

Natã Goulart da Silva

**FORMULAÇÕES E HEURÍSTICAS PARA O
PROBLEMA INTEGRADO DE
DIMENSIONAMENTO DE LOTES E DE
EMPACOTAMENTO DE PRODUTOS**

Belo Horizonte

2018

Natã Goulart da Silva

**FORMULAÇÕES E HEURÍSTICAS PARA O
PROBLEMA INTEGRADO DE DIMENSIONAMENTO
DE LOTES E DE EMPACOTAMENTO DE PRODUTOS**

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Escola de Engenharia da Universidade Federal de Minas Gerais, para fins de obtenção do título de doutorado

Universidade Federal de Minas Gerais - UFMG

Departamento de Engenharia de Produção

Orientadores:

Prof. Maurício Cardoso de Souza

Prof. Thiago Ferreira de Noronha

Belo Horizonte

2018

FORMULAÇÕES E HEURÍSTICAS PARA O PROBLEMA INTEGRADO DE
DIMENSIONAMENTO DE LOTES E DE EMPACOTAMENTO DE PRODUTOS

Natã Goulart da Silva.

Belo Horizonte, 2018

80 p.

Orientadores:

Prof. Maurício Cardoso de Souza

Prof. Thiago Ferreira de Noronha

Tese (Doutorado) – Universidade Federal de Minas Gerais - UFMG

Departamento de Engenharia de Produção, 2018.

1. Problemas integrados de produção e entrega.

2. Dimensionamento de Lotes.

3. Empacotamento de Produtos.

I. Universidade Federal de Minas Gerais.

II. Programa de Pós-Graduação em Engenharia de Produção.

CDU 62.002




FOLHA DE APROVAÇÃO

Formulações e Heurísticas para o Problema Integrado de Dimensionamento de Lotes e de Empacotamento de Produtos


NATA GOULART DA SILVA

Tese submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA DE PRODUÇÃO, como requisito para obtenção do grau de Doutor em ENGENHARIA DE PRODUÇÃO, área de concentração PESQUISA OPERACIONAL E INTERVENÇÃO EM SISTEMAS SOCIOTÉCNICOS, linha de pesquisa Mod. e Algorit. de Otimiz. para Sistemas em Redes e de Prod..

Aprovada em 20 de dezembro de 2018, pela banca constituída pelos membros:


Prof(a). Maurício Cardoso de Souza - Orientador
UFMG


Prof(a). Thiago Ferreira de Noronha - Coorientador
UFMG


Prof(a). Marcone Jamilson Freitas Souza
Universidade Federal de Ouro Preto


Prof(a). Thiago Augusto de Oliveira Silva
UFOP


Prof(a). Alexandre Xavier Martins
UFOP


Prof(a). Ricardo Saraiva de Camargo
UFMG

Belo Horizonte, 20 de dezembro de 2018.

Agradecimentos

Agradeço a minha esposa Roseane pelo apoio e compreensão ao longo desta jornada. Agradeço também a minha família, amigos e colegas de trabalho. Esta tese de forma alguma seria possível sem as orientações e contribuições dos professores Maurício Cardoso de Souza e Thiago Ferreira de Noronha, meus orientadores. Um agradecimento final e muito especial para minha mãe, sempre me levando pela mão pelo bom caminho e que não mediu esforços para este objetivo fosse alcançado.

Resumo

No Problema Integrado de Dimensionamento de Lotes e Empacotamento de Produtos, as decisões relativas ao problema de Dimensionamento de Lotes para o reabastecimento de cada produto com as decisões de Empacotamento para atender aos pedidos dos clientes realizam-se de maneira integrada. Um pedido de um cliente pode ser composto por diferentes quantidades de produtos que devem ser entregues em um mesmo bin em até, no máximo, um determinado período. Pode-se entregar os pedidos antecipadamente caso seja vantajoso em termos de custos. Assume-se um conjunto suficientemente grande de bins homogêneos disponíveis em cada período capaz de atender a todos os pedidos do horizonte de planejamento, caso necessário. Os custos envolvidos são os custos fixos de compra, custos de manutenção de estoque e o custo de usar um bin em um determinado período. Todos os custos podem variar ao longo do horizonte de planejamento e o objetivo é minimizar o custo total. São propostas formulações por Programação Linear Inteira (PLI) e uma relaxação combinatória na qual não é mais necessário determinar o bin específico em que cada pedido é inserido. Neste caso, calcula-se a demanda de empacotamento agregada por período. São propostas, também, heurísticas usando diferentes estratégias para conectar o problema de Dimensionamento de Lotes e o problema de Empacotamento. Experimentos computacionais em instâncias com diferentes configurações permitiram concluir que os métodos heurísticos propostos são formas eficientes de obter bons gaps de otimalidade em tempos computacionais reduzidos.

Palavras-chave: Problemas integrados de produção e entrega; Dimensionamento de Lotes; Empacotamento de Produtos; Heurísticas.

Abstract

In the Integrated Uncapacitated Lot Sizing and Bin Packing problem we have to couple lot sizing decisions of replenishment from single products suppliers with bin packing decisions in the delivering of client orders. A client order is composed by quantities of each product, and the quantities of such order must be delivered all together no later than a given period. The quantities of an order must all be packed in a same bin, and may be delivered in advance if its advantageous in terms of costs. We assume a large enough set of homogeneous bins available at each period. The costs involved are setup and inventory holding costs and the cost to use a bin as well. All costs are variable in the planning horizon, and the objective is to minimize the total cost incurred. We propose mixed integer linear programming formulations, and a combinatorial relaxation where it is no longer necessary to keep track of the specific bin where each order is packed. An aggregate delivering capacity is computed instead. We also propose heuristics using different strategies to couple the lot sizing and the bin packing subproblems. Computational experiments on instances with different configurations show that the proposed methods are efficient ways to obtain small optimality gaps in reduced computational times.

Keywords: Integrated production-delivering problems; Lot sizing; Bin packing; Heuristics.

Lista de tabelas

Tabela 1	– Resultados de duas variantes do algoritmo de <i>branch and bound</i> padrão do CPLEX. O B&B-SF baseia-se na formulação <i>SF</i> , enquanto o B&B-MF na formulação <i>MF</i>	43
Tabela 2	– Resultados dos algoritmos B&B-SF e B&B-MF sem nenhum corte habilitado.	44
Tabela 3	– Desempenho das quatro abordagens para obter limites inferiores para o IULSBP. As duas primeiras consistem na solução da relaxação linear das formulações <i>SF</i> e <i>MF</i> , respectivamente. A terceira e quarta consistem na solução da relaxação combinatória <i>CR</i> e $CR^{w_t \geq 0}$	46
Tabela 4	– Desempenho das heurísticas UH, LSBP, CRH e BPLS.	57
Tabela 5	– Experimentos com a heurística ReF que utiliza a formulação <i>SF</i> com diferentes valores de t_i , t_s e t_k	59
Tabela 6	– Buscas locais de dois e três períodos aplicadas a uma heurística construtiva para o IULSBP	62
Tabela 7	– heurística BLN-OPT e heurística BPLS para o IULSBP	63
Tabela 8	– Resultados dos experimentos com as heurísticas FFOLT e MS-LSBP.	71
Tabela 9	– Resultados da heurística MS-LBSP com duas execuções do algoritmo genético BRKGA-LSBP.	73
Tabela 10	– Resultados da melhor heurística determinística (BLN-OPT) e da melhor heurística estocástica (BRKGA-LSBP).	74

Sumário

	Lista de tabelas	11
1	INTRODUÇÃO	13
2	TRABALHOS RELACIONADOS	17
2.1	Fundamentação Teórica	24
3	FORMULAÇÕES E LIMITES INFERIORES	33
3.1	Formulação de Fluxo Único <i>SF</i>	33
3.2	Formulação de Múltiplos Fluxos <i>MF</i>	35
3.3	Relaxação Combinatória <i>CR</i>	37
3.4	Experimentos Computacionais das Formulações e Relaxações	38
3.4.1	Instâncias de Teste	38
3.4.2	Resultados	40
4	HEURÍSTICAS DETERMINÍSTICAS PARA IULSBP	47
4.1	Heurística UH	47
4.2	Heurística LSBP	48
4.3	Heurística CRH	50
4.4	Heurística BPLS	51
4.5	Relaxar e Fixar - ReF	52
4.6	Heurísticas de Busca Local N-OPT	54
4.7	Experimentos Computacionais das Heurísticas Determinísticas	56
5	HEURÍSTICAS ESTOCÁSTICAS PARA IULSBP	65
5.1	Heurística <i>First Fit</i> - <i>FFOLT</i>	65
5.2	Heurística MS-LSBP	67
5.3	Algoritmo Genético BRKGA-LSBP	68
5.4	Experimentos Computacionais das Heurísticas Estocásticas	69
6	CONSIDERAÇÕES FINAIS	75
	REFERÊNCIAS	77

1 Introdução

O gerenciamento eficiente das atividades de compra, produção, armazenamento e distribuição é fundamental para a redução dos custos finais e, conseqüentemente, a competitividade das empresas. Neste trabalho avaliam-se formas de se resolver o problema Integrado de Dimensionamento de Lotes e de Empacotamento de Produtos que são dois importantes problemas que aparecem em muitos processos relacionados com a compra e entrega de produtos. Em uma análise integrada, as decisões de compra e empacotamento são tomadas considerando o cenário global, o que implicam em uma maior dificuldade para a solução do problema e, geralmente, em soluções mais eficientes e econômicas.

Na versão do Problema de Dimensionamento de Lotes (do inglês *Lot Sizing Problem - LSP*) (WAGNER; WHITIN, 1958) utilizada neste trabalho, deve-se determinar quando comprar um determinado produto, considerando um horizonte de planejamento finito. Neste problema, ocorrem as decisões conflitantes da redução dos custos fixos de compra com os custos de manutenção de estoque. Comprar lotes maiores, geralmente, significa reduzir os custos relacionados, mas por outro lado, pode ocorrer o aumento nos custos de manutenção de estoque. Na versão do problema de Empacotamento (do inglês *Bin Packing Problem - BPP*) (MARTELLO; TOTH, 1990) estudada neste trabalho, deve-se definir como e quando empacotar os pedidos. Então, tem-se um problema de Empacotamento com múltiplos períodos no qual um pedido pode ser empacotado em qualquer período t menor que o seu período limite de atendimento e a partir do período no qual exista estoque para todos os seus produtos. Neste problema, os bins têm tamanhos fixos, mas custos variáveis ao longo dos períodos.

Este problema integrado ocorre em situações práticas em que, por exemplo, uma empresa intermediária ou um armazém compra itens de fornecedores para compor pedidos com múltiplos produtos para um conjunto de clientes. A Figura 1 apresenta um cenário no qual ocorrem as decisões de compra de três diferentes produtos para atender demandas conhecidas dos pedidos dos clientes, que podem ser compostos por diferentes quantidades de

cada produto. O problema de Dimensionamento de Lotes e o problema de Empacotamento são avaliados de forma integrada com o objetivo de fornecerem custos totais menores do que em abordagens nas quais eles são estudados de forma isolada.

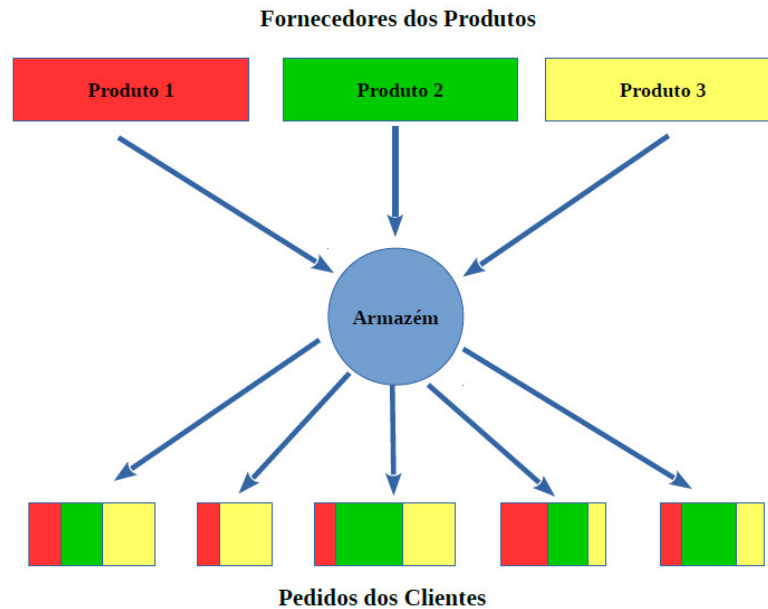


Figura 1 – Uma empresa intermediária compra itens de fornecedores para compor pedidos com múltiplos produtos para um conjunto de clientes.

Apresenta-se neste trabalho o *Problema integrado de Dimensionamento de Lotes e de Empacotamento de Produtos* (do inglês *Integrated Uncapacitated Lot Sizing and Bin Packing problem - IULSBP*). Sejam os conjuntos K de bins, P de produtos, T de períodos e V de pedidos de clientes. Um pedido $i \in V$ é empacotado e entregue até o período limite $t_i \in T$, sendo composto por quantidades q_{ip} de cada produto $p \in P$. Entregam-se os produtos de um pedido todos juntos em um período $t \leq t_i$. Seja K um conjunto de bins homogêneos com capacidade Q , disponível em cada período $t \in T$. Empacota-se os produtos de um pedido i em um mesmo bin $k \in K$ e um bin pode ser usado para empacotar mais de um pedido caso sua capacidade não for excedida. Assume-se que a quantidade de bins deve ser suficiente para que todos os pedidos possam ser empacotados e entregues em um único período, se necessário.

O IULSBP consiste em decidir de forma integrada, em um horizonte de planejamento finito, (i) o tamanho dos lotes de compra para cada produto $p \in P$ no período $t \in T$, (ii) como empacotar os pedidos em V e (iii) quando empacotar e entregar esses pedidos.

Há um custo c_{pt} quando compra-se um lote de um produto p no período t e um custo de estoque e_{pt} para cada unidade de produto p presente no estoque ao final do período t . Disponíveis todas as quantidades q_{ip} de todos os produtos que o compõe um pedido, empacote-se este pedido $i \in V$ em qualquer bin para ser entregue em um $t = 1, \dots, t_i$. Assume-se que, caso o pedido i seja empacotado em um período $t < t_i$, ainda assim existe o custo de manutenção de estoque para as quantidades q_{ip} nos períodos $t, t + 1, \dots, t_i - 1$, já que repõe-se o capital investido apenas na data de vencimento do pedido t_i . Além disso, existe um custo f_t para cada bin utilizado no período $t \in T$. O IULSBP consiste em atender a todos os pedidos dos clientes com um custo total mínimo.

Neste trabalho são propostas uma formulação por Programação Linear Inteira (PLI) para o problema e, em seguida, uma reformulação usando a chamada formulação por múltiplos fluxos das variáveis do problema de Dimensionamento de Lotes. Propõe-se também uma relaxação combinatória com o objetivo de obter bons limites inferiores em um período de tempo reduzido. Com relação aos limites superiores, desenvolveram-se inicialmente heurísticas segundo duas estratégias: (i) Resolver primeiro o problema de Dimensionamento do Lotes; e (ii) Resolver primeiro o problema de Empacotamento. Em seguida, utilizam-se heurísticas baseadas nas formulações para o problema. Finalmente, desenvolveram-se algoritmos multipartida, heurísticas com buscas locais e um algoritmo genético. Através de experimentos computacionais realizados em instâncias com diferentes configurações, obtiveram-se gaps de otimalidade médios da ordem de 1%.

Os principais objetivos deste trabalho são: (i) Avançar no conhecimento sobre problemas integrados de produção e entrega; (ii) Apresentar formulações e relaxações para geração de limites inferiores; (iii) Desenvolver heurísticas para a solução do problema.

No próximo capítulo apresentam-se trabalhos relacionados com os problemas de Dimensionamento de Lotes, Empacotamento de Produtos e problemas integrados. Propõem-se no Capítulo 3 formulações por PLI e para obter limites inferiores. Heurísticas determinísticas são propostas no Capítulo 4. No Capítulo 5 apresentam-se algoritmos multipartida e um algoritmo genético. Considerações finais são colocadas no último capítulo.

2 Trabalhos Relacionados

Neste capítulo apresentam-se inicialmente alguns trabalhos da literatura sobre o problema de Dimensionamento de Lotes. Em seguida, são discutidos trabalhos que tratam do problema de Empacotamento. Posteriormente, expõem-se trabalhos que abordam problemas de Dimensionamento de Lotes e Gerenciamento de Estoques integrados a problemas de Roteamentos de Veículos e Agregação de Produtos. Na última parte do capítulo, Seção 2.1, tem-se trabalhos com métodos utilizados nas heurísticas para o problema integrado como técnicas utilizadas nos algoritmos multipartida e um algoritmo genético de chaves aleatórias.

Dentre os trabalhos que consideram a produção de um único item no problema de Dimensionamento de Lotes, destacam-se os seguintes. [Wagner e Whitin \(1958\)](#) utilizaram um algoritmo de programação dinâmica com complexidade $O(|T|^2)$ para a versão sem capacidade, sendo $|T|$ o número de períodos. [Aggarwal e Park \(1993\)](#) propuseram algoritmos com complexidade de tempo linear para versões não capacitadas e com custos de produção monotonicamente crescentes ou decrescentes ao longo dos períodos. Estes autores propuseram algoritmos com complexidade de tempo de $O(|T| \log |T|)$ para problemas com custos arbitrários. [Akbalik et al. \(2015\)](#) resolveram uma versão com capacidade de produção fixa utilizando um algoritmo com complexidade $O(|T|^4)$. [Chowdhury et al. \(2018\)](#) propuseram um algoritmo que resolve a mesma versão estudada por [Wagner e Whitin \(1958\)](#) com complexidade de tempo de $O(|T|)$ encontrando caminhos mais curtos em um grafo que representa o problema.

Em [\(DREXL; KIMMS, 1997\)](#) realizou-se uma revisão bibliográfica sobre o problema de Dimensionamento de Lotes em que apresentou-se formulações e discutiram-se variações que consideram os custos de preparação e de estoque. Esse trabalho apresentou versões do problema com um e múltiplos itens, com e sem capacidade de produção, com dependência de produção entre diferentes itens, com múltiplos níveis de produção, entre outros aspectos.

Para o problema de Empacotamento, [Carvalho \(1999\)](#) utilizou um algoritmo de

branch-and-price baseado no problema de Fluxo Mínimo (FATHABADI et al., 2012) e obteve uma redução no número de soluções simétricas definindo padrões ordenados de empacotamento. Já Carvalho (2002) apresentou formulações por *PLI* e técnicas que permitiram o desenvolvimento de novos algoritmos de *branch-and-price*. Delorme et al. (2016) realizaram uma revisão bibliográfica, comparações entre formulações por *PLI* e algoritmos para o problema. Um dos algoritmos que apresentou os melhores resultados em (DELORME et al., 2016) foi o proposto por Brandao e Pedroso (2016). Estes últimos desenvolveram um algoritmo chamado *Vector Packing Solver - VPSolver*, disponível em (BRANDAO, 2017), que pode ser utilizado para resolver problemas de Empacotamento com mais de uma dimensão, problemas de Corte, problemas de Empacotamento com Conflitos, entre outros.

O algoritmo de (BRANDAO; PEDROSO, 2016) é uma extensão do trabalho de (CARVALHO, 1999). Este algoritmo utiliza técnicas de compactação possibilitando uma redução significativa no número de vértices e arcos do grafo, permitindo uma representação eficiente de cada instância do problema. Após a compactação do grafo, transforma-se o problema resultante em uma formulação por *PLI*, sendo resolvido por pacotes de otimização. A saída da solução do problema de fluxo mínimo é transformada em uma solução para a instância do problema de Empacotamento original. O método de compactação utilizado em (BRANDAO; PEDROSO, 2016) pode apresentar algumas dificuldades em grafos de instâncias em que a capacidade dos bins e o número de diferentes itens aumentam, ocasionando um grande número de padrões de empacotamento.

Ben-Khedher e Yano (1994) abordaram um problema no qual as demandas por itens são dadas em termos de contêineres, e os contêineres são embarcados em caminhões. Não há custo de configuração, sendo que, os itens são comprados de acordo com as decisões de envio.

Chan et al. (2002a) apresentaram um problema no qual um centro de distribuição recebia produtos de uma fábrica e imediatamente os entregava aos clientes (*cross-docking*). As demandas dos clientes foram geradas apenas quando o nível de estoque chegava a zero, estratégia conhecida como *ZIO*, do inglês *Zero Inventory Order*. Chan et al. (2002b)

utilizaram uma abordagem semelhante a aplicada em (CHAN et al., 2002a) para resolver o problema de Dimensionamento de Lotes não capacitado com um único produto.

Bertazzi et al. (2005) resolveram o problema *Integrated Vendor Managed Inventory System*. Neste problema, consideram-se os custos fixos e variáveis de produção, custos de estoque (na fábrica e nos clientes) e os custos de transporte. Heurísticas que decompõem o problema integrado no problema de Dimensionamento de Lotes e no problema de Roteamento de Veículos foram utilizadas. Os experimentos comprovaram os benefícios da utilização do controle de estoque integrado entre todos os clientes comparado com uma política na qual um veículo visita um cliente quando não há mais estoque.

Em (NORDEN; VELDE, 2005) realizou-se um estudo no qual a redução dos custos de transporte foi obtida através de um contrato com um custo fixo para um conjunto de R paletes. Os custos de transporte foram calculados baseados em três parâmetros: um custo fixo c_0 , um custo c_1 por palete até um limite de R paletes e um custo c_2 para os paletes extras, com $c_2 > c_1$. Foram propostas heurísticas e uma Relaxação Lagrangiana em experimentos com instâncias com até 20 produtos, 12 períodos e capacidade dos paletes igual a 100. Os resultados forneceram gaps de otimalidade menores que 1%.

Ertogral (2008) apresentou o problema *Single Stage Multi-item Uncapacitated Dynamic Lot Sizing with transportation cost*. Neste problema consideram-se os custos de estoque dos produtos, os custos fixos para cada pedido e um custo de transporte de acordo com o tamanho do lote solicitado. As restrições que modelam o custo de transporte baseiam-se em uma função linear por partes apresentada em (NEMHAUSER; WOLSEY, 1988). Os gaps obtidos entre o limite superior fornecido pela solução de uma formulação e o limite inferior fornecido pelo método do Subgradiente foram em cerca de 90% das instâncias inferiores a 1%.

