_{jt} \in \{0, 1\}, \quad \forall j, t \\
 & y_{jt} \in \{0, 1\}, \quad \forall j, t \\
 & z_{ijt} \in \{0, 1\}, \quad \forall i, j, t.
 \end{aligned}$$

O valor de função objetivo referente à solução encontrada via procedimento RFH é dado por $f = \sum_{l=1}^L f^l$. Note que as decisões com respeito à abertura de linhas de produção são tomadas na primeira iteração (subproblema S_1), levando em consideração a parte de dimensionamento de lotes do problema. Em seguida, na segunda iteração, as variáveis de dimensionamento ($w_{l,jt}$) são novamente otimizadas, agora em conjunto com as variáveis binárias de sequenciamento. Assim, as variáveis $w_{l,jt}$ são otimizadas tanto na primeira, quanto na segunda iteração, caracterizando um procedimento com sobreposição.

4.2 Heurísticas Fix-and-Optimize

Nesta Seção são propostos procedimentos iterativos de refinamento que partem de uma solução factível inicial e buscam encontrar melhores soluções por meio de busca em vizinhança.

São desenvolvidas três heurísticas do tipo *fix-and-optimize* (FO), sendo uma heurística de natureza determinística e duas de natureza estocástica. Assim como as heurísticas RF, as heurísticas FO baseiam-se na formulação matemática do problema e consistem em resolver, a cada iteração, um subproblema menor do que o original. Todos os procedimentos propostos nesta Seção utilizam o modelo CLSDPRL como base.

Primeiramente, propõe-se o procedimento FOTO, que consiste na liberação das variáveis binárias referentes a dois períodos produtivos para reotimização a cada iteração. Dada uma solução factível inicial, na primeira iteração são reotimizadas todas as variáveis binárias referentes aos períodos 1 e 2, enquanto que as demais variáveis binárias são fixadas no valor incumbente. Na segunda iteração, são reotimizadas todas as variáveis binárias referentes aos períodos 1 e 3 e assim sucessivamente, até que na iteração de número $T - 1$ são reotimizadas as variáveis referentes aos períodos 1 e T . Em seguida, na iteração T , reotimiza-se as variáveis binárias referentes aos períodos 2 e 3 e assim sucessivamente. A cada iteração, todas as variáveis contínuas ($x_{ljt}, V_{ljt}, \forall l, j, t, I_{jt}, B_{jt}, \forall j, t$) são reotimizadas.

A Figura 5 representa o esquema de funcionamento da heurística FOTO numa instância composta por quatro períodos produtivos. Os retângulos com fundo azul indicam que as variáveis binárias relacionadas ao respectivo período produtivo são liberadas para reotimização, enquanto que os retângulos com fundo branco indicam que as variáveis são fixadas no valor incumbente.

Figura 5 – Esquema de funcionamento da heurística FOTO

Um ciclo completo da heurística FOTO é composto por $\frac{T(T-1)}{2}$ iterações, sendo que, a cada $\frac{T(T-1)}{2}$ iterações realizadas, os conjuntos de variáveis a serem reotimizadas começam a se repetir. Na Figura 5, r indica o ciclo da heurística. As primeiras $\frac{T(T-1)}{2} = \frac{4 \cdot 3}{2} = 6$ iterações pertencem ao primeiro ciclo ($r = 1$), enquanto que as iterações de número 7 a 12 pertencem ao segundo ciclo ($r = 2$), e assim sucessivamente. O único critério de parada imposto foi o alcance do tempo máximo de execução. Os subproblemas associados a cada iteração podem ser solucionados utilizando-se um algoritmo *Branch-and-Cut* de um resolvidor.

O procedimento FOALE2 possui natureza estocástica, pois execuções distintas tendo como base uma mesma instância de teste podem resultar em diferentes soluções finais. Nesse procedimento, a cada iteração, reotimiza-se as variáveis binárias referentes a dois períodos produtivos que são selecionados aleatoriamente. Permite-se, inclusive, que os períodos selecionados numa iteração sejam iguais, de forma que, quando isso ocorrer, o subproblema associado a essa iteração poderá ser resolvido em menor tempo computacional.

Seja $(\bar{\delta}, \bar{y}, \bar{z})$ uma solução factível (incumbente) para o modelo CLSDPRL. Considere o parâmetro $\alpha_t = 1$ se o período t é selecionado para ter as suas variáveis binárias reotimizadas e $\alpha_t = 0$, caso contrário. A cada iteração da heurística FOALE2, o subproblema $\text{SUB}(\bar{\delta}, \bar{y}, \bar{z}, \alpha)$ reotimiza apenas as variáveis binárias referentes aos períodos selecionados ($\{t \in \{1, \dots, T\} | \alpha_t = 1\}$), enquanto que as demais variáveis binárias são fixadas no valor incumbente. Desse modo, o subproblema $\text{SUB}(\bar{\delta}, \bar{y}, \bar{z}, \alpha)$ consiste no modelo CLSDPRL adicionado das restrições (4.28) a (4.30).

• $\text{SUB}(\bar{\delta}, \bar{y}, \bar{z}, \alpha)$:

$$\begin{aligned} \text{Min} \quad & \sum_{t,j} (h_j I_{jt} + b_j B_{jt}) + \sum_{l,t,i,j} sc_{lij} z_{lij} + \sum_{l,t} oc_l \delta_{lt} \\ \text{s.a.} \quad & (3.37) - (3.53) \\ & (1 - \alpha_t) \delta_{lt} = (1 - \alpha_t) \bar{\delta}_{lt} \quad \forall l, t \end{aligned} \quad (4.28)$$

$$(1 - \alpha_t) y_{ljt} = (1 - \alpha_t) \bar{y}_{ljt} \quad \forall l, j, t \quad (4.29)$$

$$(1 - \alpha_t) z_{lij} = (1 - \alpha_t) \bar{z}_{lij} \quad \forall l, i, j, t. \quad (4.30)$$

Um pseudocódigo para o procedimento FOALE2 é apresentado no Algoritmo 2.

Algoritmo 2 – Heurística FOALE2

```

procedimento FOALE2( $TL, \bar{\delta}, \bar{z}, \bar{y}, \bar{f}$ )
2:   it ← 1;
   enquanto tempo ≤  $TL$  faça
4:     Sorteie aleatoriamente dois períodos produtivos  $t_1, t_2 \in \{1, \dots, T\}$ ;
     Faça  $\alpha_{t_1} = 1, \alpha_{t_2} = 1$  e  $\alpha_t = 0, \forall t \in \{1, \dots, T\} \setminus \{t_1, t_2\}$ ;
6:     Resolva o problema  $\text{SUB}(\bar{\delta}, \bar{y}, \bar{z}, \alpha)$ , obtendo uma solução  $(\delta^{it}, y^{it}, z^{it})$  com valor  $f^{it}$ ;
     Se  $f^{it} \leq \bar{f}$ , então  $\bar{f} \leftarrow f^{it}, \bar{\delta} \leftarrow \delta^{it}, \bar{y} \leftarrow y^{it}, \bar{z} \leftarrow z^{it}$ ;
8:     Atualize o tempo;
     it ← it + 1;
10:  fim enquanto
   retorna  $\bar{\delta}, \bar{z}, \bar{y}, \bar{f}$ 
12: fim procedimento

```

Quando cada subproblema $\text{SUB}(\bar{\delta}, \bar{y}, \bar{z}, \alpha)$ é resolvido até a otimalidade, o valor de função objetivo obtido em cada iteração f^{it} é sempre menor ou igual ao valor incumbente $\bar{f}$, pois a solução incumbente $(\bar{\delta}, \bar{y}, \bar{z})$ é factível para $\text{SUB}(\bar{\delta}, \bar{y}, \bar{z}, \alpha)$, para todo vetor α . Nesse caso, a condição da linha 6 do Algoritmo 2 torna-se dispensável. Porém, resolver todos os subproblemas

até a otimalidade pode ser computacionalmente inviável. Assim, normalmente, um limite de tempo de execução é imposto para solução de cada subproblema, fazendo com que seja possível obter $f^{it} > \bar{f}$.

Por fim, um último procedimento do tipo *fix-and-optimize*, FOALE3, é proposto. Trata-se de uma extensão do procedimento FOALE2 onde, a cada iteração, são aleatoriamente sorteados três períodos produtivos para terem as suas variáveis binárias reotimizadas, enquanto que as demais variáveis binárias são fixadas no valor incumbente. Dessa forma, um pseudocódigo para o procedimento FOALE3 pode ser obtido a partir do Algoritmo 2, adaptando as linhas 3 e 4 para considerar três períodos produtivos ao invés de apenas dois. O subproblema $\text{SUB}(\bar{\delta}, \bar{y}, \bar{z}, \alpha)$ continua válido para a heurística FOALE3.

4.3 Matheuristic INS

James e Almada-Lobo (2011) propuseram a *matheuristic iterative MIP-based neighbourhood search* (INS) com o objetivo de solucionar problemas de dimensionamento e sequenciamento de lotes com tempos e custos de preparação dependentes da sequência.

Assim como as heurísticas RF e FO, a INS baseia-se na ideia de decomposição do problema original em subproblemas menores que podem ser mais facilmente resolvidos via técnicas de programação matemática inteira mista. Partindo de uma solução factível inicial e definidas diferentes estruturas de vizinhança, uma busca local é realizada a cada iteração de acordo com a estrutura de vizinhança incumbente.

Dessa forma, a INS se assemelha à meta-heurística *variable neighbourhood search* (VNS) (Mladenović e Hansen (1997)), pois é composta por dois elementos principais: i) um conjunto de estruturas de vizinhança; e ii) um esquema de troca de estruturas de vizinhança.

Na INS as estruturas de vizinhança consistem em regras que possibilitam determinar um subconjunto das variáveis binárias para serem reotimizadas a cada iteração, ao passo que as demais variáveis são fixadas no valor incumbente.

Um pseudocódigo para a INS visando solucionar o PDSL-LPR é apresentado no Algoritmo 3, onde TL representa o tempo máximo de execução, I_{max} o número máximo de iterações, k indica a estrutura de vizinhança incumbente e it é o contador de iterações. O procedimento consiste num laço de repetição que se encerra ao atingir o tempo máximo de execução ou o número máximo de iterações.

Em cada iteração, realiza-se uma busca local na vizinhança V_k da solução incumbente $(\bar{\delta}, \bar{z}, \bar{y})$ visando obter uma solução candidata $(\delta^{it}, z^{it}, y^{it})$ a nova solução incumbente (linha 4 do Algoritmo 3). A solução candidata é aceita como nova solução incumbente quando o seu valor de função objetivo é menor do que o valor atual e, nesse caso, na próxima iteração, a primeira estrutura de vizinhança (chamada estrutura principal) será explorada. Já quando a

solução candidata não é aceita, troca-se a estrutura de vizinhança incumbente visando diversificar o processo de busca.

Algoritmo 3 – Matheuristic INS

```

1: procedimento INS( $V_1, \dots, V_{k_{max}}, \bar{\delta}, \bar{z}, \bar{y}, \bar{f}$ )
2:    $k \leftarrow 1$ ;
3:    $it \leftarrow 0$ ;
4:   enquanto tempo < TL e  $it < I_{max}$  faça
5:     Faça uma busca na vizinhança incumbente  $V_k$  da solução  $(\bar{\delta}, \bar{z}, \bar{y})$ , obtendo uma nova
       solução  $(\delta^{it}, z^{it}, y^{it})$  com valor  $f^{it}$  (conforme Algoritmo 4);
6:     se  $f^{it} < \bar{f}$  então
7:       Atualize o valor e a solução incumbentes:  $\bar{f} \leftarrow f^{it}$ ;  $\bar{\delta} \leftarrow \delta^{it}$ ;  $\bar{y} \leftarrow y^{it}$ ;  $\bar{z} \leftarrow z^{it}$ ;
8:       Retorne para a primeira estrutura de vizinhança:  $k \leftarrow 1$ ; [
9:     senão
10:      Troque a vizinhança incumbente: Se  $k < k_{max}$ , então  $k \leftarrow k + 1$ . Senão,  $k \leftarrow 1$ ; ]
11:    fim se
12:    Atualize o tempo disponível;
13:     $it \leftarrow it + 1$ ;
14:  fim enquanto
15:  retorna  $(\bar{\delta}, \bar{z}, \bar{y}, \bar{f})$ 
16: fim procedimento

```

A busca em vizinhança da linha 4 do Algoritmo 3 consiste na realização de um procedimento do tipo *fix-and-optimize*, onde subproblemas são iterativamente resolvidos considerando-se apenas uma parte das variáveis binárias do modelo CLSDPRL. As variáveis a serem reotimizadas são escolhidas conforme a estrutura de vizinhança incumbente e a fase de busca local é encerrada de acordo com critérios de parada que, normalmente, consistem na realização de um dado número de iterações consecutivas sem melhoria no valor da função objetivo e/ou na obtenção de uma melhor solução.

Sejam $P_1, \dots, P_{max}$ critérios de parada para a fase de busca local da *matheuristic* INS. Considere novamente o parâmetro $\alpha_t = 1$ se o período t é selecionado para ter as suas variáveis binárias reotimizadas e $\alpha_t = 0$, caso contrário. Considere, ainda, o parâmetro $\beta_l = 1$ se a linha l é selecionada para ter as suas variáveis binárias reotimizadas e $\beta_l = 0$, caso contrário. Na fase de busca local, a cada iteração, um subproblema $SUB(\delta^{it}, z^{it}, y^{it}, \alpha, \beta)$ é solucionado, onde algumas variáveis binárias do modelo CLSDPRL são reotimizadas e as demais variáveis binárias são fixadas nos valores incumbentes $(\delta^{it}, z^{it}, y^{it})$. O subproblema $SUB(\delta^{it}, z^{it}, y^{it}, \alpha, \beta)$ consiste no modelo CLSDPRL adicionado das restrições (4.31) a (4.33).

• $SUB(\bar{\delta}, \bar{z}, \bar{y}, \alpha, \beta)$:

$$\begin{aligned}
 & \text{Min} \quad \sum_{t,j} (h_j I_{jt} + b_j B_{jt}) + \sum_{l,t,i,j} sc_{lij} z_{lijt} + \sum_{l,t} oc_l \delta_{lt} \\
 & \text{s.a.} \quad (3.37) - (3.53) \\
 & \quad (1 - \alpha_t \beta_l) \delta_{lt} = (1 - \alpha_t \beta_l) \delta_{lt}^{it} \quad \forall l, t \quad (4.31)
 \end{aligned}$$

$$(1 - \alpha_t \beta_l) y_{ljt} = (1 - \alpha_t \beta_l) y_{ljt}^{it} \quad \forall l, j, t \quad (4.32)$$

$$(1 - \alpha_t \beta_l) z_{lijt} = (1 - \alpha_t \beta_l) z_{lijt}^{it} \quad \forall l, i, j, t. \quad (4.33)$$

A heurística *fix-and-optimize* utilizada na fase de busca local da INS é definida no Algoritmo 4.

Algoritmo 4 – Fase de busca local INS

```

1: procedimento FASE DE BUSCA LOCAL(V, P1, ..., Pmax,  $\bar{\delta}, \bar{z}, \bar{y}, \bar{f}$ )
2:    $it^{FO} \leftarrow 1$ ;
3:    $\delta^{it} \leftarrow \bar{\delta}$ ;  $y^{it} \leftarrow \bar{y}$ ;  $z^{it} \leftarrow \bar{z}$ ;  $f^{it} \leftarrow \bar{f}$ ;
4:   enquanto P1  $\vee \dots \vee$  Pmax não for atendido faça
5:     Determine  $\alpha$  e  $\beta$  de acordo com a estrutura de vizinhança V;
6:     Resolva SUB( $\delta^{it}, z^{it}, y^{it}, \alpha, \beta$ ), obtendo uma solução ( $\delta', y', z'$ ) com valor  $f'$ ;
7:     se  $f' \leq f^{it}$  então
8:       Atualize a solução incumbente:  $f^{it} \leftarrow f'$ ;  $\delta^{it} \leftarrow \delta'$ ;  $y^{it} \leftarrow y'$ ;  $z^{it} \leftarrow z'$ ;
9:     fim se
10:     $it^{FO} \leftarrow it^{FO} + 1$ ;
11:  fim enquanto
12:  retorna ( $\delta^{it}, z^{it}, y^{it}$ ),  $f^{it}$ 
13: fim procedimento
  
```

Neste trabalho, são propostas cinco estruturas de vizinhança visando a decomposição do problema original por períodos e por linhas de produção, para serem utilizadas na *matheuristic* INS visando solucionar o problema em estudo. As estruturas de vizinhança propostas definem como selecionar as variáveis binárias que serão reotimizadas a cada iteração da fase de busca local da INS, isto é, a cada iteração da heurística *fix-and-optimize*.

- V₁: Seleciona-se aleatoriamente três períodos do horizonte de planejamento, t_1, t_2 e t_3 , e libera-se todas as variáveis binárias referentes a esses períodos para serem reotimizadas. Nesse caso, tem-se $\alpha_{t_1} = \alpha_{t_2} = \alpha_{t_3} = 1$, $\alpha_t = 0, \forall t \in \{1, \dots, T\} \setminus \{t_1, t_2, t_3\}$ e $\beta_l = 1, \forall l$.
- V₂: Seleciona-se aleatoriamente três linhas de produção, l_1, l_2 e l_3 e libera-se todas as variáveis binárias referentes a essas linhas para serem reotimizadas. Assim, $\alpha_t = 1, \forall t$, $\beta_{l_1} = \beta_{l_2} = \beta_{l_3} = 1$ e $\beta_l = 0, \forall l \in \{1, \dots, L\} \setminus \{l_1, l_2, l_3\}$.
- V₃: Seleciona-se aleatoriamente um período $t' \in \{1, \dots, T-1\}$ e libera-se as variáveis binárias referentes aos períodos t' e $t' + 1$ para reotimização. Portanto, $\alpha_{t'} = \alpha_{t'+1} = 1$, $\alpha_t = 0, \forall t \in \{1, \dots, T\} \setminus \{t', t' + 1\}$ e $\beta_l = 1, \forall l$. No início de cada fase de busca local utilizando essa estrutura, todos os períodos possuem a mesma probabilidade de serem selecionados. Entretanto, essa probabilidade diminui de acordo com a frequência com a qual cada período foi selecionado durante a fase de busca local incumbente.
- V₄: Seleciona-se aleatoriamente um período t' no conjunto $\{2, \dots, T-1\}$ e libera-se as variáveis referentes aos períodos $t' - 1, t', t' + 1$ para reotimização. Logo, $\alpha_{t'-1} = \alpha_{t'} = \alpha_{t'+1} =$

1, $\alpha_t = 0, \forall t \in \{1, \dots, T\} \setminus \{t' - 1, t', t' + 1\}$ e $\beta_l = 1, \forall l$. Novamente, a probabilidade inicial de cada período t ser selecionado é a mesma, porém a probabilidade diminui de acordo com a frequência com que cada período foi escolhido durante o procedimento;

V₅: Seleciona-se aleatoriamente uma linha l_1 no conjunto $\{2, \dots, L - 1\}$. Em seguida, seleciona-se uma linha l_2 no conjunto $\{1, \dots, l_1 - 1\}$ e uma linha l_3 no conjunto $\{l_1 + 1, \dots, L\}$. As variáveis binárias referentes às linhas l_1 , l_2 e l_3 são liberadas para reotimização. Agora, tem-se que $\alpha_t = 1, \forall t$, $\beta_{l_1} = \beta_{l_2} = \beta_{l_3} = 1$ e $\beta_l = 0, \forall l \in \{1, \dots, L\} \setminus \{l_1, l_2, l_3\}$. A probabilidade inicial de cada linha l ser a primeira linha selecionada (l_1) é a mesma, porém essa probabilidade é reduzida de acordo com a frequência com que cada linha foi selecionada durante o procedimento *fix-and-optimize* incumbente.

Nas estruturas V₁ e V₂ as linhas de produção e os períodos são sorteados de maneira completamente aleatória e, em cada iteração, a probabilidade de cada linha e cada período ser sorteado é a mesma. Além disso, permite-se o sorteio de linhas e períodos coincidentes numa mesma iteração, produzindo assim, subproblemas menores compostos por uma ou duas linhas ou períodos, conforme o caso.

