Métodos de solução aplicados ao problema de produção e distribuição

Henrique Hiroshi Motoyama Watanabe

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Henrique Hiroshi Motoyama Watanabe

Métodos de solução aplicados ao problema de produção e distribuição

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

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Maristela Oliveira dos Santos

USP – São Carlos
Julho de 2016

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

W324m Watanabe, Henrique Hiroshi Motoyama
 Métodos de solução aplicados ao problema de
 produção e distribuição / Henrique Hiroshi Motoyama
 Watanabe; orientadora Maristela Oliveira dos
 Santos. -- São Carlos, 2016.
 50 p.

 Dissertação (Mestrado - Programa de Pós-Graduação
 em Ciências de Computação e Matemática
 Computacional) -- Instituto de Ciências Matemáticas
 e de Computação, Universidade de São Paulo, 2016.

 1. Otimização matemática. 2. Heurísticas. 3.
 Produção. 4. Roteamento. 5. Dissertação. I. Santos,
 Maristela Oliveira dos, orient. II. Título.

Henrique Hiroshi Motoyama Watanabe

**Solution methods applied to production and distribution
problem**

Master dissertation submitted to the Instituto de
Ciências Matemáticas e de Computação – ICMC-
USP, in partial fulfillment of the requirements for the
degree of the Master Program in Computer Science
and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and
Computational Mathematics

Advisor: Prof. Dr. Maristela Oliveira dos Santos

USP – São Carlos
July 2016

AGRADECIMENTOS

Agradeço primeiramente à minha família. Aos meus pais, Sérgio e Eliza, por apoiarem minhas decisões, pela educação, incentivo e confiança. Aos meus irmãos, Vanessa e Danilo, pelo carinho, companhia e cumplicidade.

Agradeço aos meus amigos pelos momentos de descontração. À minha orientadora, Maristela, pela paciência, conhecimento e orientação.

Agradeço aos professores da área de otimização, pelas aulas na graduação que motivaram a escolha da área deste trabalho. Aos amigos e colegas do LOt pelas ajudas com as disciplinas, Cplex e LaTeX.

Agradeço ao CNPq (Processos 132404/2014-1 e 476792/2013-4) pelo apoio financeiro durante os dois anos de mestrado.

RESUMO

Neste mestrado estudamos o problema integrado de produção e distribuição (PIPD). Ao resolver o PIPD, tenta-se obter de maneira integrada as decisões de produção, controle de estoque, distribuição e roteamento de veículos. Neste tipo de problema, em cada período, um único ou múltiplos itens são produzidos e distribuídos para os clientes com o objetivo de atender uma determinada demanda a um custo total mínimo. Ou seja, no PIPD deve-se decidir quando e quanto produzir de cada item e as rotas e distribuições de cada veículo tal que o custo final, que inclui custos de produção, estoque e distribuição, seja mínimo. Estudamos o problema e modelo matemático considerado em Armentano et al. (2011). Para tratar o PIPD propomos heurísticas baseadas em programação matemática e geramos instâncias com múltiplos itens para testar o desempenho destas heurísticas. Comparamos os resultados obtidos pelas heurísticas com o *solver* comercial Cplex. Os resultados mostram que algumas das abordagens propostas obtiveram soluções de boa qualidade considerando as instâncias geradas.

Palavras chave: Planejamento de produção, roteamento de veículos, métodos de solução, *relax-and-fix*, *fix-and-optimize*.

ABSTRACT

In this work, the integrated production and vehicle routing problem (IPVRP) was studied. IPRP is solved by deciding on a integrated way the production, inventory control, distribution and vehicle routing decisions. In this type of problem, in each period, a single or multiple items are produced and distributed to costumers in order to meet a certain demand while minimizing the total cost, i.e., decide when and how much to produce of each item and the routes and distribution for each vehicle such that the final cost, which includes production, inventory and distribution costs is minimal. We considered the problem and mathematical model studied in Armentano et al. (2011). In order to deal with IPRP, heuristics based on mathematical programming were proposed together with multiple items instances to test the performance of those heuristics. We compare their results with the comercial solver Cplex. Results showed that some proposed heuristics obtained good quality solutions considering the generated instances.

Key words: Production planning problem, vehicle routing, solution methods, relax-and-fix, fix-and-optimize.

SUMÁRIO

1	Introdução	p. 1
2	Revisão da Literatura	p. 5
2.1	Introdução geral	p. 5
2.2	O problema integrado	p. 7
3	Problema de Produção e Distribuição	p. 15
3.1	Descrição do problema	p. 15
3.1.1	Função objetivo	p. 15
3.1.2	Restrições	p. 16
3.2	Modelo matemático	p. 16
4	Abordagens de Solução	p. 21
4.1	<i>Relax-and-fix</i> (RF)	p. 22
4.1.1	Decomposições	p. 23
4.1.1.1	Por problemas	p. 23
4.1.1.2	Por períodos	p. 24
4.1.1.3	Mista	p. 24

4.2	Heurística construtiva de Lei et al. (2006)	p. 25
4.3	<i>Fix-and-Optimize</i> (FO)	p. 25
4.3.1	Decomposições	p. 26
4.3.2	Heurística combinada	p. 26
5	Testes Computacionais	p. 27
5.1	Instâncias utilizadas	p. 27
5.1.1	Geração de dados	p. 27
5.2	Ambiente de testes	p. 30
5.3	Testes preliminares	p. 30
5.3.1	Qualidade da solução inicial	p. 32
5.3.2	Qualidade da busca local	p. 39
5.4	Testes	p. 41
6	Considerações Finais	p. 45
	Referências Bibliográficas	p. 47

LISTA DE TABELAS

1	Resumo da literatura	p. 14
2	Pseudo algoritmo <i>relax-and-fix</i>	p. 23
3	Conjuntos de váriaveis	p. 23
4	Pseudo algoritmo <i>fix-and-optimize</i>	p. 26
1	Instâncias geradas de pequeno porte	p. 28
2	Instâncias geradas de grande porte	p. 28
3	Decomposições utilizadas	p. 31
4	Testes preliminares - <i>gaps</i> médios das heurísticas	p. 34
5	Testes preliminares - <i>gaps</i> de 5 minutos contra 10 minutos	p. 35
6	Testes preliminares - <i>gaps</i> com 5 minutos por iteração da <i>fix-and-optimize</i> (FO1 e bFO1)	p. 37
7	Testes preliminares - <i>gaps</i> da solução inicial da RF7 com várias melhorias	p. 39
8	Resultados finais	p. 42
9	Lei+FO6 nas instâncias grandes	p. 44
10	Quantidade de variáveis das instâncias	p. 44

LISTA DE FIGURAS

1	<i>Relax-and-fix</i>	p. 23
2	Decomposição por problemas	p. 24
3	Decomposição por períodos	p. 24
4	Decomposição mista por problemas e períodos	p. 24
5	Decomposição por períodos	p. 26
6	Testes preliminares variando a <i>relax-and-fix</i>	p. 32
7	Perfis de desempenho das heurísticas RF7 - RF10 com 5 e 10 minutos por iteração da <i>fix-and-optimize</i> (FO1)	p. 36
8	Vermelho (5 minutos) x Verde (10 minutos)	p. 36
9	Perfis de desempenho das heurísticas apresentadas na Tabela 6	p. 37
10	Melhores desempenhos da Figura 9	p. 38
11	Testes preliminares variando a <i>fix-and-optimize</i>	p. 39
12	Perfis de desempenho das heurísticas apresentadas na Tabela 7	p. 40
13	Figura 12 ampliada	p. 40
14	Melhores perfis de desempenho, RF7 fixo	p. 41
15	Perfis de desempenho	p. 43
16	Destaque para Cplex x RF7+FO9	p. 43

LISTA DE ABREVIATURAS E SIGLAS

PIPD	Problema Integrado de Produção e Distribuição
<i>MIP</i>	<i>Mixed Integer Programming</i>
RF	<i>Relax-and-Fix</i>
FO	<i>Fix-and-Optimize</i>
<i>GRASP</i>	<i>Greedy Randomized Adaptive Search Procedures</i>
<i>ALNS</i>	<i>Adaptative Large Neighborhood Search</i>
<i>OU</i>	<i>Order-Up to Level</i>
<i>ML</i>	<i>Maximum Level</i>

CAPÍTULO 1

INTRODUÇÃO

Os problemas de dimensionamento de lotes (*lot sizing*) e roteamento de veículos (*vehicle routing problem*), presentes em diversas empresas, são clássicos da área de otimização e foram introduzidos há mais de 50 anos, por Wagner e Whitin (1958) e Dantzig e Ramser (1959), respectivamente. Com o aumento da competição entre empresas, a busca por um planejamento mais eficiente chegou, enfim, à integração desses dois problemas, sendo objeto de estudo em Chandra e Fisher (1994), Boudia et al. (2005), Archetti et al. (2011) e outros trabalhos da literatura.

O problema de dimensionamento de lotes faz parte do planejamento de produção, no qual decidimos as quantidades de produtos que serão produzidas em cada período de um horizonte finito de tempo, com o intuito de atender demandas conhecidas e minimizar os custos envolvidos (Karimi et al. (2003)). Após a fase da produção, geralmente utiliza-se o problema de roteamento de veículos para definir as rotas de uma frota de veículos que distribuirá os produtos aos clientes, também em busca de minimizar os custos.

O problema integrado de produção e distribuição (PIPD) ainda não foi tão estudado na literatura quanto cada um dos problemas separados. Segundo Adulyasak et al. (2012), sua complexidade é um dos motivos que fez com que não recebesse tanta atenção.

No PIPD tem-se uma ou mais fábricas produzindo itens que serão entregues aos clientes, de modo a satisfazer suas demandas, por uma frota de veículos durante uma série de períodos. O objetivo é minimizar os custos envolvidos, que geralmente são formados

pela preparação das máquinas para a produção (*setup*), produção, estoque e distribuição. A principal motivação para a integração desses dois problemas é a diminuição de custos, pois ao focar na minimização de apenas uma parte do problema integrado, o custo do restante geralmente aumenta (Adulyasak et al. (2012)). Por exemplo, minimizando apenas o custo de preparação das máquinas, o custo de estoque aumenta, porque a fábrica produziria o máximo em cada período para economizar com custos de preparação, o que levaria a uma quantidade de estoque maior. Por outro lado, produções em quantidades menores diminuem o custo de estoque mas aumentam o custo de preparação e transporte, pois será preciso realizar essas tarefas com mais frequência para atender às demandas. Portanto a ideia é encontrar o equilíbrio entre todos os subproblemas.

Assim, neste trabalho de mestrado, consideramos o problema integrado de produção e distribuição. Modelos propostos na literatura foram investigados e estudamos abordagens de solução que utilizam a formulação matemática em procedimentos de decomposição. Nesta dissertação, foi considerado o problema estudado em Armentano et al. (2011), com produção de múltiplos itens em uma única fábrica com capacidade de produção limitada, múltiplos clientes com demandas conhecidas ao longo de vários períodos de tempo, e frota homogênea para realizar a distribuição.

Propusemos instâncias de múltiplos itens, e testamos várias combinações de heurísticas *relax-and-fix* com *fix-and-optimize*, comparando-as com o *solver* comercial Cplex aplicado no modelo proposto por Armentano et al. (2011). Para as instâncias maiores, as quais nem as heurísticas combinadas e nem o Cplex conseguiram resolver no limite de tempo, utilizamos a heurística construtiva da primeira fase de Lei et al. (2006) com a melhoria *fix-and-optimize* para resolvê-las.

Os resultados mostraram que, no limite de tempo de execução, a combinação de heurísticas (*relax-and-fix* e *fix-and-optimize*) escolhida obteve, em média, custos totais menores que a resolução do modelo, por meio do Cplex. Porém ao considerar a heurística de melhoria *fix-and-optimize* partindo da solução inicial obtida pela construtiva de Lei et al. (2006), os custos totais foram maiores em quase todas as instâncias.

Esta dissertação está organizada da seguinte forma: No Capítulo 2 será apresentada uma introdução do problema, junto com uma revisão da literatura. Uma formulação matemática da literatura será apresentada no Capítulo 3. As abordagens de solução serão apresentadas no Capítulo 4, com testes computacionais e resultados no Capítulo 5. O Capítulo 6 apresentará conclusões e considerações finais.

/

CAPÍTULO 2

REVISÃO DA LITERATURA

2.1 Introdução geral

Nesta dissertação, trabalhamos em um problema típico de planejamento de produção. O planejamento da produção engloba decisões de curto, médio e longo prazo. As decisões do planejamento a longo prazo são estratégicas, como por exemplo a escolha de produtos, equipamentos, seus processos, localização da(s) planta(s) e armazéns. O planejamento a médio prazo se resume a escolher como e quando utilizar os recursos disponíveis para produzir uma quantidade de itens, visando atender uma certa demanda. As decisões de curto prazo, englobam decisões do dia-a-dia, como sequenciamento de tarefas (Karimi et al. (2003)). O problema de produção considerado neste trabalho planeja as decisões de médio prazo, dado pelo problema de dimensionamento de lotes (*lot sizing*) e de curto prazo, que são as decisões de distribuição e roteamento dos itens aos clientes.