Yaman e Azen (2008) apresentaram o problema *The Manufacture's Mixed Pallet Design*, o qual consiste em a partir de padrões predefinidos de alocação de produtos em paletes, determinar qual a combinação de padrões que atendem as demandas dos clientes, para pedidos de itens que não têm demanda suficiente para preencher totalmente um palete. As demandas estão distribuídas em um horizonte de planejamento T e os custos

estão relacionados com a quantidade de produtos em estoque e a quantidade de produtos que não são entregues. Os autores propuseram uma formulação e, através de testes com algumas desigualdades válidas, reduziram em cerca de 6% o gap inicial.

Bard e Nananukul (2009) resolveram o problema *Integrated Production-Inventory-Distribution-Routing* através de uma metodologia de duas fases. A primeira fase consistiu em resolver o problema de Dimensionamento de Lotes através de uma formulação que não continha as restrições de roteamento de veículos. A solução desta formulação foi utilizada como entrada para um problema de Roteamento capacitado, resolvido para cada período por uma heurística Busca Tabu (GLOVER; LAGUNA, 1999).

Em (MOLINA et al., 2009) abordou-se uma extensão do problema proposto por (NORDEN; VELDE, 2005) em que admitiam-se atrasos no empacotamento e consideravam-se custos diferentes para cada produto e para cada período. Nesta versão do problema, pretende-se encontrar um plano de produção que apresente um custo mínimo com relação aos custos de preparação, de armazenamento, de atraso e de transporte, considerando como custo de transporte a contratação de contêineres. Apesar da aplicação de uma Relaxação Lagrangiana combinada com a Relação Surrogate, os resultados foram semelhantes aos obtidos quando utilizada apenas a Relaxação Lagrangiana.

Bard e Nananukul (2010) utilizaram diferentes métodos para resolver o problema *Integrated Production and Inventory Routing - IPIR*. Eles avaliaram uma formulação, um método *Branch and Price* e uma heurística baseada em *Branch and Price*. Instâncias com 3 veículos, 30 clientes e 5 períodos não foram resolvidas na otimalidade pelo CPLEX dentro do limite de tempo de 90 minutos. A heurística baseada em *Branch and Price* resolveu instâncias com até 50 clientes e 8 períodos com limite de tempo de execução de 60 minutos.

Armentano et al. (2011) abordaram uma versão do *IPIR* no qual consideravam-se os seguintes custos: (i) custos de preparação para a produção, (ii) custo por unidade produzida, (iii) custo de estoque (nos clientes e na fábrica), (iv) custos pela contratação de cada veículo e (v) um custo por cada cliente visitado. No artigo, foram apresentadas uma formulação e duas metaheurísticas: uma Busca Tabu com memória reduzida e uma

Busca Tabu integrada com religamento de caminhos (do inglês *Path Relinking*).

[Liberalino et al. \(2011\)](#) abordaram uma versão do *IPIR* na qual um produto era construído em uma fábrica a partir de componentes produzidos em outras unidades. Cada fábrica fornecia um item específico e um único produto final era produzido. Os itens eram transportados entre as fábricas através de um conjunto de veículos com capacidade homogênea W . Foram propostas duas heurísticas, uma derivada da heurística *Relax-and-Fix* e outra heurística gulosa chamada *Production Sequence Fusion - PSF* que faz iterações entre os problemas de Dimensionamento de Lotes e de Roteamento de Veículos. A heurística *Relax-and-Fix* foi superior a PSF para as menores instâncias. Para as maiores instâncias, a heurística PSF apresentou melhores resultados.

[Sancak e Salman \(2011\)](#) trataram um problema no qual deseja-se minimizar os custos de transporte, representados por um custo fixo pela utilização de veículos, e os custos de estoque. Neste problema, um veículo transporta até r_i paletes do tipo i e, um palete do tipo i pode acomodar até p_j itens do tipo j . No trabalho, algumas demandas eram atendidas após o período limite, para que os veículos fossem melhor utilizados.

[Lima et al. \(2013\)](#) abordaram um problema integrado no qual produtos coletados nos clientes podem ser usados como matéria-prima para a produção. O objetivo do problema é minimizar os custos relativos à aquisição de matéria-prima nova ou usada, os custos de estoques de produtos novos e usados, o custo de atraso da embalagem e os custos de transporte. Foram propostas duas formulações sendo que a primeira resolve o problema integrado em todo o horizonte de planejamento. A segunda formulação resolve o problema de Roteamento apenas para o primeiro período, utilizando aproximações nos outros. O modelo integrado foi executado com um limite de tempo de 7200 segundos e obteve soluções com gaps da ordem de 1% para instâncias com até 20 clientes e 4 períodos de planejamento.

[Molina et al. \(2013\)](#) apresentaram variações do problema introduzido em ([NORDEN; VELDE, 2005](#)). A primeira considera que os produtos têm dimensões diferentes e a segunda que os caminhões utilizados têm capacidades e custos diferentes. Foi possível resolver na otimalidade a maioria das instâncias com 20 produtos e 12 períodos com o limite de tempo

de 180 segundos. A formulação resolvida pelo *CPLEX* não conseguiu obter soluções ótimas para a maioria das instâncias com 50 e 150 produtos e 20 e 50 períodos.

[Schmid et al. \(2013\)](#) realizaram uma revisão bibliográfica sobre a integração do problema de Roteamento de Veículos com Janela de Tempo com outros problemas como o de Dimensionamento de Lotes, o de Planejamento de Produção, o de Agendamento de Máquinas, entre outros. Neste trabalho foram propostas formulações para cada problema integrado e discutidos alguns métodos de solução.

[Coelho et al. \(2014\)](#) realizaram uma revisão da literatura sobre trabalhos relacionados com o *Inventory Routing Problem - IRP* em que se considera o gerenciamento dos estoques e a distribuição dos produtos. Como extensões do problema, os autores apresentaram o *Problema de Roteamento da Produção - PRP* e o *IRP* com empacotamento direta.

[Coelho e Laporte \(2014\)](#) apresentaram o problema *Optimal Joint Replenishment, Delivery and Inventory Management Policies for Perishable Products*. Neste problema, o valor dos produtos se altera de acordo com o seu tempo de validade. Vender os produtos mais frescos pode, inicialmente, significar um maior lucro, mas futuramente, implicar em descarte pela expiração da data de validade. Foi apresentada uma formulação e um algoritmo *Branch-and-Cut* que permitiu que instâncias com 3 períodos e 30 clientes fossem resolvidas na otimalidade. Os autores apresentaram com detalhes os parâmetros para a geração das instâncias utilizadas neste artigo.

[Adulyasak et al. \(2015a\)](#) utilizaram o método de Decomposição de *Benders* para resolver um problema estocástico de Dimensionamento de Lotes integrado a um problema de Roteamento, com incertezas nas demandas. As decisões de roteamento e de quando produzir foram resolvidas pelo problema mestre de *Benders*, enquanto o subproblema foi usado para determinar as quantidades produzidas e entregues a partir do conhecimento da distribuição de probabilidades das demandas. Os resultados obtidos neste trabalho foram superiores aos resultados conhecidos na literatura que utilizavam um algoritmo de *Branch-and-Cut*.

[Adulyasak et al. \(2015b\)](#) apresentaram revisões de trabalhos relacionados com

o *Production Routing Problem - PRP* e com o IRP. Foram discutidas técnicas para a obtenção de limites inferiores, a solução de uma Relaxação Lagrangiana, solução por geração de colunas, um algoritmo exato utilizando *Branch-and-cut* e algumas heurísticas e metaheurísticas.

Em (CORDEAU et al., 2015), o IRP foi decomposto em dois subproblemas. O problema de Gerenciamento de Estoques foi resolvido através de uma formulação com os custos de transporte dados pela distância entre a fábrica e cada cliente. Em seguida, encontra-se um conjunto de rotas para cada período e para cada cliente, resolvendo-se um problema de Roteamento de Veículos. Uma terceira fase resolve novamente o subproblema inicial com os custos de transporte fornecidos pelo segundo subproblema.

Mirzaei e Seifi (2015) resolveram o IRP considerando que os produtos são perecíveis. Neste trabalho, consideram-se os custos de transporte, custos de estoque e custos gerados pelos produtos que atingem o prazo de validade. Foram propostas formulações e metaheurísticas híbridas baseadas em Simulated Annealing e Busca Tabu. As formulações resolveram apenas as menores instâncias do problema e forneceram limites inferiores para as maiores.

Molina et al. (2016) estendem o trabalho de Norden e Velde (2005) e propõe formulações para problemas de Dimensionamento de Lotes com custos de distribuição usando elementos de carga unitária, como paletes e contêineres. As decisões de dimensionamento de lotes estão sujeitas a restrições de capacidade e os itens devem ser embalados ao mesmo tempo em que são produzidos. Além disso, o custo de entrega de um caminhão é o mesmo em todos os períodos. Neste problema temos que: (i) Os produtos são reunidos em pedidos de clientes e são entregues juntos; (ii) O custo de entrega varia entre os períodos; (iii) A entrega pode ocorrer no período em que um produto é comprado ou produzido. Isso faz com que a integração entre as decisões de dimensionamento e embalagem seja desafiadora.

Uma formulação por PLI e um Algoritmo Genético - AG foram utilizados em (AZADEH et al., 2017) para resolver o IRP com produtos perecíveis e com transbordo. Neste problema, utilizou-se um veículo para atender as demandas dos clientes e admitia-se a possibilidade de contratação de empresas para realizar o transporte entre o fornecedor

e os clientes ou entre os clientes. Os autores destacaram os benefícios de um algoritmo baseado no método de Taguchi (TSAI et al., 2008) para determinação dos parâmetros para o AG.

2.1 Fundamentação Teórica

Nesta tese, para a solução do problema de Dimensionamento de Lotes utilizou-se um algoritmo de programação dinâmica proposto por (WAGNER; WHITIN, 1958). Neste algoritmo, seja a função objetivo que considera os custos de ativação de produção e de estoque de um único produto:

$$\sum_{t=1}^T c_t v_t + \sum_{t=1}^T e_t h_t \quad (2.1)$$

Onde: T é o total de períodos, c_t o custo de produção, e_t o custo de estoque, v_t a variável binária de ativação da produção e h_t o variável de estoque.

Escrevendo a variável de estoque como:

$$h_t = \sum_{i=1}^t x_i - \sum_{i=1}^t d_i \quad (2.2)$$

Com d_i representando as demandas pelos produtos em cada período, a função objetivo torna-se:

$$\sum_{t=1}^T c_t v_t + \sum_{t=1}^T e_t \left(\sum_{i=1}^t x_i - \sum_{i=1}^t d_i \right) \quad (2.3)$$

A segunda parcela do somatório pode ser reescrita, colocando a variável de produção x_t com o somatório de $1..T$, pois:

$$\sum_{t=1}^T e_t \sum_{i=1}^t x_i = e_1 * x_1 + e_2 * (x_1 + x_2) + \dots + e_T * (x_1 + x_2 + \dots + x_T) = \sum_{t=1}^T x_t \sum_{i=t}^T e_i \quad (2.4)$$

E a função objetivo pode ser escrita em função da variável de produção, com sua última parcela dada por uma constante que pode ser calculada ao final do problema:

$$\sum_{t=1}^T c_t v_t + \sum_{t=1}^T x_t \sum_{i=t}^T e_i - \sum_{t=1}^T e_t \sum_{i=1}^t d_i \quad (2.5)$$

Para um dado período k onde ocorra produção e desconsiderando a parcela constante em (2.5), temos que o custo será dado por:

$$c_k + x_k \sum_{i=k}^T e_i \quad (2.6)$$

Ou,

$$c_k + \sum_{i=k}^t d_i \sum_{i=k}^T e_i \quad (2.7)$$

Seja $H(t)$ o custo ótimo para o problema *Dimensionamento de Lotes Não Capacitado e com Único Produto* no instante t e sendo $k \leq t$ o último período em que houve produção, podemos escrever:

$$H(t) = H(k - 1) + c_k + \sum_{i=k}^T e_i \sum_{i=k}^t d_i \quad (2.8)$$

Assim, a função de recursão do *algoritmo de Wagner-Within* é dada por:

$$H(t) = \min_{1 \leq k \leq t} (H(k - 1) + c_k + \sum_{i=k}^T e_i \sum_{i=k}^t d_i) \quad (2.9)$$

Com $H(0) = 0$ e retirando-se a última parcela da expressão 2.5 na solução final.

Este algoritmo de programação dinâmica foi utilizado em nosso trabalho para resolver o problema de Dimensionamento de Lotes na metodologia de solução de duas fases.

Esta versão do problema de Dimensionamento de Lotes pode ser vista como um problema de fluxo de redes no qual pretende-se encontrar os arcos $(0, t)$ onde ocorreram produção, obtendo-se um fluxo de custo mínimo através da rede da Figura 2.

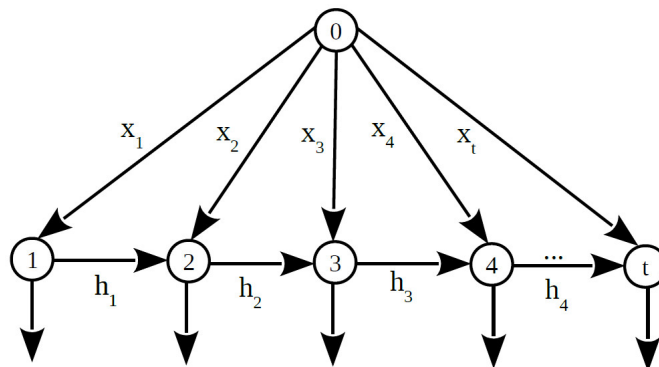


Figura 2 – Representação do problema de Dimensionamento de Lotes não Capacitado como fluxo em redes

Para o problema de Empacotamento, o principal método utilizado para solução foi o algoritmo VPsolver. O VPsolver foi proposto em (BRANDAO; PEDROSO, 2016) e deriva do trabalho de (CARVALHO, 1999) que apresentou uma formulação baseada na utilizada no Problema de Fluxo Mínimo para resolver o problema de Cortes Unidimensional e que pode ser estendida para o problema de Empacotamento. Nesta formulação, o problema de empacotamento é equivalente a encontrar um fluxo mínimo em um grafo entre os vértices inicial 0 e final W , sendo W a capacidade dos bins.

Brandao e Pedroso (2016) utilizaram novas técnicas de compactação possibilitando uma redução significativa no número de vértices e arcos do grafo, permitindo uma representação eficiente de cada instância do problema. Após a redução do tamanho, o problema de encontrar um fluxo mínimo no grafo compactado pode ser resolvido por pacotes de otimização. A saída da solução do problema de fluxo mínimo é transformada por um algoritmo em uma solução ótima para a instância do problema de empacotamento original.

Para a construção do grafo inicial que representará o problema de empacotamento, deve-se ordenar os itens que serão empacotados em ordem decrescente de acordo com o tamanho e inicia-se o grafo pela inserção do vértice de origem S . Em seguida é selecionado o primeiro vértice da lista, item com maior peso, e cria-se um arco entre a origem e este vértice. Se a demanda pelo item for maior que um e existir capacidade no caminho do grafo, arcos e vértices são adicionados. Existir capacidade no caminho implica que a soma do rótulo do último vértice do caminho mais o tamanho do item atual é menor ou igual do que a capacidade do bin e respectivamente, do rótulo do vértice final do caminho. Cada arco do grafo é rotulado com o índice do item o qual o fluxo irá atender as demandas.

Seja o exemplo da Figura 3 onde temos um grafo que representa um problema com bins com capacidade $W = 9$, itens com pesos $w_i = \{5; 3; 2\}$ e demandas pelos itens de $d_i = \{2; 3; 2\}$. O primeiro vértice a ser inserido no grafo é o vértice $w_1 = 5$ que é ligado a origem e atende ao item $i = 1$. Como não há capacidade disponível no caminho para atender outra unidade desta demanda ($5 + 5 > 9$), não serão inseridos outros arcos para este item no caminho.

Seleciona-se o próximo item da lista e um arco entre o vértice origem e o vértice

selecionado é inserido. Verifica-se a possibilidade de inserir arcos entre os vértices terminais do grafo e o vértice relacionado ao último item retirado da lista, caso exista capacidade disponível.

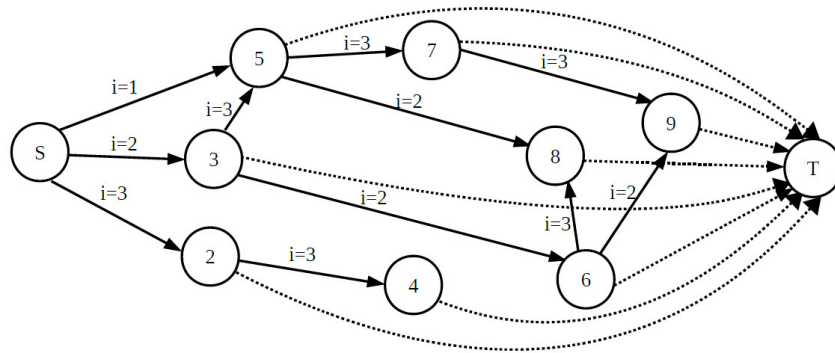


Figura 3 – Grafo Passo 1 que representa um problema de empacotamento

Outras operações serão aplicadas sobre o grafo resultante do procedimento descrito com objetivo de reduzir o número de vértices e arcos. Por este motivo, nos referimos ao grafo inicial com um grafo de Passo 1. Após a inserção do último item da lista, arcos chamados de *arcos de sobra* são inseridos entre os vértices do grafo com exceção da origem, ligando-os ao vértice destino T . Estes arcos são referentes as capacidades não utilizadas e podem ser vistos na figura representados por arcos com linhas tracejadas. Os arcos de sobra tem como convenção pesos de item iguais a zero. Sem considerar os arcos de sobra, no grafo do exemplo, temos que os conjuntos dos vértices e arcos têm cardinalidade 9 e 11, respectivamente.

Como pode ser observado na Figura 3, a escolha da inserção dos vértices considerando um lista ordenada por pesos não elimina a possibilidade soluções simétricas com relação a utilização das bins. Para reduzir a simetria, um novo grafo foi criado dividindo o grafo Passo 1 em n níveis verticais, um nível para cada tamanho de item. Este novo grafo, identificado como grafo de Passo 2, está representado pela Figura 4. No grafo de Passo 2 existem arcos no sentido horizontal ligando os vértices de mesmo nível, de acordo com o grafo de Passo 1, e arcos verticais ligando os vértices entre os níveis.

No grafo Passo 2 os rótulos são formados pelo rótulo do vértice do grafo Passo 1 mais o nível do vértice no grafo atual. Pela figura verificamos que o vértice origem é o $(0, 1)$ o qual não tem nenhum arco incidente. Com a utilização do grafo Passo 2, há a

redução de soluções simétricas ao custo do aumento no número de vértices e arcos.

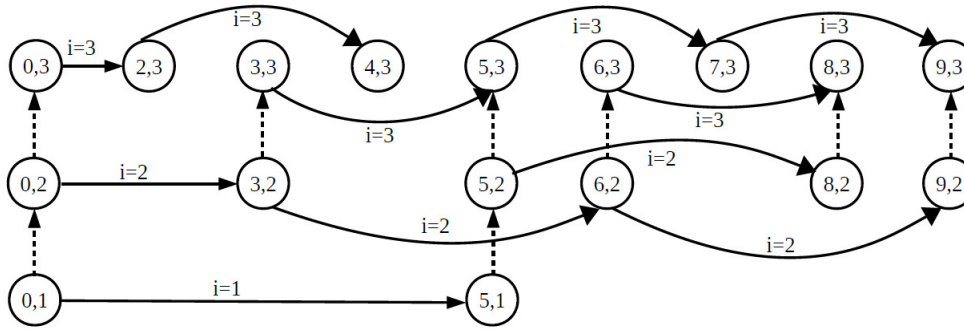


Figura 4 – Grafo Passo 2, construído a partir do grafo Passo 1

A principal técnica para redução do número de vértices e arcos é aplicada sobre o grafo Passo 2, gerando o grafo o qual chamamos de Passo 3. No grafo Passo 2, o rótulo de um vértice indica a maior distância entre este vértice e a origem. Para a construção do grafo Passo 3, utilizamos como rótulo a maior distância entre o vértice em análise e o vértice destino T . O algoritmo que gera o grafo Passo 3 se inicia com a construção de uma lista de adjacências do grafo Passo 2. Para cada vértice u do grafo Passo 2, criamos uma lista com arcos que conectam u a outros vértices v para cada item i , inclusive os vértices ligados por arcos sobra com $i = 0$.

Após a construção da lista de adjacências, um algoritmo de programação dinâmica avalia os vértices partindo dos que tem maior valor para os que tem menor valor, ou seja, do vértices do final da tabela para o início. Os rótulos $\varphi(u)$ dos vértices do grafo Passo 3 serão obtidos a partir da lista de adjacência observando a expressão (2.10):

$$\varphi(u) = \begin{cases} 0 & \text{if } u = S \\ T & \text{if } u = T \\ \min_{(u,v,i) \in A} (\varphi(v) - x_i) & \text{caso contrário} \end{cases} \quad (2.10)$$

Definindo que $\varphi(T) = T$, ou seja, a capacidade do bin e que os pesos dos arcos sobra são iguais a zero, $x_0 = 0$, encontramos um novo rótulo para os vértices a partir da expressão.

O algoritmo inicia a análise dos novos rótulos a partir no vértices de maior valor. O grafo Passo 3 obtido após a alteração dos rótulos segundo a lista de adjacência está

descrito na Figura 5. Verifica-se que para a instância de exemplo que utilizamos nesta seção, que tem itens com três pesos diferentes e que gerou um grafo inicial com 9 vértices e 11 arcos, chegamos a um grafo Passo 3 com 6 vértices e 9 arcos sem que nenhum padrão válido de empacotamento tenha sido descartado.