As estruturas V₃, V₄ e V₅ foram baseadas nas estruturas propostas em [James e Almada-Lobo \(2011\)](#). Em cada iteração, a probabilidade de cada período (ou linha de produção) ser selecionado para ter as suas variáveis binárias reotimizadas depende da frequência com que o período (linha) foi sorteado durante a fase de busca local incumbente. É utilizada uma estratégia similar à utilizada na meta-heurística *simulated annealing* (SA) para o cálculo das probabilidades de seleção de cada período (linha). Seja, por exemplo, f_t a frequência de ocorrência do período t durante a fase de busca local incumbente. A probabilidade de o período t ser sorteado, em cada iteração, é dada por $e^{-\frac{f_t}{\alpha}}$, onde α é um parâmetro com valor maior ou igual a 1, determinado de maneira empírica. Além disso, não se permite que um mesmo período (linha) seja selecionado em duas iterações consecutivas.

Quatro critérios são propostos para a interrupção da fase de busca local da INS. A fase de busca local é interrompida:

- P₁: Após a realização de cinco iterações consecutivas sem obtenção de uma melhor solução;
- P₂: Após a realização de três iterações consecutivas sem melhoria no valor da função objetivo;
- P₃: Quando todas as opções de períodos (ou linhas de produção, conforme a estrutura de vizinhança incumbente) forem testadas;
- P₄: Quando uma estrutura de vizinhança auxiliar está sendo explorada e uma melhor solução é encontrada.

Os critérios P₁ e P₂ impõem apenas um número máximo de iterações consecutivas que podem ser realizadas melhoria no valor da função objetivo, antes de proceder a troca da

estrutura incumbente. Quando o critério P_3 é satisfeito, é possível que uma solução ótima local foi encontrada e, portanto, a estrutura de vizinhança deve ser alterada.

Dado um procedimento INS utilizando as estruturas $V_{\varpi_1}, \dots, V_{\varpi_{k_{\max}}}$, define-se a primeira estrutura do procedimento V_{ϖ_1} como a estrutura principal e as demais estruturas ($V_{\varpi_2}, \dots, V_{\varpi_{k_{\max}}}$) como estruturas auxiliares. O critério P_4 é incluído quando se deseja enfatizar a estrutura principal durante o procedimento. Na presença desse critério, as demais estruturas possuem um papel auxiliar, sendo utilizadas apenas com a finalidade de escapar de ótimos locais. Assim, uma vez que uma melhor solução é encontrada e uma estrutura auxiliar está sendo utilizada, a fase de busca local é interrompida e a estrutura principal volta a ser explorada.

Denota-se por $\text{INS}(V_{\varpi_1}, \dots, V_{\varpi_{k_{\max}}}, P_{\theta_1}, \dots, P_{\theta_m})$ um procedimento do tipo INS, adotando-se as estruturas de vizinhança $V_{\varpi_1}, \dots, V_{\varpi_{k_{\max}}}$ e tal que, em cada fase de busca local, uma heurística do tipo *fix-and-optimize* é executada até que, pelo menos um dos critérios $P_{\theta_1}, \dots, P_{\theta_m}$ seja atendido.

Seis procedimentos do tipo INS são propostos combinando a utilização das estruturas de vizinhança e dos critérios de interrupção da fase de busca local desenvolvidos nesta Seção. Os procedimentos propostos são os seguintes:

- INS1: $\text{INS}(V_1, V_2, P_1)$ - as estruturas V_1 e V_2 exploram decomposições completamente aleatórias via períodos e linhas de produção, respectivamente. Cada fase de busca local é interrompida após a realização de cinco iterações do tipo *fix-and-optimize* sem melhoria no valor da função objetivo;
- INS2: $\text{INS}(V_1, V_2, P_1, P_4)$ - difere do anterior por enfatizar a estrutura V_1 durante o procedimento;
- INS3: $\text{INS}(V_3, V_4, V_5, P_2, P_3)$ - as estruturas V_3 e V_4 exploram decomposições via períodos adjacentes, enquanto que a estrutura V_5 via linhas de produção. A estrutura V_3 difere de V_4 apenas pelo número de períodos a serem reotimizados a cada iteração da fase de busca local. Os subproblemas obtidos utilizando V_3 podem ser resolvidos em menor tempo computacional quando comparado com os subproblemas obtidos utilizando V_4 . As fases de busca local são interrompidas quando três iterações consecutivas do tipo *fix-and-optimize* são realizadas sem obtenção de uma melhor solução ou quando todas as combinações de períodos (ou linhas de produção, conforme o caso) tenham sido testadas;
- INS4: $\text{INS}(V_3, V_4, V_5, P_2, P_3, P_4)$ - difere do anterior por enfatizar a estrutura V_3 durante o procedimento;
- INS5: $\text{INS}(V_4, V_5, P_2, P_3)$ - este procedimento difere do procedimento INS3 por não considerar a estrutura V_3 . Suponha que um mesmo período t tenha sido sorteado tanto adotando-se a estrutura V_3 , como V_4 . Assim, de acordo com V_3 , são liberadas para reotimização as variáveis referentes aos períodos t e $t + 1$, enquanto que, de acordo com V_4 , são liberadas

as variáveis referentes aos períodos $t - 1, t$ e $t + 1$. Assim, tem-se que, de certa forma, $V_3 \subset V_4$. Esse fato, motivou a realização de experimentos considerando apenas a “maior” estrutura de vizinhança, V_4 ;

- INS6: $\text{INS}(V_4, V_5, P_2, P_3, P_4)$ - difere do anterior por enfatizar a estrutura V_4 durante o procedimento.

RELAXAÇÃO LAGRANGIANA

De acordo com [Fisher \(2004\)](#), alguns problemas de difícil solução poderiam ser vistos como diversos problemas mais fáceis que são “complicados” após a inclusão de algumas restrições. Por exemplo, em [Toledo e Armentano \(2006\)](#) foi considerado o problema de dimensionamento de lotes (PDL) em máquinas paralelas que na ausência das restrições de capacidade, pode-se decompor o problema em diversos subproblemas com único item e capacidade ilimitada de produção. Assim, cada subproblema pode ser facilmente solucionado utilizando uma extensão do algoritmo de programação dinâmica proposto em [Wagner e Whitin \(1958\)](#) para o PDL em única máquina com capacidade ilimitada.

Abordagens baseadas em relaxação lagrangiana são caracterizadas pela relaxação (dualização) das restrições consideradas “complicadoras”, obtendo-se um subproblema que considera essas restrições apenas na função objetivo mediante uma penalização e, portanto, é mais fácil de ser resolvido ([Xiao et al. \(2015\)](#)).

Neste capítulo, apresenta-se dois esquemas de relaxação lagrangiana para o problema em estudo. Na Seção 5.1, apresenta-se uma adaptação de um esquema de decomposição lagrangiana proposto em [Xiao et al. \(2015\)](#), que consiste na duplicação das variáveis de produção (x_{ljt}) no modelo CLSDPRL, enquanto que, na Seção 5.2 apresenta-se um esquema de relaxação lagrangiana (LH) baseado no modelo CLSDPRL^w que consiste em dualizar as restrições do modelo que interligam as decisões de dimensionamento e de sequenciamento da produção. Em ambos os casos, o subproblema lagrangiano resultante pode ser decomposto em um problema de dimensionamento de lotes com decisões a respeito da abertura das linhas de produção e num problema de sequenciamento em linhas paralelas não relacionadas. O problema de sequenciamento pode ser decomposto por períodos, quando não considera-se a preservação do estado de preparação das linhas entre períodos adjacentes, e por linhas de produção, quando para cada item existe uma única linha apta a produzi-lo.

Na Seção 5.3 apresenta-se os métodos do subgradiente e do volume que foram utilizados

para solução do problema dual lagrangiano, enquanto que, na Seção 5.4 um procedimento baseado na formulação matemática do problema é proposto a fim de encontrar uma solução factível a partir da solução do subproblema lagrangiano.

5.1 Esquema de decomposição lagrangiana (LD)

Em Xiao *et al.* (2015) um esquema baseado em decomposição lagrangiana foi proposto para lidar com um problema de dimensionamento e sequenciamento de lotes em linhas paralelas e não relacionadas. No problema tratado em Xiao *et al.* (2015), todos os itens podem ser produzidos em todas as linhas, porém para cada item existe uma linha de produção de preferência, de modo que, deseja-se maximizar o bônus gerado pela produção dos itens em suas respectivas linhas de preferência, descontando-se os custos de preparação e estoque.

A decomposição lagrangiana proposta em Xiao *et al.* (2015) baseia-se na duplicação e dualização das variáveis de produção x_{ljt} . Nesta Seção, propõe-se o esquema de decomposição lagrangiana LD que é uma extensão do esquema proposto em Xiao *et al.* (2015) para lidar com o problema em estudo.

Considere o modelo CLSDPRL (dado por (3.36)-(3.53)) e seja A_{ljt} a quantidade produzida do item j na linha l durante o período t . Assim, podemos escrever as restrições (5.1) e (5.2) e as restrições de balanceamento de estoque (3.37) podem ser escritas como (5.3).

$$A_{ljt} = x_{ljt}, \quad \forall l, j, t \quad (5.1)$$

$$A_{ljt} \geq 0, \quad \forall l, j, t, \quad (5.2)$$

$$I_{j,t-1} - B_{j,t-1} + \sum_{l \in L_j} A_{ljt} = d_{jt} + I_{jt} - B_{jt}, \quad \forall j, t. \quad (5.3)$$

Além disso, é possível reformular as restrições (3.40), (3.43) e (3.44) do modelo CLSDPRL tornando as variáveis de sequenciamento (y_{ljt} e z_{lijt}) independentes das variáveis de abertura de linhas (δ_{lt}). Para tanto, substitui-se as restrições (3.40), (3.43) e (3.44) do modelo CLSDPRL pelas restrições (5.4), (5.5) e (5.6), respectivamente e inclui-se as restrições (5.7) para garantir que os itens sejam produzidos apenas em linhas ativadas.

$$\sum_{j \in P_l} y_{ljt} \leq 1, \quad \forall l, t, \quad (5.4)$$

$$\sum_{i \in P_l} \sum_{j \in P_l} z_{lijt} \leq \Psi_{lt} - 1, \quad \forall l, t, \quad (5.5)$$

$$y_{ljt} + \sum_{i \in P_l} z_{lijt} \leq 1, \quad \forall l, j \in P_l, t, \quad (5.6)$$

$$A_{ljt} \leq \frac{C_{lt}}{a_{lj}} \delta_{lt}, \quad \forall l, j \in P_l, t. \quad (5.7)$$

Para possibilitar a obtenção da decomposição lagrangiana considere a reformulação CLSDPRL^X do modelo CLSDPRL que consiste em minimizar a função objetivo (3.36), sujeito

as restrições (5.3), (3.38), (3.39), (5.4), (3.41), (3.42), (5.5), (5.6), (3.45) a (3.53), (5.1), (5.7) e (5.2).

As restrições (5.1) do modelo CLSDPRL^X são dualizadas usando os multiplicadores de Lagrange ϕ_{ljt} . Desse modo, obtém-se o subproblema lagrangiano (LD_ϕ) dado pela função objetivo (5.8) sujeito as restrições (5.3), (3.38), (3.39), (5.4), (3.41), (3.42), (5.5), (5.6), (3.45) a (3.53), (5.7) e (5.2).

• LD_ϕ :

$$\begin{aligned} z_\phi^{LD} = \text{Min} \quad & \sum_j \sum_t (h_j I_{jt} + b_j B_{jt}) + \sum_l \sum_i \sum_j \sum_t sc_{lij} z_{lijt} \\ & + \sum_l \sum_t ac_l \delta_{lt} + \sum_l \sum_j \sum_t \phi_{ljt} (A_{ljt} - x_{ljt}). \end{aligned} \quad (5.8)$$

s.a. (5.3), (3.38), (3.39), (5.4), (3.41), (3.42), (5.5), (5.6), (3.45) a (3.53), (5.7) e (5.2)

O subproblema lagrangiano (LD_ϕ) pode ser decomposto em um problema de dimensionamento de lotes com decisões a respeito da abertura de linhas de produção (LD_ϕ^1) e em um problema de sequenciamento em linhas paralelas (LD_ϕ^2) da seguinte forma.

• Problema de dimensionamento de lotes e abertura de linhas de produção (LD_ϕ^1):

$$\begin{aligned} \text{Min} \quad & \sum_j \sum_t (h_j I_{jt} + b_j B_{jt}) + \sum_l \sum_t ac_l \delta_{lt} + \sum_l \sum_j \sum_t \phi_{ljt} A_{ljt} \end{aligned} \quad (5.9)$$

s. a. (5.3), (3.46), (3.47), (3.48), (3.49), (3.53), (5.7) e (5.2)

• Problema de sequenciamento (LD_ϕ^2):

$$\begin{aligned} \text{Min} \quad & \sum_l \sum_i \sum_j \sum_t sc_{lij} z_{lijt} - \sum_l \sum_j \sum_t \phi_{ljt} x_{ljt} \end{aligned} \quad (5.10)$$

s. a. (3.38), (3.39), (5.4), (3.41), (3.42), (5.5), (5.6), (3.45), (3.50), (3.51) e (3.52)

No subproblema LD_ϕ^1 apenas as variáveis $\delta_{lt}, \forall l, t$ são binárias. Assim esse subproblema pode ser resolvido por um algoritmo *Branch-and-Cut* de um resolvidor comercial em razoável tempo computacional. Já o subproblema de sequenciamento LD_ϕ^2 pode ser decomposto tanto por linhas de produção, como por períodos.

5.2 Esquema de relaxação de restrições (LH)

Nesta Seção, introduz-se outro esquema lagrangiano, LH, que consiste na inserção e dualização de restrições que interligam as decisões de dimensionamento e abertura de linhas com as decisões de sequenciamento do problema. Dessa forma, assim como na abordagem LD, obtém-se um subproblema lagrangiano que pode ser decomposto em um subproblema de dimensionamento com abertura de linhas e em um subproblema de sequenciamento em linhas paralelas. Entretanto, na abordagem LH busca-se considerar mais informações no subproblema

de dimensionamento visando obter padrões de abertura de linhas que resultem em melhores soluções factíveis (conforme procedimento de factibilização proposta na Seção 5.4) e melhores limitantes duais.

Para a definir a abordagem LH utiliza-se o modelo CLSDPRL^w que foi introduzido na Seção 4.1.2 do Capítulo 4. Considere as variáveis binárias w_{ljt} onde $w_{ljt} = 1$ se o item j é produzido na linha l durante o período t e $w_{ljt} = 0$, caso contrário. Note que, cada item pode ser produzido em uma linha e período se, e somente se, este item for sequenciado nessa mesma linha e período. Logo, pode-se escrever as restrições (5.11).

$$w_{ljt} = y_{lij} + \sum_i z_{lij}, \quad \forall l, j, t. \quad (5.11)$$

O modelo CLSDPRL^w pode ser obtido a partir do modelo CLSDPRL ((3.36)-(3.53)) considerando as restrições (5.11), introduzindo as restrições de domínio $w_{ljt} \in \{0, 1\}, \forall l, j, t$ e substituindo algumas expressões da forma $y_{ljt} + \sum_i z_{lij}$ por w_{ljt} (conforme (5.11)). Assim, o modelo CLSDPRL^w consiste em minimizar a função objetivo (5.12) sujeito as restrições (5.13) a (5.27).

• CLSDPRL^w:

$$\text{Min} \quad \sum_j \sum_t (h_j I_{jt} + b_j B_{jt}) + \sum_l \sum_i \sum_j \sum_t sc_{lij} z_{lij} + \sum_l \sum_t ac_l \delta_{lt} \quad (5.12)$$

$$\text{s.a.} \quad I_{j,t-1} - B_{j,t-1} + \sum_l x_{ljt} - d_{jt} = I_{jt} - B_{jt}; \quad \forall j, t, \quad (5.13)$$

$$m_{lj} w_{ljt} \leq x_{ljt} \leq \frac{C_{lt}}{a_{lj}} w_{ljt}; \quad \forall l, j, t, \quad (5.14)$$

$$I_{jt} \leq \sum_{s=t+1}^{t+sl_j} d_{js}; \quad \forall j, t, \quad (5.15)$$

$$\sum_l r_{kl} \delta_{lt} \leq R_k; \quad \forall k, t, \quad (5.16)$$

$$\sum_j w_{ljt} \leq \Psi_{lt} \delta_{lt}; \quad \forall l, t, \quad (5.17)$$

$$\sum_j a_{lj} x_{ljt} + \sum_i \sum_j st_{lij} z_{lij} \leq C_{lt}; \quad \forall l, t, \quad (\lambda_{lt}) \quad (5.18)$$

$$w_{ljt} = y_{ljt} + \sum_i z_{lij}; \quad \forall l, j, t, \quad (\alpha_{ljt}) \quad (5.19)$$

$$y_{ljt} + \sum_i z_{lij} \geq \sum_i z_{lij}; \quad \forall l, j, t, \quad (5.20)$$

$$V_{ljt} \geq V_{lit} + 1 - \Psi_{lt} * (1 - z_{lij}); \quad \forall l, i, j, t, \quad (5.21)$$

$$\sum_j y_{ljt} \leq 1; \quad \forall l, t, \quad (5.22)$$

$$B_{jT} = 0; \quad \forall j, \quad (5.23)$$

$$I_{jt}, B_{jt} \geq 0; \quad \forall j, t, \quad x_{ljt} \geq 0; \quad \forall l, j, t, \quad (5.24)$$

$$\delta_{lt} \in \{0, 1\}; \quad \forall l, t, \quad w_{ljt} \in \{0, 1\}; \quad \forall l, j, t, \quad (5.25)$$

$$V_{ljt} \geq 0; \quad \forall l, j, t, \quad (5.26)$$

$$y_{ljt} \in \{0, 1\}; \quad \forall l, j, t, \quad z_{lij} \in \{0, 1\}; \quad \forall l, i, j, t. \quad (5.27)$$

Sejam $\lambda_{lt} \geq 0$ e α_{ljt} as variáveis duais associadas às restrições (5.18) e (5.19), respectivamente. O subproblema lagrangiano $LS(\lambda, \alpha)$ resultante da dualização dessas restrições é dado pela função objetivo (5.28) sujeito as restrições (5.13) a (5.17) e (5.20) a (5.27).

• $LS(\lambda, \alpha)$:

$$\begin{aligned} \bar{z}_{\lambda, \alpha} = \text{Min} \quad & \sum_j \sum_t (h_j I_{jt} + b_j B_{jt}) + \sum_l \sum_i \sum_j \sum_t sc_{lij} z_{lij} + \sum_l \sum_t ac_l \delta_{lt} \\ & + \sum_l \sum_t \lambda_{lt} \left(\sum_j a_{lj} x_{ljt} + \sum_i \sum_j st_{lij} z_{lij} - C_{lt} \right) \\ & + \sum_l \sum_j \sum_t \alpha_{ljt} \left(w_{ljt} - y_{ljt} - \sum_i z_{lij} \right) \end{aligned} \quad (5.28)$$

s. a. (5.13) a (5.17)
(5.20) a (5.27)

O subproblema lagrangiano $LS(\lambda, \alpha)$ pode ser decomposto em um problema de dimensionamento de lotes com decisões acerca da montagem das linhas de produção, $LS^1(\lambda, \alpha)$, consistindo da função objetivo (5.29) sujeito as restrições (5.13) a (5.17), (5.23), (5.24) e (5.25), e em um problema de sequenciamento em linhas paralelas, $LS^2(\lambda, \alpha)$, consistindo da função objetivo (5.30) sujeito as restrições (5.20) a (5.22), (5.26) e (5.27).