Existem vários trabalhos na literatura que abordam o problema de dimensionamento de lotes, como Wagner e Whitin (1958), Manne (1958), Billington et al. (1983), Trigeiro et al. (1989), Federgruen e Tzur (1991), Wagelmans et al. (1992), Aggarwal e Park (1993). Trigeiro et al. (1989), por exemplo, estudam o efeito que os tempos de preparação de máquinas têm em um problema de dimensionamento de lotes com uma máquina, múltiplos itens, e custos variáveis em relação ao tempo.

O foco deste trabalho é o planejamento de produção em ambientes com limitação de

capacidade, por exemplo: limitação de mão de obra, máquinas, tempo, matéria-prima, espaço. Em Karimi et al. (2003), listam as principais características que definem um problema de dimensionamento de lotes, e neste trabalho estamos interessados em problemas com o horizonte de planejamento finito, com múltiplos produtos simples (sem sequenciamento de produção, chamado monoestágio), fábricas com limites de capacidades (produção e estoque), demandas predefinidas, custos de preparação de máquinas e sem nenhum tipo de atraso (*backlogging*).

A seguir, apresentamos um modelo de programação inteira mista para um problema de dimensionamento de lotes monoestágio com limites de capacidade. O problema consiste em decidir a quantidade produzida para cada item em cada período de modo a satisfazer a demanda enquanto minimiza o custo total, que é a soma dos custos de estoque, produção e preparação de máquinas. Neste modelo, a capacidade produtiva da fábrica é utilizada apenas para a produção dos itens e, como os custos de preparação, é constante em todos os períodos. O número de itens é J e temos T períodos.

Índices

- t Índice de períodos ($t = 1, \dots, T$);
 j Índice de itens ($j = 1, \dots, J$);

Parâmetros

- B_t Capacidade de produção na fábrica no período t ;
 b_j Tempo de produção do item j ;
 c_j^p Custo de produção do item j ;
 f_j^p Custo de preparação da máquina para produzir item j ;
 h_j Custo de estoque do item j ;
 d_{jt} Demanda do item j no período t ;
 M Número muito grande.

Variáveis do problema

- p_{jt} Quantidade do item j produzida no período t ;
 I_{jt} Estoque do item j no final do período t ;
 y_{jt} Variável binária de produção do item j no período t .
 ($y_{jt} = 1$ se $p_{jt} > 0$ e $y_{it} = 0$, caso contrário)

$$\text{Minimize} \quad \sum_{t=1}^T \sum_{j=1}^J (h_k I_{jt} + c_j^p p_{jt} + f_j^p y_{jt}) \quad (2.1)$$

$$\text{sujeito a: } p_{jt} + I_{j,t-1} - I_{jt} = d_{jt} \quad 1 \leq t \leq T, 1 \leq j \leq J \quad (2.2)$$

$$\sum_{j=1}^J b_j p_{jt} \leq B_t \quad 1 \leq t \leq T \quad (2.3)$$

$$p_{jt} \leq M y_{jt} \quad 1 \leq t \leq T, 1 \leq j \leq J \quad (2.4)$$

$$p_{jt}, I_{jt} \geq 0, y_{jt} \in \{0, 1\}, 1 \leq t \leq T, 1 \leq j \leq J \quad (2.5)$$

A função objetivo (2.1) minimiza os custos de estoque, produção e preparação de máquinas. As restrições (2.2) garantem que a demanda será atendida para cada item e período. As restrições (2.3) garantem que a produção não ultrapassa a capacidade da fábrica e as restrições (2.4) ativam o custo de preparação quando existir produção. As restrições (2.5) são os domínios das variáveis.

Na literatura, geralmente realiza-se a produção para depois efetuar a distribuição dos itens, com uma frota de veículos homogêneos ou heterogêneos, para os clientes. Essa entrega pode ser parcial, quando um cliente pode ser atendido por veículos diferentes, ou não, quando cada cliente pode ser atendido por, no máximo, um veículo em cada período. No caso da frota homogênea, os veículos são idênticos, ou seja, todos os veículos possuem uma mesma capacidade de transporte (limite na quantidade de itens que podem carregar), as mesmas restrições de rotas (distância percorrida por período, por exemplo) e custos.

Este trabalho considera que as decisões de produção e distribuição são integradas. Como em Armentano et al. (2011), a produção será de múltiplos itens em uma única fábrica com capacidade de produção limitada. A entrega não é parcial, os veículos são homogêneos e só podem realizar uma rota por período. A seguir, alguns trabalhos que consideram o problema integrado são apresentados.

2.2 O problema integrado

Apesar de cada problema geralmente ser considerado separado na literatura, o interesse pelo PIPD está crescendo. Um dos primeiros estudos que considerou a integração dos dois problemas foi Chandra e Fisher (1994). O problema considerado tinha múltiplos itens, uma única fábrica com limitação de produção e frota homogênea com entrega parcial.

Chandra e Fisher (1994) primeiramente resolveram o problema desacoplado, com um método derivado de Barany et al. (1984) e Leung et al. (1989) para a produção e uma heurística construtiva com busca local para a distribuição e roteamento. Em seguida, utilizaram uma abordagem mudando apenas o procedimento de busca local. Este novo procedimento permitiu que o planejamento da produção fosse modificado, com um novo teste de factibilidade, o que considera a integração entre os dois subproblemas. Foram geradas 132 instâncias para verificar o comportamento dos métodos propostos e em quais casos o planejamento integrado leva maior vantagem sobre o desacoplado. As instâncias variaram na quantidade de clientes, períodos, itens, custos de preparação, estoque, veículo e transporte, e nas capacidades dos veículos e produção. Os custos de distribuição (custo fixo de utilização do veículo e custos relacionados a seus percursos) foram as características que menos influenciaram, ou seja, a variação de seus valores não teve muito efeito na vantagem do planejamento integrado sobre o desacoplado. Quanto maior a quantidade de clientes, períodos ou itens, maior é a diferença entre os métodos. Porém ao aumentar os custos de preparação, estoque e a capacidade da fábrica, a vantagem da integração diminui.

Fumero e Vercellis (1999) consideraram o problema com produção de múltiplos itens, uma única fábrica com limitação de capacidade e frota homogênea que entrega os itens para clientes cujas demandas são conhecidas. Permitiram a entrega parcial, que é a possibilidade de mais de um veículo entregar itens a um cliente em um mesmo período, mas cada veículo está limitado a uma rota por período, isto é, não podem passar na fábrica para reabastecer itens durante o período. Utilizaram relaxação lagrangiana para relaxar as restrições de acoplamento entre os dois subproblemas, assim eles puderam ser resolvidos com mais facilidade. Foram geradas instâncias com até 12 clientes, 8 períodos e 10 itens. Os autores compararam seus resultados com um método que resolve o problema separadamente, utilizando o resultado do problema de produção como dados de entrada para o problema de distribuição. Seus resultados mostraram que, de fato, integrar a produção com a distribuição resulta em soluções com custos menores e que quanto maior o tamanho da instância (clientes, períodos, itens), maior a redução nos custos totais.

Boudia et al. (2005), Boudia et al. (2007), Archetti et al. (2007), Boudia e Prins (2009), Bard e Nananukul (2009), Archetti et al. (2011), Absi et al. (2014) e Adulyasak et al. (2012) consideraram o problema com produção de um item em uma fábrica com limitação de capacidade, frota homogênea e com limite de uma rota por período por veículo. Archetti et al. (2007) permitem entregas parciais, os outros não.

Boudia et al. (2005) geraram instâncias de 20 períodos com 50, 100 e 200 clientes, que serviram de *benchmark* para Boudia et al. (2007), Boudia e Prins (2009), Bard e Nananukul (2009) e Adulyasak et al. (2012).

Em Boudia et al. (2005), foi desenvolvida uma heurística que resolve o PIPD separadamente, utilizando o método de Wagner e Whitin (1958) para planejar a produção, e a distribuição é feita a partir disso. Depois alteraram essa heurística, de modo que a produção fosse reajustada após determinar o roteamento, gerando uma pequena integração entre os dois problemas. Os autores nomearam esses métodos de H1 e H2, respectivamente. Como melhoria para o roteamento em ambas, aplicaram trocas 2-opt (Lin e Kernighan (1973)).

Em Boudia et al. (2007), os autores escolheram o *GRASP* (*Greedy Randomized Adaptive Search Procedures* - Feo e Resende (1995)) por sua simplicidade. A heurística construtiva possui duas fases: determina, para cada dia separadamente, o dimensionamento de lotes e a distribuição dos produtos na primeira fase e, na segunda, tenta deslocar a produção para evitar custos de preparação desnecessários. Para movimentos da busca local, consideraram o 2-opt, troca e reinserção de nós (clientes). Depois propuseram um *GRASP* reativo e um *GRASP* com *path relinking*.

Boudia e Prins (2009) propuseram um algoritmo memético (*memetic algorithm* - Moscato et al. (1989)) com controle de população. Num algoritmo memético, uma população inicial (grupo de soluções) é gerada aleatoriamente e em seguida alterada com uma série de operações. Foi utilizado um critério de distância para garantir que não há indivíduos (soluções) iguais. Em cada iteração, o método começa com uma população inicial (ordenada) e, a partir de torneios, escolhe pares de soluções para realizar um *crossover*. *Crossover* é uma mistura de duas soluções para obter uma nova (C). A solução C passa por uma busca local com uma probabilidade π fixa. Uma solução na pior metade da população é escolhida (R) e comparada com a melhor solução da população (P), se a diferença entre elas for maior que uma distância δ_g (diferente em cada geração g), substitui-se R por C . Se o teste falhar, compara-se a função objetivo de C e P e, se C for melhor (custo menor), ela substitui R .

Bard e Nananukul (2009) propuseram uma abordagem em duas fases, uma solução inicial é procurada na primeira fase e, em seguida, utilizam uma busca tabu (*tabu search* - Glover (1989)) para melhorá-la. A busca tabu utiliza uma "lista tabu" para evitar a repetição de soluções durante a busca, proibindo o método de voltar em uma solução que já foi visitada nas últimas k iterações, k é um parâmetro escolhido.

Adulyasak et al. (2012) escolheram uma variação da heurística *ALNS* (*Adaptive Large Neighborhood Search*), proposta em Ropke e Pisinger (2006), em que os autores nomearam *Op-ALNS*. Primeiro eles criam várias soluções iniciais, obtidas pela resolução dos subproblemas de produção-distribuição e roteamento, depois realizam uma busca local nelas.

Dentre os trabalhos que utilizaram as instâncias de Boudia et al. (2005), a heurística de duas fases H1 de Boudia et al. (2005) foi a mais rápida, porém com os maiores custos. Seu tempo médio foi de 0,1, 0,54 e 2,1 segundos para instâncias de 50, 100 e 200 clientes, respectivamente. As metaheurísticas de Boudia et al. (2007) e Boudia e Prins (2009) obtiveram resultados melhores que Boudia et al. (2005) em todos os casos, como era esperado, pois o método H1 não resolveu o problema de forma integrada. Mas seus tempos de resolução foram muito maiores, com o *GRASP* de Boudia et al. (2007) atingindo 30 minutos e o algoritmo memético de Boudia e Prins (2009) passando de uma hora nas instâncias de 200 clientes. Conseguiram soluções que foram, em média, 17,29% e 23,64% melhores, respectivamente, quando comparadas à H1 nas instâncias de 200 clientes. A busca tabu de Bard e Nananukul (2009) foi melhor que o *GRASP* de Boudia e Prins (2009), com a diferença na função objetivo variando de 10% a 20%, mas com um tempo de 3 a 5 vezes maior. E Adulyasak et al. (2012) obteve, até agora, uma média de custos menor que os outros trabalhos para as instâncias propostas por Boudia et al. (2005).

Adulyasak et al. (2012) resolveram, também com soluções de boa qualidade, as instâncias propostas por Archetti et al. (2011). Archetti et al. (2007) e Absi et al. (2014) também utilizaram esse conjunto de instâncias.

Archetti et al. (2007) e Archetti et al. (2011) estudaram duas abordagens diferentes em relação ao estoque. O primeiro trabalho escolheu a abordagem *Order-Up to Level* (*OU*), que força os estoques dos clientes visitados no período a atingirem os seus limites máximos. No segundo, a abordagem de *Maximum Level* (*ML*), que é mais comum na literatura, na qual os estoques dos clientes não podem ultrapassar suas respectivas capacidades máximas. Archetti et al. (2007) utilizaram um algoritmo *branch-and-cut* para a abordagem *OU*. O algoritmo começa construindo uma solução inicial, com inserção de nós (clientes), em seguida é aplicado um processo de melhoria, retirando e reinserindo um par de clientes. Este processo é repetido enquanto o custo total for diminuindo. Archetti et al. (2011) propuseram uma heurística para resolver o problema utilizando a abordagem *ML*. Os autores decomposeram o problema, em produção e distribuição, resolveram primeiro a distribuição com uma heurística de inserção de nós (clientes) e depois a produção, tam-

bém separando em subproblemas diferentes para cada cliente. Os resultados mostraram que a abordagem *ML* obtém soluções com custos menores do que a abordagem *OU*, o que era esperado já que a abordagem *OU* é um caso particular da *ML*.