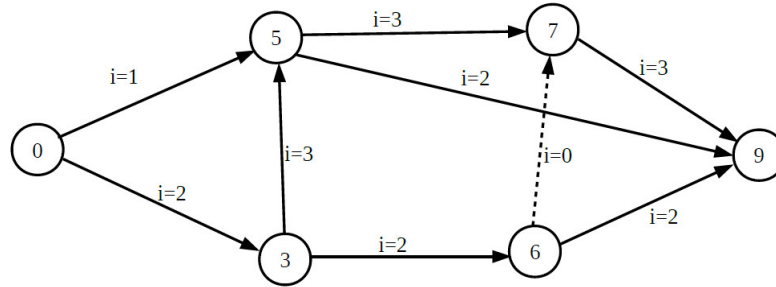


Figura 5 – Grafo Passo 3, construído após a alteração dos rótulos

Um último procedimento, que irá gerar um grafo Passo 4, será aplicado alterando os rótulos dos vértices do grafo Passo 3. Este procedimento é baseado na expressão (2.11) que calcula a distância máxima até a origem para cada vértice.

$$\varphi(u) = \begin{cases} 0 & \text{if } u = S \\ X & \text{if } u = T \\ \min_{(u,v,i) \in A} (\varphi(v) - x_i) & \text{caso contrário} \end{cases} \quad (2.11)$$

No grafo, a escolha dos caminhos podem levar a soluções onde as demandas sejam atendidas com excesso, sem implicar em aumento do custo da solução. É simples criar um procedimento que desconsidera as demandas em excesso.

Na prática, a geração do grafo Passo 3 ocorre segundo os conceitos descritos mas sem a necessidade efetiva de gerar o grafo Passo 2, que pode ter dimensões consideráveis. Todos os algoritmos para os procedimentos descritos nesta seção estão presentes em (??). Segundo os autores, os procedimentos permitiram a redução do tamanho das instâncias, para o grupo utilizado no trabalho, em até 97%.

Este método de compactação pode apresentar algumas dificuldades em grafos de instâncias em que a capacidade dos bins e o número de tipos de itens aumentam, ocasionando um grande número de padrões de empacotamento.

Dentre as heurísticas utilizadas nesta tese e que resolve o problema integrado em um procedimento iterativo, tem-se a heurística MS-LSBP. Esta heurística se baseia no *Noising Method*, apresentado nos trabalhos de (CHARON; HUDRY, 1993; CHARON; HUDRY, 2000). Neste método realiza-se uma perturbação nos valores dos custos do problema, promovendo uma diversificação no espaço de busca de soluções e, em seguida, é aplicado algum método de melhora com busca local, gerando uma intensificação. Os valores da perturbação são gradativamente reduzidos até atingir uma perturbação nula. Em (CHARON; HUDRY, 1993) o método foi utilizado para resolver um problema de Partições de Cliques em grafos em que a perturbação era aplicada aos valores dos custos dos arcos. A mesma ideia de perturbação dos custos é proposta como uma alternativa ao processo de construção de soluções na aplicação da metaheurística GRASP em (GLOVER; KOCHENBERGER, 2003).

Em (CHARON; HUDRY, 2000) o *Noising Method* foi utilizado para resolver o problema do caixeiro viajante. Nesse trabalho, após a construção de uma solução inicial, eram aplicadas perturbações que alteravam os benefícios em cada troca, ao se executar uma busca local com vizinhança *2-opt*. Em (MARTÍ et al., 2013) realizou-se uma revisão de métodos multipartida (do inglês *Multi-start methods*), na qual os autores dividiram os métodos em os que mantêm alguma memória sobre as soluções encontradas e os que não mantêm. Nesse trabalho, o *Noising Method* é destacado como um procedimento que não utiliza esta memória, juntamente com os métodos *simulated annealing* e *GRASP*.

Outro método utilizado para resolver o problema integrado nesta tese é um Algoritmo Genético com Chaves Aleatórias Bayesianas (do inglês *Biased Random Key Genetic Algorithm - BRKGA*) (GONÇALVES; RESENDE, 2011). O BRKGA utiliza uma população de cromossomos que consistem em vetores de números reais com valores no intervalo $[0, 1]$. Ao longo de várias iterações conhecidas também como gerações, segundo diferentes estratégias, novos cromossomos são gerados usando informações da geração anterior. Define-se como *chave* cada elemento do vetor que é gerado aleatoriamente na geração inicial. A qualidade de um cromossomo é definida pelo custo da solução fornecida por um método chamado *decoder*, que recebe como um de seus dados de entrada, o vetor

de chaves do cromossomo e devolve uma solução viável para o problema.

Uma das formas de gerar novos cromossomos ao longo da execução do algoritmo genético é utilizar operações conhecidas como recombinação ou cruzamento. O cruzamento utilizado no BRKGA é o Cruzamento Uniforme Parametrizado (do inglês *Parameterized Uniform Crossover*), proposto em (SPEARS; DEJONG, 1991). Sejam dois cromossomos c_1 e c_2 , um novo cromossomo c_{new} é gerado da seguinte forma: assumindo-se que o cromossomo c_1 é mais apto que o cromossomo c_2 , o novo cromossomo c_{new} recebe uma chave aleatoriamente da chave correspondente em c_1 com probabilidade p_{rec} e da chave correspondente em c_2 com probabilidade $1 - p_{rec}$, sendo p_{rec} um parâmetro. Ou seja, para cada uma das posições do vetor, gera-se um número aleatório no intervalo $[0, 1]$. Caso o valor gerado seja menor ou igual a p_{rec} , a chave é herdada do cromossomo c_1 ; caso contrário, é herdada do cromossomo c_2 .

Outra operação utilizada para gerar elementos de uma nova população é conhecida como mutação. A mutação no BRKGA ocorre da seguinte forma. A cada geração, uma parte dos cromossomos cujo *decoder* retornou os piores valores de solução são excluídos. Novos cromossomos com valores de *chaves* geradas aleatoriamente são criados para substituí-los.

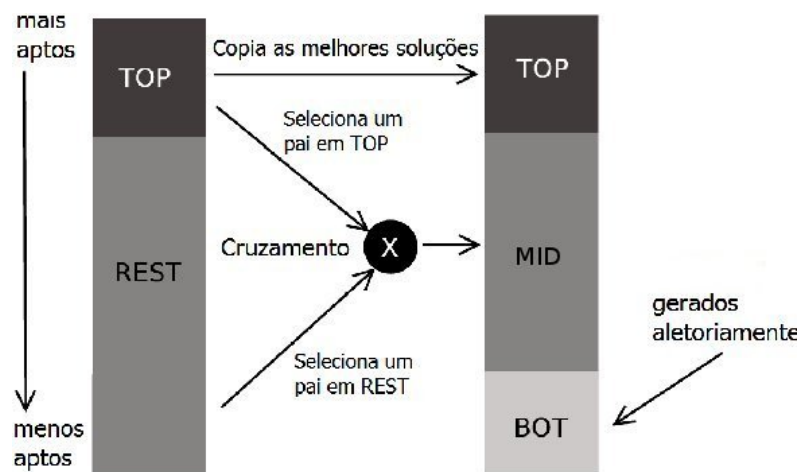


Figura 6 – Evolução entre duas gerações sucessivas do BRKGA-LSBP

Na Figura 6 tem-se uma representação da evolução entre duas gerações do BRKGA. Em uma nova geração, a população é dividida em três partes: *TOP*, *MID* e *BOT*. Os cromossomos que fornecem as melhores soluções são mantidos no conjunto *TOP* e o conjunto *MID* contém cromossomos com soluções de qualidade intermediária. O tamanho

dos conjuntos *TOP*, *MID* e *BOT* são parâmetros do algoritmo. Para gerar uma nova população, copia-se os cromossomos do conjunto *TOP* da geração corrente para a próxima. Os cromossomos do conjunto *BOT* são substituídos por novos cromossomos gerados aleatoriamente e os cromossomos no conjunto *MID* são gerados pelo cruzamento de dois cromossomos, sendo que o primeiro é escolhido aleatoriamente do conjunto *TOP* e o segundo entre todos os elementos da população, possivelmente sendo parte do conjunto *TOP* também.

3 Formulações e Limites Inferiores

Neste capítulo são apresentadas formulações por PLI e relaxações para o IULSBP bem como os experimentos relacionados. Propõe-se na Seção 3.1 uma formulação por Programação Linear Inteira (PLI) chamada de formulação de Fluxo Único (SF , do inglês *Single Flow*). Em seguida, Seção 3.2, é apresentada uma reformulação baseada na formulação de Múltiplos Fluxos (MF , do inglês *Multicommodity Flow*) para o problema de Dimensionamento de Lotes (POCHET; WOLSEY, 2006). Na Seção 3.3 tem-se uma Relaxação Combinatória (CR , do inglês *Combinatorial Relaxation*) da formulação SF com o objetivo de obter bons limites inferiores em um período de tempo reduzido. Na Seção 3.4 são apresentadas as instâncias utilizadas para a realização dos experimentos. Em seguida, têm-se os resultados das formulações e das relaxações resolvidas pelo CPLEX para o problema.

3.1 Formulação de Fluxo Único SF

Seja o conjunto dos períodos $\Pi_i = \{1, \dots, t_i\}$ nos quais o pedido $i \in V$ pode ser atendido. Uma formulação para o IULSBP é obtida através das variáveis (i) v_{pt} , tal que $v_{pt} = 1$ se for colocada uma ordem de compra para o produto $p \in P$ no período $t \in T$ e $v_{pt} = 0$ caso contrário, e variáveis (ii) y_{ik}^t , tal que $y_{ik}^t = 1$ se o pedido $i \in V$ for empacotado em $k \in K$ e entregue no período $t \in \Pi_i$, e $y_{ik}^t = 0$ caso contrário. As seguintes variáveis auxiliares também são necessárias. Variáveis $x_{pt} \in \mathbb{R}$ que representam quantas unidades do produto $p \in P$ são compradas em $t \in T$, e as variáveis h_{pt} que representa quantas unidades do produto $p \in P$ são deixadas no estoque ao final do período $t \in T$. Além disso, utilizam-se variáveis z_{kt} , tal que $z_{kt} = 1$ se o bin $k \in K$ for utilizado no período $t \in T$ e $z_{kt} = 0$ caso contrário.

$$\min \sum_{p \in P} \sum_{t \in T} c_{pt} v_{pt} + \sum_{p \in P} \sum_{t \in T} e_{pt} h_{pt} + \sum_{k \in K} \sum_{t \in T} f_t z_{kt} \quad (3.1)$$

$$x_{pt} \leq M_{pt} v_{pt} \quad \forall p \in P, \forall t \in T \quad (3.2)$$

$$h_{pt-1} + x_{pt} - h_{pt} = \sum_{i \in V: t=t_i} q_{ip} \quad \forall p \in P, \forall t \in T \quad (3.3)$$

$$\sum_{u \in \{1, \dots, t\}} x_{pu} \geq \sum_{u \in \{1, \dots, t\}} \sum_{i \in V: u \in \Pi_i} q_{ip} \sum_{k \in K} y_{ik}^u \quad \forall p \in P, \forall t \in T \quad (3.4)$$

$$z_{kt} \geq y_{ik}^t \quad \forall k \in K, \forall i \in V, \forall t \in \Pi_i \quad (3.5)$$

$$\sum_{k \in K} \sum_{t \in \Pi_i} y_{ik}^t = 1 \quad \forall i \in V \quad (3.6)$$

$$\sum_{i \in V: t \in \Pi_i} \sum_{p \in P} q_{ip} y_{ik}^t \leq Q z_{kt} \quad \forall k \in K, \forall t \in T \quad (3.7)$$

$$\sum_{i \in V: t \in \Pi_i} y_{ik}^t \leq \sum_{i \in V: t \in \Pi_i} y_{ik-1}^t \quad \forall k \in K, \forall t \in T \quad (3.8)$$

$$0 \leq z_{kt} \leq 1 \quad \forall k \in K, \forall t \in T \quad (3.9)$$

$$y_{ik}^t \in \{0, 1\} \quad \forall k \in K, \forall i \in V, \forall t \in \Pi_i \quad (3.10)$$

$$v_{pt} \in \{0, 1\} \quad \forall p \in P, \forall t \in T \quad (3.11)$$

$$x_{pt} \geq 0 \quad \forall p \in P, \forall t \in T \quad (3.12)$$

$$h_{pt} \geq 0 \quad \forall p \in P, \forall t \in T \quad (3.13)$$

A formulação resultante, chamada formulação de Fluxo Único SF , é definida por (3.1)-(3.13). Dada uma instância do IULSBP caracterizada pela tupla $\langle V, P, T, K, Q, q_{ip}, t_i, c_{pt}, e_{pt}, f_t \rangle$, a função objetivo (3.1) minimiza a soma dos custos de compra, custos de manutenção de estoque e custos de empacotamento, respectivamente. As restrições em (3.2) impõem que o custo c_{pt} é ativado se ocorrer uma compra, sendo que a constante $M_{pt} = \sum_{i \in V: t_i \geq t} q_{ip}$ é um limite superior para os valores de x_{pt} . As restrições de saldo do estoque são impostas por (3.3). As restrições em (3.4) impõem que as unidades de produto $p \in P$ compradas até o período $t \in T$ sejam suficientes para atender todas as demandas até t . As restrições em (3.5) exigem que um custo f_t seja cobrado se um bin $k \in K$ for usado para empacotar um pedido $i \in V$ em um período $t \in T$. A restrição que um pedido $i \in V$ deve ser empacotado em um único bin é imposto por (3.6), enquanto as restrições de capacidade dos bins são impostas por (3.7). As restrições de redução de simetria (3.8) são adicionadas para reduzir o número de soluções equivalentes no espaço de busca. Finalmente, as restrições em (3.9)-(3.13) definem o domínio das variáveis.

3.2 Formulação de Múltiplos Fluxos MF

Nesta seção propõe-se uma reformulação de SF com o objetivo de melhorar sua relaxação linear. Primeiramente, suprimiram-se as variáveis de estoque h_{pt} , para todos os $p \in P$ e $t \in T$, por meio das seguintes manipulações algébricas.

Reescrevem-se as restrições em (3.3) para obter h_{pt} que é dado pela diferença de tudo que foi produzido do produto p até o período t e a demanda deste mesmo produto até o período t .

$$h_{pt} = \sum_{u \in \{1, \dots, t\}} x_{pu} - \sum_{u \in \{1, \dots, t\}} \sum_{i \in V : u=t_i} q_{ip}.$$

Em seguida, substituem-se as variáveis h_{pt} no segundo componente de (3.1), ou seja,

$$\begin{aligned} \sum_{p \in P} \sum_{t \in T} e_{pt} h_{pt} &= \sum_{p \in P} \sum_{t \in T} e_{pt} \left(\sum_{u \in \{1, \dots, t\}} x_{pu} - \sum_{u \in \{1, \dots, t\}} \sum_{i \in V : u=t_i} q_{ip} \right) = \\ &= \sum_{p \in P} \sum_{t \in T} e_{pt} \sum_{u \in \{1, \dots, t\}} x_{pu} - \sum_{p \in P} \sum_{t \in T} e_{pt} \sum_{u \in \{1, \dots, t\}} \sum_{i \in V : u=t_i} q_{ip}. \end{aligned}$$

Têm-se que

$$\begin{aligned} \sum_{p \in P} \sum_{t \in T} e_{pt} \sum_{u \in \{1, \dots, t\}} x_{pu} &= e_{p1} x_{p1} + e_{p2} (x_{p1} + x_{p2}) + e_{p3} (x_{p1} + x_{p2} + x_{p3}) + \dots \\ &+ e_{pT} (x_{p1} + x_{p2} + x_{p3} + \dots + x_{pT}) = x_{p1} (e_{p1} + e_{p2} + \dots + e_{pT}) + x_{p2} (e_{p2} + \dots + e_{pT}) + \dots + x_{pT} e_{pT}. \\ &= \sum_{p \in P} \sum_{t \in T} x_{pt} \sum_{u \in \{t, \dots, T\}} e_{pu}. \end{aligned}$$

Assim, segue-se que

$$\sum_{p \in P} \sum_{t \in T} e_{pt} h_{pt} = \sum_{p \in P} \sum_{t \in T} x_{pt} \sum_{u \in \{t, \dots, T\}} e_{pu} - \sum_{p \in P} \sum_{t \in T} e_{pt} \sum_{u \in \{1, \dots, t\}} \sum_{i \in V : u=t_i} q_{ip}$$

Então, a função objetivo (3.1) pode ser reescrita como (3.14), com $\bar{e}_{pt} = \sum_{u \in \{t, \dots, T\}} e_{pu}$.

$$\sum_{p \in P} \sum_{t \in T} c_{pt} v_{pt} + \sum_{p \in P} \sum_{t \in T} \bar{e}_{pt} x_{pt} + \sum_{k \in K} \sum_{t \in T} f_t z_{kt} \quad (3.14)$$

Como a componente

$$- \sum_{p \in P} \sum_{t \in T} e_{pt} \sum_{u \in \{1, \dots, t\}} \sum_{i \in V : u=t_i} q_{ip}$$

é constante, ela é omitida de (3.14). Como as restrições (3.4) e (3.6) garantem que as demandas de todos os pedidos em V são atendidas, as variáveis h_{pt} e as restrições (3.3) podem ser removidas da formulação.

Além disso, substituem-se em MF as variáveis x_{pt} (presentes em FU) pelas variáveis $x_{\tau t}^{ip}$ usadas na formulação de múltiplos fluxos para o problema de Dimensionamento de Lotes (POCHET; WOLSEY, 2006), tal que $x_{\tau t}^{ip} \in \mathbb{R}$ é igual ao número de unidades de produto $p \in P$ compradas no período $\tau \in \{1, \dots, t\}$ a ser entregue no período $t \in \{1, \dots, t_i\}$ para atender a demanda do pedido $i \in V$.

$$\min \sum_{p \in P} \sum_{t \in T} c_{pt} v_{pt} + \sum_{p \in P} \sum_{\tau \in T} \bar{e}_{p\tau} \sum_{i \in V : \tau \in \Pi_i} \sum_{t \in \{\tau, \dots, t_i\}} x_{\tau t}^{ip} + \sum_{k \in K} \sum_{t \in T} f_t z_{kt} \quad (3.15)$$

$$x_{\tau t}^{ip} \leq q_{ip} v_{p\tau} \quad \forall p \in P, \forall i \in V, \forall t \in \Pi_i, \tau \in \{1, \dots, t\} \quad (3.16)$$

$$\sum_{\tau \in \{1, \dots, t\}} x_{\tau t}^{ip} = q_{ip} \sum_{k \in K} y_{ik}^t \quad \forall i \in V, \forall p \in P, \forall t \in \Pi_i \quad (3.17)$$

$$x_{\tau t}^{ip} \geq 0 \quad \forall p \in P, \forall i \in V, t \in \Pi_i, \forall \tau \in \{1, \dots, t\} \quad (3.18)$$

A formulação resultante, chamada de formulação de Múltiplos Fluxos (MF), é definida por (3.15)-(3.18) e as restrições (3.5)-(3.11) da formulação anterior. Dada uma instância do IULSBP caracterizada pela tupla $\langle V, P, T, K, Q, q_{ip}, t_i, c_{pt}, e_{pt}, f_t \rangle$, a função objetivo (3.1) minimiza a soma dos custos de compra, custos de manutenção de estoque e custos de empacotamento, respectivamente. As restrições em (3.16) impõem que o custo c_{pt} é ativado se ocorrer a compra de unidades de produto $p \in P$ no período $\tau \in \{1, \dots, t\}$ a ser entregue no período $t \in \Pi_i$ para atender à demanda do pedido $i \in V$. As restrições em (3.17) asseguram que, caso o pedido $i \in V$ seja atendido no período $t \in \Pi_i$, o número de unidades do produto $p \in P$, dos períodos de 1 a t , para atender a este pedido, é igual a sua demanda. Finalmente, as restrições em (3.18) definem o domínio das variáveis $x_{\tau t}^{ip}$. Têm-se o mesmo número de variáveis binárias nas formulações SF e MF .

3.3 Relaxação Combinatória CR

Nesta seção é apresentada uma Relaxação Combinatória (CR, do inglês *Combinatorial Relaxation*) da formulação SF com o objetivo de obter bons limites inferiores em um período de tempo reduzido. Relaxa-se a restrição de que todos os produtos de um pedido devem ser empacotados em um mesmo bin. Neste caso, não é mais necessário determinar qual o bin em que cada produto é empacotado, mas apenas definir quantos bins são necessários em cada período. A formulação resultante é obtida removendo as variáveis z_{kt} e substituindo as variáveis binárias y_{ik}^t , para todos os $k \in K$, $i \in V$ e $t \in \Pi_i$, pelas variáveis y_{it} , tal que $y_{it} = 1$ se o pedido $i \in V$ for empacotado no período $t \in \Pi_i$ e $y_{it} = 0$ caso contrário. Além disso, adicionam-se as variáveis inteiras $w_t \in \mathbb{N}$, representando quantos bins com capacidade Q são necessários em cada período $t \in T$.

$$\min \sum_{p \in P} \sum_{t \in T} c_{pt} v_{pt} + \sum_{p \in P} \sum_{t \in T} e_{pt} h_{pt} + \sum_{t \in T} f_t w_t \quad (3.19)$$

$$\sum_{u \in \{1, \dots, t\}} x_{pu} \geq \sum_{u \in \{1, \dots, t\}} \sum_{i \in V: u \in \Pi_i} q_{ip} y_{ui} \quad \forall p \in P, \forall t \in T \quad (3.20)$$

$$\sum_{t \in \Pi_i} y_{it} = 1 \quad \forall i \in V \quad (3.21)$$

$$\frac{1}{Q} \sum_{i \in V} \sum_{t \in \Pi_i} \sum_{p \in P} q_{ip} y_{it} \leq w_t \quad \forall t \in T \quad (3.22)$$

$$y_{it} \in \{0, 1\} \quad \forall i \in V, \forall t \in \Pi_i \quad (3.23)$$

$$w_t \in \mathbb{N} \quad \forall t \in T \quad (3.24)$$

A formulação resultante é definida por (3.19)-(3.24), pelas restrições (3.2),(3.3) e pelas restrições (3.11)-(3.13) da formulação SF . Dada uma instância de IULSBP caracterizada por $\langle V, P, T, K, Q, q_{ip}, t_i, c_{pt}, e_{pt}, f_t \rangle$, a função objetivo (3.19) minimiza a soma dos custos de compra, custos de manutenção de estoque e custos aproximados para o empacotamento, respectivamente. As restrições em (3.20) impõem que as unidades de produto $p \in P$ produzidas até o período $t \in T$ sejam suficientes para atender todas as demandas até t . As restrições que impõem que um pedido $i \in V$ deve ser entregue em um

único período são (3.21), enquanto o valor correto de w_t é imposto por (3.22). Finalmente, as restrições em (3.23) e (3.24) definem o domínio das variáveis y_{it} e w_t , respectivamente.