• Problema de dimensionamento de lotes com montagem de linhas de produção ($LS^1(\lambda, \alpha)$):

$$\begin{aligned} z_{\lambda, \alpha}^1 = \text{Min} \quad & \sum_j \sum_t (h_j I_{jt} + b_j B_{jt}) + \sum_l \sum_t ac_l \delta_{lt} \\ & + \sum_l \sum_j \sum_t \lambda_{lt} a_{lj} x_{ljt} + \sum_l \sum_j \sum_t \alpha_{ljt} w_{ljt} \end{aligned} \quad (5.29)$$

s. a. (5.13) a (5.17), (5.23), (5.24) e (5.25)

• Problema de sequenciamento ($LS^2(\lambda, \alpha)$):

$$\begin{aligned} z_{\lambda, \alpha}^2 = \text{Min} \quad & \sum_l \sum_i \sum_j \sum_t (sc_{lij} + \lambda_{lt} st_{lij} - \alpha_{ljt}) z_{lij} \\ & - \sum_l \sum_j \sum_t \alpha_{ljt} y_{ljt} \end{aligned} \quad (5.30)$$

s. a. (5.20) a (5.22), (5.26) e (5.27)

O valor ótimo ($\bar{z}_{\lambda, \alpha}$) do subproblema lagrangiano $LS(\lambda, \alpha)$ é dado por $\bar{z}_{\lambda, \alpha} = z_{\lambda, \alpha}^1 + z_{\lambda, \alpha}^2 - \sum_l \sum_t \lambda_{lt} C_{lt}$.

O problema $LS^1(\lambda, \alpha)$ é ainda NP-difícil, porém não considera o sequenciamento da produção e pode ser mais facilmente resolvido do que o problema original CLSDPRL. Utiliza-se

um algoritmo do tipo *Branch-and-Cut* de um resolvidor comercial munido de uma regra de ramificação que consiste em ramificar as variáveis δ_{lt} antes das variáveis w_{ljt} para resolver o problema $LS^1(\lambda, \alpha)$ (ver Seção 6.4 do Capítulo 6).

Para instâncias de grande porte, em algumas iterações dos métodos utilizados para resolver o problema dual, o subproblema $LS^1(\lambda, \alpha)$ pode não ser resolvido até a otimalidade. Neste caso, utiliza-se uma solução factível para prosseguir com as iterações lagrangianas. Nessas iterações, não obtém-se um novo limitante inferior (dual). Soluções factíveis foram utilizadas no lugar da solução ótima do subproblema lagrangiano em [Xiao et al. \(2015\)](#), onde o subproblema lagrangiano foi resolvido heurísticamente em todas as iterações.

O subproblema $LS^2(\lambda, \alpha)$ pode, ainda, ser decomposto num conjunto de $L \cdot T$ subproblemas com única linha de produção e único período produtivo. Portanto, o sub-problem $LS^2(\lambda, \alpha)$ pode ser facilmente resolvido utilizando um resolvidor comercial.

O problema de dimensionamento de lotes LD_ϕ^1 (obtido pela abordagem LD) é mais fácil de ser resolvido do que o problema $LS^1(\lambda, \alpha)$, uma vez que não considera as variáveis binárias de produção w_{ljt} e não possui limite máximo de número de itens que podem ser produzidos em cada período. Essa é a principal diferença entre as abordagens LH e LD.

A seguir, na Seção 5.3 apresenta-se o método do subgradiente e o algoritmo do volume que serão utilizados para solução dos problemas duais associados às abordagens LD e LH.

5.3 Resolvendo o problema dual lagrangiano

De acordo com a Definição 2.2 (Seção 2.2.2 do Capítulo 2), o problema dual lagrangiano consiste em determinar valores para os multiplicadores lagrangianos que oferecem o maior valor para limitante dual obtido por meio da resolução do subproblema lagrangiano. Sejam D^{LH} e D^{LD} os problemas duais obtidos pelas abordagens LH e LD, respectivamente. Então:

$$\begin{aligned} D^{LH} : \quad & \max_{\lambda \geq 0, \alpha} \bar{z}_{\lambda, \alpha} \\ D^{LD} : \quad & \max_{\phi} z_{\phi}^{LD}. \end{aligned}$$

Na Seção 5.3.1 apresenta-se o método do subgradiente para solução dos problemas D^{LH} e D^{LD} , enquanto que o algoritmo do volume é apresentado na Seção 5.3.2.

5.3.1 Método do subgradiente

- Resolvendo o problema D^{LH} :

O método subgradiente parte de vetores iniciais de multiplicadores $\lambda^0 \geq 0$ e α^0 e os atualiza de acordo com as expressões (5.31) e (5.32), onde β_v^u, S_v^u , $v = 1, 2$ são os tamanhos dos passos e os vetores subgradientes na iteração u , respectivamente.

$$\lambda^{u+1} = \max\{0, \lambda^u + \beta_1^u S_1^u\}; \quad \forall l, t, \quad (5.31)$$

$$\alpha^{u+1} = \alpha^u + \beta_2^u S_2^u; \quad \forall l, j, t. \quad (5.32)$$

Os vetores subgradientes são dados por

$$(S_1^u)_{lt} = \sum_j a_{lj} x_{ljt}^u + \sum_i \sum_j s_{lij} z_{lijt}^u - C_{lt}$$

e

$$(S_2^u)_{ljt} = w_{ljt}^u - y_{ljt}^u - \sum_i z_{lijt}^u,$$

onde x^u, w^u, y^u e z^u são as soluções ótimas do problema $LS(\lambda^u, \alpha^u)$.

Os valores adotados para os tamanhos dos passos $\beta_v^u, \forall u, v$ são de fundamental importância para a convergência do método do subgradiente. Em particular, é preciso que $\beta_v^u \rightarrow 0$, $v = 1, 2$ para garantir a convergência teórica. Os tamanhos dos passos foram atualizados conforme a equação (5.33), proposta em [Held, Wolfe e Crowder \(1974\)](#), onde $\bar{z}_{\lambda, \alpha}^u$ é a solução ótima de $LS(\lambda, \alpha)$ e z_f^u é um limitante primal para o problema obtido na iteração incumbente u por meio do procedimento de factibilização que será apresentado na Seção 5.4.

$$\beta_v^u = \sigma^u \frac{z_f^u - \bar{z}_{\lambda, \alpha}^u}{\|S_v^u\|^2}; \quad v = 1, 2, \quad (5.33)$$

O parâmetro $0 \leq \sigma^u \leq 2$ foi atualizado conforme o procedimento proposto [Carvalho e Nascimento \(2016\)](#) e testes foram realizados levando em consideração diversas configurações de parâmetros. Na Seção 6.5.3.1 do Capítulo 6 apresenta-se os testes realizados para determinar a melhor configuração para os parâmetros.

- Resolvendo o problema D^{LD} :

O método do subgradiente visando a solução do problema D^{LD} atualiza as variáveis duais conforme (5.34).

$$\phi_{ljt}^{u+1} \leftarrow \phi_{ljt}^u + s \left(A_{ljt}^u - x_{ljt}^u \right) \quad \forall l, j, t. \quad (5.34)$$

Os valores A_{ljt}^u e x_{ljt}^u são as soluções ótimas do subproblema lagrangiano na iteração u . Novamente, atribuímos o valor inicial zero para todas as variáveis duais, isto é, $\phi_{ljt} = 0, \forall l, j, t$.

O subgradiente na iteração u é dado por $S_{ljt}^u = A_{ljt}^u - x_{ljt}^u, \forall l, j, t$. O tamanho do passo s é calculado conforme (5.35), onde σ é um parâmetro no intervalo $[0, 2]$, z^f é o valor de uma solução factível e z^u é o valor do limitante dual incumbente.

$$s = \sigma \frac{z^f - z^u}{\|S\|^2}. \quad (5.35)$$

5.3.2 Algoritmo do volume

Além do algoritmo do subgradiente, utilizamos, nesta tese, o algoritmo do volume, descrito na Seção 2.2.2.2 do Capítulo 2, para solução dos problemas duais lagrangianos D^{LH} e D^{LD} obtidos nas heurísticas lagrangianas LH e LD, repectivamente.

- Resolvendo o problema D^{LH} :

O algoritmo do volume implementado, a menos do procedimento de factibilização, segue os passos descritos na Seção 2.2.2.2 do Capítulo 2. Mais especificamente, adotou-se os seguintes passos:

P0: $\bar{\lambda}_{lt} \leftarrow 0, \forall l, t, \bar{\alpha}_{ljt} \leftarrow 0, \forall l, j, t$ e $it \leftarrow 0$;

P1: Resolva $LS(\bar{\lambda}, \bar{\alpha})$, obtendo as soluções ótimas $\delta^1, \bar{x}, \bar{w}, \bar{y}$ e $\bar{z}$ e o valor ótimo $\overline{DB}$;

P2: $x^0 \leftarrow \bar{x}, w^0 \leftarrow \bar{w}, y^0 \leftarrow \bar{y}, z^0 \leftarrow \bar{z}, DB^0 \leftarrow \overline{DB}, it \leftarrow 1$;

P3: Se necessário, aplique o procedimento de factibilização (Seção 5.4) utilizando o padrão de abertura de linhas δ^{it} e determinando uma solução factível PB ;

P4: $(v_1^{it})_{lt} \leftarrow \sum_j a_{lj} \bar{x}_{ljt} + \sum_i \sum_j s t_{lij} \bar{z}_{lij} - C_{lt}, \forall l, t, (v_2^{it})_{ljt} \leftarrow \bar{w}_{ljt} - \bar{y}_{ljt} - \sum_i \bar{z}_{lij}, \forall l, j, t$;

P5: $\lambda^{it} \leftarrow \bar{\lambda} + s * v_1^{it}$ e $\alpha^{it} \leftarrow \bar{\alpha} + s * v_2^{it}$ onde s é o tamanho do passo;

P6: Resolva $LS(\lambda^{it}, \alpha^{it})$, obtendo as soluções $\delta^{it}, x^{it}, w^{it}, y^{it}$ e z^{it} e o valor ótimo DB^{it} ;

P7: $\bar{x} \leftarrow \gamma x^{it} + (1 - \gamma) \bar{x}, \bar{w} \leftarrow \gamma w^{it} + (1 - \gamma) \bar{w}, \bar{y} \leftarrow \gamma y^{it} + (1 - \gamma) \bar{y}, \bar{z} \leftarrow \gamma z^{it} + (1 - \gamma) \bar{z}$, onde γ é um escalar entre 0 e 1;

P8: Se $DB^{it} > \overline{DB}$, então $\bar{\lambda} \leftarrow \lambda^{it}, \bar{\alpha} \leftarrow \alpha^{it}$ e $\overline{DB} \leftarrow DB^{it}$;

P9: $it \leftarrow it + 1$ e volte ao passo P3.

No passo P0, atribui-se o valor zero para as variáveis duais. Já no passo P1, o subproblema lagrangiano $LS(\bar{\lambda}, \bar{\alpha})$ é resolvido por meio da decomposição num subproblema de dimensionamento de lotes com abertura de linhas de produção ($LS^1(\bar{\lambda}, \bar{\alpha})$) e num subproblema de sequenciamento ($LS^2(\bar{\lambda}, \bar{\alpha})$) e um limitante dual é obtido.

Um limitante primal é obtido no passo P3 por meio do procedimento de factibilização que será apresentado na Seção 5.4, ainda neste capítulo. O procedimento de factibilização é realizado numa iteração it apenas quando existe alteração na matriz de abertura de linhas, isto é, quando $\delta^{it} \neq \delta^{it-1}$.

As direções de busca, $v_i, i = 1, 2$, são atualizadas no passo P4, enquanto que as variáveis duais são atualizadas no passo P5. Considere a direção completa $v = (v_1, v_2)$, o tamanho do

passo s (utilizado em P5) é obtido por meio da equação 5.36.

$$s = \sigma \frac{PB - \overline{DB}}{\|v\|^2} \quad (5.36)$$

No passo P6 um novo limitante dual é obtido, novamente pela resolução do subproblema lagrangiano. Uma estimativa para as soluções primais é calculada no passo P7. De acordo com Barahona e Anbil (2000), o parâmetro γ utilizado em P7 é obtido da seguinte forma.

- i) Defina uma limitante superior γ_{max} ;
- ii) Calcule $\gamma_{opt} = \min_{\gamma} \{\|\gamma v^{it} + (1 - \gamma)\bar{v}\|\}$, onde $v^{it} = (v_1^{it}, v_2^{it})$ com $(v_1^{it})_{lt} = \sum_j a_{lj} x_{ljt}^{it} + \sum_i \sum_j st_{lij} z_{lijt}^{it} - C_{lt}, \forall l, t$, $(v_2^{it})_{ljt} = w_{ljt}^{it} - y_{ljt}^{it} - \sum_i z_{lijt}^{it}, \forall l, j, t$ e $\bar{v} = (\bar{v}_1, \bar{v}_2)$ com $(\bar{v}_1)_{lt} = \sum_j a_{lj} \bar{x}_{ljt} + \sum_i \sum_j st_{lij} \bar{z}_{lijt} - C_{lt}, \forall l, t$, $(\bar{v}_2)_{ljt} = \bar{w}_{ljt} - \bar{y}_{ljt} - \sum_i \bar{z}_{lijt}, \forall l, j, t$;
- iii) Se $\gamma_{opt} < 0$, então $\gamma = \frac{\gamma_{max}}{10}$. Caso contrário, $\gamma = \min\{\gamma_{max}, \gamma_{opt}\}$.

Uma fórmula pode ser obtida para o cálculo do parâmetro γ_{opt} . Considere a função $f(\gamma) = \|\gamma v^{it} + (1 - \gamma)\bar{v}\|^2$, onde $v^{it} = (v_1^{it}, v_2^{it})$ e $\bar{v} = (\bar{v}_1, \bar{v}_2)$. Por simplicidade, vamos considerar v^{it} e $\bar{v}$ como sendo vetores indexados por um único índice (k) e representar as suas componentes por $(v^{it})_k$ e $(\bar{v})_k$, respectivamente. Considerando a norma vetorial euclidiana, tem-se que:

$$\begin{aligned} f(\gamma) &= \sum_k (\gamma(v^{it})_k + (1 - \gamma)(\bar{v})_k)^2 \\ &= \sum_k (\gamma^2 (v^{it})_k^2 + 2\gamma(1 - \gamma)(v^{it})_k(\bar{v})_k + (1 - \gamma)^2 (\bar{v})_k^2) \\ &= \sum_k (\gamma^2 (v^{it})_k^2 + 2\gamma(v^{it})_k(\bar{v})_k - 2\gamma^2 (v^{it})_k(\bar{v})_k + (\bar{v})_k^2 - 2\gamma(\bar{v})_k + \gamma^2 (\bar{v})_k^2) \\ &= \sum_k [\gamma^2 ((v^{it})_k^2 - 2(v^{it})_k(\bar{v})_k + (\bar{v})_k^2) + 2\gamma((v^{it})_k(\bar{v})_k - (\bar{v})_k^2) + (\bar{v})_k^2] \\ &= \left[\sum_k ((v^{it})_k - (\bar{v})_k)^2 \right] \gamma^2 + 2 \left[\sum_k ((v^{it})_k(\bar{v})_k - (\bar{v})_k^2) \right] \gamma + \sum_k (\bar{v})_k^2. \end{aligned}$$

Assim, tem-se que $f(\gamma)$ é uma função quadrática de γ . Como $\sum_k ((v^{it})_k - (\bar{v})_k)^2 \geq 0$, segue que:

- i) se $\sum_k ((v^{it})_k - (\bar{v})_k)^2 = 0$, então $f(\gamma) \rightarrow -\infty$;
- ii) se $\sum_k ((v^{it})_k - (\bar{v})_k)^2 > 0$, então $f(\gamma)$ tem um ponto de mínimo dado por (5.37).

$$\gamma_{opt} = \frac{\sum_k ((\bar{v})_k^2 - (v^{it})_k(\bar{v})_k)}{\sum_k ((v^{it})_k - (\bar{v})_k)^2}. \quad (5.37)$$

Reescrevendo (5.37) utilizando as componentes originais ($v^{it} = (v_1^{it}, v_2^{it})$ e $\bar{v} = (\bar{v}_1, \bar{v}_2)$) dos vetores de direção, obtém-se (5.38).

$$\gamma_{opt} = \frac{\sum_l \sum_t ((\bar{v}_1)_{lt}^2 - (v_1^{it})_{lt}(\bar{v}_1)_{lt}) + \sum_l \sum_j \sum_t ((\bar{v}_2)_{ljt}^2 - (v_2^{it})_{ljt}(\bar{v}_2)_{ljt})}{\sum_l \sum_t ((v_1^{it})_{lt} - (\bar{v}_1)_{lt})^2 + \sum_l \sum_j \sum_t ((v_2^{it})_{ljt} - (\bar{v}_2)_{ljt})^2}. \quad (5.38)$$

O Algoritmo 5 apresenta o procedimento utilizado para a determinação do valor do parâmetro γ a cada iteração. O parâmetro $\gamma_{max} \in (0, 1)$ é um limitante superior para o valor de γ . Os testes apresentados em Barahona e Anbil (2000) utilizaram $\gamma_{max} = 0, 1$, porém outros valores podem ser utilizados.

Algoritmo 5 – Cálculo do parâmetro γ (passo P7 do método do volume - problema D^{LH})

```

1: procedimento CÁLCULO DO PARÂMETRO  $\gamma(\gamma_{max}, v^{it} = (v_1^{it}, v_2^{it}), \bar{v} = (\bar{v}_1, \bar{v}_2))$ 
2:   se  $\sum_l \sum_t ((v_1^{it})_{lt} - (\bar{v}_1)_{lt})^2 + \sum_l \sum_j \sum_t ((v_2^{it})_{ljt} - (\bar{v}_2)_{ljt})^2 > 0$  então
3:     Calcule  $\gamma_{opt}$  pela equação (5.38);
4:     se  $\gamma_{opt} < 0$  então
5:        $\gamma \leftarrow \frac{\gamma_{max}}{10}$ ;
6:     senão
7:        $\gamma \leftarrow \min\{\gamma_{opt}, \gamma_{max}\}$ ;
8:     fim se
9:   senão
10:     $\gamma \leftarrow \frac{\gamma_{max}}{10}$ ;
11:  fim se
12:  retorna  $\gamma$ 
13: fim procedimento

```

- Resolvendo o problema D^{LD} :

O problema dual D^{LD} foi solucionado da mesma forma que o problema D^{LH} . Mais especificamente, utilizamos o procedimento definido pelos passos P0' a P9'.

P0': $\bar{\phi}_{ljt} \leftarrow 0, \forall l, j, t$ e $it \leftarrow 0$;

P1': Resolva $LD_{\bar{\phi}}$, obtendo as soluções ótimas $\delta^1, \bar{x}$ e $\bar{A}$ e o valor ótimo $\overline{DB}$;

P2': $x^0 \leftarrow \bar{x}, A^0 \leftarrow \bar{A}, DB^0 \leftarrow \overline{DB}, it \leftarrow 1$;

P3': Se necessário, aplique o procedimento de factibilização utilizando o padrão de abertura de linhas δ^{it} e determinando uma solução factível com valor PB para o problema original;

P4': $v_{ljt}^{it} \leftarrow \bar{A}_{ljt} - \bar{x}_{ljt} \forall l, j, t$;

P5': $\phi^{it} \leftarrow \bar{\phi} + s * v^{it}$ onde s é o tamanho do passo;

P6': Resolva $LD_{\phi^{it}}$, obtendo as soluções δ^{it}, x^{it} e A^{it} e o valor ótimo DB^{it} (limitante dual incumbente);

P7': $\bar{x} \leftarrow \gamma x^{it} + (1 - \gamma)\bar{x}$ e $\bar{A} \leftarrow \gamma A^{it} + (1 - \gamma)\bar{A}$, onde γ é um escalar entre 0 e 1;

P8': Se $DB^t > \overline{DB}$, então $\bar{\phi} \leftarrow \phi^{it}$ e $\overline{DB} \leftarrow DB^{it}$;

P9': $it \leftarrow it + 1$ e volte ao passo P3.

O tamanho do passo, s , da etapa P5' é calculado conforme a equação (5.36) e o parâmetro γ do passo P7' é obtido pelo Algoritmo 6, onde $v_{ljt}^{it} = A_{ljt}^{it} - x_{ljt}^{it}$ e $\bar{v}_{ljt} = \bar{A}_{ljt} - \bar{x}_{ljt} \forall l, j, t$.