Absi et al. (2014) propuseram o *Iterative Method*, que funciona em duas fases. Na primeira fase, o método resolve um problema de produção com um custo de visita, para acoplar a parte de distribuição, planejando quanto e quando produzir e entregar para cada cliente, assim como em quais períodos realizar essas tarefas. Na segunda, otimiza rotas para os veículos disponíveis, como múltiplos problemas do caixeiro viajante, e ajusta o custo de visita para a próxima primeira fase, se necessária. Essas fases são realizadas repetidamente até atingir um critério de parada, que pode ser um número máximo de iterações ou quando a solução parar de melhorar. Os autores utilizaram um *solver* de otimização para obter soluções na primeira fase e implementam a heurística de Lin e Kernighan (1973) para a segunda. Nos testes computacionais nas instâncias de Archetti et al. (2011), obtiveram soluções cujo *gap* médio foi melhor que dos outros trabalhos.

Lei et al. (2006), Piewthongngam et al. (2013) e Darvish et al. (2016) basearam seus trabalhos em problemas reais, com múltiplas fábricas. Enquanto o problema de Lei et al. (2006) tem apenas um item, os outros dois consideraram múltiplos itens. Lei et al. (2006) e Piewthongngam et al. (2013) possuem frotas heterogêneas, e Darvish et al. (2016) trabalhou com veículos terceirizados.

Lei et al. (2006) consideraram um problema com distribuição marítima. Propuseram um método que chamaram de *two-phase*. O método começa resolvendo o problema integrado, utilizando o *solver* Cplex na formulação *MIP*, com distribuição direta, ou seja, os veículos se locomovem apenas entre a fábrica e os clientes, sem fazer nenhum percurso entre dois clientes. A solução obtida na primeira fase é sempre factível e, portanto, um limitante superior para o problema de minimização. E a segunda fase consiste em um algoritmo de melhoria que aplica duas heurísticas nas rotas da atual solução e tenta melhorá-las. A primeira heurística testa soluções vizinhas trocando o veículo que passa por um certo cliente, e a segunda reconstrói as rotas, utilizando o método do vizinho mais próximo, começando pelo “melhor” veículo, até atingir o seu limite e em seguida muda para o próximo veículo, repetindo o processo. Existem dois critérios para determinar o *ranking* dos veículos, sua velocidade e o seu custo por distância percorrida. Os testes realizados foram comparados com os do *solver* Cplex. A vantagem que eles levaram sobre o Cplex foi em tempo, cuja diferença foi muito mais significativa do que o *gap* entre as soluções. O maior *gap* foi de 11%, mas houve instâncias em que o Cplex, em 2 horas,

não encontrou nenhuma solução factível enquanto o *two-phase* resolveu cada instância em segundos.

Piewthongngam et al. (2013) fizeram um modelo baseado na produção e distribuição de rações de porcos à fazendas. As fábricas, rações e fazendas representam, respectivamente, as fábricas, produtos e clientes. Cada fazenda possui um modo diferente de armazenar as rações (caixas, sacos ou ambos), ou seja, as capacidades de estoque variam. Propuseram uma heurística que resolve o problema em duas partes, estimando a demanda (com ajuda de uma base de dados) e planejando a produção e quantidade de veículos necessários para cada grupo de fazendas na primeira parte, e o roteamento na segunda. Comparam a performance da heurística implementada com a solução dada pela resolução do modelo matemático proposto utilizando o *solver LINGO 11*, em 10 simulações pequenas (2 fábricas e 2 clientes). Obteve resultados que eram, no máximo, 5% piores que o *solver*, porém mais rápidos.

Darvish et al. (2016) trabalharam juntos com um companhia que produz e vende móveis e possui janelas de tempo para as entregas. Trataram apenas das entregas aos clientes online, o outro tipo de entrega é para lojas, portanto apenas a fábrica (armazém) pode estocar itens de um período para o próximo. Atrasos não são permitidos e as fábricas podem trocar itens entre si, com um custo de deslocamento. Como a companhia utiliza um serviço terceirizado para fazer as entregas, o problema não trata da parte de roteamento, apenas a distribuição com um custo proporcional à quantidade entregue por período. Escolheram um algoritmo de *branch-and-bound* para abordar o problema, com o tempo limite de uma hora. Utilizando os dados da companhia, criaram diferentes situações para realizar uma análise de sensibilidade em relação ao tamanho da janela de entregas, intervalo de tempo entre produções, custos e capacidades de produção e estoque. Apesar de não encontrarem soluções ótimas para o problema, o *gap* foi pequeno (menos de 3%), e o algoritmo escolhido foi melhor que o método utilizado pela companhia, possibilitando uma economia de 4% a 14% de custos totais.

Armentano et al. (2011) e Brahimi e Aouam (2015) estudaram o problema com uma fábrica, múltiplos itens e frotas homogêneas sem entrega parcial. Brahimi e Aouam (2015), porém, modelou o problema permitindo atrasos (*backordering*). Devido à falta de instâncias com múltiplos itens, Armentano et al. (2011) testou suas buscas tabu nas instâncias, de um item, propostas por Boudia et al. (2005) para comparar com outros trabalhos da literatura como, por exemplo, Boudia e Prins (2009) e Bard e Nananukul (2009). Brahimi e Aouam (2015) também utilizou instâncias de um item só, mas escolheu as de Archetti

et al. (2011). Ambos criaram suas próprias instâncias de múltiplos itens.

Armentano et al. (2011) geraram instâncias de pequeno porte, com até 15 clientes, 14 períodos e 5 itens, para comparar as soluções das buscas tabu propostas com as soluções exatas, dada pelo *solver* Cplex 10. E instâncias maiores, com até 100 clientes, 24 períodos e 10 itens, que foram utilizadas para comparar com um método que resolve o problema de forma desacoplada, justamente para verificar que ocorre uma melhoria quando a integração é utilizada.

Brahimi e Aouam (2015) propuseram uma heurística híbrida baseada no *relax-and-fix* para resolver o problema. Primeiro resolvem o problema de dimensionamento de lotes com o método *relax-and-fix* e depois aplicam uma busca local, implementada por Groër et al. (2010), para otimizar as rotas, fazendo ajustes na produção, se necessário. O tamanho de suas instâncias variaram de 5, 6 e 14 a 25, 12 e 50 itens, períodos e clientes, respectivamente.

Adulyasak et al. (2015) e Mostafa e Eltawil (2015) fizeram revisões dos trabalhos dessa área. No primeiro, separaram os trabalhos por categorias, como métodos por decomposições, exatos e metaheurísticas. Além disso, apresentaram um modelo matemático para o problema PIPD com um único item, uma fábrica, frota homogênea e sem entrega parcial. Discutiram algumas maneiras de tratar as restrições que eliminam as sub-rotas em problemas de roteamento de veículos e a inserção delas no modelo apresentado. No segundo, consideraram apenas trabalhos mais recentes (de 2006 à 2014) e específicos, ou seja, os que consideram produção, inventário, distribuição e roteamento. Citam a escassez de artigos que estudam um caso real ou problemas mais complicados, com múltiplas fábricas, produtos e frota heterogênea. Propuseram um modelo com uma fábrica que produz múltiplos itens, cujos custos e tempos de preparação e produção variam, inventários na fábrica e clientes com capacidade, frota de veículos heterogênea, e sem atraso de entrega.

A Tabela 1 mostra um resumo dos trabalhos apresentados nesta revisão e enquadra o problema em estudo na literatura.

Tabela 1: Resumo da literatura

Trabalho	Fábrica	Item	Frota	Entrega parcial	<i>backordering</i>
Chandra e Fisher (1994)	única	múltiplos	homogênea	sim	não
Fumero e Vercellis (1999)	única	múltiplos	homogênea	sim	não
Boudia et al. (2005)	única	único	homogênea	não	não
Lei et al. (2006)	múltiplas	único	heterogênea	sim	não
Archetti et al. (2007)	única	único	homogênea	sim	não
Boudia et al. (2007)	única	único	homogênea	não	não
Boudia e Prins (2009)	única	único	homogênea	não	não
Bard e Nananukul (2009)	única	único	homogênea	não	não
Archetti et al. (2011)	única	único	homogênea	não	não
Armentano et al. (2011)	única	múltiplos	homogênea	não	não
Adulyasak et al. (2012)	única	único	homogênea	não	não
Piewthongngam et al. (2013)	múltiplas	múltiplos	heterogênea	não	sim
Absi et al. (2014)	única	único	homogênea	não	não
Brahimi e Aouam (2015)	única	múltiplos	homogênea	não	sim
Darvish et al. (2016)	múltiplas	múltiplos	terceirizada	não	não
Esta dissertação	única	múltiplos	homogênea	não	não

CAPÍTULO 3

PROBLEMA DE PRODUÇÃO E DISTRIBUIÇÃO

3.1 Descrição do problema

O problema de produção e distribuição tem variações que dependem do tipo de produto e fábrica que estamos a tratar, o que pode mudar completamente tanto a função objetivo como algumas das restrições. Neste trabalho consideramos o problema como apresentado em Armentano et al. (2011), ou seja, com possibilidade de produção de vários itens em uma única fábrica com limite na capacidade de produção, sem atrasos, frota homogênea e sem entrega parcial.

3.1.1 Função objetivo

Nesse problema integrado, deve-se decidir como organizar a produção de vários tipos de itens em uma única fábrica durante um horizonte finito de tempo. E também planejar o roteamento com uma quantidade fixa de veículos homogêneos disponíveis, assim como a distribuição realizada por cada um deles, de modo que as demandas de todos os clientes sejam satisfeitas sem nenhum atraso. O objetivo é minimizar o custo total, dado pelo custo da produção, do estoque e transporte. Na parte de produção, temos o custo fixo de preparação das máquinas (*setup*) se produzimos naquele período e o custo de produção

de cada item. Os custos de estoque podem ser diferentes em cada local (clientes ou fábrica), porém invariantes em relação aos períodos. Os veículos têm um custo fixo se forem utilizados e mais um custo de transporte proporcional ao tamanho da rota que ele percorre.

Outras características poderiam ser consideradas. Por exemplo, podemos citar custos de produção e estoque que variam em diferentes períodos (preço da matéria-prima depende da época do ano, produtos agrícolas por exemplo), custo de preparação de máquinas diferente para cada item, uma parcela a mais no custo de transporte que depende da quantidade de itens transportada e até mesmo a possibilidade de contratar uma empresa terceirizada para realizar parte da distribuição que excede a capacidade da frota disponível.

3.1.2 Restrições

Vamos considerar que a capacidade de produção seja limitada, e que não existe um tempo de preparação antes de começar a produzir, ou seja, a capacidade produtiva já vem descontada do tempo de preparação médio. As demandas não podem ser atendidas com atraso. Existe uma capacidade máxima e uma quantidade mínima de estoque de segurança para cada item em cada local. A frota é homogênea, os veículos têm um limite no tamanho da rota percorrida, e só podem percorrer uma por período, isto é, não podem voltar à fábrica para recarregar itens e entregar no mesmo período, e dois veículos não podem visitar um mesmo cliente em um mesmo período.

Outras características poderiam ser consideradas. Por exemplo, podemos permitir atraso, com penalização da função objetivo, frotas heterogêneas com veículos de tamanhos diferentes e/ou que têm características (custo de transporte) distintas, múltiplas rotas, entrega parcial (quando um cliente pode ser atendido por mais de um veículo em cada período).

3.2 Modelo matemático

Para modelar o nosso problema, vamos definir representação, parâmetros e as variáveis. $G = (W, E)$ é um grafo completo que representa o problema, $W = \{0, 1, \dots, N\}$ é o conjunto de nós que representa a fábrica e os clientes e $E = \{(k, l) \mid k, l \in W, k \neq l\}$ o conjunto de arestas que são as rotas entre fábrica e clientes, e clientes entre si. A fábrica, representada pelo nó $k = 0$, produz os itens $j = 1, \dots, J$ que são levados aos clientes

$k = 1, \dots, N$ pelos veículos $v = 1, \dots, V$. Denotamos os períodos por $t = 1, \dots, T$.

A capacidade de produção da fábrica é B unidades de tempo e o tempo de produção para o item j é b_j com custo c_j^p . O custo de preparação das máquinas para o item j é f_j^p e o estoque da unidade do item j no nó k é h_{jk} , sendo que os estoques máximo e mínimo são U_{jk} e L_{jk} , respectivamente. Na parte de transporte, os veículos tem capacidade C , a distância máxima que cada um pode percorrer é L , o custo para utilizar cada veículo tem a parte fixa f^v e a parte variável c_{kl}^v , se o veículo passar pela aresta (k, l) , e a demanda do item j no cliente k no período t é d_{jkt} .