Uma relaxação combinatória alternativa pode ser obtida relaxando a integralidade das variáveis w_t , ou seja, trocando (3.24) por (3.25). A formulação resultante, chamada $CR^{w_t \geq 0}$, é definida por (3.2), (3.3), (3.11)-(3.13), (3.19)-(3.23), (3.25). Essa relaxação é mais fácil de resolver do que a formulação CR , por ter um menor número de variáveis inteiras. Experimentos computacionais mostraram que seus limites inferiores são melhores do que os fornecidos pela relaxação linear das formulações SF e MF .

$$w_t \geq 0, \quad \forall t \in T \quad (3.25)$$

3.4 Experimentos Computacionais das Formulações e Relaxações

Nesta seção são apresentados resultados de experimentos realizados com as formulações SF e MF , bem como para as relaxações combinatórias CR e $CR^{w_t \geq 0}$. Estas formulações foram implementadas na linguagem OPL (Optimization Programming Language), que é suportada pelo IBM ILOG CPLEX Optimization Studio (versão 12.6) (ILOG, 2018). Os experimentos computacionais foram executados em uma máquina com processador E5405 Intel (R) Xeon (R) quad-core com 2,00 GHz de clock e 16 GB de memória RAM, rodando o sistema operacional Linux Ubuntu (versão 14.04).

Um novo conjunto de instâncias de teste é introduzido na Seção 3.4.1. Em seguida, na Seção 3.4.2, são apresentados (i) os resultados do algoritmo de branch and bound padrão do CPLEX para as formulações SF e MF , (ii) a qualidade dos limites inferiores obtidos pelas relaxações lineares de SF e MF , assim como aquelas obtidas pela relaxação combinatória de CR e $CR^{w_t \geq 0}$. Realizam-se também considerações sobre os resultados das formulações e relaxações.

3.4.1 Instâncias de Teste

Uma instância do IULSBP é caracterizada pela tupla $\langle V, P, T, K, Q, q_{ip}, t_i, c_{pt}, e_{pt}, f_t \rangle$. As instâncias de teste usadas nos experimentos computacionais deste trabalho são geradas

a partir de três parâmetros: $\eta = |V|$, $\rho = |P|$, $\mu = |T|$. A data limite de atendimento t_i de cada pedido $i \in V$ é obtida a partir de $U_i[1; |T|]$, em que $U_i[a; b]$ denota um número pseudo-aleatório inteiro gerado, com uma distribuição uniforme, no intervalo $[a; b]$ (MATSUMOTO; NISHIMURA, 1998). O valor de Q foi fixado em 10.000 e o cálculo dos valores q_{ip} , para cada $i \in V$ e $p \in P$, foi inspirado pelas instâncias conhecidas como *Triplets* para o problema de Empacotamento (FALKENAUER, 1996). Essas instâncias são geradas de forma que sua solução ótima tenha exatamente três itens por bin e nenhum espaço não utilizado. Seguindo a metodologia proposta por (FALKENAUER, 1996), primeiro constrói-se uma instância do problema Empacotamento, cuja solução ótima tenha $|K| = \eta/3$ bins, dividindo cada bin em três partes. O tamanho do primeiro item é definido como $U_r[0, 38 \cdot Q; 0, 49 \cdot Q]$, enquanto o tamanho do segundo é definido como $U_r[0, 25 \cdot Q; Q/2]$. $U_r[a; b]$ denota um número pseudo-aleatório real gerado com uma distribuição uniforme, no intervalo $[a; b]$. O espaço restante no bin corresponde ao tamanho do terceiro item. Em seguida, relaciona-se aleatoriamente cada item com o peso \bar{q}_i a um pedido $i \in V$. Tem-se que $\bar{q}_i = \sum_{p \in P} q_{ip}$, e calcula-se o valor exato de q_{ip} como segue. Primeiro, $q_{ip} = 0$ para todos os $i \in V$ e $p \in P$. Então, enquanto $\sum_{p \in P} q_{ip} \leq \bar{q}_i$, seleciona-se iterativamente uma demanda $q_{ip} = \min\{\bar{q}_i - \sum_{p \in P} q_{ip}, U_r[0.2 \cdot \bar{q}_i; 0.5 \cdot \bar{q}_i]\}$ para um produto p escolhido aleatoriamente em P . Essa abordagem permite cenários em que pedidos diferentes têm demandas para diferentes subconjuntos de itens.

Foram gerados seis grupos de instâncias que diferem entre si pela forma como os custos de compra c_{pt} , custos de estoque e_{pt} , e o custos de empacotamento f_t são calculados. No primeiro grupo de instâncias, define-se $c_{pt} = U_r[12000; 13000]$ e o custo de manutenção do estoque $e_{pt} = U_r[0, 4; 0, 6]$ para cada unidade do produto $p \in P$ no período $t \in T$. O custo de empacotamento para cada período $t \in T$ foi definido como $f_t = \theta/\log(|K|)$, com $\theta = U_r[11000; 14000]$. Esses valores de c_{pt} , e_{pt} e f_t foram escolhidos de modo que a contribuição na função objetivo da soma dos custos de compra e estoque sejam aproximadamente a mesma que a dos custos de empacotamento. Isso leva a instâncias mais difíceis, pois o problema Dimensionamento de Lotes e o problema de Empacotamento devem ambos ser bem resolvidos para encontrar soluções ótimas ou quase ótimas.

Os cinco grupos de instâncias restantes foram gerados como aqueles do primeiro grupo. No entanto, no segundo grupo, os valores de c_{pt} e f_t para todos os produtos $p \in P$ e todos os períodos $t \in T$ que sejam números pares foram incrementados em 30%. Além desse aumento de 30% em c_{pt} e f_t nos períodos pares, as instâncias do terceiro grupo tiveram o valor de c_{pt} e e_{pt} incrementado em mais 20% em todos os períodos. As instâncias do quarto grupo tiveram o valor de c_{pt} e e_{pt} aumentado em 20% em todos os períodos, enquanto que as do quinto grupo tiveram o valor do custo de empacotamento aumentado em 20% em todos os períodos. No sexto grupo, os valores de c_{pt} e f_t para todos os produtos $p \in P$ e todos os períodos $t \in T$ que sejam números pares, foram aumentados em 30% e o valor do custo de empacotamento foi aumentado por 20% em todos os períodos.

Cada um dos seis grupos de instâncias é dividido em seis conjuntos de dez instâncias geradas aleatoriamente com os seguintes valores de η , ρ e μ . O primeiro e segundo conjuntos têm o mesmo valor de $\rho = 6$ e $\mu = 4$, mas com $\eta = 36$ e $\eta = 48$, respectivamente, enquanto o terceiro e quarto conjuntos têm $\rho = 6$ e $\mu = 6$, com $\eta = 72$ e $\eta = 90$, respectivamente. Além disso, o quinto e sexto conjuntos têm o mesmo valor de $\rho = 8$ e $\mu = 8$, com $\eta = 120$ e $\eta = 144$, respectivamente. Cada conjunto de instâncias é identificado pelo número do grupo e pelos valores de η , ρ e μ . Por exemplo, o nome g1-36-6-4 refere-se a um conjunto de instâncias do primeiro grupo que foi gerado com $\eta = 36$, $\rho = 6$ e $\mu = 4$. As 360 instâncias resultantes estão disponíveis para download em (GOULART et al., 2018)

3.4.2 Resultados

No primeiro experimento, avaliou-se o desempenho de duas variantes do algoritmo de *branch and bound* padrão do CPLEX. Uma variante chamada B&B-SF, baseada na formulação *SF*, e a outra baseada na formulação *MF*, chamada B&B-MF. Os objetivos deste experimento foram: (i) Identificar as instâncias que podem ser resolvidas na otimalidade por estes dois algoritmos; e (ii) Avaliar seus gaps médios ($[ub - lb]/ub$) em 3600 segundos, com ub e lb sendo, respectivamente, o melhor limite superior e o melhor limite inferior obtido por cada algoritmo dentro dos 3600 segundos.

Os resultados são apresentados na Tabela 1. A primeira coluna identifica cada

conjunto de 10 instâncias. A segunda coluna apresenta o número médio de variáveis binárias das duas formulações. Os gaps médios de otimalidade relativos ao algoritmo B&B-SF são apresentados na terceira coluna. Além disso, o número de instâncias solucionadas com garantia de otimalidade pelo algoritmo B&B-SF e seus tempos médios de execução nessas instâncias são relatados nas quarta e quinta colunas, respectivamente. Caso não seja encontrada nenhuma solução ótima em 3600 segundos, este limite de tempo é apresentado na quinta coluna. Na sexta coluna tem-se o número de nós explorados pelo algoritmo. Os mesmos dados são reportados para o algoritmo B&B-MF nas últimas quatro colunas.

Pela última linha da Tabela 1, observa-se que o algoritmo B&B-SF obteve um gap médio de 2,3%, enquanto o algoritmo B&B-MF alcançou um gap médio de 3,4%. Esses resultados são inesperados porque a formulação *MF* tem uma melhor relaxação linear que é resolvida em um tempo inferior a relaxação da formulação *SF*, como pode ser visto na Tabela 3. O número de instâncias resolvidas com garantia de otimalidade foi 34 para o algoritmo B&B-SF e 27 para o algoritmo B&B-MF.

Realizou-se um segundo experimento alterando parâmetros para os dois algoritmos de *branch and bound*. Na Tabela 2 são apresentados experimentos nos quais todos os cortes do CPLEX foram desabilitados. Com exceção da coluna variáveis binárias, a Tabela 2 possui os mesmos campos da Tabela 1. Com todos os cortes desabilitados, o algoritmo B&B-SF apresentou um gap médio de 5,6% enquanto o algoritmo B&B-MF um gap médio de 3,2%. Estes resultados indicam que o algoritmo B&B-MF não obtém redução significativa dos gaps de otimalidade com o uso de cortes, o que pode ser explicado pelo fato de que ele utiliza cortes que não resultam em incremento no valor do limite inferior da formulação. Por outro lado, o algoritmo B&B-SF tem uma redução significativa do gap com a utilização dos cortes (de 5,6% para 2,3%).

Durante os experimentos não foi possível identificar quais foram os cortes que mais influenciaram na performance do algoritmo B&B-MF. O CPLEX tem, na versão atual, doze diferentes tipos de cortes e avaliar todas as possíveis combinações seria inviável com relação a tempo computacional para todo o conjunto de instâncias. Durante a análise da utilização de alguns tipos de cortes em um conjunto reduzido de instâncias, verificou-se

que o algoritmo B&B-MF não utilizou nenhum corte *Flow Path*.

Assim, em um terceiro experimento no qual a execução dos dois algoritmos foi realizada com apenas a aplicação automática de cortes *Flow Path*, os resultados de gap foram idênticos aos apresentados no primeiro experimento, Tabela 1. Deste terceiro experimento, conclui-se que utilizar apenas o corte *Flow Path* apresenta os mesmos benefícios para o algoritmo B&B-SF do que deixar todos os cortes ativados. O CPLEX possui quatro possíveis configurações para a aplicação dos cortes que varia de cortes desabilitados, a utilização padrão que aplica cortes enquanto ocorrer melhora no limite inferior, aplicação de cortes de maneira moderada e agressiva. Nos experimentos com cortes habilitados, foi utilizada a aplicação padrão.

Em pesquisa realizada na literatura não foi possível identificar a razão pela qual o algoritmo B&B-MF não utilizou os cortes *Flow Path*. Estes cortes são gerados considerando um conjunto de restrições contendo as variáveis contínuas que definem uma estrutura de caminho em uma rede, em que as restrições são nós e as variáveis contínuas são fluxos de entrada e saída. Os fluxos estarão ativados ou desativados dependendo das configurações das variáveis binárias associadas.

No quarto experimento desta seção, avalia-se o desempenho de quatro abordagens para obter limites inferiores para o IULSBP. As duas primeiras consistem em resolver a relaxação linear das formulações *SF* e *MF*, respectivamente, enquanto a terceira e quarta consistem na solução da relaxação combinatória da formulação *CR* e da formulação $CR^{w_i \geq 0}$. O objetivo deste experimento é avaliar a qualidade dos limites inferiores fornecidos por cada abordagem e os correspondentes tempos de execução. Os resultados são exibidos como um intervalo de otimalidade ($[ub^* - lb]/ub^*$) entre os limites inferiores lb obtidos por cada abordagem e o custo ub^* da melhor solução de limite superior obtida para cada instância (encontrado por qualquer um dos algoritmos estudados neste trabalho). Portanto, quanto menor for o gap exibido para uma determinada abordagem, melhor é seu limite inferior.

Os resultados são apresentados na Tabela 3, na qual a primeira coluna identifica cada conjunto de 10 instâncias. O gap médio da relaxação linear de *SF*, calculado através

Tabela 1 – Resultados de duas variantes do algoritmo de *branch and bound* padrão do CPLEX. O B&B-SF baseia-se na formulação *SF*, enquanto o B&B-MF na formulação *MF*.

Name	int vars	B&B-SF				B&B-MF			
		g(%)	#	t(s)	nós	g(%)	#	t(s)	nós
g1-36-6-4	1118	1,4	5	334,8	57276	1,4	5	1220,0	74778
g1-48-6-4	1990	1,6	0	3600,0	75710	1,8	0	3600,0	83588
g1-72-6-6	6023	1,5	0	3600,0	22249	1,7	0	3600,0	37419
g1-90-6-6	9655	1,2	0	3600,0	15531	1,6	0	3600,0	19208
g1-120-8-8	21850	4,5	0	3600,0	1689	7,7	0	3600,0	2845
g1-144-8-8	31350	7,6	0	3600,0	538	11,6	0	3600,0	564
g2-36-6-4	1128	1,7	3	12,0	110869	1,2	5	207,0	89240
g2-48-6-4	2016	1,5	2	1161,0	113226	1,6	2	290,0	57990
g2-72-6-6	6246	1,3	0	3600,0	25127	1,4	0	3600,0	28708
g2-90-6-6	9847	1,5	0	3600,0	25769	1,4	0	3600,0	19583
g2-120-8-8	22114	1,6	0	3600,0	3618	3,1	0	3600,0	3202
g2-144-8-8	32444	3,1	0	3600,0	688	5,4	0	3600,0	1387
g3-36-6-4	1132	1,3	5	360,0	90678	1,2	3	1325,0	67944
g3-48-6-4	2016	1,3	1	2315,0	90835	1,5	1	339,0	48242
g3-72-6-6	6208	0,9	0	3600,0	28191	1,1	1	3015,0	37572
g3-90-6-6	9655	0,9	1	3142,0	21376	1,1	1	1719,0	18251
g3-120-8-8	22166	2,2	0	3600,0	2499	3,5	0	3600,0	2999
g3-144-8-8	31892	4,7	0	3600,0	612	5,6	0	3600,0	1384
g4-36-6-4	1124	1,6	3	90,0	87258	2,0	1	286,0	84318
g4-48-6-4	2059	0,9	4	1932,0	78514	1,4	0	3600,0	75269
g4-72-6-6	6287	1,0	1	1377,0	27816	1,3	0	3600,0	39294
g4-90-6-6	9613	1,4	0	3600,0	20620	2,0	0	3600,0	16738
g4-120-8-8	22302	3,1	0	3600,0	2485	8,8	0	3600,0	2765
g4-144-8-8	32084	6,3	0	3600,0	544	9,3	0	3600,0	992
g5-36-6-4	1178	2,0	3	353,0	95989	1,7	4	999,0	68187
g5-48-6-4	1963	2,0	1	322,0	78998	2,4	0	3600,0	56248
g5-72-6-6	6011	1,4	0	3600,0	24568	1,6	0	3600,0	32329
g5-90-6-6	9583	1,5	0	3600,0	15036	1,7	0	3600,0	17947
g5-120-8-8	21702	2,7	0	3600,0	1894	8,3	0	3600,0	2792
g5-144-8-8	31475	7,8	0	3600,0	548	12,1	0	3600,0	623
g6-36-6-4	1170	1,8	4	901,0	128379	2,1	3	572,0	102871
g6-48-6-4	2010	1,6	1	2,5	104233	1,8	1	1846,0	74531
g6-72-6-6	6191	1,2	0	3600,0	22519	1,5	0	3600,0	28037
g6-90-6-6	9790	1,5	0	3600,0	19968	1,4	0	3600,0	25316
g6-120-8-8	21666	2,1	0	3600,0	2213	2,6	0	3600,0	3553
g6-144-8-8	30683	4,4	0	3600,0	614	6,6	0	3600,0	1408
Média:		2,3				3,4			

Tabela 2 – Resultados dos algoritmos B&B-SF e B&B-MF sem nenhum corte habilitado.

Name	B&B-SF				B&B-MF			
	g(%)	#	t(s)	nos	g(%)	#	t(s)	nos
g1-36-6-4	2,3	3	2,4	925607	2,8	2	701,0	814514
g1-48-6-4	2,7	0	3600,0	743455	2,5	1	1591,0	609651
g1-72-6-6	2,1	0	3600,0	275373	2,1	0	3600,0	166296
g1-90-6-6	2,9	0	3600,0	76369	2,2	0	3600,0	70537
g1-120-8-8	11,8	0	3600,0	15051	7,2	0	3600,0	8029
g1-144-8-8	15,1	0	3600,0	7733	6,7	0	3600,0	5546
g2-36-6-4	2,0	3	44,9	898921	2,6	1	1535,0	543708
g2-48-6-4	2,6	0	3600,0	542195	2,2	0	3600,0	446287
g2-72-6-6	1,8	0	3600,0	122719	2,2	0	3600,0	115250
g2-90-6-6	2,1	0	3600,0	79195	2,4	0	3600,0	61829
g2-120-8-8	9,1	0	3600,0	8810	3,5	0	3600,0	6289
g2-144-8-8	12,6	0	3600,0	3780	4,7	0	3600,0	4723
g3-36-6-4	1,6	4	119,0	1208775	2,1	3	2235,0	438339
g3-48-6-4	2,2	0	3600,0	643457	2,1	0	3600,0	421587
g3-72-6-6	1,6	0	3600,0	155863	1,9	0	3600,0	112723
g3-90-6-6	1,6	0	3600,0	79904	1,7	0	3600,0	71544
g3-120-8-8	10,1	0	3600,0	8295	3,1	0	3600,0	14621
g3-144-8-8	12,8	0	3600,0	4863	4,5	0	3600,0	5692
g4-36-6-4	1,8	3	1056,0	1050338	2,8	1	3303,0	824029
g4-48-6-4	2,3	0	3600,0	1049409	2,4	0	3600,0	681696
g4-72-6-6	2,0	0	3600,0	235095	2,1	0	3600,0	163585
g4-90-6-6	2,6	0	3600,0	94500	2,3	0	3600,0	80486
g4-120-8-8	12,8	0	3600,0	13396	4,7	0	3600,0	13377
g4-144-8-8	16,8	0	3600,1	6229	6,0	0	3600,0	5791
g5-36-6-4	2,6	3	1724,0	759300	2,9	2	3343,0	837111
g5-48-6-4	2,6	0	3600,0	870271	2,8	0	3600,0	711975
g5-72-6-6	2,2	0	3600,0	178431	2,3	0	3600,0	165851
g5-90-6-6	2,9	0	3600,0	72784	2,5	0	3600,0	96536
g5-120-8-8	11,2	0	3600,0	19218	5,3	0	3600,0	14978
g5-144-8-8	14,3	0	3600,0	6548	7,3	0	3600,0	5051
g6-36-6-4	2,9	2	0,7	770751	2,3	2	866,0	387471
g6-48-6-4	2,6	1	1,4	589854	2,6	1	9,0	359232
g6-72-6-6	2,1	0	3600,0	147606	1,9	0	3600,0	115272
g6-90-6-6	2,1	0	3600,0	68307	2,1	0	3600,0	70386
g6-120-8-8	9,1	0	3600,0	7682	3,5	0	3600,0	12538
g6-144-8-8	11,8	0	3600,0	3018	5,1	0	3600,0	5418
Média:	5,6				3,2			

do limite inferior com relação ao ub^* é mostrada na segunda coluna. O tempo que o CPLEX levou para resolver essa formulação é exibido na terceira coluna. Os mesmos dados são apresentados para a formulação MF na quarta e quinta colunas, respectivamente.

A relaxação combinatória CR não pôde ser resolvida na otimalidade, dentro do tempo de 3600 segundos, para todas as instâncias. Por este motivo apresenta-se na sexta coluna o gap obtido no CPLEX para os grupos de instâncias. O gap médio do limite inferior da relaxação combinatória com relação a ub^* é mostrado na sétima coluna. Além disso, o número de instâncias que foram resolvidas na otimalidade por CR e seus tempos médios de execução são exibidos na oitava e na nona colunas, respectivamente. Os mesmos dados apresentados para as relaxações das formulações SF e MF estão nas duas últimas colunas para a relaxação $CR^{wt \geq 0}$.

Observa-se que, conforme esperado, os limites inferiores obtidos pela relaxação linear da formulação MF possuem menor gap médio (5,0%) do que a relaxação linear da formulação SF (19,7%). Por outro lado, seus tempos de execução foram menores que os da relaxação linear da formulação SF , apesar da formulação MF possuir um número maior de variáveis e restrições. Isso é explicado pelo fato de que, após o pré-processamento do CPLEX, o número de variáveis em SF e MF estão na mesma ordem de grandeza e o número de variáveis com coeficientes não zero na relaxação linear da formulação MF é, em média, cerca da metade dos coeficientes na relaxação linear da formulação SF , o que torna a relaxação linear da formulação MF mais fácil de resolver.

