

Tese apresentada à Pró-Reitoria de Pós-Graduação do Instituto Tecnológico de Aeronáutica e da Universidade Federal de São Paulo, como parte dos requisitos para obtenção do título de Doutor em Ciências no Programa de Pós-Graduação em Pesquisa Operacional, Área de Métodos de Otimização.

**Tiago Tiburcio da Silva**

**NOVAS FORMULAÇÕES DE FLUXO PARA  
PROBLEMAS DE OTIMIZAÇÃO  
COMBINATÓRIA**

Tese aprovada em sua versão final pelos abaixo assinados:



Prof. Dr. Antônio Augusto Chaves

Orientador



Prof. Dr. Horacio Hideki Yanasse

Coorientador

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

São José dos Campos, SP - Brasil  
2020

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

Silva, Tiago Tiburcio da  
Novas formulações de fluxo para problemas de otimização combinatória / Tiago Tiburcio da Silva.  
São José dos Campos, 2020.  
164f.

Tese de Doutorado – Curso de Pesquisa Operacional. Área de Métodos de Otimização – Instituto Tecnológico de Aeronáutica e Instituto de Ciência e Tecnologia da Universidade Federal de São Paulo, 2020. Orientador: Prof. Dr. Antônio Augusto Chaves. Coorientador: Prof. Dr. Horacio Hideki Yanasse.

1. Problema do caixeiro viajante. 2. Modelos matemáticos. 3. Otimização. I. Instituto Tecnológico de Aeronáutica. II. Universidade Federal de São Paulo. III. Título.

## **REFERÊNCIA BIBLIOGRÁFICA**

SILVA, Tiago Tiburcio da. **Novas formulações de fluxo para problemas de otimização combinatória**. 2020. 164f. Tese de Doutorado – Instituto Tecnológico de Aeronáutica e Universidade Federal de São Paulo, São José dos Campos.

## **CESSÃO DE DIREITOS**

NOME DO AUTOR: Tiago Tiburcio da Silva

TÍTULO DO TRABALHO: Novas formulações de fluxo para problemas de otimização combinatória.

TIPO DO TRABALHO/ANO: Tese / 2020

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

---

Tiago Tiburcio da Silva  
Av. Cesare Monsueto Giulio Lattes, 1201  
12.247-014 – São José dos Campos–SP

# NOVAS FORMULAÇÕES DE FLUXO PARA PROBLEMAS DE OTIMIZAÇÃO COMBINATÓRIA

Tiago Tiburcio da Silva

Composição e Assinatura da Banca Examinadora:

Prof. Dr. Luiz Leduino de Salles Neto	Presidente	-	Unifesp
Prof. Dr. Antônio Augusto Chaves	Orientador	-	UNIFESP
Prof. Dr. Maria do Socorro Nogueira Rangel	Externo	-	Unesp
Prof. Dr. Leandro Callegari Coelho	Externo	-	Université Laval
Prof. Dr. Luis Felipe Cesar da Rocha Bueno	Interno	-	Unifesp

Tiago Tiburcio da Silva

NOVAS FORMULAÇÕES DE FLUXO PARA PROBLEMAS DE OTIMIZAÇÃO  
COMBINATÓRIA

Tese apresentada à Universidade Federal São Paulo  
como requisito parcial para obtenção do título de  
Doutorado em Ciências/Ciências/Engenharia de  
Produção com ênfase em Pesquisa Operacional.

Área de Concentração: Pesquisa Operacional

Aprovada em 18 de fevereiro de 2020.

**Presidente da Banca:**



Prof. Dr. Antônio Augusto Chaves

**Banca Examinadora:**

Prof. Dr. Luiz Leduino de Salles Neto

Prof(a). Dr(a). Maria do Socorro Nogueira Rangel

Prof. Dr. Leandro Callegari Coelho

Prof. Dr. Luis Felipe Cesar da Rocha Bueno

Este trabalho é dedicado aos meus pais, Valdecir e Abigail, que me ensinaram à admirá-los. À minha esposa Aline por seu amor e paciência. E ao meu pequeno companheiro Miguel (meu filho).