Índices

- k Índice de fábrica e clientes ($k = 0, 1, \dots, N$);
- v Índice de veículos ($v = 1, \dots, V$).
- t Índice de períodos ($t = 1, \dots, T$);
- j Índice de itens ($j = 1, \dots, J$);

Parâmetros

- B Capacidade de produção na fábrica;
- b_j Tempo de produção do item j ;
- c_j^p Custo de produção do item j ;
- f_j^p Custo de preparação da máquina para produzir item j ;
- h_{jk} Custo de estoque do item j no cliente k ;
- L_{jk} Estoque mínimo (segurança) do item j no cliente k ;
- U_{jk} Estoque máximo (capacidade) do item j no cliente k ;
- C Capacidade dos veículos;
- L Tamanho máximo da rota percorrida por cada veículo;
- f^v Custo fixo para cada veículo utilizado em cada período;
- c_{kl}^v Custo de transporte na aresta (k, l) ;
- d_{jkt} Demanda do item j no cliente k no período t ;
- M Número muito grande.

Variáveis do problema

- p_{jt} Quantidade do item j produzida no período t ;
- I_{jkt} Estoque do item j em k no final do período t ;
- q_{jkt}^v Quantidade do item j entregue ao cliente k pelo veículo v no período t ;
- x_{jkt}^v Quantidade do item j transportado pelo veículo v no período t na aresta (k, l) ;
- y_{jt} Variável binária de produção do item j no período t ;

- ($y_{jt} = 1$ se $p_{jt} > 0$ e $y_{it} = 0$, caso contrário)
- z_{klt}^v Variável binária de transporte do veículo v no período t na aresta (k, l) .
 ($z_{klt}^v = 1$ se v passa por (k, l) no período t e $z_{klt}^v = 0$, caso contrário)

O modelo a seguir foi proposto em Armentano et al. (2011).

$$\text{Minimize} \quad \sum_{t=1}^T \left\{ \sum_{j=1}^J \left[\sum_{k=0}^N h_{jk} I_{jkt} + c_{jt}^p p_{jt} + f_j^p y_{jt} \right] + \sum_{v=1}^V \left[\sum_{l=1}^N f^v z_{0lt}^v + \sum_{l=0, l \neq k}^N c_{kl}^v z_{klt}^v \right] \right\} \quad (3.1)$$

$$\text{sujeito a:} \quad p_{jt} + I_{j0,t-1} - I_{j0t} = \sum_{k=1}^N \sum_{v=1}^V q_{jkt}^v \quad 1 \leq t \leq T, 1 \leq j \leq J \quad (3.2)$$

$$\sum_{v=1}^V q_{jkt}^v + I_{jk,t-1} - I_{jkt} = d_{jkt} \quad 1 \leq t \leq T, 1 \leq j \leq J, 0 \leq k \leq N \quad (3.3)$$

$$\sum_{j=1}^J b_j p_{jt} \leq B \quad 1 \leq t \leq T \quad (3.4)$$

$$p_{jt} \leq M y_{jt} \quad 1 \leq t \leq T, 1 \leq j \leq J \quad (3.5)$$

$$\sum_{\substack{i=0 \\ i \neq k}}^N x_{jikt}^v - \sum_{\substack{m=0 \\ m \neq k}}^N x_{jkm t}^v = q_{jkt}^v \quad 1 \leq t \leq T, 1 \leq j \leq J, 1 \leq v \leq V, 0 \leq k \leq N \quad (3.6)$$

$$\sum_{i=1}^N \sum_{v=1}^V x_{ji0t}^v - \sum_{m=1}^N \sum_{v=1}^V x_{j0mt}^v = - \sum_{k=1}^N \sum_{v=1}^V q_{jkt}^v \quad 1 \leq t \leq T, 1 \leq j \leq J \quad (3.7)$$

$$\sum_{j=1}^J x_{jkl t}^v \leq C z_{klt}^v \quad 1 \leq t \leq T, 1 \leq v \leq V, 0 \leq k, l \leq N \mid k \neq l \quad (3.8)$$

$$\sum_{k=0}^N \sum_{\substack{l=0 \\ l \neq k}}^N c_{kl}^v z_{klt}^v \leq L \quad 1 \leq t \leq T, 1 \leq v \leq V \quad (3.9)$$

$$\sum_{k=1}^N z_{0kt}^v \leq 1 \quad 1 \leq t \leq T, 1 \leq v \leq V \quad (3.10)$$

$$\sum_{\substack{i=0 \\ i \neq k}}^N z_{ikt}^v - \sum_{\substack{m=0 \\ m \neq k}}^N z_{kmt}^v = 0 \quad 1 \leq t \leq T, 1 \leq v \leq V, 0 \leq k \leq N \quad (3.11)$$

$$\sum_{k=0}^N \sum_{v=1}^V z_{klt}^v \leq 1 \quad 1 \leq t \leq T, 0 \leq l \leq N \quad (3.12)$$

$$L_{jk} \leq I_{jkt} \leq U_{jk} \quad 1 \leq t \leq T, 1 \leq j \leq J, 0 \leq k \leq N \quad (3.13)$$

$$p_{jt}, q_{jkt}^v, x_{jkl t}^v \geq 0, y_{jt}, z_{klt}^v \in \{0, 1\}, 1 \leq t \leq T, 1 \leq j \leq J, 0 \leq k, l \leq N \quad (3.14)$$

A primeira parte da função objetivo (3.1) tem os custos relacionados ao dimensionamento de lotes: estoque, produção, preparação e a segunda tem a parte de distribuição,

com os custos de utilização dos veículos e transporte. As restrições (3.2) representam o balanço entre produção, estoque e quantidade de itens entregues na fábrica. As restrições (3.3) também são de balanço, mas agora entre a quantidade de itens entregues, estoque e demanda no cliente. As restrições (3.4) garantem que a capacidade da fábrica é respeitada e as restrições (3.5) mostram quando a preparação de máquinas é necessária. As restrições (3.6) e (3.7) representam o fluxo de itens que passa pelos clientes e fábrica. As restrições (3.8) e (3.9) são as capacidades dos veículos, de carga e percurso (o custo de transporte na restrição funciona como se fosse combustível, limitando a distância percorrida e minimizando o seu gasto). As restrições (3.10) estabelecem no máximo uma rota para cada veículo por período e as restrições (3.11) forçam que os veículos terminem os períodos na fábrica. As restrições (3.12) garantem que no máximo um veículo visita cada cliente por período, proibindo a entrega parcial. As restrições (3.13) garantem que os limites de estoque serão respeitados. As restrições (3.14) definem os domínios das variáveis do problema. Fumero e Vercellis (1999) afirmaram que as restrições (3.6) e (3.7) eliminam as sub-rotas, que é quando um veículo passa duas vezes por um mesmo cliente num único período. Da restrição (3.6) junto com a não negatividade da variável q_{jkt}^v , eliminamos a possibilidade de um veículo pegar itens de um cliente e levar a outro.

CAPÍTULO 4

ABORDAGENS DE SOLUÇÃO

Devido ao tamanho do problema descrito no Capítulo 3, a resolução do modelo *MIP* (*Mixed Integer Programming*) apresentado utilizando o *solver* Cplex é limitada. Portanto utilizamos heurísticas, como as de decomposição *relax-and-fix* e *fix-and-optimize*. Ambas heurísticas já foram bastante utilizadas na literatura, principalmente nos problemas de dimensionamento de lotes (*lot sizing*), o que serve como indicativo de que são métodos cujos resultados são competitivos nesse tipo de problema.

Entre os estudos relacionados ao dimensionamento de lotes, Pochet e Wolsey (2006) utilizam *relax-and-fix* para obter limitantes de forma rápida. Furlan (2011) e Baldo (2010) encontram uma solução inicial com *relax-and-fix* e, em seguida, aplicam a *fix-and-optimize* como busca local. Furlan (2011) ainda combina *fix-and-optimize* com o algoritmo de abelhas (Pham et al. (2011)). Sahling et al. (2009) e Helber e Sahling (2010) utilizaram a *fix-and-optimize* pela sua flexibilidade e adaptabilidade, como a heurística necessita de uma solução incumbente, eles simplificaram com uma solução inicial cujas variáveis binárias (de preparação) são todas iguais a 1. Chen (2015) combina a *fix-and-optimize* com um método VNS (*Variable Neighborhood Search*) para tentar fugir de ótimos locais. Com exceção de Chen (2015), que fez uma decomposição genérica que agrupa as variáveis binárias de acordo com seus níveis de proximidade (influência entre as variáveis), os trabalhos citados utilizam decomposições baseadas nas características do problema como, por exemplo: períodos, itens, máquinas, processos, etc.

Em artigos referentes ao problema integrado, podemos citar Liberalino et al. (2011) que utilizaram heurísticas baseadas na *relax-and-fix*. O problema considerado por Liberalino et al. (2011) possui múltiplas fábricas e itens, porém cada fábrica só pode produzir um único item e alguns itens podem ser necessários para a produção de outros (multietágio). Toda a produção é direcionada a um nó (várias fábricas produzindo itens para um cliente em comum).

Neste capítulo apresentamos os métodos propostos e suas implementações.

4.1 *Relax-and-fix* (RF)

A *relax-and-fix* (Pochet e Wolsey (2006)) é uma heurística construtiva de decomposição. É uma decomposição, pois dividimos um conjunto de variáveis inteiras (S) em S^* subconjuntos disjuntos, ou seja,

$$\begin{aligned} S(1) \cup S(2) \cup \dots \cup S(S^*) &= I, \\ S(1) \cap S(2) \cap \dots \cap S(S^*) &= \emptyset \end{aligned}$$

e construtiva, pois a solução é construída enquanto resolvemos subproblemas.

Uma melhoria para o método é a inclusão de sobreposição (*overlapping*), porém o custo computacional nem sempre vale a pena. A sobreposição consiste em integralizar variáveis a mais, sem fixá-las no final da iteração, o que aproxima os subproblemas do problema original e pode resultar em soluções mais interessantes.

Um algoritmo para um procedimento da heurística *relax-and-fix* é dado na Tabela 2. No subproblema $MIP(i)$ do Passo 3, as variáveis dos conjuntos $S(0)$, $S(1)$, ..., $S(i-1)$ já foram fixadas com valores inteiros (ou binários) em $MIPs$ anteriores. As variáveis do conjunto $S(i)$, e outras variáveis de *overlapping*, são integralizadas, e as variáveis restantes continuam relaxadas.

No exemplo da Figura 1, as partições verde-escuras são as que serão fixadas no fim da iteração, as verde-claras são variáveis sobrepostas e as partições pretas já estavam fixadas antes da iteração. Na primeira iteração, por exemplo, a partição $S(2)$ foi separada em duas pois a sobreposição não precisa ser, necessariamente, a próxima partição toda. E a sobreposição não precisa conter variáveis de apenas uma partição, ela poderia ter elementos das partições $S(2)$ e $S(3)$, por exemplo.

Tabela 2: Pseudo algoritmo *relax-and-fix*

Passo 0	Escolher uma decomposição (em S^* subconjuntos disjuntos) para um conjunto de variáveis inteiras S ;
Passo 1	Relaxar todas as variáveis de S (podem assumir valores reais);
Passo 2	$i = 0$;
Passo 3	Resolver o subproblema ($MIP(i)$) obtido após integralizar as variáveis da partição i (e <i>overlap</i>);
Passo 4	Fixar apenas as variáveis da partição $S(i)$, considerando ou não <i>overlap</i> (variáveis fixadas não podem ser alteradas na resolução dos próximos subproblemas);
Passo 5	$i++$;
Passo 6	Se $i \leq S^*$, voltar para o Passo 3. Terminar caso contrário.

Figura 1: *Relax-and-fix*

4.1.1 Decomposições

Neste problema, aplicaremos a *relax-and-fix* nas variáveis binárias: preparação de produção $Y = \{y_{jt} \mid j = 1, \dots, J, t = 1, \dots, T\}$, e distribuição $Z = \{z_{klt}^v \mid v = 1, \dots, V, t = 1, \dots, T, k = 0, \dots, N, l = 0, \dots, N\}$.

Vamos definir os conjuntos $Y_t = \{y_{jt}, j = 1, \dots, J\}$ e $Z_t = \{z_{klt}^v, v = 1, \dots, V, k = 0, \dots, N, l = 0, \dots, N\}$ como mostra a Tabela 3. A seguir, apresentamos as decomposições utilizadas.

Tabela 3: Conjuntos de váriaveis

Períodos	1	2	3	...	T
Produção	Y_1	Y_2	Y_3	...	Y_T
Distribuição	Z_1	Z_2	Z_3	...	Z_T

4.1.1.1 Por problemas

Essa decomposição separa a produção do roteamento, sendo a partição $S(1)$ composta pelas variáveis de preparação e $S(2)$ pelas de roteamento. Ou seja, $S(1) = Y$ e $S(2) = Z$. Ao aplicar esta decomposição (Figura 2), o primeiro *MIP* é resolvido com todas as variáveis binárias de roteamento ($S(2)$) relaxadas (com valores reais entre 0 e 1 ao invés

de binárias), e o segundo *MIP* tem as variáveis binárias de produção ($S(1)$) fixadas de acordo com a solução obtida na *MIP*(1), com as de roteamento voltando a ser binárias.