Pela última linha da Tabela 3, verifica-se que o gap médio da relaxação combinatória CR foi de 0,8%, mas com tempos computacionais elevados. Por outro lado, a relaxação combinatória $CR^{wt \geq 0}$ apresentou um gap médio de 2,0%, obtido com uma fração do tempo necessário para calcular a relaxação de SF e MF . Esta última relaxação é utilizada para desenvolver uma heurística construtiva. Foram realizados experimentos para a Relaxação Combinatória aplicada a formulação MF . Estes resultados demonstraram que esta relaxação apresentava resultados de gap semelhantes, mas com um aumento considerável do tempo computacional.

Tabela 3 – Desempenho das quatro abordagens para obter limites inferiores para o IULSBP. As duas primeiras consistem na solução da relaxação linear das formulações SF e MF , respectivamente. A terceira e quarta consistem na solução da relaxação combinatória CR e $CR^{wt \geq 0}$.

Name	SF		MF		CR				$CR^{wt \geq 0}$	
	g(%)	t(s)	g(%)	t(s)	gC(%)	g(%)	#	t(s)	g(%)	t(s)
g1-36-6-4	10,7	0,2	2,8	0,1	0,0	0,4	10	11,8	0,9	0,0
g1-48-6-4	15,1	0,6	5,5	1,0	0,4	0,5	9	385,7	2,2	0,0
g1-72-6-6	21,3	4,1	5,9	7,2	0,0	0,2	10	462,6	2,2	0,2
g1-90-6-6	23,3	9,3	5,4	10,6	0,0	0,3	10	259,1	1,9	0,2
g1-120-8-8	28,9	39,7	7,2	34,5	0,5	1,6	7	1352,6	2,6	0,4
g1-144-8-8	29,3	59,9	6,8	41,1	0,5	1,6	8	1248,0	2,5	0,4
g2-36-6-4	8,5	0,2	1,9	0,3	0,0	0,4	10	0,6	1,2	0,0
g2-48-6-4	11,5	0,9	3,7	1,0	0,0	0,5	10	1,9	2,1	0,0
g2-72-6-6	17,4	5,5	5,1	3,9	0,0	0,5	10	5,0	2,4	0,1
g2-90-6-6	18,8	13,1	4,2	7,2	0,0	0,6	10	3,2	2,3	0,1
g2-120-8-8	23,6	55,4	5,5	19,1	0,0	1,0	10	12,1	2,1	0,2
g2-144-8-8	24,2	90,7	5,0	33,0	0,0	0,9	10	11,8	2,0	0,3
g3-36-6-4	9,1	0,2	2,6	0,4	0,5	0,2	7	1204,0	0,7	0,0
g3-48-6-4	12,2	0,9	4,6	0,9	0,7	0,6	3	2827,5	2,0	0,0
g3-72-6-6	19,4	5,5	5,0	4,4	0,6	0,6	3	3105,0	2,0	0,1
g3-90-6-6	19,2	11,8	4,3	6,9	1,0	0,8	1	3396,3	1,8	0,1
g3-120-8-8	26,1	57,5	6,8	17,9	4,2	1,8	0	3600,0	2,1	0,3
g3-144-8-8	26,5	85,9	5,8	32,0	3,9	2,0	0	3600,0	2,1	0,3
g4-36-6-4	12,5	0,2	3,5	0,3	0,0	0,8	10	8,8	1,6	0,0
g4-48-6-4	18,2	0,9	5,7	1,1	0,0	0,5	10	26,9	2,6	0,0
g4-72-6-6	24,0	4,4	5,7	4,8	0,0	0,2	10	29,1	2,0	0,1
g4-90-6-6	24,8	8,9	5,8	7,5	0,0	0,6	10	26,3	2,1	0,2
g4-120-8-8	30,8	42,8	6,9	22,5	0,0	1,0	10	429,5	1,9	0,4
g4-144-8-8	31,8	61,7	6,9	36,8	0,0	1,5	10	793,7	2,3	0,4
g5-36-6-4	13,7	0,4	3,6	0,4	0,0	0,5	10	13,1	1,3	0,0
g5-48-6-4	13,4	1,0	4,7	0,9	0,0	0,5	10	167,8	1,8	0,0
g5-72-6-6	19,3	4,1	5,2	4,7	0,4	0,3	8	1180,9	2,3	0,1
g5-90-6-6	21,1	9,1	4,9	7,5	0,2	0,3	9	948,4	2,1	0,1
g5-120-8-8	26,4	38,9	7,0	20,8	0,7	1,7	3	2709,3	2,8	0,4
g5-144-8-8	27,0	64,8	6,5	34,7	0,6	1,8	4	2790,8	2,7	0,5
g6-36-6-4	9,6	0,3	2,6	0,4	0,0	0,6	10	0,8	1,3	0,0
g6-48-6-4	11,0	0,9	3,3	0,9	0,0	1,2	10	1,4	2,4	0,0
g6-72-6-6	15,5	5,4	4,3	4,4	0,0	0,2	10	27,0	2,2	0,1
g6-90-6-6	16,1	12,9	3,1	7,8	0,0	0,3	10	5,4	2,1	0,1
g6-120-8-8	21,5	54,1	5,4	17,5	0,0	1,0	10	224,7	2,2	0,2
g6-144-8-8	21,3	88,1	5,3	30,7	0,0	1,5	10	33,0	2,6	0,3
Média:	19,7		5,0			0,8			2,0	

4 Heurísticas Determinísticas para IULSBP

Para solucionar o IULSBP foram propostas neste capítulo: (i) Heurísticas de duas fases nas quais os problemas de Dimensionamento de Lotes e de Empacotamento são solucionados sequencialmente; e (ii) Heurísticas baseadas nas formulações apresentadas no Capítulo 3. Estes métodos de solução recebem como entrada uma instância do problema caracterizada pela tupla $\langle V, P, T, K, Q, q_{ip}, t_i, c_{pt}, e_{pt}, f_t \rangle$, e retornam uma solução caracterizado pela tupla $\langle T_p, S \rangle$. O subconjunto $T_p \subseteq T$ representa os períodos nos quais há compras dos produtos $p \in P$ e S é um conjunto de tuplas $\langle b, t \rangle$, em que $b \subseteq V$ é um subconjunto de pedidos que são empacotados no mesmo bin e entregues no mesmo período $t \in T$. Como o problema de Dimensionamento de Lotes é não capacitado, o valor de quantas unidades do produto $p \in P$ são compradas no período $t \in T$ é igual a soma das demandas para este produto entre t e o próximo período em que ocorrer a compra.

Nas heurísticas de duas fases o problema de Dimensionamento de Lotes é resolvido pelo algoritmo de Programação Dinâmica proposto em (WAGNER; WHITIN, 1958). Nas heurísticas em que o problema de Empacotamento é resolvido em sua forma clássica, empacotando os pedidos sem considerar o período limite de empacotamento, o problema foi resolvido pelo VPSolver (BRANDAO; PEDROSO, 2016).

4.1 Heurística UH

A primeira heurística para o IULSBP, chamada de heurística Desacoplada - UH (do inglês, Uncouple Heuristic), resolve os problemas de Dimensionamento de Lotes e de Empacotamento de forma completamente independente. Os pedidos dos clientes são sempre entregues em sua data limite e a solução do problema de Empacotamento não tem nenhum impacto sobre o problema de Dimensionamento do Lotes e vice-versa. Nenhum esforço é realizado para combinar as soluções de ambos os problemas com o objetivo de encontrar melhores soluções para o problema integrado.

O pseudocódigo da UH é descrito no Algoritmo 1. Na Linha 1, a instância do problema de Dimensionamento de Lotes caracterizada pela tupla $\langle V, P, T, q_{ip}, t_i, c_{pt}, e_{pt} \rangle$ é resolvida pelo algoritmo WW (WAGNER; WHITIN, 1958). O conjunto S é inicializado na Linha 2. O laço das linhas 3-7 é executado para cada período $t \in T$. O subconjunto $V_t \subseteq V$ de pedidos $i \in V_t$ cuja data de vencimento $t_i = t$ é identificada na Linha 4. Um problema de Empacotamento é resolvido para os pedidos em V_t , em que a capacidade dos bins é Q , e o peso de cada pedido i é $\bar{q}_i = \sum_{p \in P} q_{ip}$. A solução do problema de Empacotamento resultante é armazenada em $B_t \subset 2^V$ na Linha 5. O laço das linhas 6-7 insere cada bin $b \in B_t$ na solução S . A heurística é encerrada na Linha 8, e a solução $\langle T_p, S \rangle$ é retornada.

Algoritmo 1: $\text{UH}(V, P, T, K, Q, q_{ip}, t_i, c_{pt}, e_{pt}, f_t)$

1. $T_p \leftarrow \text{WW}(V, P, T, q_{ip}, t_i, c_{pt}, e_{pt})$
 2. $S \leftarrow \emptyset$
 3. **Para** $t = 1$ **até** $|T|$ **faça**
 4. $V_t \leftarrow \{i \in V : t_i = t\}$
 5. $B_t \leftarrow \text{BP}(V_t, Q, \bar{q}_i)$
 6. **Para cada** $b \in B_t$ **faça**
 7. $S \leftarrow S \cup \langle b, t \rangle$
 8. **retorna** $\langle T_p, S \rangle$
-

4.2 Heurística LSBP

A heurística LSBP (do inglês *Lot Sizing and Bin Packing*) para o problema integrado resolve inicialmente uma instância do problema de Dimensionamento de Lotes considerando todo o horizonte de planejamento. Em seguida, ela resolve vários problemas de Empacotamento para decidir como e quando empacotar os pedidos dos clientes. Esses pedidos podem ser entregues antes de suas datas limite (desde que existam recursos suficientes para atender às suas demandas) a fim de minimizar os custos de empacotamento. A redução dos custos pode ser obtida reduzindo o espaço não utilizado nos bins, empacotando em períodos com menor custo de empacotamento e, possivelmente, reduzindo o número de bins.

Algoritmo 2: LSPB($V, P, T, K, Q, q_{ip}, t_i, c_{pt}, e_{pt}, f_t$)

-
01. $T_p \leftarrow \text{WW}(V, P, T, q_{ip}, t_i, c_{pt}, e_{pt})$
 02. $t'_i \leftarrow \max_{p \in P: q_{ip} > 0} \max_{t \in T_p: t \leq t_i} t, \quad \forall i \in V$
 03. $S \leftarrow \emptyset$
 04. **Para** $t = 1$ **até** $|T|$ **faça**
 05. $V_t \leftarrow \{i \in V : t'_i \leq t \wedge \nexists \langle b, t_b \rangle \in S : i \in b\}$
 06. $B_t \leftarrow \text{BP}(V_t, Q, \bar{q}_i)$
 07. **Para cada** $b \in B_t$ **faça**
 08. $u_b \leftarrow \min\{t_i : i \in b\}$
 09. **Se** $t = u_b$ **ou** $f_t < f_u, \forall u \in \{t + 1, \dots, u_b\}$, **então**
 10. $S \leftarrow S \cup \langle b, t \rangle$
 11. **retorna** $\langle T_p, S \rangle$
-

O pseudocódigo da heurística LSBP é descrito no Algoritmo 2. Na Linha 1, a instância do problema de Dimensionamento de Lotes caracterizada pela tupla $\langle V, P, T, q_{ip}, t_i, c_{pt}, e_{pt} \rangle$ é resolvida por WW (WAGNER; WHITIN, 1958). Então, o período t'_i que é o primeiro período do horizonte de planejamento para o qual existem produtos disponíveis para empacotamento (de acordo com a solução T_p do problema de Dimensionamento de Lotes) do pedido $i \in V$ é computado na Linha 2. O conjunto S é inicializado na Linha 3. O laço nas linhas 4-10 é executado para cada período $t \in T$ e decide quais pedidos são empacotados juntos e entregues em t e quais são adiados para períodos posteriores. O subconjunto $V_t \subseteq V$ dos pedidos ainda não empacotados e que podem ser entregues em $t'_i \leq t$ é identificado na Linha 5. Um problema de Empacotamento é resolvido para as ordens em V_t , em que a capacidade dos bins é Q e o peso de cada pedido i é $\bar{q}_i = \sum_{p \in P} q_{ip}$. A solução problema de Empacotamento resultante é armazenada em $B_t \subset 2^V$ na Linha 6. O laço nas linhas 7-10 decide, para cada bin $b \in B_t$, se o bin é entregue em t ou se os pedidos em b são adiados para períodos posteriores. O período mais recente $u_b \geq t$ no qual o bin b pode ser entregue é identificado na Linha 8. Se u_b for igual a t ou se t for o período mais barato disponível para empacotar os pedidos em b (Linha 9), a tupla $\langle b, t \rangle$ é adicionada a S na Linha 10, indicando que b é entregue em t . Caso contrário, as ordens em

b são entregues em um período posterior. A heurística LSBP para na Linha 11, fornecendo a solução $\langle T_p, S \rangle$.

4.3 Heurística CRH

A heurística CRH (do ingles *Combinatorial Relaxation Heuristic*) utiliza inicialmente a formulação $CR^{wt \geq 0}$ para decidir (i) os períodos em que há ordens de compras de cada produto em P e (ii) a data de empacotamento de cada pedido em V . Em seguida, ela resolve um problema de Empacotamento para cada período em T para decidir como empacotar os pedidos.

Algoritmo 3: $CRH(V, P, T, K, Q, q_{ip}, t_i, c_{pt}, e_{pt}, f_t)$

1. $\langle T_p, V_t \rangle \leftarrow \text{Solve-CR}^{wt \geq 0}(V, P, T, Q, q_{ip}, t_i, c_{pt}, e_{pt}, f_t)$
 2. $S \leftarrow \emptyset$
 3. **Para** $t = 1$ **até** $|T|$ **tal que** $V_t \neq \emptyset$ **faça**
 4. $B_t \leftarrow \text{BP}(V_t, Q, \bar{q}_i)$
 5. **Para cada** $b \in B_t$ **faça**
 6. $S \leftarrow S \cup \langle b, t \rangle$
 7. **retorna** $\langle T_p, S \rangle$
-

O pseudocódigo da heurística CRH é descrita no Algoritmo 3. Na Linha 1, a formulação $CR^{wt \geq 0}$ é resolvida para a instância caracterizada pela tupla $\langle V, P, T, Q, q_{ip}, t_i, c_{pt}, e_{pt}, f_t \rangle$. Os períodos em que há compras do produto $p \in P$ são armazenados em $T_p \subseteq T$, e os pedidos que são entregues no período t são armazenados em $V_t \subseteq V$, para todos os $t \in T$. O conjunto S é inicializado na Linha 2. O laço nas linhas 3-6 é executado para cada período $t \in T$, no qual há pedidos a serem entregues. Um problema de Empacotamento é resolvido para todas os pedidos programados para serem entregues no período t , sendo a capacidade dos bins igual a Q e o peso de cada pedido $i \in V_t$ é $\bar{q}_i = \sum_{p \in P} q_{ip}$. A solução do problema de Empacotamento resultante é armazenada em $B_t \subset 2^V$ na Linha 4. O laço nas linhas 5-6 insere cada bin $b \in B_t$ na solução S . A heurística CRH termina na Linha 7, e a solução $\langle T_p, S \rangle$ é retornada.

4.4 Heurística BPLS

A heurística BPLS (do inglês *Bin Packing and Lot Sizing*) inicialmente gera o conjunto Δ com todas as partições de T em um conjunto de intervalos contendo no máximo β períodos. Dada uma partição $\delta \in \Delta$, $[l, u] \in \delta$ denota um subconjunto de períodos consecutivos em que l e u , com possivelmente $l = u$, são os primeiro e os últimos períodos, respectivamente. Ou seja, $\Delta = \{\delta : \bigcap_{[l,u] \in \delta} \mathcal{T}([l, u]) = \emptyset \wedge \bigcup_{[l,u] \in \delta} \mathcal{T}([l, u]) = T \wedge |\mathcal{T}([l, u])| \leq \beta, \forall [l, u] \in \delta\}$, e $\mathcal{T}([l, u]) \subseteq T$ é o conjunto com todos os períodos no intervalo $[l, u] \in \delta$. O valor de β é um parâmetro a ser ajustado. Essa heurística cria uma solução diferente para cada $\delta \in \Delta$ da seguinte maneira. Primeiro, ela resolve uma instância de Empacotamento para cada subconjunto de δ com $V_{[l,u]} = \{i \in V : l \leq t_i \leq u\}$. Em seguida, ela resolve uma instância do subproblema de Dimensionamento de Lotes em que as demandas de cada produto $p \in P$ em cada período $t \in T$ são definidas de acordo com a forma como os pedidos foram empacotados na etapa anterior. Finalmente, depois que o subproblema de dimensionamento do lote foi resolvido, o BPLS tenta identificar se há um intervalo de períodos em que um bin pode ser entregue com menor custo.

O pseudocódigo da heurística BPLS é descrito no Algoritmo 4. Ela recebe uma instância do problema caracterizada pela tupla $\langle V, P, T, K, Q, q_{ip}, t_i, c_{pt}, e_{pt}, f_t \rangle$, e retorna a melhor solução conhecida caracterizada pela tupla $\langle T_p^*, S^* \rangle$. O valor de quantas unidades do produto $p \in P$ são compradas no período $t \in T$ pode ser calculado diretamente a partir de $\langle T_p^*, S^* \rangle$. Na linha 1, o conjunto Δ é inicializado. O laço das linhas 2-12 cria uma solução $\langle T_p, S \rangle$ para cada partição $\delta \in \Delta$. Um problema de Empacotamento é resolvido para cada $[l, u] \in \delta$, e $V_{[l,u]} = \{i \in V : t_i \in [l, u]\}$, sendo a capacidade dos bins é Q e o peso de cada pedido $i \in V_{[l,u]}$ é $\bar{q}_i = \sum_{p \in P} q_{ip}$. Os problemas de Empacotamento também são resolvidos pelo VPSolver (BRANDAO; PEDROSO, 2016). Na Linha 3, a união de todas as soluções resultantes do problema de Empacotamento é armazenada em $B \subset 2^V$. Este conjunto é tal que $\bigcup_{b \in B} b = V$. O período mais recente $u_b \in T$, no qual o bin $b \in B$ pode ser entregue, é computado na Linha 4. Como todos os pedidos em b devem ser entregues no mesmo período, uma nova data limite de empacotamento $\bar{t}_i = u_b$ é definida para cada pedido $i \in b$ na Linha 5. Uma instância do problema de Dimensionamento de Lotes caracterizada

pela tupla $\langle V, P, T, q_{ip}, \bar{t}_i, c_{pt}, e_{pt} \rangle$ é resolvida por WW (WAGNER; WHITIN, 1958) na Linha 6, a fim de garantir recursos suficientes para atender às demandas de cada pedido i in \bar{t}_i . O conjunto S é inicializado na Linha 7, e o laço nas linhas 8-11 decide a data de empacotamento de cada bin $b \in B$. O período l_b representando a primeira data de empacotamento (de acordo com a solução T_p do problema de Dimensionamento de Lotes) para o bin $b \in B$ é calculado na Linha 9, e o período $t^* \in [l_b, u_b]$ com o menor custo de empacotamento é identificado na Linha 10. Em seguida, a tupla $\langle b, t^* \rangle$ é adicionada a S na Linha 11, indicando que b é entregue em t^* . A melhor solução conhecida $\langle T_p^*, S^* \rangle$ é atualizada na Linha 12 e retornada na Linha 13 no final da heurística.

Algoritmo 4: BPLS($V, P, T, K, Q, q_{ip}, t_i, c_{pt}, e_{pt}, f_t, \beta$)

01. Seja Δ o conjunto de partições de T em um conjunto de intervalos com no máximo β períodos
 02. **Para Cada** $\delta \in \Delta$ **faça**
 03. $B \leftarrow \bigcup_{[l,u] \in \delta} \text{BP}(V_{[l,u]} = \{i \in V : t_i \in [l, u]\}, Q, \bar{q}_i)$
 04. $u_b \leftarrow \min\{t_i : i \in b\}, \forall b \in B$
 05. $\bar{t}_i \leftarrow u_b, \forall i \in b : b \in B$
 06. $T_p \leftarrow \text{WW}(V, P, T, q_{ip}, \bar{t}_i, c_{pt}, e_{pt})$
 07. $S \leftarrow \emptyset$
 08. **Para Cada** $b \in B$ **faça**
 09. $l_b \leftarrow \max_{i \in b} \max_{p \in P: q_{ip} > 0} \max_{t \in T_p: t \leq t_i} t$
 10. $t^* \leftarrow \arg \min_{t \in [l_b, u_b]} f_t$
 11. $S \leftarrow S \cup \{\langle b, t^* \rangle\}$
 12. **Atualiza**($\langle T_p, S \rangle, \langle T_p^*, S^* \rangle$)
 13. **retorna** $\langle T_p^*, S^* \rangle$
-

4.5 Relaxar e Fixar - ReF

Nesta seção é descrito um procedimento iterativo para a solução do problema integrado em que ocorre a relaxação de variáveis inteiras da formulação SF , fixando parte das variáveis de acordo com soluções de iterações anteriores. Este procedimento consiste em a cada iteração resolver a formulação dividindo o conjunto de períodos em duas partes. Seja o período t_i com valor inicial igual a um e o parâmetro $t_s < T$, que representa a quantidade de períodos a partir de t_i em que as variáveis são mantidas como inteiras. Relaxa-se as variáveis inteiras para valores de $t > t_s$ resolvendo-se a formulação. Seja $t_k < t$

outro parâmetro do procedimento. Após a primeira iteração, as variáveis inteiras entre t_i e t_k são fixadas de acordo com a solução obtida na iteração anterior, sendo $t_k \leq t_s - t_i$. O período inicial t_i para a próxima iteração é dado por $t_i = t_i + t_k$.