Algoritmo 6 – Cálculo do parâmetro γ (passo P7' do método do volume - problema D^{LD})

```

1: procedimento PARÂMETRO  $\gamma(\gamma_{max}, v^{it}, \bar{v})$ 
2:   se  $\sum_l \sum_j \sum_t \left( v_{ljt}^{it} - \bar{v}_{ljt} \right)^2 > 0$  então
3:     Calcule  $\gamma_{opt} = \frac{\sum_l \sum_j \sum_t \left( \bar{v}_{ljt}^2 - v_{ljt}^{it} \bar{v}_{ljt} \right)}{\sum_l \sum_j \sum_t \left( v_{ljt}^{it} - \bar{v}_{ljt} \right)^2}$ ;
4:     se  $\gamma_{opt} < 0$  então
5:        $\gamma \leftarrow \frac{\gamma_{max}}{10}$ ;
6:     senão
7:        $\gamma \leftarrow \min\{\gamma_{opt}, \gamma_{max}\}$ ;
8:     fim se
9:   senão
10:     $\gamma \leftarrow \frac{\gamma_{max}}{10}$ ;
11:  fim se
12:  retorna  $\gamma$ 
13: fim procedimento
```

5.4 Procedimento de factibilização da solução

Com o objetivo de determinar uma solução de boa qualidade, desenvolvemos um procedimento baseado na formulação matemática do problema que parte da solução ótima do subproblema lagrangiano e determina uma solução factível para o problema original. Seja δ^u uma solução ótima do subproblema lagrangiano ($LS^1(\lambda^u, \alpha^u)$ na abordagem LH e $LD^1(\phi^u)$ na abordagem LD) na iteração u . Considere o modelo CLSDPRL^w ((5.12)-(5.27)) introduzido na Seção 5.2. Se fixarmos as variáveis δ nos valores δ^u , isto é, se adicionarmos as restrições $\delta_{lt} = \delta_{lt}^u, \forall l, t$ ao modelo CLSDPRL^w, o problema resultante ($FP(\delta^u)$) pode ser decomposto por linhas de produção, uma vez que δ^u satisfaz as restrições (5.16).

Denota-se por $FP^l(\delta^u)$ o subproblema referente a linha de produção l . Note que se $\delta_{lt'}^u = 0$ para algum período t' , então não pode haver produção na linha l durante este período, assim, tem-se $x_{ljt'} = y_{ljt'} = w_{ljt'} = 0, \forall l, j$ e $z_{lijt'} = 0, \forall l, i, j$. Portanto, cada subproblema $FP^l(\delta^u)$ possui um número de períodos com produção menor ou igual a T .

Dessa forma, cada subproblema $FP^l(\delta^u)$ possui uma única linha de produção e possui um número de períodos produtivos menor (ou, no pior caso, igual) do que o problema original. Além disso, não se faz necessário resolver, completamente, o problema $FP(\delta^u)$ em todas as iterações lagrangianas, uma vez que, o valor da solução factível altera se, e somente se, o valor incumbente das variáveis δ é alterado. Assim, em cada iteração u , só é necessário resolver o subproblema $FP^l(\delta^u)$ se $\delta_{lt}^u \neq \delta_{lt}^{u-1}$ para algum t .

Testes computacionais preliminares mostraram que é possível permanecer algumas iterações sem resolver nenhum subproblema $FP^l(\delta^u)$, uma vez que, o padrão de aberturas de linhas δ^u não é alterado em todas as iterações lagrangianas. Desse modo, o tempo computacional total consumido pelo procedimento de factibilização proposto não é muito elevado.

A Figura 6 apresenta um esquema completo das abordagens lagrangianas de solução introduzidas nesta tese.

Figura 6 – Um esquema de funcionamento das abordagens lagrangianas propostas

RESULTADOS COMPUTACIONAIS

Apresenta-se neste Capítulo os resultados computacionais obtidos a fim de verificar a eficiência dos modelos e dos métodos de solução propostos nesta tese para solução do PDSL-LPR. Na Seção 6.1 são propostas instâncias de teste geradas aleatoriamente para representação de diversos cenários do problema em estudo e, em seguida, na Seção 6.2 descreve-se o ambiente utilizado para realização dos testes computacionais. Na Seção 6.3, apresentam-se as análises de sensibilidade e de desempenho dos modelos, enquanto que, na Seção 6.4, são propostas regras de ramificação das variáveis binárias do problema para melhorar a performance de algoritmos do tipo *Branch-and-Bound* para solução do PDSL-LPR. Na Seção 6.5, apresenta-se as análises de desempenho dos métodos de solução e, por fim, na Seção 6.6 apresenta-se uma análise comparativa entre os métodos que foram identificados como promissores.

6.1 Instâncias

Nesta Seção são apresentados dois conjuntos de instâncias de teste para o PDSL-LPR. O primeiro conjunto é formado por 95 exemplares gerados com o objetivo de estudar o comportamento dos modelos propostos frente a variações de parâmetros, enquanto que o segundo conjunto de instâncias é composto por 100 exemplares introduzidos com a finalidade de analisar a performance computacional dos modelos e dos métodos de solução desenvolvidos. Na Seção 6.1.1 apresenta-se as características utilizadas para a geração das instâncias do Conjunto 1, enquanto que na Seção 6.1.2 caracteriza-se as instâncias pertencentes ao Conjunto 2. Por fim, na Seção 6.2, descreve-se o ambiente utilizado na implementação computacional dos modelos e métodos e os equipamentos que foram utilizados para a realização dos experimentos computacionais.

6.1.1 Instâncias para análise de sensibilidade

As 95 instâncias do Conjunto 1 são propostas para possibilitar uma análise de sensibilidade dos modelos desenvolvidos, isto é, verificar como que variações nos valores dos

parâmetros de entrada, tais como, número de linhas, períodos, produtos e recursos afetam o desempenho computacional dos modelos propostos ao serem solucionados por um algoritmo do tipo *Branch-and-Cut* de um resolvidor comercial.

Utiliza-se a notação $p \in U[a, b]$ para indicar que um valor inteiro foi aleatoriamente escolhido para o parâmetro p com $a \leq p \leq b$ e $p \in U\{a_1, \dots, a_n\}$ para indicar que o valor do parâmetro p foi aleatoriamente escolhido no conjunto $\{a_1, \dots, a_n\}$. Seguindo a mesma notação introduzida na Tabela 1 do Capítulo 3, os exemplares de teste do Conjunto 1 foram gerados de acordo com as seguintes especificações de parâmetros:

1. Número de períodos $T = 10$, número de linhas de produção $L \in \{3, 5, 7, 9\}$, número de recursos $K = 6$ e quantidade do recurso k requerida para montagem da linha l , $r_{kl} \in U[0, 2], \forall t, k > 0$ e $r_{0t} \in U[5, 10], \forall t$;
2. Tempo de preparação $st_{lij} \in U[15, 45], \forall l, i, j \in P_l$, custo de preparação $sc_{lij} = 2 * st_{lij}, \forall l, i, j \in P_l$, lote mínimo $m_{lj} = 2, \forall l, j \in P_l$ e tempo de produção $a_{lj} = 1, \forall l, j$;
3. Custo unitário de estoque $h_j \in U[1, 10]$, custo unitário de atraso no atendimento das demandas (por período) $b_j = 10 * h_j, \forall j$ e tempo de perecibilidade (*shelf life*) $sl_j \in U[4, T]$;
4. Capacidade produtiva $C_{lt} = 480, \forall l, t$, correspondendo a um dia produtivo (8 horas = 480 minutos);
5. Número máximo de itens que podem ser produzidos em cada linha e em cada período $\Psi \in \{4, 6, 8, 10\}$ e fixou-se o mesmo valor para todas as linhas em todos os períodos, isto é, $\Psi_{lt} = \Psi, \forall l, t$;
6. O número de itens foi calculado como $J = \Psi(L + \alpha)$, onde α é um parâmetro escolhido no conjunto $\{0, 1, 2, 3\}$;
7. Custo de montagem das linhas de produção $ac_l = \sum_k rc_k * r_{kl}$, onde rc_k é um parâmetro positivo que representa o custo do recurso k . O valor de rc_k foi escolhido no conjunto $U[\phi^b, \phi^e], \forall k$ com ϕ^b e ϕ^e sendo parâmetros não negativos que representam os custos mínimo e máximo do recurso k ;
8. $d_{jt} \in U\left[0, \frac{C_{lt} - \max_{i,j} \{st_{lij}\} * \Psi_{lt} - \phi^d}{|P_l|}\right], \forall t, l, j \in P_l$, onde ϕ^d é um parâmetro não negativo que representa uma redução na demanda máxima de cada item em cada período e foi escolhido no conjunto $\{80, 90, 100, 110\}$;
9. $R_k = \max \{ \max_{l=1, \dots, L} \{r_{kl}\}, \phi^r \sum_{l \in [L]} r_{kl} \}, \forall k$, onde $\phi^r \in [0, 1]$ é um parâmetro para representar a redução total dos recursos;
10. Para cada item, existe uma única linha de produção, selecionada aleatoriamente, apta a produzi-lo.

Os exemplares de teste foram agrupados em seis classes caracterizadas de acordo com a escolha dos valores dos parâmetros $L, \Psi, \varphi^d, \varphi^r, \alpha, \phi^b$ e ϕ^e . Em cada classe, existem três grupos compostos por cinco exemplares cada. Existe, ainda, um grupo padrão, também composto por cinco instâncias de teste, que compõe todas as classes, caracterizado pela seguinte escolha de parâmetros

$$P_0 = (5, 6, 100, 0.8, 0, 0, 0) = (L, \Psi, \varphi^d, \varphi^r, \alpha, \phi^b, \phi^e).$$

A Tabela 4 apresenta os valores de parâmetros adotados em cada grupo e classe.

Tabela 4 – Parâmetros utilizados nas instâncias do Conjunto 1

		L	Ψ	φ^d	φ^r	α	ϕ^b	ϕ^e	J	Exemplares
	Grupo Padrão	5	6	100	0.8	0	0	0	30	5
Classe 1	Grupo 1	3	6	100	0.8	0	0	0	18	5
	Grupo 2	7	6	100	0.8	0	0	0	42	5
	Grupo 3	9	6	100	0.8	0	0	0	54	5
Classe 2	Grupo 1	5	4	100	0.8	0	0	0	20	5
	Grupo 2	5	8	100	0.8	0	0	0	40	5
	Grupo 3	5	10	100	0.8	0	0	0	50	5
Classe 3	Grupo 1	5	6	110	0.8	0	0	0	30	5
	Grupo 2	5	6	90	0.8	0	0	0	30	5
	Grupo 3	5	6	80	0.8	0	0	0	30	5
Classe 4	Grupo 1	5	6	100	0.7	0	0	0	30	5
	Grupo 2	5	6	100	0.9	0	0	0	30	5
	Grupo 3	5	6	100	1	0	0	0	30	5
Classe 5	Grupo 1	5	6	100	0.8	1	0	0	36	5
	Grupo 2	5	6	100	0.8	2	0	0	42	5
	Grupo 3	5	6	100	0.8	3	0	0	48	5
Classe 6	Grupo 1	5	6	100	0.8	0	50	100	30	5
	Grupo 2	5	6	100	0.8	0	50	150	30	5
	Grupo 3	5	6	100	0.8	0	50	200	30	5
Total										95

Nota-se que, em cada classe todos os parâmetros foram fixados no mesmo valor do grupo padrão, com exceção a alguns parâmetros que foram modificados visando alterar uma única característica do problema dentre as seguintes: número de linhas, número de máximo de diferentes itens que podem ser produzidos em cada período, demandas, quantidade de recursos disponíveis, número de produtos e custo para abertura das linhas. Assim, por exemplo, por meio das instâncias da classe 1 (juntamente com as instâncias do grupo padrão) é possível observar o comportamento dos modelos quando (apenas) o número de linhas de produção é alterado.

6.1.2 Instâncias para análise de desempenho

O Conjunto 2 é formado por 100 exemplares de teste gerados aleatoriamente com o objetivo de testar a performance computacional dos modelos e métodos de solução propostos,

identificando as abordagens mais promissoras. Os exemplares foram gerados de acordo com as seguintes definições de parâmetros:

1. Número de períodos $T \in \{10, 12, 14\}$, número de linhas de produção $L \in \{7, 10\}$, número de itens $J \in \{45, 80, 90, 110\}$, número máximo de itens que podem ser produzidos em cada linha num mesmo período $\Psi \in \{6, 8, 10\}$. Tomou-se $\Psi_{lt} = \Psi, \forall l, t$, isto é, um mesmo número máximo de itens foi utilizado para todas as linhas e períodos;
2. Número de recursos considerados $K \in \{5, 6, 7\}$ e número de unidades do recurso k necessárias para montagem da linha l , $r_{kl} \in U[0, 2], \forall k > 0, l$ e $r_{0l} \in U[5, 10], \forall l$. Número total de recursos do tipo k disponíveis dado por $R_k = \max \left\{ \max_{l=1, \dots, L} \{r_{kl}\}, \varphi_k^r \sum_{l \in [L]} r_{kl} \right\}$, onde φ_k^r é um parâmetro pertencente ao intervalo $[0, 1]$ que representa a redução percentual da quantidade disponível do recurso k com relação ao total de unidades do recurso k necessárias para montagem de todas as linhas de produção simultaneamente. Assim, se $\varphi_k^r = 1, \forall k$, tem-se um problema de dimensionamento e sequenciamento de lotes tradicional onde todas as linhas permanecem aptas a produção durante todos os períodos;
3. Tempo de preparação $st_{lij} \in U[15, 45]$ e custo de preparação $sc_{lij} = 2 * st_{lij}, \forall l, i, j$;
4. Lote mínimo de produção $m_{lj} = 2, \forall l, j$;
5. Custo de estoque $h_j \in U[1, 10]$ e custo unitário de atraso no atendimento das demandas $b_j = 10 * h_j, \forall j$;
6. Tempo de perecibilidade (*shelf-life*) $sl_j \in U[4, T]$ e tempo para produção de uma unidade do cada produto em cada linha $a_{lj} = 1, \forall l, j$;
7. Capacidade $C_{lt} = 480$ minutos, representado um período produtivo de um dia;
8. Demanda de cada item em cada período $d_{jt} \in U \left[0, \frac{C_{lt} - \max_{i,j} \{st_{lij}\} * \Psi - \varphi^d}{|P_l|} \right], \forall t, l, j \in P_l$, onde φ^d é um parâmetro (não negativo) redutor da demanda máxima de cada produto em cada período;
9. Para cada item, foi selecionada aleatoriamente uma única linha de produção capaz de produzi-lo;
10. Custo de montagem das linhas de produção $ac_l = \sum_k rc_k * r_{kl}$, onde rc_k é um parâmetro positivo que representa o custo do recurso k . O valor de rc_k foi escolhido no conjunto $U[\phi^b, \phi^e], \forall k$ com ϕ^b e ϕ^e sendo parâmetros não negativos que representam os custos mínimo e máximo do recurso k ;

O Conjunto 2 foi dividido em cinco grupos (C2G1, ..., C2G5) compostos por 20 exemplares cada, totalizando 100 exemplares de teste. A Tabela 6 apresenta as definições de parâmetros adotadas em cada grupo de instâncias do Conjunto 2.

Tabela 6 – Conjunto 2: descrição das instâncias de teste

Grupo	ϕ^b	ϕ^e	T	L	K	J	Ψ	ϕ_k^r	ϕ_0^r	ϕ^d	Exemplares
C2G1	0	0	10	7	5	45	6	0,8	0,60	100	20
C2G2	0	0	10	10	6	80	8	0,8	0,60	100	20
C2G3	0	0	14	10	6	90	8	0,6	0,50	90	20
C2G4	100	150	12	10	7	110	8	0,60	0,55	90	20
C2G5	50	150	14	10	7	110	8	0,60	0,55	90	20

6.2 Ambiente de teste

Os modelos propostos foram implementados em linguagem C++ em conjunto com as bibliotecas disponíveis na ferramenta *Concert Technology* do resolvidor comercial Cplex 12.6 com parametrização padrão. Assim, os modelos foram solucionados via um algoritmo do tipo *Branch-and-Cut*. Os métodos de solução também foram implementados em linguagem C++ e o mesmo resolvidor foi utilizado para solução dos subproblemas obtidos em cada método.

Os testes foram realizados num computador equipado com dois processadores Intel Xeon E5-2680v2 de 2.8 GHz com dez núcleos cada e com 128 GB DDR3 1866 MHz de memória RAM. Foi fixado um tempo máximo de execução de uma hora para a solução de cada instância em cada modelo/método de solução e os melhores limitantes primal e dual foram observados.

O desvio médio percentual entre a solução factível (z^f) e o limitante dual (z^d) encontrados (GAP) é definido como $GAP = 100 * \frac{z^f - z^d}{z^f}$.

6.3 Análise dos modelos

Nesta Seção apresenta-se resultados computacionais com o objetivo de identificar a formulação mais promissora para o PDSL-LPR. Na Seção 6.3.1, a análise de sensibilidade dos modelos é apresentada com base as instâncias do Conjunto 1 e na Seção 6.3.2, apresenta-se a análise de desempenho dos modelos realizada com base nas instâncias do Conjunto 2.

6.3.1 Análise de sensibilidade dos modelos

- Comparação entre o PDSL e o PDSL-LPR:

A Figura 7 mostra o impacto na performance computacional dos modelos propostos frente a variações nos parâmetros ϕ_k^r que definem a quantidade disponível dos recursos necessários para

montagem das linhas de produção. Nas instâncias pertencentes ao grupo 3 da classe 4, existem recursos suficientes para montagem de todas as linhas de produção simultaneamente ($\phi_k^r = 1, \forall k$ conforme Tabela 4), de modo que tem-se um tradicional problema de dimensionamento e sequenciamento de lotes (PDSL) em linhas paralelas. Já nos demais grupos da classe 4 e no grupo padrão, a quantidade de recursos disponíveis não é suficiente para a montagem de todas as linhas num mesmo período.

Verifica-se na Figura 7-(a) que tempo médio de execução foi constante (3600 segundos) para todos os grupos no modelo GLSPPRL e para os grupos 1, 2 (Classe 4) e grupo padrão no modelo CCPRL. Esse fato ocorreu porque esses modelos (GLSPPRL e CCPRL) não foram capazes de provar a otimalidade nessas instâncias. O modelo CLSDPRL foi capaz de provar a otimalidade na maioria dos exemplares e obteve o menor tempo médio de execução para todos os grupos. Nos modelos CLSDPRL, CCFLPRL e CCPRL o tempo médio de execução para o tradicional PDSL foi significativamente menor do que o tempo médio de execução para o PDSL-LPR.

Na Figura 7-(b), é possível concluir que o percentual de atraso no atendimento das demandas foi menor no PDSL do que no PDSL-LPR. Consequentemente, o valor da função objetivo foi substancialmente menor. A Figura 7-(c) mostra que para os modelos CLSDPRL, CCFLPRL e CCPRL o GAP médio obtido para o PDSL foi menor do que o GAP médio obtido para o PDSL-LPR, de modo que quando o parâmetro ϕ^r aumentou, o GAP médio reduziu. Porém, no modelo GLSPPRL observa-se um comportamento inverso. Quando ϕ^r aumentou, o GAP médio do modelo GLSPPRL também aumentou. Nota-se que o GAP médio obtido pelo modelo GLSPPRL para o tradicional PDSL foi de cerca de 80%, enquanto que para o PDSL-LPR foi, em média, de 20%. Esse fato ocorreu porque os limitantes duais obtidos pelo modelo GLSPPRL para o PDSL são mais fracos do que os limitantes duais obtidos para o PDSL-LPR.

Figura 7 – Comparação entre o PDSL e o PDSL-LPR - foram consideradas as instâncias da classe 4 e do grupo padrão

Na Figura 8 observa-se o comportamento dos modelos com respeito a alterações nos custos de montagem das linhas de produção. A Figura 8-(b) mostra que o modelo GLSPRL apresentou menor GAP médio quando os custos de abertura de linhas foram considerados, porém o tempo médio de execução não foi alterado quando os custos de abertura foram modificados (Figura 8-(a)), uma vez que, o GLSPRL falhou em encontrar soluções ótimas. Já o modelo CLSDPRL conseguiu provar a otimalidade na maioria das instâncias, apresentando o menor GAP médio com relação aos demais modelos. Quando os custos para abertura das linhas de produção foram relativamente próximos uns dos outros, o modelo CLSDPRL gastou mais tempo para encontrar a solução ótima. De modo geral, o modelo CLSDPRL apresentou melhor performance quando não considerou-se os custos de abertura de linhas, enquanto que os modelos CCFLPRL e GLSPRL apresentaram melhor performance quando os custos de abertura de linhas foram considerados. O modelo CCPRL mostrou-se praticamente invariante com relação a consideração dos custos de aberturas de linhas.