Figura 2: Decomposição por problemas

4.1.1.2 Por períodos

Esta decomposição pode ser aplicada progressivamente no tempo, iniciando no primeiro período (*forward*), ou regressivamente começando pelo último período (*backward*). Sejam $S(i) = \{y_{jt}, z_{klt}^v \mid j = 1, \dots, J, v = 1, \dots, V, k = 0, \dots, N, l = 0, \dots, N, t \in T_i\}$ os conjuntos de variáveis, onde T_i são as partições de períodos tais que $T_1 \cup T_2 \cup \dots \cup T_n = \{1, 2, \dots, T\}$. Na Figura 3, um exemplo agrupando os períodos em pares, sem *overlap*:

Figura 3: Decomposição por períodos

4.1.1.3 Mista

Também é possível combinar as duas decomposições (por problemas e períodos), separando as variáveis em dois subconjuntos (y_{jt} e z_{klt}^v) e também por períodos, como ilustra a Figura 4.

Figura 4: Decomposição mista por problemas e períodos

É claro que existem muito mais formas de particionar o conjunto das variáveis binárias. Além das citadas, podemos decompor por itens, clientes, veículos e combinações entre elas.

4.2 Heurística construtiva de Lei et al. (2006)

Testamos outra forma de obter uma solução inicial, pois a *relax-and-fix* teve dificuldades com instâncias grandes (a partir de 30 clientes), que é a solução inicial obtida na fase 1 proposta por Lei et al. (2006). O modelo (3.1) - (3.14) é resolvido acrescentando a restrição (4.1).

$$z_{klt}^v = 0 \quad 1 \leq t \leq T; 1 \leq v \leq V; 1 \leq k, l \leq N \quad (4.1)$$

Ou seja, esse novo problema permite apenas rotas diretas entre a fábrica e os clientes. O objetivo é reduzir o problema e encontrar uma solução factível com mais facilidade, em um espaço de soluções reduzido, e testar se a heurística de melhoria consegue encontrar uma solução competitiva.

4.3 *Fix-and-Optimize* (FO)

A heurística *fix-and-optimize* (Pochet e Wolsey (2006), Sahling et al. (2009)) é uma heurística de melhoria que também particiona o conjunto de variáveis. Ela pode utilizar as mesmas decomposições propostas para a *relax-and-fix*, porém necessita de uma solução inicial por ser uma heurística de melhoria.

A Tabela 4 apresenta um algoritmo para um procedimento da heurística *fix-and-optimize*. No subproblema do Passo 3, todas as variáveis, exceto as do conjunto $S(i)$ e outras variáveis de *overlapping*, estão fixadas.

A Figura 1 também pode ser utilizada para ilustrar a *fix-and-optimize*, com as partições verde (claras e escuras) representando as variáveis relaxadas e fixadas ao término da iteração, e as pretas são as variáveis que já foram fixadas pela *fix-and-optimize*. Porém existe a chance do método retornar à primeira iteração (que seria a $(R+1)$ -ésima iteração) após a última.

Tabela 4: Pseudo algoritmo *fix-and-optimize*

Passo 0	Escolher uma decomposição (em S^* subconjuntos disjuntos) para um conjunto de variáveis inteiras S ;
Passo 1	Encontrar solução inicial e fixar as variáveis de S (variáveis fixadas não podem ser alteradas na resolução de subproblemas);
Passo 2	$i = 0, j = 0$;
Passo 3	Resolver o subproblema obtido após liberar (continuam inteiras) as variáveis da partição i (e <i>overlap</i>);
Passo 4	Se a solução nova for melhor, fixar as variáveis que foram liberadas no Passo 3 e $j = 1$;
Passo 5	$i++$;
Passo 6	Se $i \leq S^*$, voltar para o Passo 3. Se ($i > S^*$ e $j > 0$), voltar para o Passo 2, terminar caso contrário.

4.3.1 Decomposições

As decomposições utilizadas durante a *relax-and-fix* podem ser utilizadas na *fix-and-optimize*.

A seguir apresenta-se um exemplo de decomposição por períodos, considerando que cada partição contém as variáveis de um único período t , ou seja, em cada subproblema, as variáveis de um t -ésimo período são liberadas, enquanto as outras permanecem fixadas no valor da solução incumbente. Na Figura 5, as variáveis dos períodos verdes são desafixadas para a resolução da *MIP*, e a $(T+1)$ -ésima iteração é a primeira do segundo ciclo, supondo que seja necessário.

Figura 5: Decomposição por períodos

4.3.2 Heurística combinada

A combinação de heurísticas consiste em encontrar uma solução inicial com a heurística construtiva de Lei et al. (2006) ou *relax-and-fix*, e realizar uma busca local com a *fix-and-optimize*. Todas as combinações utilizadas são apresentadas no Capítulo 5.

CAPÍTULO 5

TESTES COMPUTACIONAIS

5.1 Instâncias utilizadas

5.1.1 Geração de dados

O modelo utilizado nesta dissertação foi o apresentado na Seção 3.2. Como não encontramos instâncias de múltiplos itens disponíveis na literatura, tivemos que gerar instâncias para testar o modelo e os métodos. Nossas instâncias geradas foram baseadas em Armentano et al. (2011).

Foram geradas instâncias de pequeno porte, cujas quantidade de clientes, períodos e itens foram $\{5, 10, 15\}$, $\{7, 14\}$ e $\{3, 5\}$, respectivamente (Tabela 1). E instâncias maiores, com quantidades de clientes, períodos e itens aumentadas para $\{30, 50, 100\}$, $\{12, 24\}$ e $\{5, 10\}$ (Tabela 2), o que nos dá um total de 12 combinações distintas para cada tipo de instância. Foram geradas 10 instâncias para cada uma das 12 combinações, sendo metade delas com um estoque inicial reduzido pela metade, totalizando 240 instâncias. Como não consideramos entregas parciais, a quantidade de veículos disponíveis foi igual a quantidade de clientes.

As demandas, coordenadas dos clientes, limites de estoque, estoques iniciais e custos de estoque nos clientes são gerados aleatoriamente, com distribuição aleatória uniforme. Não existe tempo de preparação e o custo de preparação é proporcional ao custo de estoque na fábrica. Seja $D_{total} = \sum_j \sum_k \sum_t d_{jkt}$.

Tabela 1: Instâncias geradas de pequeno porte

Instâncias	N (V)	T	J
1 .. 10	5	7	3
11 .. 20	5	7	5
21 .. 30	5	14	3
31 .. 40	5	14	5
41 .. 50	10	7	3
51 .. 60	10	7	5
61 .. 70	10	14	3
71 .. 80	10	14	5
81 .. 90	15	7	3
91 .. 100	15	7	5
101 .. 110	15	14	3
111 .. 120	15	14	5

Tabela 2: Instâncias geradas de grande porte

Instâncias	N (V)	T	J
121 .. 130	30	12	5
131 .. 140	30	12	10
141 .. 150	30	24	5
151 .. 160	30	24	10
161 .. 170	50	12	5
171 .. 180	50	12	10
181 .. 190	50	24	5
191 .. 200	50	24	10
201 .. 210	100	12	5
211 .. 220	100	12	10
221 .. 230	100	24	5
231 .. 240	100	24	10

A seguir, apresentamos os parâmetros utilizados para geração das instâncias.

Demanda dos clientes	$d_{jkt} \in [10, 100]$
Estoque mínimo nos clientes	$L_{jk} \in [50, 150]$
Estoque máximo nos clientes	$U_{jk} = L_{jk} + g_{jk} \sum_{t \in T} (d_{jkt}/T)$
Parâmetro de peso	$g_{jk} \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$
Estoque inicial nos clientes	$I_{jk0} \in [0, 100]$
Estoque inicial na fábrica	$I_{j00} = \sum_{k \in N} (U_{jk} - L_{jk})$
Custo de estoque nos clientes	$h_{jk} \in [0.1, 1]$
Custo de estoque na fábrica	$h_{j0} = \{0.3, 0.8\}$
Custo de preparação das máquinas	$f_j^p = 1000h_{j0}$
Tempo de produção dos itens	$b_j = 1$
Capacidade de produção da fábrica	$B = 2(D_{total}/T)$
Custo de transporte fixo	$f^v = (N + 1) \max_{k,l \in N} c_{kl}^v$
Custo de transporte	$c_{kl}^v = \lfloor \sqrt{(x_k - x_l)^2 + (y_k - y_l)^2} \rfloor$
Posição da fábrica	$x_0 = y_0 = 0$
Posição dos clientes	$x_k \in [0, 500]$ e $y_k \in [0, 1000]$
Capacidade dos veículos	$C = J(\max_{j \in J, k \in N} U_{jk})$
Comprimento máximo de rota	$L = r(\max_{k,l \in N} c_{kl}^v), r \in [5, 20]$

As adaptações realizadas para a geração das instâncias foram reduções do estoque inicial nos clientes, da capacidade de produção da fábrica e do peso g_{jk} , que afeta o estoque inicial na fábrica, além de um novo multiplicador na capacidade dos veículos para compensar a redução de U_{jk} . Essas alterações foram necessárias pois Armentano et al. (2011) e Bertazzi et al. (2002) focaram mais nas partes de estoque e roteamento, dando menos importância à produção. Com as adaptações realizadas, espera-se obter instâncias mais “apertadas” na parte de produção, ou seja, queremos que a fábrica tenha um estoque inicial menor para aumentar a produção de itens.

Os códigos (gerador de instâncias e heurística) utilizados nesse mestrado, assim como as instâncias e resultados obtidos, podem ser encontrados em <http://www.otm.icmc.usp.br/index.php/pt/pesquisa/instancias-de-teste>

5.2 Ambiente de testes

Pesquisa desenvolvida com utilização dos recursos computacionais do Centro de Ciências Matemáticas Aplicadas à Indústria (CeMEAI) financiados pela FAPESP. As heurísticas foram implementadas em C++ e resolvidas, com o auxílio do *solver* Cplex 12.6, no *cluster* de alto desempenho Euler, do CeMEAI localizado no Instituto de Ciências Matemáticas e de Computação (ICMC) da USP São Carlos. O *cluster* possui 104 lâminas computacionais IP113, com 2 processadores Intel Xeon E5-2680v2 de 2.8 GHz com dez núcleos e 128 GB DDR3 1866MHz de memória por lâmina, totalizando 2080 núcleos de processamento (4120 com Hyper-threading) e 13 TB de memória. Também possui um nó dedicado com 2 Processadores Intel Xeon E5-2680v2 de 2.8 GHz com dez núcleos, 128 GB DDR3 1866MHz de memória e 1 Xeon Phi C2108-RP2 (8GB, 1.053 GHz, 60 núcleos de processamento). Foi dado um tempo limite de uma hora para a resolução de cada instância, tanto para as heurísticas quanto para o Cplex. Para as heurísticas, foi dado um tempo máximo de 20 minutos para a *relax-and-fix* e 5 minutos para a heurística construtiva de Lei et al. (2006), e o resto do tempo utilizado pela melhoria *fix-and-optimize*, com 5 ou 10 minutos por subproblema (iteração).

5.3 Testes preliminares

Como a qualidade de cada heurística depende dos parâmetros (decomposições e tempo máximo de cada subproblema por exemplo) escolhidos, mostramos, nesta seção, os testes preliminares realizados em algumas das instâncias geradas que serviram para verificar qual combinação de decomposições obteve melhores resultados no nosso problema em particular. As variáveis binárias do nosso problema são y_{jt} e z_{klt}^v , ou seja, têm apenas o índice t em comum. Então escolhemos decomposições por períodos para agrupá-las. Como estamos interessados mais na qualidade da solução do que no tempo computacional, e o limite de tempo é baixo (60 minutos), não vamos apresentar as médias de tempos para os testes preliminares.

As decomposições podem ser combinadas de diversas maneiras. Nesta dissertação, foram consideradas as variáveis binárias particionadas por períodos consecutivos, com α períodos por subproblema, sendo β períodos de sobreposição. α variou entre 1 e 4, enquanto β ficou no intervalo de 0 a 3 (e $\beta \leq \alpha$). As combinações testadas estão na Tabela 3. A RF6, por exemplo, relaxa as variáveis binárias de 3 períodos por iteração, resolve o subproblema e fixa as variáveis de 1 período, pois os outros 2 períodos são apenas

de sobreposição. Já a FO6 libera e fixa as variáveis binárias de 3 períodos por iteração, porém só avança 1 período na próxima iteração. A Figura 3 é um exemplo da RF2, e a Figura 5 da FO1.

Também utilizamos essas decomposições regressivas no tempo, tanto para a *relax-and-fix* quanto para a *fix-and-optimize*, ou seja, as variáveis do período T estão na primeira partição, e a última partição contém as variáveis do período 1. Denotamos estas heurísticas com os índices bRF/bFO1 - bRF/bFO10, seguindo a mesma ordem, ou seja, bRF1 tem a mesma decomposição que RF1 porém com o passo regressivo no tempo.