Desta forma, a partir da segunda iteração da solução da formulação SF com algumas variáveis relaxadas, as variáveis inteiras com índices $t \in [0; t_i)$ têm seus valores fixados por iterações anteriores e as variáveis com índices $t > t_s$ são relaxadas. Como exemplo, seja uma instância do problema com $|T| = 12$ períodos e sejam os parâmetros $t_s = 4$ e $t_k = 2$. Na primeira iteração, $t_i = 1$ e todas as variáveis inteiras com índices $t \in (t_s; T]$, ou seja $t \in (4; 12]$ são relaxadas. Na segunda iteração, as variáveis inteiras nos períodos menores que $t_i + t_k$ são fixadas de acordo com os valores obtidos na primeira iteração e $t_i = t_i + t_k = 3$. Além disso, as variáveis inteiras indexadas com períodos $t \in (6; 12]$ são relaxadas. Na terceira iteração, o período inicial é $t_i = 5$ e as variáveis nos períodos menores que $t_i + t_k$ são fixadas de acordo com os valores obtidos na primeira e na segunda iterações. As variáveis indexadas com períodos $t \in (8; 12]$ são relaxadas. Este processo é repetido enquanto $t_s < T$. A Figura 4.5 apresenta as três primeiras iterações deste exemplo.

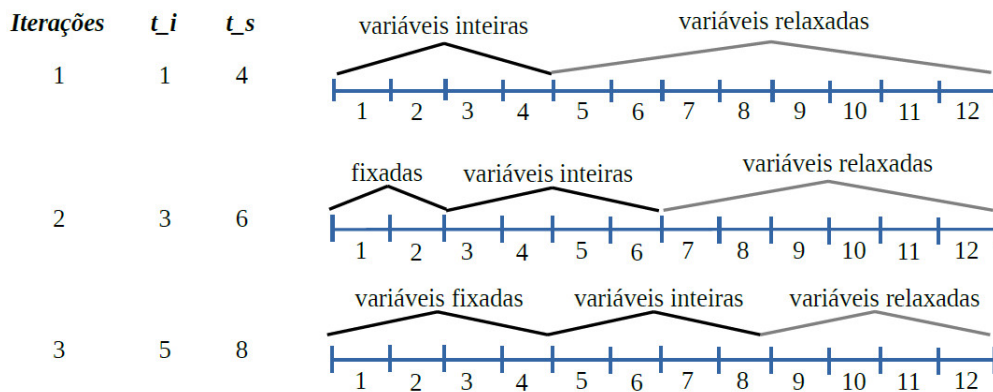


Figura 7 – Iterações do procedimento Relaxar e Fixar - ReF

Em experimentos preliminares avaliou-se execuções nas quais iniciava-se a fixação da variáveis a partir do período inicial e também, a partir do período final. A cada execução de uma iteração do procedimento ReF foi definido como critério de parada o tempo de execução. Ou seja, não foi fixado um tempo limite para a execução total do procedimento,

apenas para cada iteração.

Baseado nos resultados apresentados em experimentos preliminares, verificou-se que as execuções do procedimento que começavam fixando as variáveis do período final para o inicial apresentaram melhores resultados do que os que começavam fixando as variáveis do período inicial. Os melhores resultados para o procedimento foram obtidos partindo do período final e deixando apenas as variáveis de um período fixadas a cada iteração. Tendo os problemas resolvidos a cada iteração menos variáveis binárias devido ao fato de que o número de variáveis inteiras é menor para períodos próximos ao período final. Como exemplo, apenas existem variáveis y_{ikt} no período final para os pedidos que têm tempo limite de empacotamento neste período. Por outro lado, no período inicial, existe uma variável para cada pedido, veículo e período.

O procedimento ReF apresentou problemas com relação ao tempo computacional devido ao fato de que em muitas instâncias, as iterações não puderam ser resolvidas na otimalidade no tempo limite de execução. Um dos fatores que podem explicar este comportamento é o tempo elevado necessário para resolver a relaxação linear das grandes instâncias.

4.6 Heurísticas de Busca Local N-OPT

Nesta seção são apresentadas buscas locais que utilizam as heurísticas construtivas LSBP e CRH. Os valores de gaps médios destas heurísticas construtivas são da ordem de 3% com tempos de execução inferiores a um segundo. A partir de uma solução inicial, é definida uma vizinhança com tamanho N e então aplicado o algoritmo de *branch and bound* padrão do CPLEX utilizando a formulação SF . Todas as variáveis da formulação que não pertencem à vizinhança, têm seus valores fixados pela solução inicial na primeira execução, e pelas soluções fornecidas pelas buscas locais em caso de melhora da solução, a posteriori. As decisões tomadas pela busca local são definir como e quando realizar o empacotamento dos pedidos dentro de uma vizinhança e, para todo o horizonte de planejamento, quando comprar. Em cada busca local, as decisões de compra para qualquer

período são revistas porque o problema de Dimensionamento de Lotes é eficientemente resolvido na formulação SF .

A primeira busca local consiste em encontrar soluções de empacotamento que diferenciem da solução corrente por no máximo dois períodos consecutivos e para qualquer decisão de compra. A busca se inicia a partir dos dois primeiros períodos e após encontrar um ótimo local ou, devido a dificuldade do problema, um critério de parada seja atingido, é reiniciada na próxima vizinhança. Então, inicia-se uma nova busca para o segundo e terceiro períodos. O processo é repetido de forma sequencial até que o limite de tempo de execução seja alcançado ou toda a vizinhança seja explorada. Esta busca local foi denominada de heurística BL2-OPT. A segunda busca local é aplicada de forma semelhante a primeira mas utilizando uma vizinhança de três períodos consecutivos. A segunda busca local foi chamada de heurística BL3-OPT.

Em seguida, utilizou-se uma heurística para promover uma busca local em todas as possíveis vizinhanças com até N períodos do problema integrado, sendo N um parâmetro a ser ajustado. A esta heurística que é baseada na heurística VND, da-se o nome de BLN-OPT. Como um ótimo local com relação a uma vizinhança não é necessariamente um ótimo local com relação a outra vizinhança, trocas de vizinhanças podem ser realizadas durante a fase de busca local. A heurística VND é uma versão da metaheurística VNS proposta por (HANSEN; MLADENOVIC, 2001).

Como a estrutura de vizinhança utilizada é baseada em períodos consecutivos e como as instâncias de interesse para o problema tem número limitado de períodos, optou-se por uma heurística com busca de vizinhança não aleatorizada. A heurística BLN-OPT parte de uma solução inicial e realiza buscas locais em diferentes vizinhanças. Neste trabalho, este processo inicia-se com uma busca sequencial em vizinhança 2-OPT e, após explorar toda uma vizinhança, realiza-se o incremento do tamanho da vizinhança limitado ao parâmetro N e a um tempo de execução. Em uma heurística BLN-OPT padrão, sempre que uma busca local melhora a solução global, o processo de busca é reiniciado. Neste trabalho, como uma busca local em vizinhança $N+1$ -OPT engloba a busca em uma vizinhança N -OPT, optou-se por não reiniciar a busca local devido ao alto custo

computacional das buscas locais. Heurísticas BLN-OPT foram também utilizadas para resolver problemas relacionados com o problema de Roteamento, como em (TAGMOUTI et al., 2010; JARBOUI et al., 2013).

4.7 Experimentos Computacionais das Heurísticas Determinísticas

Nesta seção são apresentados os resultados dos experimentos com as heurísticas determinísticas UH, LSBP, CRH, BPLS, ReF, BL2-OPT, BL3-OPT e BLN-OPT. Pretende-se através dos experimentos avaliar a qualidade dos limites superiores fornecidos por cada heurística e os correspondentes tempos de execução. Os gaps relativos de otimalidade são calculados pela expressão $([ub - lb^*]/ub)$, em que ub é o limite superior obtido por cada heurística e lb^* o melhor valor de limite inferior conhecido para cada instância (encontrado por qualquer uma das abordagens exibidas na Tabela 3). Portanto, quanto menor o gap de uma determinada heurística, melhor é seu limite superior. Os parâmetros e ambiente de testes, bem como a instâncias utilizadas são os mesmos descritos na Seção 3.4.

O primeiro experimento, Tabela 4, foi realizado com as heurísticas UH, LSBP, CRH e BPLS. Na primeira coluna tem-se o nome das instâncias. O gap médio de otimalidade de UH é mostrado na segunda coluna e seu tempo médio de execução na terceira coluna. Os mesmos dados são relatados para a heurística LSBP na quarta e quinta colunas, para CRH na sexta e sétima colunas e para a heurística BPLS nas oitava e nona colunas, respectivamente. Para estes experimentos, $\beta = 8$, que corresponde ao número de períodos das maiores instâncias.

Observar-se, pela última linha da tabela, que as médias dos gaps de otimalidade das heurísticas LSBP (3,1%) e CRH (2,3%) são significativamente inferiores do que os da UH (7,8%). Além disso, seus tempos de execução nunca são superiores a um segundo. Os resultados da heurística BPLS são ainda melhores, com gaps médios de otimalidade também inferiores (1,1%) aos gaps das heurísticas LSBP e CRH. Embora os tempos máximos de execução da heurística BPLS sejam superiores as das outras heurísticas, eles são de cerca de apenas um minuto.

Tabela 4 – Desempenho das heurísticas UH, LSBP, CRH e BPLS.

Name	UH		LSBP		CRH		BPLS	
	g(%)	t(s)	g(%)	t(s)	g(%)	t(s)	g(%)	t(s)
g1-36-6-4	8,5	0,2	3,5	0,1	3,4	0,2	0,5	1,1
g1-48-6-4	5,3	0,2	3,1	0,2	2,9	0,3	0,7	1,3
g1-72-6-6	4,9	0,3	2,5	0,3	2,8	0,5	0,9	7,6
g1-90-6-6	4,4	0,3	2,8	0,3	3,4	0,6	1,4	8,9
g1-120-8-8	4,6	0,4	3,3	0,4	3,5	0,9	2,0	47,9
g1-144-8-8	3,3	0,5	2,7	0,4	2,8	0,9	1,7	58,6
g2-36-6-4	12,9	0,2	2,1	0,1	2,2	0,3	0,6	1,1
g2-48-6-4	11,8	0,2	1,5	0,2	1,5	0,2	0,7	1,3
g2-72-6-6	11,2	0,3	3,6	0,2	1,8	0,4	1,3	7,6
g2-90-6-6	10,7	0,3	3,2	0,2	1,2	0,4	1,2	9,0
g2-120-8-8	9,9	0,5	3,2	0,3	1,5	0,6	1,5	48,6
g2-144-8-8	9,2	0,5	3,4	0,4	1,1	0,9	1,0	57,8
g3-36-6-4	12,4	0,2	3,9	0,1	1,5	0,3	0,6	1,1
g3-48-6-4	9,6	0,2	1,4	0,1	1,0	0,2	0,7	1,2
g3-72-6-6	9,1	0,3	2,5	0,2	1,2	0,4	0,9	7,6
g3-90-6-6	9,4	0,3	2,6	0,2	1,2	0,4	0,9	9,1
g3-120-8-8	8,7	0,4	3,7	0,4	1,8	0,8	1,5	49,1
g3-144-8-8	8,5	0,5	3,7	0,4	1,2	0,8	1,2	58,2
g4-36-6-4	7,4	0,2	3,4	0,1	2,2	0,2	0,8	1,3
g4-48-6-4	4,3	0,2	2,0	0,2	2,0	0,3	0,8	1,3
g4-72-6-6	4,4	0,3	1,8	0,2	2,8	0,6	1,0	8,0
g4-90-6-6	3,6	0,4	2,6	0,3	3,3	0,6	1,8	9,2
g4-120-8-8	3,7	0,4	2,2	0,4	2,6	0,9	1,4	48,9
g4-144-8-8	3,4	0,5	2,3	0,5	3,0	1,0	1,6	60,2
g5-36-6-4	8,2	0,2	4,7	0,1	4,1	0,3	0,4	1,1
g5-48-6-4	6,2	0,2	2,6	0,2	3,5	0,3	0,8	1,3
g5-72-6-6	5,7	0,3	2,6	0,2	3,0	0,5	1,3	7,5
g5-90-6-6	5,6	0,3	3,0	0,3	3,8	0,6	1,2	8,9
g5-120-8-8	4,7	0,4	3,7	0,4	3,3	0,8	2,0	48,1
g5-144-8-8	4,1	0,5	3,2	0,4	3,4	0,9	2,1	58,5
g6-36-6-4	12,3	0,2	3,7	0,1	2,5	0,2	1,0	1,1
g6-48-6-4	10,7	0,2	2,7	0,1	1,6	0,2	0,8	1,3
g6-72-6-6	9,7	0,3	3,2	0,2	1,6	0,4	0,7	7,5
g6-90-6-6	9,9	0,3	3,2	0,2	1,4	0,4	1,2	8,9
g6-120-8-8	11,2	0,4	6,4	0,4	1,8	0,7	1,5	47,8
g6-144-8-8	11,3	0,5	6,7	0,4	1,5	0,7	1,5	59,2
Média:	7,8	0,32	3,1	0,25	2,3	0,52	1,1	22,2

No segundo experimento desta seção, têm-se os resultados da heurística Relaxar e Fixar - ReF com cada iteração tendo como critério de parada o tempo limite de 300 segundos. A Tabela 5 apresenta resultados para diferentes valores de t_i , t_k e t_s . O nome das instâncias estão apresentados na primeira coluna. Nas colunas segunda à quinta têm-se os resultados nos quais as iterações se iniciavam com as variáveis dos últimos períodos relaxadas, $t_i = 1$. Nas colunas sexta a nona, os resultados para execuções nas quais as iterações se iniciavam com as variáveis dos primeiros períodos relaxadas, $t_i = t_k + t_s$.

Na Tabela 5, a segunda e terceira colunas apresentam os valores de gap e de tempo de execução para os valores $t_k = t_s = 1$. Assim, para os resultados destas colunas, a heurística inicia com quase todas as variáveis inteiras relaxadas, com exceção das variáveis do primeiro período. Nas colunas quarta e quinta tem-se os valores de gap e de tempo de execução para $t_k = t_s = 2$. Nas colunas sexta e sétima, têm-se o gap e o tempo de execução para $t_k = -1$ e $t_s = |T|$. Finalmente, nas colunas oitava e nona têm-se o gap e o tempo de execução para $t_k = -2$ e $t_s = |T|$.

Verifica-se que os melhores valores de gap (2,6%) são obtidos quando as variáveis dos primeiros períodos da formulação são relaxadas e fixa-se a cada iteração variáveis de apenas um período, $t_s = |T|$ e $t_k = -1$, colunas sexta e sétima. Outro ponto verificado durante os experimentos é que mesmo relaxando a maioria das variáveis da formulação, a heurística demanda um alto tempo computacional. Isso pode ser observado pelos tempos médios necessários para a solução da heurística, concluindo-se que este método não é eficiente.

No terceiro experimento desta seção foram realizados testes com as heurísticas BL2-OPT e BL3-OPT utilizando a heurística construtiva LSBP, tendo como critérios de parada tempo de execução e gap. O número máximo de iterações (diferentes vizinhanças) é dado por $|T| - 1$ para a heurística BL2-OPT e $|T| - 2$ para a heurística BL3-OPT, sendo $|T|$ o número de períodos. Os nomes das instâncias são apresentados na primeira coluna da Tabela 6. Na segunda, terceira e quarta colunas tem-se os valores de gap, número de iterações e tempo de execução para a heurística BL2-OPT com critério de parada de 1% para cada iteração. Na quinta, sexta e sétima colunas tem-se os valores de gap, número

Tabela 5 – Experimentos com a heurística ReF que utiliza a formulação SF com diferentes valores de t_i , t_s e t_k

Nome	$t_k = 1$		$t_k = 2$		$t_k = -1$		$t_k = -2$	
	$t_s = 1$		$t_s = 2$		$t_s = T $		$t_s = T $	
	g(%)	t(s)	g(%)	t(s)	g(%)	t(s)	g(%)	t(s)
g1-36-6-4	3,5	4,8	2,8	4,9	9,0	5,8	4,5	37,0
g1-48-6-4	2,7	14,7	2,4	41,3	3,2	70,7	1,6	97,6
g1-72-6-6	1,9	65,0	1,0	187,7	5,0	145,3	2,2	249,8
g1-90-6-6	2,8	246,2	1,1	315,1	3,4	157,4	2,4	369,8
g1-120-8-8	5,2	456,8	4,5	539,2	4,1	318,1	2,8	569,0
g1-144-8-8	5,6	603,5	4,6	644,3	4,6	618,9	5,6	686,9
g2-36-6-4	3,9	135,2	2,4	29,2	2,1	13,9	2,1	30,5
g2-48-6-4	3,2	292,8	1,6	190,6	1,4	74,7	1,4	132,0
g2-72-6-6	2,6	534,3	1,2	490,1	0,9	341,6	1,0	462,7
g2-90-6-6	2,7	578,4	1,6	610,1	0,9	506,2	1,1	571,3
g2-120-8-8	4,6	1086,0	3,0	1029,7	1,3	893,1	2,9	919,3
g2-144-8-8	5,8	1108,4	5,5	982,9	3,6	1176,7	4,8	916,3
g3-36-6-4	4,3	148,1	2,5	68,2	1,7	8,0	1,4	61,8
g3-48-6-4	3,1	311,7	1,4	157,6	0,9	98,9	1,0	173,6
g3-72-6-6	2,1	506,7	1,1	429,4	0,8	400,4	0,7	441,5
g3-90-6-6	2,3	612,5	1,3	614,9	0,5	413,5	0,8	565,8
g3-120-8-8	4,4	1188,5	3,5	947,6	1,5	1110,9	3,7	933,5
g3-144-8-8	5,7	1154,6	6,0	1047,4	3,6	1350,9	6,0	1067,7
g4-36-6-4	2,9	42,0	2,8	98,1	2,1	12,0	1,8	195,7
g4-48-6-4	1,8	75,9	1,4	258,3	1,1	69,4	0,6	217,9
g4-72-6-6	1,5	346,0	1,8	606,0	0,9	295,4	0,9	465,8
g4-90-6-6	1,9	534,4	1,9	729,6	1,4	537,1	2,0	705,6
g4-120-8-8	3,5	1354,1	4,4	1051,2	3,0	1056,1	5,1	998,0
g4-144-8-8	11,0	1416,0	24,5	823,8	5,4	1341,3	8,4	1009,2
g5-36-6-4	3,9	18,9	3,8	91,3	1,9	6,0	1,6	168,3
g5-48-6-4	2,8	60,2	2,6	272,3	1,9	29,7	1,6	299,4
g5-72-6-6	1,9	383,5	2,0	593,7	1,0	293,0	1,1	524,4
g5-90-6-6	1,5	636,5	1,7	693,0	1,6	491,9	1,3	617,9
g5-120-8-8	4,2	1419,3	10,2	1005,6	3,0	1098,1	8,8	986,9
g5-144-8-8	8,2	1577,1	27,3	753,7	13,8	1370,3	23,0	981,3
g6-36-6-4	4,9	182,3	3,2	45,7	2,4	52,4	1,9	96,5
g6-48-6-4	4,0	392,9	1,9	229,6	1,2	77,3	1,2	88,5
g6-72-6-6	1,8	572,3	0,9	502,1	0,4	254,1	0,5	378,3
g6-90-6-6	2,3	653,6	1,3	616,3	0,8	425,8	1,3	582,4
g6-120-8-8	4,8	1111,7	3,5	1013,4	1,1	941,9	3,6	1018,3
g6-144-8-8	11,8	1093,9	19,5	898,3	1,7	1204,9	5,2	986,4
Média:	3,9	581,1	4,5	517,0	2,6	479,5	3,2	516,9

de iterações executadas e tempo de execução para a heurística BL2-OPT com critério de parada de cada iteração limitada a 60 segundos. Nas colunas oitava a décima terceira tem-se os mesmos campos das colunas segunda a sétima para a heurística BL3-OPT. O tempo de execução total destas heurísticas foi limitado a 600 segundos.

Pelos resultados da Tabela 6 conclui-se que o critério de parada para cada execução da busca local limitado ao tempo de 60 segundos apresentou resultados médios superiores do que quando a execução era encerrada pelo gap de 1%. Além disso, verifica-se que com o critério de parada de 60 segundos, as heurísticas BL2-OPT e BL3-OPT apresentaram resultados semelhantes. A aplicação das buscas locais permitiu reduzir o gap médio obtido pela heurística construtiva LSBP que foi de 3% para 1,3%.

No experimento final desta seção, a Tabela 7 apresenta os resultados da execução da heurística BLN-OPT e da heurística BPLS com tempo de execução total limitado a 600 segundos. Cada iteração da busca local na heurística BLN-OPT foi limitada a 60 segundos. Os nomes das instâncias são apresentados na primeira coluna da tabela. Na segunda, terceira e quarta colunas tem-se os valores de gap, número de iterações e tempo de execução para a heurística BLN-OPT. Nas colunas quinta, sexta e sétima, tem-se os valores médios para gap, qual foi o número de divisões do horizonte de planejamento que forneceu a melhor solução e o tempo para a heurística BPLS.

Pelos resultados, verifica-se que os gaps médios das duas heurísticas foram iguais. A heurística BPLS tem um tempo de execução inferior, com valores máximos da ordem de 60 segundos. Por outro lado, a heurística BLN-OPT tem um melhor desempenho com relação ao gap para as maiores instâncias de cada grupo. Se forem considerados apenas as instâncias com 90, 120 e 144 pedidos, os gaps médios são da ordem de 1,1% para a heurística BLN-OPT e 1,5% para a heurística BPLS. Além disso, a heurística BPLS é resolvida por subintervalos, na otimalidade. Isso implica que não é possível melhorar o seu resultado. Por outro lado, com um tempo maior de execução, a heurística BLN-OPT pode apresentar resultados melhores.