Figura 8 – Sensibilidade quanto aos custos para aberturas de linhas

- (a) Tempo médio de execução
 • Comportamento dos modelos com respeito a demais alterações de parâmetros:

Nas Figuras 9-(a) e (b), apresenta-se os resultados computacionais para as instâncias da classe 1 e grupo padrão com o objetivo de verificar o impacto no desempenho dos modelos frente a alterações no número de linhas de produção. Nota-se que o modelo CLSDPRL conseguiu resolver otimamente todas as instâncias com 3 e 5 linhas de produção, enquanto que apenas 60% das instâncias com 7 linhas de produção e 40% das instâncias com 9 linhas puderam ser resolvidas na otimalidade (Figura 9-(a)). O modelo GLSPRL não conseguiu provar a otimalidade nessas instâncias e o modelo CCPRL provou a otimalidade em apenas 40% das instâncias com 3 linhas de produção. O modelo CCFLPRL obteve a solução ótima em 80% das instâncias com 3 linhas e em 20% das instâncias com 5 linhas.

Observa-se na Figura 9-(b) que o tempo médio de execução do modelo CLSDPRL foi menor do que dos demais modelos para todos os grupos. Naturalmente, um aumento no número de linhas de produção acarretou aumento no tempo médio de execução e reduziu o percentual de instâncias resolvidas até a otimalidade em todos os modelos.

Figura 9 – Sensibilidade frente a alterações no número de linhas de produção

(a) Porcentagem de instâncias resolvidas até a otimalidade

(b) Tempo médio de execução

A Figuras 10-(a) e (b) resumem os resultados observados para as instâncias da classe 2 (e do grupo padrão), permitindo uma análise sobre a sensibilidade dos modelos quanto a alterações no número máximo de itens que podem ser produzidos numa mesma linha e período (Ψ). O modelo CLSDPRL resolveu a maioria das instâncias consideradas até a otimalidade (Figura 10-(a)), contudo, houve substancial aumento no tempo médio de execução quando o parâmetro

Ψ foi aumentado (Figura 10-(b)), de modo que para $\Psi = 4$ dispendeu-se apenas cerca de 17 segundos para resolver cada instância, enquanto que todo o tempo disponível (3600 segundos) foi consumido para resolver cada instância com $\Psi = 10$. Nos demais modelos, foram observados comportamentos análogos. Por exemplo, no modelo GLSPPRL, os tempos médios de execução foram constantes para todos os grupos (3600 segundos), porém quando Ψ foi aumentado, o GAP médio foi significativamente aumentado.

O valor médio da função objetivo foi reduzido quando Ψ foi aumentado, uma vez que maiores valores para Ψ introduzem maior flexibilidade para confecção dos planos de produção (Figura 10-(c)). Contudo, no modelo CCPRL o valor médio da função objetivo quando $\Psi = 10$ foi maior do que o valor médio da função objetivo quando $\Psi = 8$, pois maiores valores para Ψ aumentam os tamanhos das sequências de produção, tornando o problema mais difícil de ser resolvido, o que resultou em elevado GAP médio no modelo CCPRL quando $\Psi = 10$ (em torno de 30%).

Figura 10 – Sensibilidade com relação ao número de itens que podem ser produzidos numa mesma linha e período (parâmetro Ψ)

Os resultados referentes as instâncias da classe 3 (e grupo padrão) estão descritos na Figura 11, onde é possível analisar o comportamento dos modelos quando as demandas dos clientes foram alteradas (parâmetro de controle ϕ^d). Nota-se que os modelos CLSDPRL e CCFLPRL não apresentaram variações significativas no GAP médio e no tempo médio de execução frente alterações nas demandas dos clientes, enquanto que os modelos CCPRL e GLSPPRL apresentaram pequenos aumentos no GAP médio quando as demandas foram aumentadas.

Esse aumento no GAP médio ocorreu porque os modelos CCPRL e GLSPRL não apresentaram bons limitantes duais nas instâncias com maiores demandas (Figura 11-(d)), uma vez que encontraram soluções factíveis similares aos demais modelos (Figura 11-(c)).

Figura 11 – Sensibilidade referente a alterações nas demandas dos clientes

Finalmente, na Figura 12 apresenta-se a análise de sensibilidade dos modelos com relação a alterações no número total de itens considerados (parâmetro de controle α). Verifica-se que os modelos CLSDPRL e CCFLPRL apresentaram elevações significativas no tempo médio de execução quando o número de itens (por linha de produção) foi aumentado (Figura 12-(a)), porém o GAP médio manteve-se estável (Figura 12-(b)). Já os modelos CCPRL e GLSPRL apresentaram elevações significativas no GAP médio.

Figura 12 – Sensibilidade com relação ao número de itens considerados

6.3.2 Análise de desempenho dos modelos

Nesta Seção, apresenta-se os resultados computacionais obtidos pelos modelos propostos no Capítulo 3 tendo como base as instâncias de teste pertencentes ao Conjunto 2.

A Tabela 8 apresenta, para cada modelo, o número de instâncias nas quais obteve-se: i) soluções factíveis (SF), ii) o melhor (menor) valor para a função objetivo (MFO), iii) o melhor (maior) valor para o limitante dual (MLD), e iv) solução ótima (OT). Apresenta-se também, o desvio médio (GAP) entre os limitantes primal e dual incumbentes ao final da execução para cada modelo.

Verifica-se que os modelos CCFLPRL e GLSPRL foram capazes de fornecer soluções factíveis para todas as instâncias dos grupos C2G1 e C2G2 (com até 10 períodos, 6 recursos e 80 itens) e para apenas algumas instâncias dos demais grupos. O modelo CCFLPRL encontrou soluções para apenas 42 instâncias no total e o modelo GLSPRL, para 56 instâncias. Já os demais modelos (CCPRL e CLSDPRL), foram capazes de fornecer soluções factíveis para todas as instâncias.

O modelo GLSPRL foi capaz de oferecer o melhor valor para a função objetivo em 2 instâncias do grupo C2G3, enquanto que o modelo CCPRL para 3 instâncias do mesmo grupo. Para todas as demais instâncias (95 exemplares), melhores soluções foram obtidas pelo modelo CLSDPRL.

O modelo CLSDPRL também foi capaz de fornecer o melhor limitante dual para todas as instâncias consideradas nessa análise. Nesse sentido, apresentou GAP's médios significativamente menores do que os demais modelos. Por exemplo, o modelo CLSDPRL apresentou GAP's médios de 0,12%, 3,34%, 77,70%, 52,42%, 55,11% e 37,37% para os grupos C2G1, C2G2, C2G3, C2G4 e C2G5 e no geral, respectivamente, enquanto que o modelo CCPRL apresentou GAP's médios de 12,57%, 35,26%, 92,69%, 85,64%, 85,18% e 62,27%, respectivamente. Além disso, soluções ótimas foram encontradas em 7 instâncias do grupo C2G1 pelo modelo CLSDPRL, enquanto que os demais modelos não foram capazes de provar a otimalidade dos exemplares considerados.

As instâncias do grupo C2G3 (com 14 períodos, 10 linhas de produção, 6 recursos e 90 itens) foram as mais desafiadoras para todos os modelos considerados, apresentando os maiores GAP's médios - cerca de 91%, 92%, 83% e 77% para os modelos GLSPRL, CCPRL, CCFLPRL e CLSDPRL, respectivamente. Esse fato ocorreu devido a acentuada escassez de recursos, pois nessas instâncias o parâmetro ϕ_0^r foi fixado 0,5, valor este menor do que os adotados nas demais instâncias. Por outro lado, as instâncias do grupo C2G1, que possuem menores números de linhas, recursos e produtos puderam ser mais facilmente resolvidas. Todos os modelos encontraram soluções factíveis para todos os exemplares desse grupo e obtiveram os menores GAP's.

De maneira geral, o modelo CLSDPRL destacou-se com relação aos demais modelos tendo em vista a capacidade de obtenção de boas soluções factíveis e de bons limitantes duais, apresentando os menores GAP's para todos os grupos de instâncias considerados. Assim, o

modelo CLSDPRL pode ser apontado como o modelo mais promissor e portanto, foi utilizado como base para o desenvolvimento dos procedimentos de solução propostos nesta tese.

Tabela 8 – Resultados Computacionais - Classe 2

		C2G1	C2G2	C2G3	C2G4	C2G5	Média
GLSPRL	SF	20/20	20/20	6/20	4/20	6/20	56/100
	MFO	0/20	0/20	2/20	0/20	0/20	2/100
	MLD	0/20	0/20	0/20	0/20	0/20	0/100
	OT	0/20	0/20	0/20	0/20	0/20	0/100
	GAP (%)	22,89	47,95	91,09	83,52	86,28	50,27
CCPRL	SF	20/20	20/20	20/20	20/20	20/20	100/100
	MFO	0/20	0/20	3/20	0/20	0/20	3/100
	MLD	0/20	0/20	0/20	0/20	0/20	0/100
	OT	0/20	0/20	0/20	0/20	0/20	0/100
	GAP (%)	12,57	35,26	92,69	85,64	85,18	62,27
CCFLPRL	SF	20/20	20/20	2/20	0/20	0/20	42/100
	MFO	0/20	0/20	0/20	0/20	0/20	0/100
	MLD	0/20	0/20	0/20	0/20	0/20	0/100
	OT	0/20	0/20	0/20	0/20	0/20	0/100
	GAP (%)	15,44	34,89	83,43	-	-	-
CLSDPRL	SF	20/20	20/20	20/20	20/20	20/20	100/100
	MFO	20/20	20/20	15/20	20/20	20/20	95/100
	MLD	20/20	20/20	20/20	20/20	20/20	100/100
	OT	7/20	0/20	0/20	0/20	0/20	7/100
	GAP(%)	0,12	3,34	77,70	52,42	55,11	37,73

6.4 Regras de ramificação para acelerar a convergência de algoritmos do tipo *Branch-and-Bound*

Esta Seção destina-se a apresentação de regras de ramificação das variáveis binárias dos modelos CLSDPRL e CLSDPRL^w para serem utilizadas em algoritmos do tipo *Branch-and-Bound* com a finalidade de melhorar a performance computacional desses métodos de solução. As regras propostas baseiam-se na análise da estrutura dos modelos considerados e podem ser facilmente implementadas em resolvedores comerciais como o Cplex. Assim, é possível usufruir simultaneamente de diversas heurísticas gerais e técnicas eficientes de pré-processamento existentes nesses resolvedores e das regras de ramificação propostas.

- Uma regra de ramificação para o modelo CLSDPRL:

Considere o modelo CLSDPRL, definido por (3.36) a (3.53) - Capítulo 3. Note que, se uma determinada linha de produção l não for ativada durante um período t , isto é, $\delta_{lt} = 0$, então não haverá produção na linha l durante o período t . Assim, pode-se concluir por meio das restrições (3.43) e (3.44), que $y_{ljt} = 0, \forall j$ e $z_{lijt} = 0, \forall i, j$. Dessa forma, se num dado nó da

árvore de busca de algoritmo *Branch-and-Bound* é fixado $\delta_{lt} = 0$ para algum l e algum t , então $J + J^2$ variáveis binárias serão também fixadas em 0.

Esse fato sugere que ramificar as variáveis δ antes das demais variáveis binárias do modelo CLSDPRL pode ser uma estratégia eficiente. Assim, define-se a regra de ramificação R1 da seguinte forma.

R1: Em cada nó da árvore de busca, se na solução ótima da relaxação linear existir alguma variável δ_{lt} que não assume um valor binário, então a ramificação é realizada, primeiramente, numa dessas variáveis δ_{lt} para algum l e t , antes das demais variáveis.

Note que a regra R1 não determina sozinha qual dentre as variáveis δ será ramificada. Porém, a regra R1 foi desenvolvida com o objetivo de integrar um algoritmo *Branch-and-Cut* de um resolvidor e portanto, utiliza-se as regras existentes no algoritmo do resolvidor para determinar exatamente qual variável será ramificada em cada nó. Desse modo, a inclusão da regra R1 implica apenas numa imposição de precedência de ramificação, isto é, as variáveis y e z só poderão ser ramificadas num dado nó da árvore de busca se na solução ótima da relaxação linear associada ao nó incumbente, todas as variáveis δ assumirem valores binários.

A Figura 13 apresenta uma comparação entre o funcionamento de um algoritmo *Branch-and-Bound* com e sem a utilização da regra de ramificação R1 para solucionar o modelo CLSDPRL. Foi considerada uma instância de teste gerada conforme as especificações das instâncias do Conjunto 2, porém contendo apenas 4 períodos, 3 linhas produção, 10 produtos e 3 recursos necessários para montagem das linhas. É possível perceber que com apenas 6 nós explorados em cada algoritmo, o limitante dual incumbente obtido ao utilizar a regra R1 foi cerca de 21 % maior do que o limitante dual incumbente quando a regra R1 não foi considerada. Nessa instância, o tempo consumido para provar a otimalidade utilizando a regra R1 foi cerca de 8% menor do que o tempo consumido quando R1 não foi utilizada.

- Uma regra de ramificação para o modelo CLSDPRL^w:

Considere, agora, o modelo CLSDPRL^w, definido por (5.12) a (5.27) - Capítulo 5. Além da regra de ramificação R1, é possível propor uma nova relação de precedência para ramificação das variáveis.

Observe que, se um dado item j não for produzido numa linha l durante um período t , isto é, $w_{ljt} = 0$, então este item não será sequenciado na linha l durante o período t . Mais especificamente, supondo que em um dado nó da árvore de busca de um algoritmo *Branch-and-Bound* seja fixado $w_{ljt} = 0$ para algum l , j e t , então pelas restrições (5.19), tem-se que $y_{ljt} = 0$ e $z_{lijt} = 0, \forall i$, e pelas restrições (5.20) tem-se que $z_{lijt} = 0, \forall i$, isto é, se w_{ljt} é fixado em zero, então outras $1 + 2J$ variáveis binárias também terão seus valores fixados em zero.

Esse fato motivou a proposição da regra de ramificação R2, da seguinte forma.

Figura 13 – Impacto da utilização da regra R1 num algoritmo *Branch-and-Bound* para resolver o modelo CLSDPRL

R2: Em cada nó da árvore de busca, se, na solução ótima da relaxação linear incumbente, existirem variáveis δ_{lt} com valores fracionários, então a ramificação deve ser realizada, primeiramente, em uma das variáveis δ_{lt} antes das demais variáveis. Se todas as variáveis δ_{lt} assumirem valores binários, mas existirem variáveis w_{ljt} com valores fracionários, então a ramificação deve ser realizada em uma das variáveis w_{ljt} antes das variáveis y_{ljt} e z_{lijt} .

Novamente, a regra de ramificação R2 representa apenas uma relação de precedência e não permite determinar, exatamente, qual variável será ramificada em cada nó da árvore de busca. Contudo, essa regra de ramificação pode ser utilizada em conjunto com as demais regras de ramificação de algoritmos *Branch-and-Bound* de resolvedores comerciais.

• Experimentos computacionais:

A fim de identificar o impacto gerado pela utilização das regras de ramificação desenvolvidas, quatro experimentos computacionais foram realizados com base nas instâncias de teste do Conjunto 2. Os experimentos são os seguintes:

- CLSDPRL: Utiliza-se o algoritmo *Branch-and-Cut* padrão do resolvedor Cplex 12.60 para resolver o modelo CLSDPRL;
- CLSDPRL-R1: Utiliza-se a regra de ramificação R1 juntamente com o algoritmo *Branch-and-Cut* padrão do resolvedor Cplex para solucionar o modelo CLSDPRL;

- CLSDPRL^w: Consiste em solucionar o modelo CLSDPRL^w utilizando-se o algoritmo *Branch-and-Cut* padrão do resolver Cplex 12.60;
- CLSDPRL^w-R2: Consiste na utilização da regra de ramificação R2 em conjunto com o algoritmo *Branch-and-Cut* padrão do resolver Cplex 12.60 para solucionar o modelo CLSDPRL^w.

Em todos os experimentos, o tempo máximo de solução para cada instância foi fixado em 3600 segundos e, com exceção às regras de ramificação, foi utilizada a configuração padrão do resolvidor Cplex 12.60.

A Tabela 10 apresenta os resultados computacionais obtidos em cada experimento. Denota-se por FO, LD, GAP e TM os valores médios obtidos para a função objetivo, o limitante dual, o desvio percentual e o tempo de execução, respectivamente, enquanto que, OTM representa o número de instâncias resolvidas até a otimalidade.

Ao considerar o modelo CLSDPRL, é possível observar que a utilização da regra R1 proporcionou uma redução de aproximadamente 30% no valor médio da função objetivo e um aumento de 1,90% no valor do limitante dual encontrado. Além disso, o GAP médio geral observado no experimento CLSDPRL-R1 foi de 28,34% enquanto que, sem considerar a regra R1 (experimento CLSDPRL), o GAP médio geral foi de 37,73%.

O experimento CLSDPRL^w-R2 apresentou valor médio para a função objetivo cerca de 4,5% menor do que o valor médio geral da função objetivo observado no experimento CLSDPRL^w, enquanto que, o limitante dual médio obtido no experimento CLSDPRL^w-R2 foi cerca de 1,58% maior do que o limitante dual médio obtido no experimento CLSDPRL^w. O GAP médio geral referente ao experimento CLSDPRL^w-R2 foi de 18,51%, enquanto que, quando a regra de ramificação R2 não foi utilizada, o GAP médio geral foi de 21,26%. Também foi possível observar que a utilização da regra R2 (para solucionar o modelo CLSDPRL^w) proporcionou uma redução de 1,76% no tempo computacional médio.

Os resultados também mostraram que o desempenho apresentado pelo modelo CLSDPRL^w foi bastante superior ao desempenho apresentado pelo modelo CLSDPRL em todos os aspectos observados (valores da função objetivo e do limitante dual, GAP, tempo de execução e número de instâncias resolvidas até a otimalidade). Em particular, o GAP médio geral foi em torno de 21% para o experimento CLSDPRL^w e em torno de 37% para o experimento CLSDPRL. Contudo, ressalta-se que a explicitação da regra de ramificação R2 para solução do modelo CLSDPRL^w proporcionou melhores resultados.

6.5 Procedimentos de solução

Esta Seção dedica-se a uma análise de desempenho dos métodos heurísticos de solução que foram propostos nesta tese para solucionar o PDSL-LPR. Para tanto, são considerados os resultados de testes efetuados com base nas instâncias do Conjunto 2. A seguir, na Seção 6.5.1,

Tabela 10 – Regras de ramificação

		C2G1	C2G2	C2G3	C2G4	C2G5	Geral
CLSDPRL	FO	50935,40	64410,30	702140,55	360070,75	484220,05	344130,98
	LD	50871,70	50871,70	112152,55	162832,85	209744,40	123441,32
	GAP(%)	0,12	3,34	77,70	52,42	55,11	37,73
	TM	2791,25	3600	3600	3600	3600	3574,16
	OTM	7	0	0	0	0	7
CLSDPRL-R1	FO	50935,40	63680,25	183517,60	364701,05	493137,15	237822,63
	LD	50881,65	63291,64	112986,49	172313,39	209744,35	125838,24
	GAP(%)	0,10	0,61	32,64	48,44	55,81	28,34
	TM	2750,77	3600	3600	3600	3600	3565,66
	OTM	7	0	0	0	0	7
CLSDPRL ^w	FO	50935,40	63641,65	171871,70	231204,65	480090,00	204725,21
	LD	50934,29	63286,07	113559,43	165144,65	210714,44	124657,02
	GAP(%)	0,01	0,55	30,00	26,33	46,56	21,26
	TM	667,35	3491,4805	3600	3600	3600	3105,35
	OTM	18	1	0	0	0	19
CLSDPRL ^w -R2	FO	50935,60	63602,85	157795,75	226320,55	453555,65	195428,63
	LD	50932,06	63570,62	114077,05	173925,05	210705,83	126667,17
	GAP(%)	0,01	0,05	24,96	21,47	43,73	18,51
	TM	560,95	3336,727	3600	3600	3600	3050,51
	OTM	18	3	0	0	0	21

apresenta-se os resultados obtidos para os procedimentos construtivos (*D*, RF1, RF2 e RFH) e na Seção 6.5.2, apresenta-se os resultados obtidos pelas heurísticas de melhoramento (FOTO, FOALE2, FOALE3, INS1, ..., INS6). Por fim, na Seção 6.5.3 apresenta-se os resultados obtidos pelas heurísticas lagrangianas (LH e LD).