Tabela 3: Decomposições utilizadas

Nome	Tipo	α	β
RF1/FO1	períodos	1	0
RF2/FO2	períodos	2	0
RF3/FO3	períodos	3	0
RF4/FO4	períodos	2	1
RF5/FO5	períodos	3	1
RF6/FO6	períodos	3	2
RF7/FO7	períodos	4	3
RF8/FO8	períodos	4	2
RF9/FO9	períodos	4	1
RF10/FO10	períodos	$T/2$	$\alpha/2$
RF11	mista	3	0

Para comparar as heurísticas entre si, e verificar as melhores combinações de acordo com a Tabela 3, utilizamos os perfis de desempenhos propostos por Dolan e Moré (2002). O perfil de desempenho de uma heurística é uma função de distribuição calculada pela comparação de seu desempenho com a de outras. Neste trabalho, o desempenho das heurísticas que estamos interessados é o custo total das soluções, e para calcular o perfil de desempenho de uma heurística, definimos

$s_{p,h}$ = função objetivo da instância p encontrada pela heurística h .

e calculamos o quociente de performance:

$$r_{p,h} = \frac{s_{p,h}}{\min\{s_{p,h} | h \in H\}}$$

onde H é o conjunto de heurísticas. Ou seja, estamos calculando o quão perto a solução de h está da melhor solução obtida pelas heurísticas em H , com $r_{p,h} = 1$ quando h obteve a melhor solução para o problema p , e $r_{p,h} > 1$ caso contrário. É definido um r_M tal que

$r_M \geq r_{p,h}, \forall p, h$, e $r_{p,h} = r_M$ se, e somente se, h não conseguir uma solução factível para p .

Com os quocientes de performance, construímos o gráfico de desempenho que mostra a porcentagem de instâncias p nas quais o método h obteve $r_{p,h} \leq \tau$, com τ variando no eixo horizontal. Dolan e Moré (2002) utilizaram a notação $\rho_h(\tau)$ para indicar essa porcentagem, nós utilizamos $\rho(h, \tau)$. Logo, $\rho(h, 1)$ é a quantidade de instâncias que o método h encontrou a melhor solução, e $\rho(h, r_M) = 1$ em todas as heurísticas.

5.3.1 Qualidade da solução inicial

Combinando uma decomposição para a heurística de melhoria *fix-and-optimize* com várias decomposições para a *relax-and-fix*, conforme ilustra a Figura 6. O objetivo desta comparação é encontrar e selecionar a melhor *relax-and-fix*, quando combinada com uma *fix-and-optimize* que utiliza a decomposição por períodos com passos progressivos (FO1) ou regressivos (bFO1). Observa-se que $\alpha_{FO1} = \alpha_{bFO1} = 1$ e $\beta_{FO1} = \beta_{bFO1} = 0$.

Figura 6: Testes preliminares variando a *relax-and-fix*

A Tabela 4 mostra os testes com buscas locais FO1 e bFO1, e soluções iniciais conforme a Tabela 3, em um grupo de 60 instâncias conforme a primeira coluna indica. Cada linha mostra a média, das 10 instâncias correspondentes, do *gap* entre as soluções encontradas pelas heurísticas e o limitante inferior encontrado pelo Cplex, em uma hora, enquanto resolvia o modelo. Os números em negrito são as menores médias em cada linha. O Cplex leva vantagem nas instâncias pequenas (1-30), obtendo os melhores *gaps*, algumas heurísticas chegam perto dos valores obtidos pelo Cplex. Levando apenas a Tabela 4 em consideração, escolheríamos RF7, RF8, RF9 ou RF10 como melhores heurísticas *relax-and-fix*, cujos *gap* médio estão entre 4,84% e 4,86%, que são melhores que o *gap* de 5,05% do Cplex.

As soluções foram melhorando conforme α aumentava, por isso testamos a decomposição RF10 para ver sua performance em instâncias com uma quantidade de períodos maior. RF11 foi uma decomposição mista que obteve valores de solução muito piores que RF3 (ambas têm $\alpha = 3$ e $\beta = 0$), então não investimos mais tempo nas decomposições mistas (apesar de um caso particular ruim não garantir que todas serão ruins). A *relax-and-fix* utilizando partições por problemas também apresentou soluções de baixa qualidade e não foi apresentada nos resultados.

Tabela 4: Testes preliminares - *gaps* médios das heurísticas

Instâncias	N(V)	T	J	RF1	RF2	RF3	RF4	RF5	RF6	RF7	RF8	RF9	RF10	RF11	bRF1	bRF2	bRF3	bRF4	bRF5	bRF6	bRF11
1..10	5	7	3	9,86%	5,19%	8,12%	5,03%	4,12%	3,37%	3,38%	3,38%	3,38%	3,59%	13,19%	4,74%	4,66%	4,21%	4,04%	3,77%	3,46%	4,55%
11..20	5	7	5	11,10%	5,86%	7,96%	5,09%	4,55%	3,07%	2,96%	2,96%	2,96%	3,01%	13,12%	5,96%	6,31%	5,55%	5,00%	4,79%	3,18%	6,49%
21..30	5	14	3	10,75%	6,41%	6,39%	6,57%	6,50%	5,04%	4,87%	4,87%	4,87%	4,86%	11,11%	7,72%	7,67%	7,94%	7,48%	6,31%	5,61%	9,16%
31..40	5	14	5	9,18%	5,44%	6,29%	5,85%	6,30%	4,99%	4,62%	4,61%	4,61%	4,64%	9,83%	6,69%	6,80%	6,14%	5,76%	5,83%	5,08%	8,33%
41..50	10	7	3	9,93%	7,88%	8,17%	7,76%	7,81%	7,65%	7,58%	7,59%	7,60%	7,53%	9,84%	8,14%	8,63%	8,20%	8,45%	7,59%	7,94%	8,56%
51..60	10	7	5	7,86%	6,13%	6,31%	5,85%	6,10%	5,55%	5,67%	5,63%	5,70%	5,55%	10,24%	6,30%	6,43%	7,71%	6,54%	7,19%	5,79%	6,28%
Total				9,78%	6,15%	7,21%	6,03%	5,90%	4,95%	4,85%	4,84%	4,86%	4,86%	11,22%	6,59%	6,75%	6,62%	6,21%	5,91%	5,18%	7,23%

Testes com FO1

Instâncias	N(V)	T	J	RF1	RF2	RF3	RF4	RF5	RF6	RF7	RF8	RF9	RF10	RF11	bRF1	bRF2	bRF3	bRF4	bRF5	bRF6	Cplex
1..10	5	7	3	9,89%	5,23%	8,12%	5,07%	4,12%	3,34%	3,45%	3,45%	3,49%	3,65%	13,12%	4,57%	4,68%	4,21%	4,04%	3,77%	3,55%	3,04%
11..20	5	7	5	11,10%	5,89%	7,96%	5,12%	4,53%	3,08%	3,00%	3,00%	3,00%	3,05%	13,01%	6,06%	6,31%	5,58%	4,99%	4,78%	3,70%	2,79%
21..30	5	14	3	10,75%	6,40%	6,38%	6,36%	6,49%	6,11%	4,96%	4,96%	6,05%	4,93%	10,81%	7,71%	7,68%	7,93%	7,51%	6,31%	6,31%	4,86%
31..40	5	14	5	9,20%	5,45%	6,29%	5,86%	6,31%	4,98%	4,64%	4,64%	5,53%	4,71%	9,77%	6,72%	6,83%	6,14%	5,76%	5,83%	5,20%	4,81%
41..50	10	7	3	9,24%	7,97%	8,29%	7,86%	7,84%	8,23%	7,75%	7,84%	7,69%	7,71%	10,16%	8,19%	8,68%	8,13%	8,40%	7,53%	8,36%	8,19%
51..60	10	7	5	8,37%	6,09%	6,77%	5,89%	6,35%	5,41%	5,85%	6,00%	5,95%	6,01%	9,80%	6,96%	6,39%	7,71%	5,85%	7,21%	5,66%	6,60%
Total				9,76%	6,17%	7,30%	6,03%	5,94%	5,19%	4,94%	4,98%	5,29%	5,01%	11,11%	6,70%	6,76%	6,62%	6,09%	5,90%	5,46%	5,05%

Testes com bFO1

Nos primeiros testes, que resultaram na Tabela 4, limitamos em 20 minutos, dentro de um tempo máximo de 60 minutos, o tempo para a *relax-and-fix* encontrar uma solução inicial, com o restante do tempo para a *fix-and-optimize* fazer melhorias, com um limite de 10 minutos por subproblema. Nos testes seguintes, diminuimos o tempo limite por subproblema da *fix-and-optimize* para 5 minutos cada. Com essa alteração, vamos aumentar a quantidade de subproblemas resolvidos, ou seja, estamos procurando soluções em uma quantidade maior de vizinhanças, ao invés de gastar muito tempo procurando o ótimo local em cada subproblema. O *gap* médio total melhorou para 3 das 4 heurísticas testadas, como mostra a Tabela 5, escolhemos apenas as heurísticas RF7, RF8, RF9 e RF10 porque foram as que se destacaram mais até o momento. A partir daí, o resto dos testes foi realizado com esses novos parâmetros, ou seja, 20 minutos para a *relax-and-fix* e 5 minutos por subproblema da *fix-and-optimize*.

Tabela 5: Testes preliminares - *gaps* de 5 minutos contra 10 minutos

Instâncias	N (V)	T	J	RF7	RF8	RF9	RF10	RF7	RF8	RF9	RF10
1 .. 10	5	7	3	3,38%	3,59%	3,41%	3,59%	3,38%	3,38%	3,38%	3,59%
11 .. 20	5	7	5	2,96%	3,01%	2,98%	3,01%	2,96%	2,96%	2,96%	3,01%
21 .. 30	5	14	3	4,87%	4,84%	5,45%	4,90%	4,87%	4,87%	4,87%	4,86%
31 .. 40	5	14	5	4,61%	4,41%	5,11%	4,66%	4,62%	4,61%	4,61%	4,64%
41 .. 50	10	7	3	7,47%	7,42%	7,52%	7,44%	7,58%	7,59%	7,60%	7,53%
51 .. 60	10	7	5	5,41%	5,56%	5,54%	5,57%	5,67%	5,63%	5,70%	5,55%
Total				4,79%	4,80%	5,00%	4,86%	4,85%	4,84%	4,86%	4,86%

5 minutos de FO1

10 minutos de FO1

Os gráficos nas Figuras 7 e 8, que comparam os perfis de desempenhos das heurísticas com 5 e 10 minutos por iteração na *fix-and-optimize* também destacam, com exceção da RF9, a vantagem da nova parametrização de 5 minutos. A Figura 8 tem as mesmas linhas da Figura 7, apenas pintamos as linhas de 5 minutos de vermelho e as de 10 minutos de verde para facilitar a visualização.

Figura 7: Perfis de desempenho das heurísticas RF7 - RF10 com 5 e 10 minutos por iteração da *fix-and-optimize* (FO1)

Figura 8: Vermelho (5 minutos) x Verde (10 minutos)

Tabela 6: Testes preliminares - *gaps* com 5 minutos por iteração da *fix-and-optimize* (FO1 e bFO1)

Instâncias	N (V)	T	J	RF1	RF2	RF3	RF4	RF5	RF6	RF7	RF8	RF9	RF10	bRF1	bRF2	bRF3	bRF4	bRF5	bRF6
1 .. 10	5	7	3	7,25%	4,80%	6,65%	3,78%	3,60%	3,37%	3,38%	3,38%	3,38%	3,59%	4,31%	4,08%	3,74%	3,60%	3,28%	3,34%
11 .. 20	5	7	5	7,05%	4,48%	5,11%	3,19%	3,09%	3,07%	2,96%	2,96%	2,96%	3,01%	4,03%	4,52%	4,18%	3,38%	3,10%	3,08%
21 .. 30	5	14	3	7,69%	5,50%	5,54%	5,08%	5,21%	5,04%	4,87%	4,87%	4,87%	4,86%	6,68%	6,83%	6,90%	6,66%	5,30%	6,11%
31 .. 40	5	14	5	6,45%	4,58%	5,41%	5,11%	5,23%	5,00%	4,62%	4,61%	4,61%	4,64%	5,56%	5,82%	5,06%	4,94%	5,03%	4,98%
41 .. 50	10	7	3	9,75%	7,64%	8,05%	7,37%	7,62%	7,45%	7,58%	7,59%	7,60%	7,53%	8,08%	8,50%	7,87%	8,36%	8,29%	8,02%
51 .. 60	10	7	5	6,60%	5,71%	5,79%	5,47%	5,14%	5,57%	5,67%	5,63%	5,70%	5,55%	5,96%	5,90%	5,97%	5,24%	5,48%	5,29%
Total				7,46%	5,45%	6,09%	5,00%	4,98%	4,92%	4,85%	4,84%	4,86%	4,86%	5,77%	5,94%	5,62%	5,36%	5,08%	5,14%