Experimentos realizados com as heurísticas BL2-OPT, BL3-OPT e com a heurística BLN-OPT, utilizando a heurística construtiva CRH, apresentaram resultados de gaps

Tabela 6 – Buscas locais de dois e três períodos aplicadas a uma heurística construtiva para o IULSBP

Nome	BL 2 OPT						BL 3 OPT					
	Gap 1%			60 s			Gap 1%			60s		
	g(%)	it	t(s)	g(%)	it	t(s)	g(%)	it	t(s)	g(%)	it	t(s)
g1-36-6-4	2,5	3	170,6	2,5	3	60,6	2,6	1,9	388,0	2,5	2	71,8
g1-48-6-4	1,4	3	325,2	1,4	3	89,7	1,4	1,9	538,0	1,3	2	110,5
g1-72-6-6	1,0	5	424,1	1,1	5	239,5	1,3	1,9	599,9	0,6	4	236,5
g1-90-6-6	2,2	3	520,7	1,0	5	283,8	2,0	1,5	591,2	0,9	4	238,2
g1-120-8-8	2,0	6	439,6	1,8	7	431,5	2,6	2,8	600,0	1,8	6	370,1
g1-144-8-8	1,9	6	476,5	1,6	7	437,8	2,2	2,4	600,0	2,0	6	376,0
g2-36-6-4	1,9	3	87,7	1,8	3	32,6	1,5	2,0	262,8	1,5	2	53,2
g2-48-6-4	1,1	3	166,4	1,1	3	94,5	1,1	1,9	462,4	1,1	2	102,2
g2-72-6-6	1,5	5	218,6	1,0	5	222,7	1,4	2,7	523,0	0,7	4	214,4
g2-90-6-6	1,5	4	427,5	0,9	5	267,0	1,1	2,4	571,7	0,8	4	237,8
g2-120-8-8	1,5	6	387,7	1,0	7	406,8	1,9	4,3	554,3	1,1	6	369,1
g2-144-8-8	1,1	6	402,8	0,9	7	430,9	2,2	3,2	600,0	1,0	6	375,3
g3-36-6-4	2,2	3	71,0	2,2	3	36,4	2,2	2,0	180,6	2,2	2	55,1
g3-48-6-4	0,9	3	99,4	0,9	3	57,6	0,9	2,0	341,5	0,9	2	98,2
g3-72-6-6	0,9	5	237,3	1,0	5	197,7	1,1	2,7	479,9	0,8	4	236,4
g3-90-6-6	1,0	4	355,8	0,7	5	259,1	1,4	2,6	501,8	0,5	4	235,5
g3-120-8-8	1,1	7	263,2	1,1	7	405,1	1,2	5,0	523,9	1,2	6	368,9
g3-144-8-8	1,1	7	336,2	1,2	7	430,2	1,6	4,8	580,9	1,2	6	374,8
g4-36-6-4	1,8	3	26,3	1,8	3	30,2	1,7	2,0	223,8	1,7	2	73,8
g4-48-6-4	0,9	3	239,9	0,7	3	103,3	0,6	1,7	536,3	0,6	2	113,9
g4-72-6-6	0,9	5	353,5	0,8	5	237,7	1,0	2,2	457,8	0,3	4	240
g4-90-6-6	1,8	3	554,7	1,3	5	290,6	1,7	2,0	600,0	1,1	4	241,3
g4-120-8-8	1,3	6	409,3	1,2	7	419,8	1,5	3,0	600,0	1,4	6	370,2
g4-144-8-8	1,6	7	442,0	1,4	7	438,6	1,8	4,3	600,0	1,6	6	375,2
g5-36-6-4	2,5	3	158,6	2,5	3	62,0	2,6	1,9	452,6	2,5	2	96,4
g5-48-6-4	1,8	3	289,2	1,8	3	96,7	1,8	2,0	599,7	1,8	2	112,0
g5-72-6-6	0,9	5	359,7	0,8	5	251,5	1,4	2,5	548,2	0,6	4	237,1
g5-90-6-6	1,6	4	504,9	1,0	5	302,8	1,6	1,8	595,4	1,2	4	242,3
g5-120-8-8	2,8	4	556,4	1,8	7	431,8	2,8	2,1	600,0	2,0	6	370,1
g5-144-8-8	2,4	5	513,4	1,8	7	438,3	2,5	2,4	576,6	2,3	6	375,3
g6-36-6-4	1,9	3	104,4	1,9	3	36,6	1,5	2,0	337,4	1,5	2	58,5
g6-48-6-4	1,7	3	170,9	1,4	3	85,3	1,4	1,9	466,3	1,4	2	101,7
g6-72-6-6	1,2	4	273,6	0,7	5	196,9	1,3	2,9	460,9	0,5	4	232,8
g6-90-6-6	1,9	3	483,8	1,0	5	255,9	1,6	1,8	600,0	0,7	4	234,9
g6-120-8-8	1,2	6	330,6	1,0	7	424,3	1,5	3,6	600,0	1,2	6	370,7
g6-144-8-8	1,4	6	435,5	1,2	7	421,7	2,9	2,6	600,0	1,2	6	375,9
Média	1,6			1,3			1,7			1,3		

Tabela 7 – heurística BLN-OPT e heurística BPLS para o IULSBP

Nome	BLN-OPT			BPLS		
	g(%)	it	t(s)	g(%)	it	t(s)
g1-36-6-4	1,7	6	177,0	0,5	1,0	1,1
g1-48-6-4	1,2	6	257,5	0,7	1,6	1,3
g1-72-6-6	0,5	11	599,9	0,9	3,3	7,6
g1-90-6-6	0,5	11	599,6	1,4	3,0	8,9
g1-120-8-8	1,6	10	600,0	2,0	4,4	47,9
g1-144-8-8	1,5	10	600,0	1,7	5,1	58,6
g2-36-6-4	1,1	6	131,9	0,6	1,0	1,1
g2-48-6-4	1,1	6	257,9	0,7	1,9	1,3
g2-72-6-6	0,7	12	596,3	1,3	3,2	7,6
g2-90-6-6	0,8	11	599,9	1,2	3,9	9,0
g2-120-8-8	1,0	11	600,0	1,5	4,7	48,6
g2-144-8-8	1,0	10	600,0	1,0	5,0	57,8
g3-36-6-4	1,9	6	141,2	0,6	1,0	1,1
g3-48-6-4	0,9	6	205,8	0,7	2,1	1,2
g3-72-6-6	0,5	12	599,7	0,9	4,0	7,6
g3-90-6-6	0,4	11	599,8	0,9	3,4	9,1
g3-120-8-8	0,9	11	600,0	1,5	4,7	49,1
g3-144-8-8	1,1	10	600,0	1,2	4,8	58,2
g4-36-6-4	1,3	6	146,1	0,8	1,0	1,3
g4-48-6-4	0,4	6	282,8	0,8	2,2	1,3
g4-72-6-6	0,4	12	599,8	1,0	3,6	8,0
g4-90-6-6	1,0	10	600,0	1,8	4,0	9,2
g4-120-8-8	1,2	10	600,0	1,4	4,6	48,9
g4-144-8-8	1,6	10	600,0	1,6	5,4	60,2
g5-36-6-4	2,1	6	191,6	0,4	1,0	1,1
g5-48-6-4	1,8	6	269,4	0,8	1,3	1,3
g5-72-6-6	0,5	12	599,8	1,3	3,2	7,5
g5-90-6-6	0,9	10	600,0	1,2	3,0	8,9
g5-120-8-8	1,8	10	600,0	2,0	4,3	48,1
g5-144-8-8	1,6	10	600,0	2,1	4,9	58,5
g6-36-6-4	1,1	6	156,4	1,0	1,0	1,1
g6-48-6-4	1,1	6	249,8	0,8	1,8	1,3
g6-72-6-6	0,2	13	599,8	0,7	3,2	7,5
g6-90-6-6	0,7	11	600,0	1,2	4,1	8,9
g6-120-8-8	1,0	10	600,0	1,5	4,6	47,8
g6-144-8-8	1,1	10	600,0	1,5	5,4	59,2
Média	1,1			1,1		

médios semelhantes aos presentes nas Tabelas 6 e 7, e por isso não são apresentados.

5 Heurísticas Estocásticas para IULSBP

Neste capítulo são avaliadas heurísticas estocásticas para a solução do IULSBP. Apresenta-se inicialmente na Seção 5.1 uma heurística multipartida chamada *First Fit Ordenada pelo Limite de Tempo - FFOLT* em que resolve-se o problema de Empacotamento por um algoritmo *First Fit*. Em seguida, na Seção 5.2, tem-se uma heurística multipartida baseada na heurística LSBP, chamada MS-LSBP. Finalmente, na Seção 5.3, tem-se uma heurística que utiliza um Algoritmo Genético de Chaves Aleatórias Baesyano (GONÇALVES; RESENDE, 2011). Esta heurística também baseia-se na heurística LSBP e é chamada BRKGA-LSBP. Todas as heurísticas utilizam o algoritmo de Programação Dinâmica WW (WAGNER; WHITIN, 1958) para resolver o problema de Dimensionamento de Lotes. As heurísticas MS-LBSP e BRKGA-LSBP utilizam o algoritmo VPsolver (BRANDAO; PEDROSO, 2016) para resolver o problema de Empacotamento.

5.1 Heurística *First Fit - FFOLT*

FFOLT é uma heurística multipartida que tem como critério de parada o tempo de execução. Ela se inicia agrupando os pedidos dos clientes de acordo com o tempo limite de atendimento e os empacotando através de um algoritmo *First Fit*. No algoritmo *First Fit* para o problema de Empacotamento, dada uma lista com os elementos que serão empacotados, retira-se um item da lista e o insere no primeiro bin que contenha espaço disponível. Caso não exista um bin com espaço suficiente para inserir o item, um novo bin vazio é adicionado na solução e o item é inserido. Após empacotar todos os pedidos, resolve-se o problema de Dimensionamento de Lotes.

Como exemplo, com o agrupamento dos pedidos, pode-se iniciar o empacotamento pelos pedidos com limite de atendimento no período $t = 1$, posteriormente, os com limite de atendimento no período $t = 2$, até que todos os pedidos sejam empacotados. Como diferentes ordens no empacotamento dos pedidos com mesmo período limite de atendimento podem

levar a diferentes soluções, a heurística foi executada em um procedimento multipartida. Assim, em cada execução, define-se a ordem dos pedidos com um mesmo período limite de atendimento através de uma escolha pseudoaleatória.

Após a solução do problema de Empacotamento para todos os pedidos, o tempo limite de atendimento de cada pedido precisa ser verificado, sendo adicionado um novo tempo limite de atendimento, t_i^b , utilizado para definir a data limite de compra dos produtos. Verifica-se para todos os pedidos contidos em um bin, qual tem o tempo limite de atendimento mais próximo do período inicial, aqui definido como t^f . Todos os pedidos que estão neste bin terão o tempo limite t_i^b alterados para o valor t^f . Os custos de estoque são calculados considerando o período limite original de cada pedido. Após verificar todos os bins, resolve-se um problema de Dimensionamento de Lotes considerando como tempo limite para a compra dos produtos, os períodos t_i^b . A ordenação dos pedidos tem por objetivo reduzir as diferenças dos tempos de atendimento dos pedidos dentro de um mesmo bin. Um bin contendo pedidos com tempo de limite de atendimento diferentes pode incorrer em um alto custo de estoque no problema de Dimensionamento de Lotes.

Durante o processo iterativo multipartida, sempre que ocorre uma melhora no custo da solução do problema integrado, realiza-se uma análise verificando se é possível adiantar os períodos em que os pedidos foram empacotados. Para isso, após a solução do problema de Empacotamento e do problema de Dimensionamento de Lotes, definida como Fase1, verifica-se quando um determinado pedido foi empacotado e quando ocorreu a compra para atendê-lo. Se houver uma janela de atendimento que permita a escolha de um melhor custo de empacotamento para todos os pedidos presentes em um bin, a troca do período de empacotamento é realizada. A melhora obtida na troca dos períodos de empacotamento é chamada de Fase2. Como o custo de estoque dos produtos são contabilizados desde a compra até o tempo limite de atendimento, não se pode fazer nenhuma alteração sobre ele.

Sempre que há uma melhora de solução na Fase1, avalia-se a possibilidade de melhorar os custos de empacotamento na Fase2, mesmo que a solução da Fase1 seja inferior a melhor solução conhecida. Assim, evita-se que soluções da Fase1 próximas, mas piores do que a melhor solução conhecida e que podem ser melhoradas, sejam descartadas.

5.2 Heurística MS-LSBP

Esta seção apresenta uma heurística multipartida chamada MS-LSBP que baseia-se na heurística *LSBP*. A cada execução dessa heurística gera-se uma perturbação na solução do problema de Dimensionamento de Lotes, de forma a avaliar o comportamento da solução do problema de Empacotamento. Como soluções de pior qualidade para o problema de Dimensionamento de Lotes podem gerar soluções de boa qualidade para o problema de Empacotamento, estas perturbações podem gerar soluções de melhor qualidade para o problema integrado.

Resolver o problema integrado em duas fases, solucionando inicialmente o problema de Dimensionamento de Lotes de forma exata, pode apresentar uma solução distante da melhor solução do problema original. A distância entre os valores da solução em duas fases e a solução integrada poderá variar em função da diferença entre os valores de custo dos dois problemas e também devido a qual problema é solucionado primeiro. Se, por exemplo, os custos do problema de Dimensionamento de Lotes são significativamente inferiores aos custos do problema de Empacotamento, resolver o problema de Dimensionamento de Lotes primeiro sem considerar as decisões de empacotamento, deve gerar uma solução para o problema integrado de baixa qualidade.

Verifica-se que os valores de custo do problema de Dimensionamento de Lotes na solução integrada que seja ótima são sempre maiores ou iguais aos valores para o mesmo problema na solução independente. E, baseado no fato de que diferentes soluções do problema de Dimensionamento de Lotes podem representar redução dos custos do problema de Empacotamento, optou-se por avaliar procedimentos em que os custos de estoque e compra tiveram seus valores originais alterados através de perturbações. A amplitude destas perturbações é dada pela escolha de valores que são multiplicados por cada elemento das matrizes de custos. A utilização destas perturbações baseia-se nos trabalhos de (CHARON; HUDRY, 1993; CHARON; HUDRY, 2000) que apresentaram o *Noising Method*.

Seja α um parâmetro que define a amplitude máxima de perturbação que aplica-se

aos custos do problema de Dimensionamento de Lotes, gerando a perturbação. A heurística MS-LBSP utiliza valores escolhidos de forma pseudoaleatória no intervalo $[1 - \alpha; 1 + \alpha]$. A heurística tem como critério de parada o tempo limite de execução e promove uma intensificação ou redução da diversificação do espaço de busca, alterando o valor de α , caso ocorra um determinado número de execuções sem melhora da solução integrada.

5.3 Algoritmo Genético BRKGA-LSBP

Os métodos multipartida descritos nas Seções 5.1 e 5.2 não utilizam nenhuma estrutura de memória com informações sobre a qualidade das soluções encontradas ao longo das iterações. Estes métodos apresentaram bons resultados para o IULSBP e motivado por estes resultados, decidiu-se avaliar o uso de um Algoritmo Genético na busca por soluções de melhor qualidade. O algoritmo utilizado é o Algoritmo Genético com Chaves Aleatórias Bayesiano, do inglês *Biased Random-Key Genetic Algorithm (BRKGA)* (GONÇALVES; RESENDE, 2011).

Na implementação do BRKGA neste trabalho, inicialmente utilizou-se um método baseado na heurística FFOLT, Seção 5.1, como *decoder*. Nesta abordagem, as chaves dos cromossomos do algoritmo genético definem a ordem em que o algoritmo *First Fit* realiza o empacotamento dos pedidos. Em experimentos preliminares, a solução fornecida por este método apresentou resultados de baixa qualidade e o algoritmo genético não convergiu de forma satisfatória. Desta forma, este método não foi utilizado em testes em todo o conjunto de instâncias.

Em seguida, avaliou-se utilizar o algoritmo VPsolver para minimizar o número de conflitos em um problema de Empacotamento com Conflitos (KHANAFER et al., 2012; SADYKOV; VANDERBECK, 2013). Neste caso foi definido como conflito o empacotamento de dois pedidos em um mesmo bin se a diferença dos tempos limites de atendimento fosse superior a um parâmetro τ . Porém, na implementação não se penalizava todos os conflitos e, através das chaves dos cromossomos do algoritmo genético, definia-se quais conflitos seriam penalizados na função de custo. O objetivo é que o algoritmo genético forneça uma

solução do problema integrado, resolvendo inicialmente o problema de Empacotamento e, posteriormente, a solução do problema de Dimensionamento de Lotes seria obtida pelo algoritmo WW (WAGNER; WHITIN, 1958). Testes preliminares permitiram concluir que o algoritmo VPsolver não é tão eficiente na solução do problema de Empacotamento com Conflitos, como para resolver o problema de Empacotamento clássico.

Finalmente, utilizou-se o algoritmo genético BRKGA-LSBP para definir a perturbação dos vetores de custo do problema de Dimensionamento de Lotes na heurística LSBP, em substituição ao *Noising Method* utilizado na heurística multiparida MS-LSBP. Utilizar a heurística LSBP como *decoder* apresentou os melhores resultados em testes preliminares e por isso ela foi utilizada nos testes da próxima seção. Nesta estratégia cada chave do cromossomo altera os valores de custo do problema de Dimensionamento de Lotes, gerando a perturbação dos custos.

5.4 Experimentos Computacionais das Heurísticas Estocásticas

Nesta seção apresentam-se os experimentos com as heurísticas FFOLT, MS-LSBP e o algoritmo genético BRKGA-LSBP. Os objetivos destes experimentos são avaliar a qualidade dos limites superiores fornecidos por cada heurística e os correspondentes tempos de execução. Novamente, os resultados são exibidos como um intervalo de otimalidade $([ub - lb^*]/ub)$ entre os limites superiores ub obtidos por cada heurística e o melhor valor lb^* do limite inferior conhecido, considerando cada instância (encontrado por qualquer uma das abordagens exibidas na Tabela 3). Os parâmetros e o ambiente de testes, bem como as instâncias utilizadas são os mesmos descritos na Seção 3.4.

No primeiro experimento desta seção, apresenta-se na Tabela 8 os resultados das heurísticas multipartida FFOLT e MS-LBSP. Nesta tabela, na primeira coluna tem-se o nome das instâncias. Da segunda a quarta colunas têm-se os gaps, o tempo de execução e número de iterações executadas pela heurística FFOLT. Da quinta a oitava colunas têm-se os gaps, o tempo de execução, o número de iterações do algoritmo em que o problema de Empacotamento foi resolvido e o número de iterações para o problema de

Dimensionamento de Lotes para a heurística MS-LBSP. Este algoritmo apenas resolve o problema de Empacotamento para soluções diferentes do problema de Dimensionamento de Lotes.

Ambas as heurísticas foram executadas pelo tempo limite de 600 segundos, sendo que para algumas instâncias, a heurística MS-LBSP foi encerrada pelo número de iterações sem melhora da solução, definido com 50 iterações. Pelos resultados, observa-se que a heurística MS-LBSP apresentou melhores resultados de gap (1,3%) do que a heurística FFOLT (2,4%). A ordenação dos pedidos por período utilizada no algoritmo *First Fit* tem o objetivo de reduzir a diferença de tempos de atendimento dentro de um bin e a consequente redução dos custos de estoque. Porém, isso não tornou o método mais eficiente.

No segundo experimento desta seção, Tabela 9, tem-se os resultados de uma execução da heurística MS-LBSP com duas execuções do algoritmo genético BRKGA-LSBP. Para o algoritmo genético, realizou-se experimentos com tamanhos de população de 30 e 100 indivíduos. A primeira coluna da tabela contém o nome das instâncias. Da segunda a quinta têm-se os gaps, o tempo de execução, o número de iterações do algoritmo em que o problema de Empacotamento foi resolvido e o número de iterações para o problema de Dimensionamento de Lotes para a heurística MS-LBSP. Na sexta e sétima colunas têm-se o gap e o número de gerações do algoritmo genético com tamanho de população igual a 100 indivíduos. Nas colunas oitava e nona têm-se as mesmas informações para o algoritmo genético com tamanho de população igual a 30 indivíduos. Todas as execuções dos algoritmos genéticos foram encerradas pelo tempo de execução de 600 segundos. Uma geração implica em resolver um problema integrado para cada elemento da população. Para o algoritmo genético BRKGA neste experimento, o tamanho das parcelas da população *TOP* e *BOT* foram definidas como 70% e 25% respectivamente. A probabilidade de que um indivíduo da população mais apta seja escolhido em um cruzamento é de 70%. Esses são os valores usuais de parâmetros utilizados por autores que citam o trabalho (GONÇALVES; RESENDE, 2011).

Pelos resultados, verifica-se que o algoritmo genético apresentou valores de gaps ligeiramente melhores (1,1%) do que os gaps da heurística MS-LBSP (1,3%). Porém,

Tabela 8 – Resultados dos experimentos com as heurísticas FFOLT e MS-LSBP.

Nome	FFOLT			MS-LSBP			
	g(%)	t(s)	it	g(%)	t(s)	it-E	it-DL
g1-36-6-4	3,7	600,0	9157555	0,8	300,9	28	2598294
g1-48-6-4	1,5	600,0	5737183	1,2	512,5	206	2010821
g1-72-6-6	1,6	600,0	2914260	0,8	600,0	207	18341
g1-90-6-6	1,7	600,0	2002308	1,3	600,0	224	14151
g1-120-8-8	2,4	600,0	1195063	2,0	600,0	60	891
g1-144-8-8	2,7	600,0	728690	2,0	600,0	52	806
g2-36-6-4	2,8	600,0	9177297	0,2	120,7	22	984807
g2-48-6-4	1,6	600,0	5724549	0,5	455,7	66	2729192
g2-72-6-6	2,0	600,0	2910609	1,2	564,3	134	26753
g2-90-6-6	2,4	600,0	2004796	1,7	600,0	118	159294
g2-120-8-8	2,6	600,0	1195371	1,2	600,0	65	940
g2-144-8-8	3,6	600,0	727473	2,0	600,0	45	691
g3-36-6-4	3,5	600,0	8411000	0,3	121,6	16	969466
g3-48-6-4	1,1	600,0	5728911	0,3	360,6	65	2536240
g3-72-6-6	1,8	600,0	2940826	1,0	600,0	182	27671
g3-90-6-6	1,7	600,0	2008390	1,1	600,0	159	109274
g3-120-8-8	3,0	600,0	1195345	2,1	600,0	73	1368
g3-144-8-8	3,2	600,0	723862	1,8	600,0	42	950
g4-36-6-4	2,9	600,0	8392027	1,6	420,2	54	2817194
g4-48-6-4	0,7	600,0	5745468	0,8	548,0	161	2027996
g4-72-6-6	1,1	600,0	2925323	1,1	600,0	201	479945
g4-90-6-6	1,5	600,0	2003202	1,4	600,0	156	30714
g4-120-8-8	2,0	600,0	1197355	1,4	600,0	145	1655
g4-144-8-8	2,4	600,0	726068	1,7	600,0	33	421
g5-36-6-4	3,6	600,0	5263531	0,8	480,1	115	2426546
g5-48-6-4	1,9	600,0	3556022	1,5	540,6	317	2430236
g5-72-6-6	1,8	600,0	2102243	1,2	600,0	274	281803
g5-90-6-6	2,0	600,0	1528026	1,3	600,0	146	16415
g5-120-8-8	2,6	600,0	951989	2,2	600,0	48	552
g5-144-8-8	3,0	600,0	619951	2,0	600,0	35	644
g6-36-6-4	3,8	600,0	5273189	0,5	241,5	57	1638101
g6-48-6-4	2,4	600,0	3515529	0,9	372,0	34	1728057
g6-72-6-6	2,4	600,0	2098254	0,5	540,0	189	577988
g6-90-6-6	2,6	600,0	1523397	1,4	600,0	141	155562
g6-120-8-8	3,3	600,0	953151	2,3	600,0	55	850
g6-144-8-8	3,7	600,0	621086	2,0	600,0	62	866
Média:	2,4			1,3			

mesmo com várias tentativas de se tornar cada solução do *decoder* LSBP mais eficiente, um número reduzido de gerações foram possíveis dentro do tempo limite de execução.