6.5.1 Heurísticas construtivas

Nesta Seção apresentam-se os resultados obtidos pelos procedimentos construtivos *D*, RF1, RF2 e RFH propostos na Seção 4.1 do Capítulo 4. O objetivo é identificar as abordagens mais promissoras para construção de soluções visando, tanto alimentar procedimentos de melhoria, como solucionar o PDSL-LPR de maneira completa. Quando o objetivo é alimentar um procedimento de melhoria, é interessante que o procedimento de construtivo seja capaz de fornecer soluções em baixo tempo computacional, possibilitando que o procedimento de melhoria tenha tempo suficiente para encontrar boas soluções. Por outro lado, se o procedimento construtivo for utilizado como único método de solução do problema, este pode consumir todo o tempo computacional disponível.

As heurísticas do tipo *relax-and-fix* (RF1, RF2 e RFH) foram testadas fixando o tempo máximo de execução (*tlim*) em 600, 1800, 2100 e 3600 segundos, sucessivamente, a fim de verificar a capacidade de obtenção de soluções factíveis e de boa qualidade em cada caso. Além disso, a heurística RFH também foi testada utilizando *tlim* = 3200 segundos. Para a heurística *D*, os tempos máximos de execução utilizados foram 800 e 1800 segundos.

Nos procedimentos RF1 e RF2, o tempo máximo de execução foi igualmente dividido entre cada iteração. Assim, por exemplo, como a heurística RF1 é composta por T iterações, então o tempo máximo para solução do subproblema associado a cada iteração foi fixado em $\frac{tlim}{T}$ segundos. Já para as heurísticas RFH e D , foi destinado maior tempo para a solução da primeira iteração (modelo AL no caso da heurística D). Uma vez verificado, por meio de testes preliminares, que a solução até a otimalidade dos subproblemas relacionados à segunda iteração desses procedimentos, em geral, não excede o limite de 200 segundos para os exemplares propostos, foram fixados os tempos máximos de solução em $tlim - 200$ segundos na primeira iteração e em 200 segundos na segunda iteração.

A notação RF1tlim600 é utilizada para representar os resultados obtidos pela heurística RF1 com tempo máximo de execução fixado em 600 segundos. Notação análoga é empregada nos demais casos.

O parâmetro Θ da heurística D , introduzido na equação (4.17) (Seção 4.1.1 do Capítulo 4), influencia diretamente na estimativa do tempo gasto com as preparações para produção. Valores muito elevados para Θ tendem a fornecer soluções ruins, enquanto que valores muito pequenos para Θ podem encontrar soluções infactíveis para o problema original. Testes preliminares foram efetuados com as instâncias dos grupos C2G1, C2G2 e C2G3, utilizando os valores 2, 3 e 4 para o parâmetro Θ . O valor $\Theta = 2$ foi escolhido como valor padrão, uma vez que proporcionou melhores soluções factíveis.

Na Tabela 12 apresenta-se os resultados computacionais referentes a cada heurística. Foram analisados os números de exemplares de teste nos quais as heurísticas encontraram uma solução factível, os tempos médios de execução e os desvios médios (GAP's). Para calcular o GAP utilizou-se o limitante dual obtido pelo resolvidor Cplex 12.6 ao resolver o modelo CLSDPRL com tempo máximo de execução de 3600 segundos.

As heurísticas RF1 e RF2 foram capazes de fornecer soluções factíveis apenas para as instâncias dos grupos C2G1 e C2G2 com limite máximo de tempo de execução fixado em 600 segundos. Quando o limite de tempo foi fixado em 1800 e 2100 segundos, a heurística RF1 forneceu soluções para as instâncias dos grupos C2G1, C2G2, C2G3, enquanto que a heurística RF2 forneceu ainda, soluções para algumas instâncias do grupo C2G4. Quando todo o tempo disponível (3600 segundos) foi utilizado, as heurísticas RF1 e RF2 forneceram soluções factíveis para 79 e 87 instâncias no geral, respectivamente. Em nenhum exemplar do grupo C2G5, a heurística RF1 foi capaz de prover soluções factíveis. Já as heurísticas RFH e D conseguiram fornecer soluções factíveis para a maioria das instâncias de todos os grupos.

Dessa forma, as heurísticas RF1 e RF2, que operam via tradicionais partições por períodos, se mostraram úteis apenas para solucionar instâncias de menor porte, pertencentes aos grupos C2G1, C2G2 e C2G3. Nesses casos, destaca-se que os tempos médios de execução para a heurística RF2 foram menores do que para a heurística RF1. Por exemplo, fixando $tlim = 2100$

segundos, a heurística RF1 obteve tempo médio de execução de 88,08, 603, 86 e 1963,09 segundos para os grupos C2G1, C2G2 e C2G3, respectivamente, enquanto que para a heurística RF2, esses tempos médios foram de 47,88, 361,14 e 1082,38 segundos, respectivamente. Além disso, a heurística RF2 também obteve GAP's médios ligeiramente menores para os grupos C2G1 e C2G2 (5,85% e 7,06%, respectivamente) quando comparada com a heurística RF1 (6,38% e 7,58%, respectivamente). Com relação ao grupo C2G3, a situação foi inversa. A heurística RF1 obteve melhores soluções (GAP de 27,71%) do que a heurística RF2 (GAP de 30,09%).

A heurística RFH mostrou-se sensível frente alterações no tempo máximo de execução para as instâncias dos grupos C2G3, C2G4 e C2G5. Essa sensibilidade impactou diretamente nos desvios médios (GAP's), de modo que o GAP médio geral da heurística RFH foi de 36,89%, 10,18%, 8,67%, 7,26% e 6,96% quando o tempo máximo de execução foi fixado em 600, 1800, 2100, 3200 e 3600 segundos, respectivamente. Por outro lado, com relação aos GAP's médios, a heurística D apresentou baixa variação frente alterações no tempo máximo de execução, de modo que o GAP médio observado utilizando $tlim = 800$ segundos foi de 9,02% e utilizando $tlim = 1800$, foi de 9%.

O tempo médio geral de execução da heurística D ($tlim = 800$) foi de apenas 423,27 segundos, tempo este, inferior aos tempos médios de execução da heurística RFH em todos os casos ($tlim = 600, 1800, 2100, 3200, 3600$), que foram de 499, 1205,39, 1379,91, 2051,45 e 2290,20 segundos, respectivamente. Além disso, a heurística D ($tlim = 800$) superou a heurística RFH, com relação ao GAP médio, quando o tempo de execução foi limitado em 600 e 1800 segundos. Nos demais casos ($tlim = 2100, 3200, 3600$) os GAP's obtidos pela heurística RFH (8,67%, 7,26% e 6,96%, respectivamente) foram ligeiramente melhores do que o obtido pela heurística D com $tlim = 800$ segundos (9,02%).

Dessa forma, os resultados obtidos mostram que as heurísticas RFH e D possuem maior potencial do que as heurísticas RF1 e RF2 para serem utilizadas como procedimento de solução para o problema estudado. A heurística D, por apresentar moderado GAP médio, baixa variação frente alterações no tempo máximo de execução e por conseguir encontrar soluções em menor tempo computacional do que as demais, mostra-se adequada para ser utilizada como um procedimento para construção de soluções iniciais visando alimentar procedimentos de melhoria. Por outro lado, pela alta capacidade de obtenção de soluções factíveis e pela boa qualidade das soluções obtidas, a heurística RFH mostrou-se adequada para ser utilizada como procedimento único de solução.

Tabela 12 – Resultados computacionais: Heurísticas construtivas

Heurísticas construtivas para determinação de soluções iniciais																		
Heurística	Número de instâncias resolvidas						Tempo médio de execução (seg)						GAP(%)					
	C2G1	C2G2	C2G3	C2G4	C2G5	Geral	C2G1	C2G2	C2G3	C2G4	C2G5	Geral	C2G1	C2G2	C2G3	C2G4	C2G5	Geral
RF1tlim600	20	18	0	0	0	38	85,25	541,34	—	—	—	—	6,38	9,23	—	—	—	—
RF1tlim1800	20	20	6	0	0	46	85,66	608,71	1895,94	—	—	—	6,38	7,58	75,38	—	—	—
RF1tlim2100	20	20	20	0	0	60	88,08	603,86	1963,09	—	—	—	6,38	7,58	27,71	—	—	—
RF2tlim600	20	20	0	0	0	40	49,49	320,17	—	—	—	—	5,85	14,48	—	—	—	—
RF2tlim1800	20	20	20	12	0	72	48,32	603,86	1004,83	1744,79	—	—	5,85	7,06	51,93	19,83	—	—
RF2tlim2100	20	20	20	13	9	82	47,88	361,14	1082,38	1821,17	2205,95	894,60	5,85	7,06	30,09	19,41	30,81	16,95
RFHtlim600	20	20	20	20	19	99	49,84	412,26	600,00	600,00	600,00	499,00	0,19	2,25	62,60	58,56	62,08	36,89
RFHtlim1800	20	20	20	20	20	100	50,34	476,55	1751,09	1813,98	1800,00	1205,39	0,19	2,22	16,60	13,82	18,05	10,18
RFHtlim2100	20	20	20	20	20	100	49,74	475,51	2050,04	2109,12	2215,14	1379,91	0,19	2,22	15,64	12,77	12,53	8,67
RFHtlim3200	20	20	20	20	20	100	50,98	478,30	3159,96	3200,00	3200,00	2051,45	0,19	2,22	12,36	10,31	11,23	7,26
Dtlim800	20	20	19	20	19	98	13,62	71,06	423,57	718,22	892,17	423,27	2,59	3,83	14,50	13,07	10,76	9,02
Dtlim1800	20	20	19	20	19	98	13,01	70,15	500,55	1129,93	1747,14	688,45	2,59	3,83	14,50	12,95	10,76	9,00
Heurísticas construtivas como procedimento único de solução																		
RF1tlim3600	20	20	20	19	0	79	85,25	603,58	1989,02	2941,32	—	—	6,38	7,58	25,74	16,05	—	—
RF2tlim3600	20	20	20	14	13	87	47,85	366,86	1282,28	1897,34	2719,37	1101,78	5,85	7,06	27,14	19,69	20,28	15,41
RFHtlim3600	20	20	20	20	20	100	49,80	474,57	3554,76	3600,00	3600,00	2290,20	0,19	2,22	12,01	10,10	10,30	6,96

6.5.2 Heurísticas de melhoria e *matheuristics*

Nesta Seção, apresentam-se as análises de desempenho das heurísticas de melhoramento FOTO, FOALE2 e FOALE3, que foram propostas na Seção 4.2 do Capítulo 4 e das *matheuristics* INS1, ..., INS6, que foram propostas na Seção 4.3 do Capítulo 4. Esses procedimentos partem de uma solução factível inicial e, iterativamente, buscam melhores soluções numa dada vizinhança da solução incumbente. Utilizou-se soluções iniciais obtidas tanto pela heurística *D* como pela heurística RFH. Para a heurística *D* utilizou-se tempo máximo de execução de 800 segundos. Já para a heurística RFH, o tempo máximo de execução foi fixado em 3200 segundos para que se pudesse obter soluções de melhor qualidade do que a heurística *D* para a maioria das instâncias.

Os procedimentos de natureza estocástica (FOALE2, FOALE3 e INS) foram executados três vezes para cada exemplar e tomou-se a média dos três valores obtidos para a função objetivo para efetuar o cálculo do desvio médio, enquanto que o procedimento determinístico FOTO foi executado uma única vez.

A cada iteração das heurísticas de melhoramento e *matheuristics* propostas, um subproblema é resolvido contendo um número de variáveis binárias inferior ao do problema original. Esses subproblemas foram solucionados utilizando o algoritmo *Branch-and-Cut* do resolvidor Cplex 12.60 em sua configuração padrão e com tempo máximo de execução fixado em 300 segundos para todas as iterações. Dessa forma, alguns subproblemas, oriundos de instâncias de grande porte, não foram resolvidos até a otimalidade.

Os resultados obtidos são apresentados na Tabela 15. Os desvios médios (GAP's) relativos a cada método testado foram calculados utilizando-se o limitante dual obtido via algoritmo *Branch-and-Cut* do resolvidor Cplex 12.60 ao solucionar o modelo CLSDPRL com tempo máximo de execução limitado em uma hora.

Denota-se, por exemplo, por D-FOTO e por RFH-FOTO, os resultados obtidos pela heurística FOTO partindo de uma solução inicial obtida pelos procedimentos D e RFH, respectivamente. Notação análoga é aplicada aos demais procedimentos.

Os procedimentos FOTO, FOALE2, INS(V_1, V_2, T_1) e INS(V_1, V_2, T_1, T_4) apresentaram melhores resultados quando a solução inicial foi encontrada via heurística RFH, enquanto que, os demais procedimentos foram capazes de encontrar melhores soluções quando a heurística *D* foi utilizada para obtenção de uma solução inicial.

Quando a solução inicial foi obtida via heurística RFH, o procedimento mais promissor foi o FOTO (com GAP médio de 6,78%), enquanto que, quando a solução inicial foi obtida via heurística *D*, destacou-se o procedimento INS(V_3, V_4, V_5, T_2, T_3) com GAP médio de 6,45%.

Para o grupo de instâncias C2G1, os procedimentos RFH-INS(V_1, V_2, T_1) e RFH-INS(V_4, V_5, T_2, T_3) apresentaram o menor GAP médio dentre todos os métodos testados, em torno de 0,13%. Já para o grupo C2G2, o menor GAP médio obtido de 1,50% (via procedimentos RFH-FOTO, RFH-FOALE2, RFH-INS(V_1, V_2, T_1), RFH-INS(V_1, V_2, T_1, T_4), RFH-INS(V_3, V_4, V_5, T_2, T_3), RFH-

INS($V_3, V_4, V_5, T_2, T_3, T_4$), RFH-INS(V_4, V_5, T_2, T_3) e RFH-INS(V_4, V_5, T_2, T_3, T_4)), para o grupo C2G3 foi de 11,31% (via procedimento D-INS(V_4, V_5, T_2, T_3)), para o grupo C2G4 foi de 9,79% (via procedimento RFH-FOTO) e para o grupo C2G5 foi de 9,36% (via procedimento D-INS(V_3, V_4, V_5, T_2, T_3)).

O tempo máximo de execução (3600 segundos) foi atingido em todas as instâncias para os métodos FOTO, FOALE2, FOALE3, INS(V_1, V_2, T_1) e INS(V_1, V_2, T_1, T_4) com solução inicial partindo de ambas as heurísticas construtivas utilizadas (RFH e D). Já os demais métodos, só não atingiram o tempo máximo de execução para algumas instâncias do grupo C2G1, cujos tempos médios observados para esse grupo são descritos na Tabela 14.

Tabela 14 – Tempos de execução

Heurística	C2G1
D-INS(V_3, V_4, V_5, T_2, T_3)	1187
D-INS($V_3, V_4, V_5, T_2, T_3, T_4$)	1309
D-INS(V_4, V_5, T_2, T_3)	1151
D-INS(V_4, V_5, T_2, T_3, T_4)	1465
RFH-INS(V_3, V_4, V_5, T_2, T_3)	1054
RFH-INS($V_3, V_4, V_5, T_2, T_3, T_4$)	1254
RFH-INS(V_4, V_5, T_2, T_3)	1128
RFH-INS(V_4, V_5, T_2, T_3, T_4)	1357

O procedimento D-INS(V_3, V_4, V_5, T_2, T_3) apresentou o menor GAP médio geral (6,45%), seguido do procedimento D-INS(V_4, V_5, T_2, T_3) que apresentou GAP geral de 6,47%.

Na Figura 14 apresenta-se um gráfico do tipo *box plot* dos desvios médios de todos os procedimentos considerados nesta Seção. Cada procedimento é designado na Figura 14 pela sigla atribuída na Tabela 15. É possível verificar que o procedimento D-INS(V_4, V_5, T_2, T_3) apresentou o menor GAP máximo (em torno de 24%) dentre todos os métodos considerados, seguido do procedimento D-INS(V_4, V_5, T_2, T_3, T_4), com GAP máximo em torno de 26%.

Também verifica-se que, de modo geral, os procedimentos que partiram de uma solução factível obtida pela heurística *D* apresentaram menores GAP's máximos do que os procedimentos que iniciaram com uma solução obtida via heurística RFH. Porém, os GAP's medianos apresentaram baixa variação entre os métodos propostos.

Tabela 15 – Procedimentos de melhoria

Sigla	Heurística	GAP (%)						Redução (%) do GAP inicial					
		C2G1	C2G2	C2G3	C2G4	C2G5	Geral	C2G1	C2G2	C2G3	C2G4	C2G5	Geral
D-INS1	D-FOTO	0,56	1,79	12,36	11,04	9,94	7,06	2,02	1,36	2,91	2,03	0,82	1,83
	D-FOALE2	0,79	1,91	12,93	11,96	10,10	7,46	1,80	1,24	2,34	1,11	0,66	1,43
	D-FOALE3	0,35	1,81	12,13	11,07	9,76	6,94	2,24	1,34	3,14	2,00	1,00	1,95
	D-INS(V ₁ ,V ₂ ,T ₁)	0,50	2,04	12,99	11,68	9,97	7,35	2,09	1,11	2,28	1,39	0,79	1,54
	D-INS(V ₁ ,V ₂ ,T ₁ ,T ₄)	0,53	2,11	13,32	11,55	10,10	7,43	2,06	1,04	1,95	1,52	0,66	1,46
	D-INS(V ₃ ,V ₄ ,V ₅ ,T ₂ ,T ₃)	0,27	1,59	11,37	10,04	9,36	6,45	2,32	1,56	3,90	3,03	1,40	2,44
	D-INS(V ₃ ,V ₄ ,V ₅ ,T ₂ ,T ₃ ,T ₄)	0,24	1,69	11,74	10,58	9,80	6,73	2,35	1,46	3,53	2,49	0,96	2,16
	D-INS(V ₄ ,V ₅ ,T ₂ ,T ₃)	0,26	1,62	11,31	10,04	9,49	6,47	2,33	1,53	3,96	3,03	1,27	2,42
RFH-INS6	D-INS(V ₄ ,V ₅ ,T ₂ ,T ₃ ,T ₄)	0,25	1,59	11,45	10,63	9,74	6,67	2,34	1,56	3,82	2,44	1,02	2,22
	RFH-FOTO	0,15	1,50	12,21	9,79	10,25	6,78	0,04	0,02	0,15	0,52	0,98	0,34
	RFH-FOALE2	0,15	1,50	12,27	10,14	11,22	7,06	0,04	0,02	0,09	0,17	0,01	0,06
	RFH-FOALE3	0,19	1,52	12,21	10,27	11,24	7,09	0,00	0,00	0,15	0,04	0,00	0,03
	RFH-INS(V ₁ ,V ₂ ,T ₁)	0,13	1,50	12,20	10,13	11,14	7,02	0,06	0,02	0,16	0,18	0,09	0,10
	RFH-INS(V ₁ ,V ₂ ,T ₁ ,T ₄)	0,14	1,50	12,19	10,12	11,11	7,01	0,05	0,02	0,17	0,19	0,12	0,11
	RFH-INS(V ₃ ,V ₄ ,V ₅ ,T ₂ ,T ₃)	0,14	1,50	12,01	9,92	11,04	6,92	0,05	0,02	0,35	0,39	0,19	0,20
	RFH-INS(V ₃ ,V ₄ ,V ₅ ,T ₂ ,T ₃ ,T ₄)	0,14	1,50	12,12	9,85	11,03	6,93	0,05	0,02	0,24	0,46	0,20	0,19
RFH-INS5	RFH-INS(V ₄ ,V ₅ ,T ₂ ,T ₃)	0,13	1,50	12,15	10,00	11,13	6,98	0,06	0,02	0,21	0,31	0,10	0,14
	RFH-INS(V ₄ ,V ₅ ,T ₂ ,T ₃ ,T ₄)	0,14	1,50	12,05	9,86	11,16	6,94	0,05	0,02	0,31	0,45	0,07	0,18

Figura 14 – Box Plot do desvio médio dos procedimentos de melhoria

6.5.3 Heurísticas lagrangianas

Apresenta-se nesta Seção, os resultados computacionais obtidos pelas heurísticas lagrangianas LH e LD propostas no Capítulo 5. Para cada heurística, foram utilizados dois métodos para solução do problema dual associado: o método do subgradiente e o algoritmo do volume. Denota-se por LH-S e LH-V os resultados obtidos pela heurística LH utilizando o método do subgradiente e o algoritmo do volume, respectivamente, para resolver o problema dual lagrangiano associado e notação análoga é utilizada para a heurística LD.