Testes com FO1

Instâncias	N (V)	T	J	RF1	RF2	RF3	RF4	RF5	RF6	RF7	RF8	RF9	RF10	bRF1	bRF2	bRF3	bRF4	bRF5	bRF6	Cplex
1 .. 10	5	7	3	7,61%	5,01%	6,82%	4,08%	3,68%	3,46%	3,45%	3,45%	3,49%	3,65%	5,24%	4,55%	3,96%	3,67%	3,54%	3,55%	3,04%
11 .. 20	5	7	5	7,87%	4,73%	5,24%	3,34%	3,12%	3,18%	3,00%	3,00%	3,00%	3,05%	4,90%	4,97%	4,45%	3,52%	3,35%	3,70%	2,79%
21 .. 30	5	14	3	7,89%	5,66%	5,58%	5,66%	5,85%	5,61%	4,96%	4,96%	6,05%	4,93%	7,25%	7,02%	7,00%	6,85%	5,39%	6,31%	4,86%
31 .. 40	5	14	5	6,68%	4,66%	5,47%	5,50%	5,34%	5,10%	4,64%	4,64%	5,53%	4,71%	6,51%	6,05%	5,18%	5,06%	5,20%	5,20%	4,81%
41 .. 50	10	7	3	9,87%	7,80%	8,32%	7,67%	7,89%	7,82%	7,75%	7,84%	7,69%	7,71%	8,37%	8,58%	8,81%	8,65%	8,80%	8,49%	8,19%
51 .. 60	10	7	5	6,72%	5,80%	6,24%	5,76%	5,62%	5,87%	5,85%	6,00%	5,95%	6,01%	6,83%	6,04%	6,48%	5,57%	6,12%	5,77%	6,60%
Total				7,77%	5,61%	6,28%	5,34%	5,25%	5,18%	4,94%	4,98%	5,29%	5,01%	6,52%	6,20%	5,98%	5,55%	5,40%	5,50%	5,05%

Testes com bFO1

Figura 9: Perfis de desempenho das heurísticas apresentadas na Tabela 6

A Tabela 6 e Figura 9 apresentam os resultados dos testes com 5 minutos por iteração da *fix-and-optimize* (FO1 e bFO1). A Figura 9 mostra o perfil de desempenho de todas as combinações da Figura 6, a linha em destaque representa o Cplex. Podemos ver 3 grupos de heurísticas:

- As linhas isoladas na parte inferior do gráfico são das *relax-and-fix* com decomposição mista (b)RF11, e (b)RF1 que só relaxam um período por vez e não têm *overlap*,

piores que as outras;

- As linhas intermediárias, todas muito próximas umas das outras;
- As linhas na parte superior, cujas heurísticas tiveram o quociente de performance ($r_{p,h}$) menor que 1.05 em mais de 95% das instâncias, ou seja, suas soluções ficaram a, no máximo, 5% de distância das melhores soluções em mais de 95% dos problemas resolvidos.

A Figura 10, que contém apenas o terceiro grupo de heurísticas, as de melhor desempenho, nos permite ver com mais facilidade as melhores heurísticas e que RF6+FO1 e RF7+FO1 são as que se destacam mais. O gráfico mostra uma das intenções de Dolan e Moré (2002), pois as combinações entre RF6 e bRF6 com FO1 e bFO1 não tiveram os melhores *gaps* médios na Tabela 4. Ou seja, essas combinações são boas mas suas médias podem ter sido influenciadas negativamente por causa de uma solução muito ruim, o que não afeta tanto o seu perfil de desempenho. Escolhemos a RF7 para continuar os testes preliminares, ainda falta testar qual decomposição para *fix-and-optimize*, combinada a *relax-and-fix* escolhida, tem o melhor desempenho.

Figura 10: Melhores desempenhos da Figura 9

5.3.2 Qualidade da busca local

Depois de escolher a decomposição para a *relax-and-fix* (RF7, $\alpha_{RF7} = 4$ e $\beta_{RF7} = 3$), combinamos ela com várias decomposições para a *fix-and-optimize*, repetindo o processo realizado para a *relax-and-fix*, com limite de uma hora por problema.

Figura 11: Testes preliminares variando a *fix-and-optimize*

A Tabela 7 apresenta os testes das combinações apresentadas na Figura 11.

Tabela 7: Testes preliminares - *gaps* da solução inicial da RF7 com várias melhorias

Instâncias	N (V)	T	J	FO1	FO2	FO3	FO4	FO5	FO6	FO7	FO8	FO9	FO10
1 .. 10	5	7	3	3,38%	3,30%	3,33%	3,28%	3,28%	3,28%	3,28%	3,21%	3,28%	4,33%
11 .. 20	5	7	5	2,96%	2,94%	2,87%	2,87%	2,87%	2,87%	2,87%	2,87%	2,87%	2,87%
21 .. 30	5	14	3	4,87%	4,85%	4,84%	4,86%	4,84%	4,81%	4,83%	4,79%	4,81%	4,78%
31 .. 40	5	14	5	4,62%	4,58%	4,53%	4,61%	4,51%	4,55%	4,50%	4,74%	4,36%	4,27%
41 .. 50	10	7	3	7,58%	7,42%	7,33%	7,43%	7,42%	7,46%	7,40%	7,41%	7,46%	7,43%
51 .. 60	10	7	5	5,67%	5,35%	5,21%	5,42%	5,30%	5,32%	5,23%	5,20%	5,33%	5,25%
Total				4,85%	4,74%	4,69%	4,75%	4,70%	4,71%	4,69%	4,70%	4,68%	4,82%

Instâncias	N (V)	T	J	bFO1	bFO2	bFO3	bFO4	bFO5	bFO6	bFO7	bFO8	bFO9	bFO10
1 .. 10	5	7	3	3,45%	3,45%	3,45%	3,45%	3,45%	3,45%	3,44%	3,45%	3,45%	3,45%
11 .. 20	5	7	5	3,00%	3,00%	3,00%	3,00%	3,00%	3,00%	3,00%	3,00%	3,00%	3,00%
21 .. 30	5	14	3	4,96%	4,96%	4,96%	4,96%	4,96%	4,92%	4,91%	4,96%	4,96%	4,96%
31 .. 40	5	14	5	4,64%	4,61%	4,61%	4,61%	4,61%	4,59%	4,60%	4,59%	4,61%	4,59%
41 .. 50	10	7	3	7,75%	7,88%	7,80%	7,81%	7,72%	7,56%	7,51%	7,68%	7,76%	7,66%
51 .. 60	10	7	5	5,85%	5,66%	5,74%	5,74%	5,81%	5,49%	5,50%	5,67%	5,92%	5,55%
Total				4,94%	4,93%	4,93%	4,93%	4,93%	4,84%	4,83%	4,89%	4,95%	4,87%

Cplex
3,04%
2,79%
4,86%
4,81%
8,19%
6,60%
5,05%

A Figura 12 mostra o perfil de desempenho das heurísticas apresentadas na Tabela 7, sendo ampliada para a Figura 13, com a linha do Cplex destacada novamente. Retiramos as piores performances e obtivemos a Figura 14.

A Figura 14 confirma o que a Tabela 7 mostrava, a combinação RF7+FO9 é uma das melhores dentre as selecionadas. O gráfico dela atinge 100% antes que as outras, em

Figura 12: Perfis de desempenho das heurísticas apresentadas na Tabela 7

Figura 13: Figura 12 ampliada

Figura 14: Melhores perfis de desempenho, RF7 fixo

$\tau < 1.02$, ou seja, suas soluções não foram mais do que 2% piores que nenhuma solução de outra combinação, em nenhum dos 60 problemas. Por isso escolhemos a RF7+FO9 para resolver as instâncias resultantes. E apesar da vantagem que o Cplex tem em instâncias pequenas (1-20), sua dificuldade com instâncias maiores faz seu desempenho ser um dos piores entre os métodos avaliados.

5.4 Testes

A Tabela 8 apresenta os resultados obtidos pelo Cplex (com a parametrização padrão) e pelas heurísticas propostas (RF7+FO9 e Lei+FO6). A RF7+FO9 foi a heurística escolhida após os testes preliminares, e a Lei+FO6 foi escolhida para encontrar soluções iniciais nas instâncias grandes. Cada linha representa a média de cada método nas 10 instâncias correspondentes. Os *gaps* foram calculados em relação ao limitante inferior encontrado pelo Cplex durante a resolução do modelo. Os números entre parênteses na primeira coluna do RF7+FO9 indicam o número de instâncias não resolvidas pela heurística, provavelmente pelo tamanho do problema. As colunas de melhoria mostram a porcentagem que a busca local (*fix-and-optimize*) melhorou da solução inicial (*relax-and-fix* e construtiva de Lei et al. (2006)). A melhoria de Lei+FO6, combinada com o seu

gap, mostra que a solução inicial tem qualidade ruim, mas a *fix-and-optimize* conseguiu melhorias que a deixaram no mesmo nível da RF7+FO9 e do Cplex para instâncias pequenas, então é possível que uma busca local mais adequada combinada com a heurística contrutiva de Lei et al. (2006) se transforme em um método mais competitivo.

Os tempos acima de uma hora ocorreram devido à uma resolução extra do modelo (5 minutos), para verificar a factibilidade da solução. As instâncias pequenas são as únicas em que o tempo computacional difere, pois os três métodos atingem o limite de tempo a partir da instância 61 ($N = V = 10, T = 14, J = 3$), com a RF7+FO9 sendo a mais rápida, seguida pela Lei+FO6. As heurísticas implementadas tiveram tempos menores que o Cplex nas instâncias pequenas, o que pode compensar a pequena diferença na qualidade da solução.

Tabela 8: Resultados finais

Instâncias	N (V)	T	J	RF7+FO9			Lei+FO6			Cplex	
				<i>gap</i>	melhoria	tempo (s)	<i>gap</i>	melhoria	tempo (s)	<i>gap</i>	tempo (s)
1 .. 10	5	7	3	3,28%	0,18%	818,9	3,24%	47,95%	1352,9	3,04%	2607,1
11 .. 20	5	7	5	2,87%	0,13%	888,1	2,82%	47,59%	1204,7	2,79%	3466,3
21 .. 30	5	14	3	4,81%	0,15%	1586,2	11,54%	42,62%	2776,5	4,86%	3600,0
31 .. 40	5	14	5	4,36%	0,26%	2505,9	15,99%	33,53%	3157,2	4,81%	3600,0
41 .. 50	10	7	3	7,46%	0,36%	2552,7	18,70%	56,77%	3160,3	8,19%	3600,0
51 .. 60	10	7	5	5,33%	0,46%	2594,1	28,00%	45,36%	3461,9	6,60%	3600,0
61 .. 70	10	14	3	5,62%	0,44%	3647,9	44,72%	25,41%	3504,5	6,99%	3600,0
71 .. 80	10	14	5	5,57%	1,06%	3776,4	41,03%	28,36%	3761,4	7,96%	3600,0
81 .. 90	15	7	3	6,35% (1)	2,61%	3442,6	34,00%	50,11%	3598,6	7,21%	3600,0
91 .. 100	15	7	5	6,75%	8,73%	3691,0	35,40%	48,08%	3886,9	7,01%	3600,0
101 .. 110	15	14	3	6,21% (4)	11,90%	3774,3	50,65%	24,91%	3897,6	14,55%	3600,0
111 .. 120	15	14	5	0,66% (7)	27,35%	3862,3	50,84%	25,27%	3908,7	25,85%	3600,0

A Figura 15, ampliada na Figura 16, mostra que, nas instâncias que resolveu, a RF7+FO9 tem desempenho um pouco melhor que o Cplex, sendo ultrapassado pelo Cplex devido às 12 instâncias (10%) que não resolveu. O gráfico também mostra que a qualidade da Lei+FO6 é bem pior, com soluções duas vezes piores em quase 10% das instâncias resolvidas, porém obteve soluções em todas elas.

Encontramos o mesmo problema que Armentano et al. (2011) nas instâncias grandes, nem o *solver* e nem a heurística conseguiram resolvê-las pois são muito grandes. Porém conseguimos encontrar soluções com a heurística construtiva de Lei et al. (2006) e, em alguns casos, melhorá-la com a *fix-and-optimize*. A Tabela 9 mostra que, apesar de encontrar solução inicial, a *fix-and-optimize* tem dificuldades para encontrar soluções melhores. Não conseguimos calcular o *gap* pois essas instâncias são muito grandes e o Cplex não conseguiu executar o modelo.

Figura 15: Perfis de desempenho

Figura 16: Destaque para Cplex x RF7+FO9

Tabela 9: Lei+FO6 nas instâncias grandes

Instâncias	N (V)	T	J	melhoria
121 .. 130	30	12	5	23,40%
131 .. 140	30	12	10	4,14%
141 .. 150	30	24	5	13,44%
151 .. 160	30	24	10	1,87%

A Tabela 10 mostra a dimensão das instâncias geradas, com a modelagem da menor instância grande ($N = V = 30, T = 12, J = 5$) utilizando quase 7 vezes a quantidade de variáveis binárias utilizadas na maior instância pequena ($N = V = 15, T = 14, J = 5$), apesar de manter praticamente a mesma quantidade de períodos e itens.