Finalmente, compara-se os resultados da melhor heurística determinística (BLN-OPT) com a melhor heurística estocástica (BRKGA-LSBP) com tamanho de população igual a 30 indivíduos. A primeira coluna da Tabela 10 apresenta o nome das instâncias. Das colunas segunda a quarta têm-se o gap, o tempo de execução e o número de iterações para a heurística BLN-OPT . Na quinta e sexta colunas têm-se o gap e o número de gerações do algoritmo genético cujos parâmetros geraram os melhores resultados.

Observa-se que os resultados dos gaps médios, última linha da tabela, são iguais para os dois métodos. A heurística BLN-OPT encerra sua execução para algumas instâncias em tempo inferior a 600 segundos e se for considerado apenas as instâncias com 90, 120 e 144 pedidos, o gap da heurística BLN-OPT é de 1,1%, enquanto o do algoritmo genético é de 1,6%.

Tabela 9 – Resultados da heurística MS-LBSP com duas execuções do algoritmo genético BRKGA-LSBP.

Nome	MS-LBSP				BRKGA-LSBP pop 100		BRKGA-LSBP pop 30	
	g(%)	t(s)	it_DL	it_E	g(%)	it	g(%)	it
g1-36-6-4	0,8	300,9	2598294	28	0,8	66	0,8	227
g1-48-6-4	1,2	512,5	2010821	206	1,3	53	1,1	179
g1-72-6-6	0,8	600,0	18341	207	1,0	35	1,0	117
g1-90-6-6	1,3	600,0	14151	224	1,6	31	1,6	103
g1-120-8-8	2,0	600,0	891	60	2,4	22	2,4	76
g1-144-8-8	2,0	600,0	806	52	2,0	19	2,0	66
g2-36-6-4	0,2	120,7	984807	22	0,2	73	0,2	238
g2-48-6-4	0,5	455,7	2729192	66	0,4	60	0,4	202
g2-72-6-6	1,2	564,3	26753	134	1,4	37	1,3	122
g2-90-6-6	1,7	600,0	159294	118	1,2	33	1,2	108
g2-120-8-8	1,2	600,0	940	65	1,6	24	1,6	84
g2-144-8-8	2,0	600,0	691	45	1,5	21	1,3	70
g3-36-6-4	0,3	121,6	969466	16	0,2	69	0,2	243
g3-48-6-4	0,3	360,6	2536240	65	0,5	59	0,3	205
g3-72-6-6	1,0	600,0	27671	182	0,9	39	0,9	134
g3-90-6-6	1,1	600,0	109274	159	0,9	32	0,9	114
g3-120-8-8	2,1	600,0	1368	73	1,9	23	1,8	82
g3-144-8-8	1,8	600,0	950	42	1,5	20	1,4	69
g4-36-6-4	1,6	420,2	2817194	54	1,7	65	1,7	231
g4-48-6-4	0,8	548,0	2027996	161	1,2	53	1,2	184
g4-72-6-6	1,1	600,0	479945	201	1,2	37	1,2	128
g4-90-6-6	1,4	600,0	30714	156	1,9	30	1,9	106
g4-120-8-8	1,4	600,0	1655	145	1,5	22	1,4	78
g4-144-8-8	1,7	600,0	421	33	1,7	19	1,6	68
g5-36-6-4	0,8	480,1	2426546	115	0,8	64	0,8	226
g5-48-6-4	1,5	540,6	2430236	317	1,3	54	1,3	187
g5-72-6-6	1,2	600,0	281803	274	1,6	36	1,5	124
g5-90-6-6	1,3	600,0	16415	146	1,4	32	1,5	109
g5-120-8-8	2,2	600,0	552	48	2,4	22	2,2	74
g5-144-8-8	2,0	600,0	644	35	2,6	19	2,4	68
g6-36-6-4	0,5	241,5	1638101	57	0,5	66	0,5	194
g6-48-6-4	0,9	372,0	1728057	34	0,9	59	0,8	169
g6-72-6-6	0,5	540,0	577988	189	0,8	38	0,7	106
g6-90-6-6	1,4	600,0	155562	141	1,2	34	1,2	96
g6-120-8-8	2,3	600,0	850	55	2,0	23	1,7	69
g6-144-8-8	2,0	600,0	866	62	1,8	19	1,6	59
	1,3				1,2		1,1	

Tabela 10 – Resultados da melhor heurística determinística (BLN-OPT) e da melhor heurística estocástica (BRKGA-LSBP).

Nome	BLN-OPT			BRKGA-LSBP		
	g(%)	t(s)	it	g(%)	t(s)	it
g1-36-6-4	1,7	177,0	6	0,8	600,0	227
g1-48-6-4	1,2	257,5	6	1,1	600,0	179
g1-72-6-6	0,5	599,9	11	1,0	600,0	117
g1-90-6-6	0,5	599,6	11	1,6	600,0	103
g1-120-8-8	1,6	600,0	10	2,4	600,0	76
g1-144-8-8	1,5	600,0	10	2,0	600,0	66
g2-36-6-4	1,1	131,9	6	0,2	600,0	238
g2-48-6-4	1,1	257,9	6	0,4	600,0	202
g2-72-6-6	0,7	596,3	12	1,3	600,0	122
g2-90-6-6	0,8	599,9	11	1,2	600,0	108
g2-120-8-8	1,0	600,0	11	1,6	600,0	84
g2-144-8-8	1,0	600,0	10	1,3	600,0	70
g3-36-6-4	1,9	141,2	6	0,2	600,0	243
g3-48-6-4	0,9	205,8	6	0,3	600,0	205
g3-72-6-6	0,5	599,7	12	0,9	600,0	134
g3-90-6-6	0,4	599,8	11	0,9	600,0	114
g3-120-8-8	0,9	600,0	11	1,8	600,0	82
g3-144-8-8	1,1	600,0	10	1,4	600,0	69
g4-36-6-4	1,3	146,1	6	1,7	600,0	231
g4-48-6-4	0,4	282,8	6	1,2	600,0	184
g4-72-6-6	0,4	599,8	12	1,2	600,0	128
g4-90-6-6	1,0	600,0	10	1,9	600,0	106
g4-120-8-8	1,2	600,0	10	1,4	600,0	78
g46-144-8-8	1,6	600,0	10	1,6	600,0	68
g5-36-6-4	2,1	191,6	6	0,8	600,0	226
g5-48-6-4	1,8	269,4	6	1,3	600,0	187
g5-72-6-6	0,5	599,8	12	1,5	600,0	124
g5-90-6-6	0,9	600,0	10	1,5	600,0	109
g5-120-8-8	1,8	600,0	10	2,2	600,0	74
g5-144-8-8	1,6	600,0	10	2,4	600,0	68
g6-36-6-4	1,1	156,4	6	0,5	600,0	194
g6-48-6-4	1,1	249,8	6	0,8	600,0	169
g6-72-6-6	0,2	599,8	13	0,7	600,0	106
g6-90-6-6	0,7	600,0	11	1,2	600,0	96
g6-120-8-8	1,0	600,0	10	1,7	600,0	69
g6-144-8-8	1,1	600,0	10	1,6	600,0	59
Média:	1,1			1,1		

6 Considerações Finais

Neste trabalho foram apresentadas e discutidas técnicas de solução para um problema integrado que considera decisões de compra e estocagem de produtos, bem como quando e como realizar o empacotamento de pedidos de clientes os quais estão condicionados a um determinado período limite de atendimento. Foram propostas formulações por Programação Linear Inteira, formulações relaxadas com o objetivo de fornecer limites inferiores para o problema e métodos heurísticos para determinação de limites superiores. Este problema foi definido como o Problema Integrado de Dimensionamento de Lotes e de Empacotamento de Produtos (do inglês *Integrated Uncapacitated Lot Sizing and Bin Packing problem - IULSBP*).

Inicialmente, realizou-se uma revisão bibliográfica com trabalhos relacionados ao problema desta tese. São apresentados trabalhos com diferentes versões: (i) do Problema de Dimensionamento de Lotes; (ii) De problemas de Empacotamento e; (iii) De problemas que integram decisões de compra, estoque, alocação de produtos em paletes e que consideram decisões de roteamento. Na última parte da revisão bibliográfica, apresentam-se trabalhos utilizados nas heurísticas desta tese, como por exemplo, algoritmos multipartida e um algoritmo genético.

Em seguida foram apresentadas duas formulações por PLI e duas formulações relaxadas, as quais definimos com formulações por Relaxação Combinatória. Nos experimentos realizados com as formulações por PLI, apenas as menores instâncias são resolvidas na otimalidade. Os gaps fornecidos pelas relaxações combinatórias, comparados com os melhores limites superiores das melhores heurísticas, foram da ordem de 1%. Baseado nos bons resultados das relaxações combinatórias, aliadas aos tempos computacionais inferiores a um segundo, desolveu-se uma heurística chamada CRH, do inglês *Combinatorial Relaxation Heuristic*.

Um grupo de heurísticas determinísticas foi apresentado. Estas heurísticas foram desenvolvidas resolvendo inicialmente o problema de Dimensionamento de Lotes e

posteriormente o problema de Empacotamento, resolvendo inicialmente o problema de Empacotamento e heurísticas baseadas nas formulações por PLI. Dentre este grupo de heurísticas, destacam-se a heurística de busca local BLN-OPT e a que inicia resolvendo o problema de Empacotamento, BPLS. Estas heurísticas forneceram gaps de otimalidade da ordem de 1%.

Finalmente, heurísticas estocásticas implementadas através de algoritmos multipartida e um algoritmo genético foram avaliadas. Dentre este grupo de heurísticas, os melhores resultados foram obtidos por uma heurística multipartida baseada na heurística LSBP, MS-LSBP e um algoritmo genético de chaves aleatórias, BRKGA-LSBP. Para o grupo de instâncias avaliadas neste trabalho, foi possível concluir que as heurísticas que são resolvidas considerando os problema de Dimensionamento de Lotes e de Empacotamento de forma colaborativa e integrada, apresentam resultados superiores aos da heurística que resolve o problema integrado, tratando cada subproblema de forma independente. Os valores de gaps médios destes dois grupos de heurísticas foram de cerca de 1% (heurísticas BPLS, BLN-OPT, MS-LSBP e BRKGA-LSBP) e 8% (heurística UH), respectivamente.

Diante dos resultados promissores obtidos com este estudo, pretende-se estender a pesquisa para problemas integrados que considere: (i) Restrições de capacidade no problema de Dimensionamento de Lotes; (ii) Problemas de empacotamento com bins heterogêneos; e (iii) Integração com problemas de Roteamento de Veículos.

Referências

- ADULYASAK, Y.; CORDEAU, J.; JANS, R. Benders decomposition for production routing under demand uncertainty. *Operations Research*, v. 63, n. 4, p. 851–867, 2015.
- ADULYASAK, Y.; CORDEAU, J.; JANS, R. The production routing problem: A review of formulations and solution algorithms. *Computers & Operations Research*, v. 55, p. 141–152, 2015.
- AGGARWAL, A.; PARK, J. K. Improved algorithms for economic lot size problems. *Operations Research*, v. 41, n. 3, p. 549–571, 1993.
- AKBALIK, A.; PENZ, B.; RAPINE, C. Capacitated lot sizing problems with inventory bounds. *Annals of Operations Research*, v. 229, p. 1–18, 2015.
- ARMENTANO, V.; SHIGUEMOTO, A.; LOKKETANGEN, A. Tabu search with path relinking for an integrated production distribution problem. *Computers & Operations Research*, v. 38, p. 1199–1209, 2011.
- AZADEH, A. et al. A genetic algorithm-taguchi based approach to inventory routing problem of a single perishable product with transshipment. *Computers & Industrial Engineering*, v. 104, p. 124–133, 2017.
- BARD, J. F.; NANANUKUL, N. The integrated production-inventory-distribution-routing problem. *Journal of Scheduling*, Springer US, v. 12, n. 3, p. 257–280, 2009.
- BARD, J. F.; NANANUKUL, N. A branch-and-price algorithm for an integrated production and inventory routing problem. *Computers & Operations Research*, v. 37, n. 12, p. 2202–2217, 2010.
- BEN-KHEDHER, N.; YANO, C. A. The multi-item joint replenishment problem with transportation and container effects. *Transportation Science*, INFORMS, v. 28, n. 1, p. 37–54, 1994.
- BERTAZZI, L.; PALETTA, G.; SPERANZA, M. Minimizing the total cost in an integrated vendor-managed inventory system. *Journal of Heuristics*, Kluwer Academic Publishers, v. 11, n. 5-6, p. 393–419, 2005.
- BRANDAO, F. *Vector Packing Solver (VPSolver)*. 2017. Disponível em: <<http://vpsolver.dcc.fc.up.pt/>>.
- BRANDAO, F.; PEDROSO, J. P. Bin packing and related problems: General arc-flow formulation with graph compression. *Computers & Operations Research*, v. 69, p. 56–67, 2016.
- BURIOL, L. S. et al. A biased random-key genetic algorithm for road congestion minimization. *Optimization Letters*, v. 4, p. 619–633, 2010.
- CARVALHO, J. M. V. Exact solution of bin-packing problems using column generation and branch-and-bound. *Annals of Operations Research*, v. 86, p. 629–659, 1999.

- CARVALHO, J. M. V. *LP models for bin packing and cutting stock problems. European Journal of Operational Research*, v. 141, n. 2, p. 253–273, 2002.
- CHAN, L. M. A. et al. On the effectiveness of zero-inventory-ordering policies for the economic lot-sizing model with a class of piecewise linear cost structures. *Operations Research*, v. 50, n. 6, p. 1058–1067, 2002.
- CHAN, L. M. A. et al. Effective zero-inventory-ordering policies for the single-warehouse multiretailer problem with piecewise linear cost structures. *Management Science*, v. 48, n. 11, p. 1446–1460, 2002.
- CHARON, I.; HUDRY, O. The noising method: a new method for combinatorial optimization. *Operations Research Letters*, v. 14, n. 3, p. 133–137, 1993.
- CHARON, I.; HUDRY, O. Application of the noising method to the travelling salesman problem. *European Journal of Operational Research*, v. 125, n. 2, p. 266–277, 2000.
- CHOWDHURY, N. T.; BAKI, M.; AZAB, A. Dynamic economic lot-sizing problem: A new $o(t)$ algorithm for the wagner-whitin model. *Computers & Industrial Engineering*, v. 117, p. 6–18, 2018.
- COELHO, L. C.; CORDEAU, J.; LAPORTE, G. Thirty years of inventory routing. *Transportation Science*, v. 48, n. 1, p. 1–19, 2014.
- COELHO, L. C.; LAPORTE, G. Optimal joint replenishment, delivery and inventory management policies for perishable products. *Computers & Operations Research*, v. 47, p. 42–52, 2014.
- CORDEAU, J. et al. A decomposition-based heuristic for the multiple-product inventory-routing problem. *Computers & Operations Research*, v. 55, p. 153–166, 2015.
- DELORME, M.; IORI, M.; MARTELLO, S. Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*, v. 255, p. 1–20, 2016.
- DREXL, A.; KIMMS, A. Lot sizing and scheduling - survey and extensions. *European Journal of Operational Research*, v. 99, n. 2, p. 221–235, 1997.
- ERTOGRAL, K. Multi-item single source ordering problem with transportation cost: A lagrangian decomposition approach. *European Journal of Operational Research*, v. 191, p. 156–165, 2008.
- FALKENAUER, E. A hybrid grouping genetic algorithm for bin packing. *Journal of Heuristics*, v. 2, n. 1, p. 5–30, 1996.
- FATHABADI, H. S.; KHODAYIFAR, S.; RAYYATPANAH, M. Minimum flow problem on network flows with time-varying bounds. *Applied Mathematical Modelling*, v. 36, n. 9, p. 4414–4421, 2012.
- GLOVER, F.; KOCHENBERGER, G. *Handbook of Metaheuristics*. [S.l.]: Springer US, 2003. (International Series in Operations Research & Management Science).
- GLOVER, F.; LAGUNA, M. *Handbook of Combinatorial Optimization Volume1-3*. [S.l.]: Springer US, 1999. 2093–2229 p.

- GONÇALVES, J. F.; RESENDE, M. G. C. Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, Springer Netherlands, v. 17, p. 487–525, 2011.
- GOULART, N. et al. *Instances for the Uncapacitated Lot Sizing and Bin Packing problem*. 2018. https://ufsj.edu.br/prof_ngoulart/instances_for_iulsbp.php. Disponível em: https://ufsj.edu.br/prof_ngoulart/instances_for_iulsbp.php.
- HANSEN, P.; MLADENOVIC, N. Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, v. 130, n. 3, p. 449 – 467, 2001.
- ILOG, I. *ILOG CPLEX Optimization Studio*. ILOG CPLEX Optimization Studio, 2018. <https://www.ibm.com/products/ilog-cplex-optimization-studio>. Disponível em: <https://www.ibm.com/products/ilog-cplex-optimization-studio>.
- JARBOUI, B. et al. Variable neighborhood search for location routing. *Computers & Operations Research*, v. 40, n. 1, p. 47 – 57, 2013.
- KHANAFER, A. et al. The min-conflict packing problem. *Computers & Operations Research*, v. 39, n. 9, p. 2122–2132, 2012.
- LIBERALINO, H. et al. The integrated lot-sizing and vehicle routing problem. In: *Computational Intelligence In Production And Logistics Systems (CIPLS), IEEE Workshop On*. [S.l.: s.n.], 2011.
- LIMA, A. P. N.; SOUZA, M. C.; RAVETTI, M. Dimensionamento de lotes com remanufatura integrado ao problema do roteamento de veículos com entrega e coletas simultâneas. In: *XLV SBPO - Simpósio Brasileiro de Pesquisa Operacional*. [S.l.: s.n.], 2013. p. 3517–3526.
- MARTÍ, R.; RESENDE, M. G. C.; RIBEIRO, C. Multi-start methods for combinatorial optimization. *European Journal of Operational Research*, v. 226, n. 1, p. 1 – 8, 2013.
- MARTELLO, S.; TOTH, P. *Knapsack Problems: Algorithms and Computer Implementations*. New York, NY, USA: John Wiley & Sons, Inc., 1990.
- MATSUMOTO, M.; NISHIMURA, T. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, v. 8, p. 3–30, 1998.
- MIRZAEI, S.; SEIFI, A. Considering lost sale in inventory routing problems for perishable goods. *Computers & Industrial Engineering*, v. 87, p. 213–227, 2015.
- MOLINA, F.; MORABITO, R.; ARAÚJO, S. A. Modelos matemáticos para problemas de dimensionamento de lotes com restrições de capacidade e custos de transporte. *Gestão e Produção*, scielo, v. 20, p. 573–86, 2013.
- MOLINA, F.; MORABITO, R.; ARAUJO, S. A. de. Mip models for production lot sizing problems with distribution costs and cargo arrangement. *Journal of the Operational Research Society*, v. 67, n. 11, p. 1395–1407, Nov 2016.
- MOLINA, F. et al. An approach using Lagrangian/surrogate relaxation for lot-sizing with transportation costs. *Pesquisa Operacional*, scielo, v. 29, p. 269–288, 2009.

- NEMHAUSER, G. L.; WOLSEY, L. A. *Integer and Combinatorial Optimization*. New York, NY, USA: Wiley-Interscience, 1988.
- NORDEN, L. V.; VELDE, S. L. V. d. Multi-product lot-sizing with a transportation capacity reservation contract. *European Journal of Operational Research*, v. 165, n. 1, p. 127–138, 2005.
- POCHET, Y.; WOLSEY, L. A. *Production Planning by Mixed Integer Programming*. [S.l.]: Springer-Verlag New York, 2006.
- SADYKOV, R.; VANDERBECK, F. Bin packing with conflicts: A generic branch-and-price algorithm. *INFORMS Journal on Computing*, v. 25, n. 2, p. 244–255, 2013.
- SANCAK, E.; SALMAN, F. S. Multi-item dynamic lot-sizing with delayed transportation policy. *International Journal of Production Economics*, v. 131, n. 2, p. 595–603, 2011.
- SCHMID, V.; DOERNER, K. F.; LAPORTE, G. Rich routing problems arising in supply chain management. *European Journal of Operational Research*, v. 224, n. 3, p. 435–448, 2013.
- SPEARS, W.; DEJONG, K. On the virtues of parameterized uniform crossover. *Proceedings of the Fourth International Conference on Genetic Algorithms*, p. 230–236, 1991.
- TAGMOUTI, M.; GENDREAU, M.; POTVIN, J.-Y. A variable neighborhood descent heuristic for arc routing problems with time-dependent service costs. *Computers & Industrial Engineering*, v. 59, n. 4, p. 954–963, 2010.
- TSAI, J.-T. et al. An improved genetic algorithm for job-shop scheduling problems using taguchi-based crossover. *The International Journal of Advanced Manufacturing Technology*, Springer, v. 38, n. 1, p. 987–994, 2008.
- WAGNER, H. M.; WHITIN, T. M. Dynamic version of the economic lot size model. *Management Science*, v. 5, p. 89–96, 1958.
- YAMAN, H.; AZEN, A. Manufacture's mixed pallet design problem. *European Journal of Operational Research*, v. 186, n. 2, p. 826–840, 2008.