Na Seção 6.5.3.1 apresenta-se as escolhas de parâmetros utilizadas em cada método de solução do problema dual (algoritmo do subgradiente e algoritmo do volume) para cada esquema lagrangiano proposto (LH e LD). Em seguida, na Seção 17, apresenta-se os resultados computacionais observados.

6.5.3.1 Determinação de parâmetros

- Método do subgradiente:

Os valores dos parâmetros adotados no método do subgradiente foram determinados por meio de experimentos computacionais realizados com 60 exemplares de testes do Conjunto 2 escolhidos aleatoriamente. Os testes realizados basearam-se em [Carvalho e Nascimento \(2016\)](#). Inicialmente, foi atribuído o valor zero para todas as variáveis duais e simulações foram realizadas com os valores para o parâmetro σ (5.33) variando no conjunto $\{1, 0, 1, 4, 1, 6, 1, 8\}$.

O parâmetro qt foi introduzido para representar o número de iterações realizadas entre cada atualização do parâmetro σ . Desse modo, a cada qt iterações, o parâmetro σ foi dividido por um valor $\eta > 1$. O parâmetro η permanece constante durante todas as iterações do método e foram realizadas simulações utilizando os seguintes valores para η : 1,5; 2,0; 2,5 e 3,0. A cada qt iterações, o próprio parâmetro qt também foi atualizado da seguinte forma

$$qt \leftarrow \max\{5, \lceil qt/\eta \rceil\}.$$

Testes foram realizados adotando-se valores iniciais para o parâmetro qt escolhidos no conjunto $\{15, 20, 30\}$. Para cada método, foi escolhida a combinação de parâmetros que forneceu o menor GAP médio geral obtido pela heurística lagrangiana.

Dessa forma, os resultados apresentados na Seção 17 foram obtidos adotando-se $\sigma = 1, 4$, $\eta = 2, 5$ e $qt = 30$ para a abordagem LH e $\sigma = 1, 0$, $\eta = 1, 5$ e $qt = 20$ para a abordagem LD.

- Algoritmo do volume:

Para obtenção de uma boa combinação de parâmetros para o algoritmo do volume, testes preliminares foram realizados considerando 4 instâncias de teste escolhidas aleatoriamente em cada um dos grupos de instâncias pertencentes ao Conjunto 2 (C2G1, ..., C2G5), totalizando 20 instâncias. Esse número de instâncias foi escolhido em virtude do elevado tempo computacional

requerido para testar diversas combinações de parâmetros, impedindo a consideração de um grande número de instâncias.

O tamanho do passo (s) é atualizado conforme (5.36). Testes foram realizados adotando, sucessivamente, os valores 0,3; 0,5; 0,7; 0,9 e 1,1 como valores iniciais para o parâmetro σ . Define-se os parâmetros d^{LH} e d^{LD} conforme as expressões (6.1) e (6.2), respectivamente.

$$d^{LH} = \sum_{l,t} \left[\left(\sum_j a_{lj} \bar{x}_{ljt} + \sum_{i,j} st_{lij} \bar{z}_{lijt} - C_{lt} \right) \cdot \left(\sum_j a_{lj} x_{ljt}^{it} + \sum_{i,j} st_{lij} z_{lijt}^{it} - C_{lt} \right) \right] \\ + \sum_{l,j,t} \left[\left(\bar{w}_{ljt} - \bar{y}_{ljt} - \sum_i \bar{z}_{lijt} \right) \cdot \left(w_{ljt}^{it} - y_{ljt}^{it} - \sum_i z_{lijt}^{it} \right) \right]; \quad (6.1)$$

$$d^{LD} = \sum_{j,t} \left[\left(\bar{A}_{jt} - \sum_l \bar{x}_{ljt} \right) \cdot \left(A_{jt}^{it} - \sum_l x_{ljt}^{it} \right) \right]. \quad (6.2)$$

O parâmetro d^{LH} refere-se a abordagem LH e d^{LD} refere-se a abordagem LD.

De acordo com Barahona e Anbil (2000), quando tem-se $DB^{it} \leq \overline{DB}$ então, não foi possível encontrar um melhor limitante durante a iteração incumbente it , e desse modo, a iteração it é dita iteração *vermelha* ou *nula*. Caso contrário, se $DB^{it} > \overline{DB}$ então, verifica-se o valor do parâmetro d^{LH} (ou d^{LD} , conforme a abordagem). Se $d^{LH} < 0$ ($d^{LD} < 0$), tem-se que it é dita iteração *amarela* e, caso contrário, it é dita uma iteração *verde*.

A cada 5 iterações vermelhas consecutivas, o parâmetro σ é dividido por uma constante $\eta > 1$ e a cada iteração verde, σ é multiplicado por um parâmetro $\vartheta > 1$. Testes preliminares foram realizados utilizando os valores 1,5 e 2,0 para η e 1,1; 1,2 e 1,3 para ϑ .

O limitante γ_{max} utilizado na determinação do parâmetro γ (Algoritmos 5 e 6) foi fixado em 0,3 para ambos os procedimentos.

Após a realização dos testes preliminares, adotou-se as seguintes definições de parâmetros:

LH: $\sigma = 0,7$; $\gamma_{max} = 0,3$; $\eta = 2$ e $\vartheta = 1,1$;

LD: $\sigma = 0,9$; $\gamma_{max} = 0,3$; $\eta = 2$ e $\vartheta = 1,1$.

6.5.3.2 Resultados

Testes computacionais foram realizados utilizando as heurísticas lagrangianas propostas para solucionar os 100 exemplares de teste pertencentes ao Conjunto 2. Os principais resultados são reportados na Tabela 17, onde as linhas PB, DB, GAP e Iter denotam os valores médios para o limitante primal, limitante dual, desvio médio e número de iterações realizadas, respectivamente. A linha SF denota o número de instâncias nas quais pelo menos uma solução factível foi encontrada.

Ainda na Tabela 17, os valores médios para os limitantes duais marcados com o sinal * indicam que os subproblemas lagrangianos não foram solucionados até a otimalidade, de modo que não se tem a garantia de que o valor encontrado representa um limitante dual. Os desvios médios (GAP's) foram calculados considerando o limitante dual oferecido pelo algoritmo *Branch-and-Cut* do resolvidor Cplex (assim como em Xiao *et al.* (2015) e Carvalho e Nascimento (2016)) ao solucionar o modelo CLSDPRL com tempo máximo de execução fixado em uma hora. Esses limitantes são novamente reportados na linha Cplex da Tabela 17.

Quando o método do subgradiente foi utilizado para solucionar o problema dual lagrangiano (abordagens LH-S e LD-S), melhores soluções factíveis foram encontradas via heurística LH do que via heurística LD, uma vez que os valores médios encontrados para os limitantes primais na abordagem LH-S foram menores do que os valores encontrados via abordagem LD-S para todos os grupos de instâncias e, conseqüentemente, no geral. A diferença observada no valor médio da solução primal impactou diretamente nos desvios médios, de modo que, foram observados os desvios de 1,08%; 2,48%; 15,05%; 11,99%; 11,73% e 8,43% na abordagem LH-S e de 5%; 5,18%; 16,15%; 14,27%; 12,88% e 10,69% na abordagem LD-S, para os grupos C2G1, ..., C2G5 e no geral, respectivamente.

De acordo com a Tabela 17 é possível verificar que a abordagem LH-V se mostrou mais vantajosa do que a abordagem LD-V quanto a qualidade das soluções factíveis obtidas, de modo que a abordagem LH-V obteve GAP médio geral de 8,63% e a abordagem LD-V de 10,84%.

As abordagens LH-S e LH-V obtiveram limitantes inferiores para as instâncias dos grupos C2G1 e C2G2. Nos demais casos, o subproblema lagrangiano, ainda que decomposto, não pôde ser resolvido até a otimalidade com tempo máximo de execução fixado em 600 segundos, não havendo assim, garantia de obtenção de limitantes duais. Essa dificuldade em obter a solução ótima do subproblema lagrangiano também foi experimentada em Xiao *et al.* (2015), onde a parte de sequenciamento do subproblema lagrangiano foi resolvida heurísticamente em todas as iterações do método e, portanto, nenhum limitante dual foi obtido. Dessa forma, ainda que promissora quanto a capacidade de obtenção de boas soluções factíveis, a abordagem LH não se mostrou eficiente na obtenção de limitantes inferiores. Nas abordagens LD-S e LD-V limitantes inferiores foram encontrados para todas as instâncias de teste, porém, em ambos os casos, os limitantes encontrados não superaram os obtidos pelo algoritmo *Branch-and-Cut* do resolvidor Cplex (ver última linha da Tabela 17).

As abordagens LD-S e LD-V permitiram a realização de um maior número de iterações (em média, 87 e 110 iterações, respectivamente) quando comparadas com as abordagens LH-S (46 iterações, em média) e LH-V (41 iterações, em média), respectivamente, indicando que o subproblema lagrangiano obtido pela decomposição lagrangiana que originou a heurística LD é mais fácil de ser resolvido do que o subproblema lagrangiano obtido via heurística LH. Contudo, a realização de um maior número de iterações não resultou na obtenção de melhores resultados.

Em geral, todas as abordagens foram capazes de encontrar soluções factíveis para a

grande maioria dos exemplares de teste, com destaque para a abordagem LD-S, que foi capaz de solucionar todas as instâncias propostas. O tempo máximo de execução (1 hora) foi atingido para todas as instâncias de teste em todas as abordagens, uma vez que não foi provada a otimalidade para os exemplares considerados.

Dentre as abordagens lagrangianas estudadas, a mais promissora foi a abordagem LH-S, que encontrou as melhores soluções factíveis para os grupos de instâncias C2G1, C2G2, C2G4 e C2G5 e no geral. Para o grupo C2G3, soluções ligeiramente melhores foram obtidas pela abordagem LH-V.

A seguir, na Seção 6.6, será apresentada uma análise comparativa entre os métodos de solução mais promissores (em suas respectivas categorias) propostos nesta tese. As heurísticas lagrangianas serão representadas pelo procedimento LH-S.

Tabela 17 – Heurísticas lagrangianas

		C2G1	C2G2	C2G3	C2G4	C2G5	Geral
LH-S	PB	51415	64279	133079	195973	237770	135480
	DB	48974	60288	*102228	*170521	*214789	*118396
	GAP(%)	1,08	2,48	15,05	11,99	11,73	8,43
	Iter	167	36	12	8	7	46
	SF	20	20	20	20	19	99
LD-S	PB	53647	66132	136102	201134	241148	139632
	DB	34060	42281	80042	129136	160151	89134
	GAP(%)	5,00	5,18	16,15	14,27	12,88	10,69
	Iter	274	82	34	24	19	87
	SF	20	20	20	20	20	100
LH-V	PB	51533	64912	133452	196237	238020	135833
	DB	48901	56588	*108518	*174975	*216698	*120290
	GAP(%)	1,31	3,37	15,03	12,11	11,82	8,63
	Iter	155	26	9	7	6	41
	SF	20	20	19	20	19	98
LD-V	PB	54335	67183	133428	200623	241148	139403
	DB	34060	42281	80177	129136	160151	89252
	GAP(%)	6,08	6,61	14,75	14,07	12,88	10,84
	Iter	376	88	38	25	20	110
	SF	20	20	19	20	20	99
Cplex	DB	50872	62665	112153	172263	209744	121539

6.6 Análise comparativa dos principais procedimentos de solução

Nesta Seção, apresenta-se uma análise comparativa entre as principais abordagens de solução propostas nesta tese para lidar com o PDSL-LPR. Considera-se, nesta análise, os seguintes experimentos computacionais:

- i. CLSDPRL^w-R2: O modelo CLSDPRL^w solucionado pelo algoritmo *Branch-and-Cut* do resolvidor Cplex 12.60 utilizando a regra de ramificação R2, proposta na Seção 6.4 deste capítulo. Nesse experimento, os valores dos limitantes primais e duais foram significativamente melhorados quando comparados com os resultados obtidos ao se resolver o modelo CLSDPRL. Nesta Seção, denota-se este procedimento por CLSDPRL^w;
- ii. RFHtlm3600: Consiste em utilizar a heurística construtiva RFH, proposta na Seção 4.1.2 e baseada no modelo CLSDPRL^w, para solucionar o problema tratado de maneira completa. Nesse caso, foram destinados 3400 segundos para a solução da primeira iteração e 200 segundos para solução da segunda iteração do procedimento RFH. Essa abordagem se destacou entre as demais heurísticas construtivas para solução completa do problema. Nesta Seção, denota-se este procedimento apenas por RFH;
- iii. D-INS(V₃,V₄,V₅,T₂,T₃): A heurística construtiva *D* é utilizada na construção de uma solução inicial para alimentar o procedimento de melhoria INS(V₃,V₄,V₅,T₂,T₃). Esse procedimento apresentou o menor GAP médio dentre as heurísticas de melhoramento propostas nesta tese. Este procedimento é denotada nesta Seção por INS;
- iv. LH-S: Consiste na heurística LH, proposta no Capítulo 5, onde o algoritmo do subgradiente é utilizado para solução do problema dual lagrangiano. Essa heurística foi a mais eficiente dentre as abordagens lagrangianas propostas e será denotada nesta Seção apenas como LH.

A seguir, na Tabela 19, apresenta-se os desvios médios percentuais obtidos em cada método, recalculados de acordo com os limitantes duais observados no procedimento CLSDPRL^w. A Tabela 19 apresenta ainda, os tempos médios de execução de cada método para cada grupo de instâncias do Conjunto 2.

Para os grupos de instâncias C2G1 e C2G2, o modelo CLSDPRL^w munido da regra de ramificação R2, foi capaz de oferecer as melhores soluções factíveis, com desvios médios de 0,01% e 0,05%, respectivamente. Para esses mesmos grupos, a heurística RFH apresentou desvios médios ligeiramente maiores, 0,08% e 0,09%, respectivamente, porém apresentou tempo médio de execução significativamente menor (de 49,80 e 474,57 segundos, respectivamente) do que todas as demais abordagens.

Já para os demais grupos de instâncias, e também no geral, melhores soluções factíveis foram obtidas via abordagem INS. O desvio médio geral foi de 5,57% para o procedimento INS, e de 5,95%, 7,58% e 18,51% para as abordagens RFH, LH e CLSDPRL^w, respectivamente. No geral, a heurística RFH se mostrou a mais rápida, consumindo, em média, 2290 segundos, enquanto que, a heurística LH consumiu todo o tempo disponível em todos os exemplares de teste.

Tabela 19 – Comparação entre as principais abordagens de solução.

	GAP (%)				Tempo médio (seg)			
	CLSDPRL ^w	RFH	INS	LH	CLSDPRL ^w	RFH	INS	LH
C2G1	0,01	0,08	0,16	0,97	560,95	49,80	1187,66	3600
C2G2	0,05	0,09	0,17	1,07	3336,73	474,57	3600	3600
C2G3	24,96	10,48	9,85	13,58	3600	3554,76	3600	3600
C2G4	21,47	9,21	9,20	11,16	3600	3600	3600	3600
C2G5	43,73	9,88	8,83	11,32	3600	3600	3600	3600
Geral	18,51	5,95	5,57	7,58	3050,51	2290,20	3109,54	3600

Na Figura 15 apresenta-se um gráfico de desempenho das abordagens consideradas, produzido conforme proposto em Dolan e Moré (2002). Dados P métodos de solução para um determinado problema e Δ exemplares de teste, define-se como $\tau_{p\xi}$ o valor obtido pelo método p ao solucionar o exemplar de teste ξ . Seja, ainda, $\Gamma_\xi = \min_{p=1,\dots,P} \{\tau_{p\xi}\}$, o menor valor obtido para o exemplar ξ . Por fim, define-se, para todo par (p, ξ) , o parâmetro $r_{p\xi} \geq 1$ como sendo o desvio relativo (quociente) entre o valor obtido para instância ξ via método p e o melhor valor obtido para ξ , isto é, $r_{p\xi} = \frac{\tau_{p\xi}}{\Gamma_\xi}$.

O gráfico de desempenho proposto em Dolan e Moré (2002) é construído colocando no eixo das abscissas os valores para um parâmetro θ e no eixo das ordenadas a porcentagem de exemplares de teste nos quais se tem $r_{p\xi} \leq \theta$. Desse modo, o valor $\theta = 1$ indica a porcentagem de exemplares de teste nas quais o método p forneceu a melhor solução. Além disso, se θ'_p é o menor valor para o qual a curva do método p atingiu a marca de 100% no eixo das ordenadas, então θ'_p é maior desvio apresentado pelo método p . Dessa forma, quanto maior a porcentagem correspondente a $\theta = 1$ e quanto menor o valor de θ'_p , mais eficiente é o método.

Observa-se que as abordagens CLSDPRL^w, RFH, INS e LH encontraram a melhor, dentre as soluções encontradas, para cerca de 39%, 56%, 42% e 20% das instâncias, respectivamente. O desvio máximo para as abordagens RFH e INS foi de cerca de 1,07, indicando que, em nenhuma instância, o valor da solução encontrada por esses métodos diferiu em mais 7% do valor mínimo encontrado. Já para a abordagem LH, o desvio máximo foi de cerca de 1,27 e para CLSDPRL^w foi de 3,52.

Dessa forma, considerando os resultados aqui apresentados, as abordagens de solução RFH e INS se mostraram bastante promissoras. A *matheuristic* INS obteve os melhores resultados gerais, se mostrando mais eficiente do que RFH para instâncias maiores. Por outro lado, a *matheuristic* INS consumiu mais tempo computacional do que a heurística RFH em três grupos de instâncias e ofereceu piores soluções em dois grupos (C2G1 e C2G2). Além disso, por ter natureza estocástica, a *matheuristic* INS pode não oferecer a mesma solução quando aplicada diversas vezes numa mesma instância de teste, fato este, não esperado na heurística RFH que possui natureza determinística.

Contudo, a *matheuristic* INS oferece maior flexibilidade de trabalho, uma vez que novas

Figura 15 – Comparação entre os principais métodos de solução.

estruturas de vizinhança podem ser propostas e novos procedimentos podem ser desenvolvidos para a obtenção de uma solução factível inicial.

CONCLUSÕES E PROPOSTAS PARA TRABALHOS FUTUROS

Este trabalho abordou um problema de dimensionamento e sequenciamento de lotes de produção com preparação dependente da sequência, inspirado em uma indústria alimentícia brasileira. O problema abordado difere dos problemas descritos na literatura, principalmente, porque diversas linhas de produção compartilham dos mesmos recursos produtivos escassos (máquinas e trabalhadores). As linhas não são fixas, mas devem ser montadas em cada período de acordo com as necessidades de produção.

A escassez dos recursos necessários para a ativação das linhas e as características da demanda impedem que todas as linhas de produção sejam montadas/ativadas simultaneamente, de modo que, os gestores necessitam decidir quais linhas de produção ativar em cada período de um horizonte de planejamento. Além disso, se faz necessário determinar os tamanhos dos lotes de produção e a sequência de produção desses lotes a fim de minimizar os custos envolvidos, garantindo o atendimento das demandas dos clientes, satisfazendo as restrições de capacidade e evitando que itens sejam deteriorados pelo prazo de validade.

Num primeiro momento, o problema estudado foi sistematicamente descrito (PDSL-LPR) e modelos clássicos da literatura para o PDSL, como o GLSP (proposto em [Fleischmann e Meyr \(1997\)](#)) e sua reformulação proposta em [Clark e Clark \(2000\)](#) e o CLSD (proposto em [Haase \(1996\)](#)), foram estendidos para a sua representação. Por meio de resultados teóricos e da análise de testes computacionais realizados com base em instâncias geradas de maneira aleatória, a extensão do modelo CLSD (intitulada CLSDPRL) foi apontada como a formulação matemática mais promissora.