Tabela 10: Quantidade de variáveis das instâncias

N	T	J	V	Binárias	Reais	Restrições
5	7	3	5	896	3.276	1.960
5	7	5	5	910	5.460	2.492
5	14	3	5	1.792	6.552	3.920
5	14	5	5	1.820	10.920	4.984
10	7	3	10	7.021	23.331	10.500
10	7	5	10	7.035	38.885	12.222
10	14	3	10	14.042	46.662	21.000
10	14	5	10	14.070	77.770	24.444
15	7	3	15	23.646	75.936	30.940
15	7	5	15	23.660	126.560	34.552
15	14	3	15	47.292	151.872	61.880
15	14	5	15	47.320	253.120	69.104
30	12	5	30	324.060	1.675.860	393.672

CAPÍTULO 6

CONSIDERAÇÕES FINAIS

Nesta dissertação apresentamos o problema integrado de produção e distribuição com uma planta, múltiplos itens e frota de veículos homogênea. Neste tipo de problema, deve-se decidir quanto e quando realizar a produção de cada item e o roteamento dos veículos disponíveis de modo a satisfazer a demanda de todos os clientes e minimizar o custo total. Não permitimos nenhum tipo de atraso ou entregas parciais. Fizemos uma revisão e enquadramos o nosso problema na literatura. Observamos que para o problema com as características consideradas, a literatura é escassa. Apresentamos um modelo matemático proposto em Armentano et al. (2011), propomos métodos de soluções, geramos instâncias e testamos os nossos métodos, comparando resultados com o *solver* comercial Cplex. A escassez de instâncias com múltiplos itens faz da geração e disponibilização delas uma contribuição importante.

Os métodos de soluções propostos foram combinações de *relax-and-fix* ou a heurística construtiva de Lei et al. (2006) como solução inicial e *fix-and-optimize* como heurística de melhoria.

Encontramos várias combinações de *relax-and-fix* e *fix-and-optimize* com perfis de desempenhos Dolan e Moré (2002) melhores e piores que o Cplex, como mostram as Figuras 12 e 9, respectivamente. Então é preciso escolher boas decomposições para que o método seja competitivo. Nos testes em instâncias pequenas (1-20), o Cplex obteve soluções melhores que todas as combinações testadas. Nas instâncias maiores, as heurísticas levam

vantagem, porém falham, em alguns casos, e não conseguem encontrar uma solução factível. Após duas séries de testes preliminares, a combinação escolhida foi a RF7+FO9 ($\alpha_{RF7} = 4$ e $\beta_{RF7} = 3$, $\alpha_{FO9} = 4$ e $\beta_{FO9} = 1$), cuja performance foi ligeiramente superior a do Cplex, considerando as instâncias que conseguiu resolver, como mostra a Figura 16.

O Cplex e as heurísticas combinadas *relax-and-fix* e *fix-and-optimize* não conseguiram resolver as instâncias grandes, com 30 clientes ou mais. O método composto pela heurística construtiva de Lei et al. (2006) com *fix-and-optimize* conseguiu obter soluções para as instâncias com 30 clientes, mas as soluções foram pouco melhoradas na *fix-and-optimize*.

Assim como Armentano et al. (2011) e Brahimi e Aouam (2015), deve-se testar a heurística proposta em instâncias da literatura, com um único item, para compará-la à outras abordagens de solução.

As combinações de *relax-and-fix* e *fix-and-optimize* têm potencial, caso os parâmetros sejam bem escolhidos. Existem muitas outras decomposições possíveis, e provavelmente muitas delas podem ter desempenho superior as implementadas neste trabalho. Além das decomposições, vimos que a mudança no limite de tempo para as iterações do *fix-and-optimize* também pode melhorar a performance do método, e como só testamos 2, pode haver espaço para melhorar por este caminho. Desta forma podemos testar, como heurísticas de melhoria, outras decomposições ou implementar uma *fix-and-optimize* parecida com a de Chen (2015), que cria uma distância entre as variáveis e, em cada iteração, escolhe uma variável e um raio, e aplica a *fix-and-optimize* nas variáveis dentro dessa vizinhança. É um problema novo e muitas abordagens ainda podem ser testadas, como outras heurísticas e metaheurísticas. GRASP (Boudia et al. (2007)), algoritmo memético (Boudia e Prins (2009)), busca tabu (Bard e Nananukul (2009), Armentano et al. (2011)), ALNS (Adulyasak et al. (2012)) são exemplos testados na literatura, porém a maioria no PIPD de um único item.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABSI, N.; ARCHETTI, C.; FEILLET, D.; DAUZÈRE-PÉRÈS, S. An iterative two-phase heuristic approach for the production routing problem. *IWLS 2012*, p. 16, 2014.
- ADULYASAK, Y.; CORDEAU, J.-F.; JANS, R. Optimization-based adaptive large neighborhood search for the production routing problem. *Transportation Science*, INFORMS, v. 48, n. 1, p. 20–45, 2012.
- ADULYASAK, Y.; CORDEAU, J.-F.; JANS, R. The production routing problem: A review of formulations and solution algorithms. *Computers & Operations Research*, Elsevier, v. 55, p. 141–152, 2015.
- AGGARWAL, A.; PARK, J. K. Improved algorithms for economic lot size problems. *Operations research*, INFORMS, v. 41, n. 3, p. 549–571, 1993.
- ARCHETTI, C.; BERTAZZI, L.; LAPORTE, G.; SPERANZA, M. G. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, INFORMS, v. 41, n. 3, p. 382–391, 2007.
- ARCHETTI, C.; BERTAZZI, L.; PALETTA, G.; SPERANZA, M. G. Analysis of the maximum level policy in a production-distribution system. *Computers & Operations Research*, Elsevier, v. 38, n. 12, p. 1731–1746, 2011.
- ARMENTANO, V. A.; SHIGUEMOTO, A.; LØKKETANGEN, A. Tabu search with path relinking for an integrated production–distribution problem. *Computers & Operations Research*, Elsevier, v. 38, n. 8, p. 1199–1209, 2011.
- BALDO, T. *O problema integrado de dimensionamento e sequenciamento de lotes no processo de fabricação da cerveja: modelos e métodos de solução*. Tese (Doutorado) — Universidade de São Paulo, 2010.
- BARANY, I.; ROY, T. J. V.; WOLSEY, L. A. Strong formulations for multi-item capacitated lot sizing. *Management Science*, INFORMS, v. 30, n. 10, p. 1255–1261, 1984.
- BARD, J. F.; NANANUKUL, N. The integrated production–inventory–distribution–routing problem. *Journal of Scheduling*, Springer, v. 12, n. 3, p. 257–280, 2009.

BERTAZZI, L.; PALETTA, G.; SPERANZA, M. G. Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science*, INFORMS, v. 36, n. 1, p. 119–132, 2002.

BILLINGTON, P. J.; MCCLAIN, J. O.; THOMAS, L. J. Mathematical programming approaches to capacity-constrained mrp systems: Review, formulation and problem reduction. *Management Science*, INFORMS, v. 29, n. 10, p. 1126–1141, 1983.

BOUDIA, M.; LOULY, M. A. O.; PRINS, C. Combined optimization of production and distribution. In: *IESM'05 (Int. Conf. on Industrial Engineering and Systems Management, Marrakech, 16-19/05/2005)*. [S.l.: s.n.], 2005.

BOUDIA, M.; LOULY, M. A. O.; PRINS, C. A reactive grasp and path relinking for a combined production–distribution problem. *Computers & Operations Research*, Elsevier, v. 34, n. 11, p. 3402–3419, 2007.

BOUDIA, M.; PRINS, C. A memetic algorithm with dynamic population management for an integrated production–distribution problem. *European Journal of Operational Research*, Elsevier, v. 195, n. 3, p. 703–715, 2009.

BRAHIMI, N.; AOUAM, T. Multi-item production routing problem with backordering: a milp approach. *International Journal of Production Research*, Taylor & Francis, p. 1–18, 2015.

CHANDRA, P.; FISHER, M. L. Coordination of production and distribution planning. *European Journal of Operational Research*, Elsevier, v. 72, n. 3, p. 503–517, 1994.

CHEN, H. Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems. *Omega*, Elsevier, v. 56, p. 25–36, 2015.

DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. *Management science*, Informs, v. 6, n. 1, p. 80–91, 1959.

DARVISH, M.; LARRAIN, H.; COELHO, L. C. A dynamic multi-plant lot-sizing and distribution problem. *International Journal of Production Research*, Taylor & Francis, p. 1–11, 2016.

DOLAN, E. D.; MORE, J. J. Benchmarking optimization software with performance profiles. *Mathematical programming*, Springer, v. 91, n. 2, p. 201–213, 2002.

FEDERGRUEN, A.; TZUR, M. A simple forward algorithm to solve general dynamic lot sizing models with n periods in $O(n \log n)$ or $O(n)$ time. *Management Science*, INFORMS, v. 37, n. 8, p. 909–925, 1991.

FEO, T. A.; RESENDE, M. G. Greedy randomized adaptive search procedures. *Journal of global optimization*, Springer, v. 6, n. 2, p. 109–133, 1995.

FUMERO, F.; VERCELLIS, C. Synchronized development of production, inventory, and distribution schedules. *Transportation science*, INFORMS, v. 33, n. 3, p. 330–340, 1999.

FURLAN, M. M. *Métodos heurísticos para o problema de dimensionamento de lotes multiestágio com limitação de capacidade*. Tese (Doutorado) — Universidade de São Paulo, 2011.

- GLOVER, F. Tabu search-part i. *ORSA Journal on computing*, INFORMS, v. 1, n. 3, p. 190–206, 1989.
- GROËR, C.; GOLDEN, B.; WASIL, E. A library of local search heuristics for the vehicle routing problem. *Mathematical Programming Computation*, Springer, v. 2, n. 2, p. 79–101, 2010.
- HELBER, S.; SAHLING, F. A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics*, Elsevier, v. 123, n. 2, p. 247–256, 2010.
- KARIMI, B.; GHOMI, S. F.; WILSON, J. The capacitated lot sizing problem: a review of models and algorithms. *Omega*, Elsevier, v. 31, n. 5, p. 365–378, 2003.
- LEI, L.; LIU, S.; RUSZCZYNSKI, A.; PARK, S. On the integrated production, inventory, and distribution routing problem. *IIE Transactions*, Taylor & Francis, v. 38, n. 11, p. 955–970, 2006.
- LEUNG, J. M.; MAGNANTI, T. L.; VACHANI, R. Facets and algorithms for capacitated lot sizing. *Mathematical programming*, Springer, v. 45, n. 1-3, p. 331–359, 1989.
- LIBERALINO, H.; DUHAMEL, C.; QUILLIOT, A.; ALOISE, D. J. Modelo exato e mateheurísticas para o problema de dimensionamento de lotes integrado ao transporte. 2011.
- LIN, S.; KERNIGHAN, B. W. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, INFORMS, v. 21, n. 2, p. 498–516, 1973.
- MANNE, A. S. Programming of economic lot sizes. *Management science*, INFORMS, v. 4, n. 2, p. 115–135, 1958.
- MOSCATO, P. et al. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, Citeseer, v. 826, p. 1989, 1989.
- MOSTAFA, N. A.; ELTAWIL, A. B. The production-inventory-distribution-routing problem: An integrated formulation and solution framework. In: IEEE. *Industrial Engineering and Operations Management (IEOM), 2015 International Conference on*. [S.l.], 2015. p. 1–10.
- PHAM, D.; GHANBARZADEH, A.; KOC, E.; OTRI, S.; RAHIM, S.; ZAIDI, M. The bees algorithm—a novel tool for complex optimisation. In: *Intelligent Production Machines and Systems-2nd I* PROMS Virtual International Conference (3-14 July 2006)*. [S.l.: s.n.], 2011.
- PIEWTHONGNGAM, K.; PATHUMNAKUL, S.; HOMKHAMPAD, S. An interactive approach to optimize production–distribution planning for an integrated feed swine company. *International Journal of Production Economics*, Elsevier, v. 142, n. 2, p. 290–301, 2013.

POCHET, Y.; WOLSEY, L. *Production Planning by Mixed Integer Programming*. Springer New York, 2006. (Springer Series in Operations Research and Financial Engineering). ISBN 9780387299594. Disponível em: <https://books.google.com.br/books?id=Uv8vn_kQZqEC>.

ROPKE, S.; PISINGER, D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, INFORMS, v. 40, n. 4, p. 455–472, 2006.

SAHLING, F.; BUSCHKÜHL, L.; TEMPELMEIER, H.; HELBER, S. Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. *Computers & Operations Research*, Elsevier, v. 36, n. 9, p. 2546–2553, 2009.

TRIGEIRO, W. W.; THOMAS, L. J.; MCCLAIN, J. O. Capacitated lot sizing with setup times. *Management science*, INFORMS, v. 35, n. 3, p. 353–366, 1989.

WAGELMANS, A.; HOESEL, S. V.; KOLEN, A. Economic lot sizing: an $O(n \log n)$ algorithm that runs in linear time in the wagner-whitin case. *Operations Research*, INFORMS, v. 40, n. 1-supplement-1, p. S145–S156, 1992.

WAGNER, H. M.; WHITIN, T. M. Dynamic version of the economic lot size model. *Management science*, INFORMS, v. 5, n. 1, p. 89–96, 1958.