Baseando-se no modelo CLSDPRL, quatro heurísticas construtivas baseadas na formulação matemática do problema foram propostas, sendo três do tipo *relax-and-fix* (RF1, RF2 e RFH) e uma baseada na decomposição do problema (*D*).

As heurísticas RF1 e RF2 são extensões de procedimentos existentes na literatura (Dillenberger *et al.* (1994), Escudero e Salmeron (2005), Sel e Bilgen (2014) e (JAMES; ALMADA-LOBO, 2011)) que exploram partições das variáveis binárias do problema por períodos. Já a heurística RFH baseia-se na reformulação CLSDPRL^w do modelo CLSDPRL, onde na primeira iteração considera-se a parte do sequenciamento de maneira linearmente relaxada e otimiza-se as variáveis binárias referentes às decisões de abertura de linhas (δ_{lt}) e de produção (w_{ljt}). Em seguida, na segunda iteração, fixa-se os valores das variáveis referentes à abertura das linhas (δ_{lt}) e otimiza-se as demais variáveis (inclusive, as variáveis w_{ljt}). O subproblema referente à segunda iteração pode ser decomposto por linhas de produção, facilitando a sua resolução.

A heurística *D* consiste na utilização de um modelo simplificado para obtenção de um padrão factível de aberturas de linhas e, em seguida, fixando-se os valores das variáveis de aberturas de linhas no modelo CLSDPRL, resolve-se um subproblema onde as demais variáveis são otimizadas. Novamente, esse subproblema pode ser solucionado de maneira decomposta por linhas de produção.

Do ponto de vista computacional, a heurística RFH foi capaz de oferecer melhores soluções factíveis do que as heurísticas RF1 e RF2 e se mostrou bastante eficiente para a solução completa do problema tratado. Por outro lado, a heurística *D* ofereceu resultados competitivos e apresentou o menor custo computacional. Dessa forma, a heurística *D* se mostrou apropriada para ser utilizada como método para obtenção de soluções iniciais visando alimentar procedimentos de melhoria.

Heurísticas de melhoramento baseadas em princípios de busca local e na formulação matemática do problema também foram investigadas. Três procedimentos do tipo *fix-and-optimize* (FOTO, FOALE2 e FOALE3) foram propostos, sendo um (FOTO) de natureza determinística e os demais de natureza estocástica. Além disso, diversas estruturas de vizinhança foram desenvolvidas a fim de obter *matheuristics* do tipo INS (James e Almada-Lobo (2011)) para lidar com o PDSL-LPR. Resultados computacionais mostraram que a utilização do procedimento *D* para fornecer soluções a um procedimento do tipo INS foi eficiente para a solução do problema estudado, apresentado um desvio médio em relação ao limitante inferior obtido pelo resolvidor Cplex em torno de apenas 5,5%.

Abordagens baseadas em relaxação lagrangiana foram propostas para solucionar o PDSL-LPR. Primeiramente, um esquema de decomposição lagrangiana proposto em Xiao *et al.* (2015) para solucionar um PDSL foi estendido para o PDSL-LPR (abordagem LD) e, em seguida, um esquema de relaxação de restrições (LH) foi proposto visando obter melhores limitantes. Em ambos os casos, o subproblema lagrangiano pôde ser decomposto em um problema de dimensionamento com decisões a respeito da abertura de linhas e em um problema de sequenciamento em linhas paralelas. Um procedimento baseado na formulação matemática do problema foi proposto para encontrar soluções factíveis para o problema original a partir da solução do subproblema dual lagrangiano. Esse procedimento de factibilização pode ser utilizado

nas duas abordagens lagrangianas (LD e LH). Testes foram realizados utilizando dois métodos para solução do problema dual lagrangiano: o método do subgradiente e o algoritmo do volume.

Computacionalmente, a abordagem LH ofereceu melhores soluções factíveis do que a abordagem LD, porém na abordagem LH, a solução exata do subproblema lagrangiano exigiu elevado tempo computacional, impedindo a obtenção de limitantes duais para algumas instâncias dentro do limite de tempo computacional disponível. Já na abordagem LD, o subproblema lagrangiano pôde ser solucionado até a otimalidade em menor tempo, porém o problema dual lagrangiano associado a abordagem LD não pôde ser bem resolvido pelos métodos utilizados (método do subgradiente e algoritmo do volume) resultando em limitantes inferiores de má qualidade.

Dessa forma, assim como os procedimentos construtivos e de melhoramento, as heurísticas lagrangianas conseguiram proporcionar apenas soluções factíveis de boa qualidade, mas não obtiveram bons limitantes inferiores.

Futuros trabalhos podem explorar diferentes esquemas de relaxação lagrangiana, visando obter subproblemas lagrangianos que possam ser resolvidos com maior facilidade, evitando a utilização do algoritmo *Branch-and-Bound*. Além disso, a estrutura do problema sugere que heurísticas baseadas em geração de colunas possam fornecer bons resultados para o problema estudado.

Observa-se ainda, que todos os métodos propostos neste trabalho operam por meio da decomposição do problema em subproblemas que podem ser solucionados utilizando uma técnica de programação matemática. Entretanto, pode ser interessante o desenvolvimento de meta-heurísticas, tais como *simulated annealing* e busca tabu, utilizando esquemas baseados em busca local com movimentos aleatórios sem que subproblemas de programação matemática inteira mista sejam solucionados a cada iteração.

Do ponto de vista do problema de planejamento da produção, com a finalidade de obter maiores ganhos na gestão da cadeia de suprimentos, pode ser investigada a integração do problema de dimensionamento e sequenciamento com o problema de planejamento da distribuição. Nesse caso, os modelos e métodos propostos nesta tese podem ser estendidos para considerar esse novo cenário.

REFERÊNCIAS

- ABSI, N.; ARCHETTI, C.; DAUZÈRE-PÉRÈS, S.; FEILLET, D. An iterative two-phase heuristic approach for the production routing problem. **Transportation Science**, p. 1–25, 2014. To appear. Citado na página 40.
- ALMADA-LOBO, B.; CLARK, A.; GUIMARÃES, L.; FIGUEIRA, G.; AMORIM, P. Industrial insights into lot sizing and scheduling modeling. **Pesquisa Operacional**, SciELO Brasil, v. 35, n. 3, p. 439–464, 2015. Citado nas páginas 23, 24 e 28.
- 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. Citado nas páginas 24 e 33.
- ALMEDER, C.; ALMADA-LOBO, B. Synchronisation of scarce resources for a parallel machine lotsizing problem. **International Journal of Production Research**, Taylor & Francis, v. 49, n. 24, p. 7315–7335, 2011. Citado nas páginas 25, 33 e 55.
- ARAÚJO, S. A. de; ARENALES, M. N.; CLARK, A. R. Lot sizing and furnace scheduling in small foundries. **Computers & Operations Research**, Elsevier, v. 35, n. 3, p. 916–932, 2008. Citado na página 33.
- BALDO, T. A.; SANTOS, M. O.; ALMADA-LOBO, B.; MORABITO, R. An optimization approach for the lot sizing and scheduling problem in the brewery industry. **Computers & Industrial Engineering**, Elsevier, v. 72, p. 58–71, 2014. Citado na página 70.
- BARAHONA, F.; ANBIL, R. The volume algorithm: producing primal solutions with a subgradient method. **Mathematical Programming**, Springer, v. 87, n. 3, p. 385–399, 2000. Citado nas páginas 46, 47, 48, 89, 90 e 116.
- BAZARAA, M. S.; SHERALI, H. D. On the choice of step size in subgradient optimization. **European Journal of Operational Research**, Elsevier, v. 7, n. 4, p. 380–388, 1981. Citado na página 46.
- BRAHIMI, N.; ABSI, N.; DAUZÈRE-PÉRÈS, S.; KEDAD-SIDHOUM, S. Models and lagrangian heuristics for a two-level lot-sizing problem with bounded inventory. **OR Spectrum**, Springer, v. 37, n. 4, p. 983–1006, 2015. Citado na página 40.
- BRAHIMI, N.; DAUZÈRE-PÉRÈS, S. A lagrangian heuristic for capacitated single item lot sizing problems. **4OR**, Springer, v. 13, n. 2, p. 173–198, 2015. Citado na página 45.
- 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. Citado nas páginas 40, 45, 49, 87, 115 e 117.
- CASERTA, M.; VOSS, S. Metaheuristics: intelligent problem solving. In: **Matheuristics**. [S.l.]: Springer, 2009. p. 1–38. Citado na página 37.

CHEN, H. Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems. **Omega**, Elsevier, v. 56, p. 25–36, 2015. Citado nas páginas 37 e 39.

CLARK, A. R.; CLARK, S. J. Rolling-horizon lot-sizing when set-up times are sequence-dependent. **International Journal of Production Research**, Taylor & Francis, v. 38, n. 10, p. 2287–2307, 2000. Citado nas páginas 31, 32, 55, 59, 60 e 123.

COPIL, K.; WÖRBELAUER, M.; MEYR, H.; TEMPELMEIER, H. Simultaneous lotsizing and scheduling problems: a classification and review of models. **OR Spectrum**, Springer, v. 39, n. 1, p. 1–64, 2017. Citado nas páginas 23, 24, 28, 33 e 34.

DILLENBERGER, C.; ESCUDERO, L. F.; WOLLENSAK, A.; ZHANG, W. On practical resource allocation for production planning and scheduling with period overlapping setups. **European Journal of Operational Research**, Elsevier, v. 75, n. 2, p. 275–286, 1994. Citado nas páginas 34, 35, 71 e 124.

DOLAN, E. D.; MORÉ, J. J. Benchmarking optimization software with performance profiles. **Mathematical programming**, Springer, v. 91, n. 2, p. 201–213, 2002. Citado na página 120.

DREXL, A.; HAASE, K. Proportional lotsizing and scheduling. **International journal of production economics**, Elsevier, v. 40, p. 73–87, 1995. Citado nas páginas 28 e 29.

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. Citado nas páginas 55 e 61.

ESCUDERO, L. F.; SALMERON, J. On a fix-and-relax framework for a class of project scheduling problems. **Annals of Operations Research**, Springer, v. 140, n. 1, p. 163–188, 2005. Citado nas páginas 34, 71 e 124.

FERREIRA, D.; MORABITO, R.; RANGEL, S. Solution approaches for the soft drink integrated production lot sizing and scheduling problem. **European Journal of Operational Research**, Elsevier, v. 196, n. 2, p. 697–706, 2009. Citado nas páginas 24, 33, 34 e 35.

FISHER, M. L. The lagrangian relaxation method for solving integer programming problems. **Management science**, INFORMS, v. 50, n. 12_supplement, p. 1861–1871, 2004. Citado nas páginas 40, 44, 45, 46 e 81.

FLEISCHMANN, B. The discrete lot-sizing and scheduling problem. **European Journal of Operational Research**, Elsevier, v. 44, n. 3, p. 337–348, 1990. Citado na página 28.

FLEISCHMANN, B.; MEYR, H. The general lotsizing and scheduling problem. **Operations-Research-Spektrum**, Springer, v. 19, n. 1, p. 11–21, 1997. Citado nas páginas 24, 28, 31, 56, 58 e 123.

FURLAN, M.; ALMADA-LOBO, B.; SANTOS, M.; MORABITO, R. Unequal individual genetic algorithm with intelligent diversification for the lot-scheduling problem in integrated mills using multiple-paper machines. **Computers & Operations Research**, Elsevier, v. 59, p. 33–50, 2015. Citado nas páginas 24, 33 e 37.

_____. Unequal individual genetic algorithm with intelligent diversification for the lot-scheduling problem in integrated mills using multiple-paper machines. **Computers & Operations Research**, Elsevier, v. 59, p. 33–50, 2015. Citado nas páginas 24 e 55.

GLOCK, C. H.; GROSSE, E. H.; RIES, J. M. The lot sizing problem: A tertiary study. **International Journal of Production Economics**, Elsevier, v. 155, p. 39–51, 2014. Citado nas páginas 23 e 27.

GUIGNARD, M. Lagrangean relaxation. **Top**, Springer, v. 11, n. 2, p. 151–200, 2003. Citado nas páginas 41, 42, 43, 44 e 45.

GUIMARÃES, L.; KLABJAN, D.; ALMADA-LOBO, B. Pricing, relaxing and fixing under lot sizing and scheduling. **European Journal of Operational Research**, Elsevier, v. 230, n. 2, p. 399–411, 2013. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221713003445>>. Citado nas páginas 23 e 39.

_____. Modeling lotsizing and scheduling problems with sequence dependent setups. **European Journal of Operational Research**, Elsevier, v. 239, n. 3, p. 644–662, 2014. Citado nas páginas 23, 33, 58, 60 e 61.

HAASE, K. Capacitated lot-sizing with sequence dependent setup costs. **Operations-Research-Spektrum**, Springer, v. 18, n. 1, p. 51–59, 1996. Citado nas páginas 24, 28, 32, 55, 62 e 123.

HAASE, K.; KIMMS, A. Lot sizing and scheduling with sequence-dependent setup costs and times and efficient rescheduling opportunities. **International Journal of Production Economics**, Elsevier, v. 66, n. 2, p. 159–169, 2000. Citado na página 24.

HELD, M.; KARP, R. M. The traveling-salesman problem and minimum spanning trees. **Operations Research**, INFORMS, v. 18, n. 6, p. 1138–1162, 1970. Citado nas páginas 40 e 45.

HELD, M.; WOLFE, P.; CROWDER, H. P. Validation of subgradient optimization. **Mathematical programming**, Springer, v. 6, n. 1, p. 62–88, 1974. Citado nas páginas 45, 46 e 87.

JAMES, R. J.; ALMADA-LOBO, B. Single and parallel machine capacitated lotsizing and scheduling: New iterative mip-based neighborhood search heuristics. **Computers & Operations Research**, Elsevier, v. 38, n. 12, p. 1816–1825, 2011. Citado nas páginas 24, 34, 35, 38, 67, 75, 78 e 124.

KARIMI, B.; GHOMI, S. M. T. F.; WILSON, J. M. The capacitated lot sizing problem: a review of models and algorithms. **Omega**, v. 31, n. 5, p. 365 – 378, 2003. Citado nas páginas 27 e 28.

KARMARKAR, U. S.; SCHRAGE, L. The deterministic dynamic product cycling problem. **Operations Research**, INFORMS, v. 33, n. 2, p. 326–345, 1985. Citado na página 28.

KOPANOS, G. M.; PUIGJANER, L.; MARAVELIAS, C. T. Production planning and scheduling of parallel continuous processes with product families. **Industrial & Engineering Chemistry Research**, ACS Publications, v. 50, n. 3, p. 1369–1378, 2010. Citado na página 33.

LAZIC, J. **New Variants of Variable Neighbourhood Search for 0-1 Mixed Integer Programming and Clustering**. [S.l.]: Brunel University, 2010. Citado na página 37.

MEYR, H. Simultaneous lotsizing and scheduling by combining local search with dual reoptimization. **European Journal of Operational Research**, Elsevier, v. 120, n. 2, p. 311–326, 2000. Citado na página 53.

_____. Simultaneous lotsizing and scheduling on parallel machines. **European Journal of Operational Research**, Elsevier, v. 139, n. 2, p. 277–292, 2002. Citado nas páginas 37 e 55.

- MEYR, H.; MANN, M. A decomposition approach for the general lotsizing and scheduling problem for parallel production lines. **European Journal of Operational Research**, Elsevier, v. 229, n. 3, p. 718–731, 2013. Citado nas páginas 24 e 25.
- MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. Integer programming formulation of traveling salesman problems. **Journal of the ACM (JACM)**, ACM, v. 7, n. 4, p. 326–329, 1960. Citado nas páginas 32 e 33.
- MLADENović, N.; HANSEN, P. Variable neighborhood search. **Computers & operations research**, Elsevier, v. 24, n. 11, p. 1097–1100, 1997. Citado nas páginas 38 e 75.
- POCHET, Y.; WOLSEY, L. A. **Production Planning by Mixed Integer Programming (Springer Series in Operations Research and Financial Engineering)**. [S.l.: s.n.], 2006. ISBN 0387299599. Citado na página 36.
- SANTOS, M. O.; MASSAGO, S.; ALMADA-LOBO, B. Infeasibility handling in genetic algorithm using nested domains for production planning. **Computers & Operations Research**, ELSEVIER, v. 37, p. 1113–1122, 2010. Citado na página 37.
- SEEANNER, F.; ALMADA-LOBO, B.; MEYR, H. Combining the principles of variable neighborhood decomposition search and the fix&optimize heuristic to solve multi-level lot-sizing and scheduling problems. **Computers & Operations Research**, Elsevier, v. 40, n. 1, p. 303–317, 2013. Citado na página 37.
- SEL, Ç.; BILGEN, B. Hybrid simulation and mip based heuristic algorithm for the production and distribution planning in the soft drink industry. **Journal of Manufacturing Systems**, Elsevier, v. 33, n. 3, p. 385–399, 2014. Citado nas páginas 34, 35, 36, 70 e 124.
- SEYEDHOSSEINI, S.; GHOREYSHI, S. An integrated model for production and distribution planning of perishable products with inventory and routing considerations. **Mathematical Problems in Engineering**, Hindawi Publishing Corporation, v. 2014, 2014. Citado na página 57.
- SHIGNEMOTO, A. L.; ARMENTANO, V. A. A tabu search procedure for coordinating production, inventory and distribution routing problems. **International Transactions in Operational Research**, Blackwell, v. 17, p. 179–195, 2010. Citado na página 37.
- SIFALERAS, A.; KONSTANTARAS, I.; MLADENović, N. Variable neighborhood search for the economic lot sizing problem with product returns and recovery. **International Journal of Production Economics**, Elsevier, v. 160, p. 133–143, 2015. Citado na página 37.
- SILVA, T. T. d. Um estudo sobre limites duais para o problema integrado de dimensionamento de lotes e sequenciamento da produção. Universidade Estadual Paulista (UNESP), 2015. Citado nas páginas 46 e 48.
- SÖRENSEN, K.; GLOVER, F. W. Metaheuristics. In: **Encyclopedia of operations research and management science**. [S.l.]: Springer, 2013. p. 960–970. Citado nas páginas 37 e 38.
- TOLEDO, C. F. M.; ARANTES, M. da S.; HOSSOMI, M. Y. B.; FRANÇA, P. M.; AKAR-TUNALI, K. A relax-and-fix with fix-and-optimize heuristic applied to multi-level lot-sizing problems. **Journal of Heuristics**, Springer, v. 21, n. 5, p. 687–717, 2015. Citado nas páginas 24 e 34.

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. Citado nas páginas 40, 45, 48 e 81.

WAGNER, H. M.; WHITIN, T. M. Dynamic Version of the Economic Lot Size Model. **MANAGEMENT SCIENCE**, v. 5, n. 1, p. 89–96, 1958. Citado na página 81.

WOLOSEWICZ, C.; DAUZÈRE-PÉRÈS, S.; AGGOUNE, R. A lagrangian heuristic for an integrated lot-sizing and fixed scheduling problem. **European Journal of Operational Research**, Elsevier, v. 244, n. 1, p. 3–12, 2015. Citado nas páginas 40 e 45.

WOLSEY, L. A. Mip modelling of changeovers in production planning and scheduling problems. **European Journal of Operational Research**, Elsevier, v. 99, n. 1, p. 154–165, 1997. Citado na página 24.

_____. **Integer programming**. [S.l.]: Wiley New York, 1998. v. 42. Citado nas páginas 41, 44 e 59.

XIAO, J.; YANG, H.; ZHANG, C.; ZHENG, L.; GUPTA, J. N. A hybrid lagrangian-simulated annealing-based heuristic for the parallel-machine capacitated lot-sizing and scheduling problem with sequence-dependent setup times. **Computers & Operations Research**, Elsevier, 2015. Citado nas páginas 24, 25, 33, 40, 43, 45, 49, 81, 82, 86, 117 e 124.

XIAO, J.; ZHANG, C.; ZHENG, L.; GUPTA, J. N. Mip-based fix-and-optimize algorithms for the parallel machine capacitated lot-sizing and scheduling problem. **International Journal of Production Research**, Taylor & Francis, v. 51, n. 16, p. 5011–5028, 2013. Citado na página 24.