# Agradecimentos

Primeiramente, agradeço a Deus, por ter me guiado e abençoado em todas as fases deste trabalho, e em todos os momentos da minha vida.

Aos professores Antônio e Horacio pela orientação, amizade e paciência. Exemplos de grandes profissionais.

Aos professores e funcionários da Unifesp e ITA, pela colaboração e atenção dispensada.

A todos os meus amigos do MeMO, que tornaram essa caminhada mais agradável.

Mais que agradecer, eu dedico este trabalho à minha família:

À minha irmã Juliana, meu cunhado Bruno e meu sobrinho Benjamin. Minha eterna gratidão pelo carinho e pelas orações.

Aos meus sogros Lindaura e Amarildo pelo apoio, carinho e pelas orações.

À minha cunhada Thays e Vinicio, e meus sobrinhos Lucas, Pedro, Helena e Esther.

Especialmente aos meus pais Valdecir e Abigail, o meu pequeno Miguel e à minha esposa e cúmplice Aline, pelo incentivo, encorajamento, e orações. Se cheguei até aqui, foi graças a eles!

À todos que torceram e oraram por mim.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro destinado a este trabalho.

Enfim, a todos que contribuíram direta ou indiretamente para a realização deste trabalho.

*“Porque o Senhor dá a sabedoria,  
e da sua boca vem o conhecimento e o entendimento.”*  
— PROVÉRBIOS 2:6

# Resumo

Neste trabalho aborda-se o Problema de Minimização de Trocas de Ferramentas (PMTF) e o Problema do Caixeiro Viajante Multiproduto com Prioridades (PCVMP). O PMTF consiste em determinar um sequenciamento de tarefas, de tal modo que a quantidade de trocas de ferramentas entre as tarefas seja a menor possível. Cada tarefa requer um conjunto de ferramentas distinto, e supõe-se que cada um destes conjuntos não contenha mais ferramentas do que suporta a máquina. Já o PCVMP consiste em determinar uma rota de entrega de mercadorias considerando ao mesmo tempo, o cliente e o vendedor, ou seja, minimizando os custos totais do vendedor e maximizando as preferências dos clientes. Neste estudo tem-se como objetivo modelar, baseado em fluxo *multicommodity*, os problemas citados. Modelos matemáticos de otimização foram propostos assim como alguns resultados teóricos foram desenvolvidos. No caso do PMTF, o melhor modelo proposto foi comparado com os modelos existentes na literatura, mostrando um melhor desempenho tanto em quantidade de instâncias resolvidas na otimalidade, quanto no valor da relaxação linear e no tempo de execução. Mostrou-se que o valor da relaxação linear nos modelos propostos corresponde a diferença entre a quantidade de ferramentas e a capacidade da máquina. Algumas heurísticas baseadas em busca por proximidade e um método exato enumerativo considerando eliminação de simetria foram propostos e comparados com os resultados da literatura. Já no caso do PCVMP, o modelo proposto se mostrou eficiente em resolver instâncias de pequeno e médio porte. Duas metaheurísticas, BRKGA e BRKGA adaptativo, ambas com busca local, também foram propostas para o PCVMP, apresentando bons resultados.



# Abstract

The Minimization of Tool Switching Problem (MTSP) and the Multicommodity Traveling Salesman Problem with Priority Prize (MTSPPP) are addressed in this study. Given a set of jobs that needs processing in a machine where each job requires a set of tools that must be in the tool magazine of the machine in order to process it, the Minimization of Tool Switches Problem (MTSP) consists in finding a sequence to process the jobs that minimizes the total number of tool switches. The tool magazine has limited capacity therefore tool switches are necessary when processing the jobs. The MTSPPP consists in determining a delivery route considering both the customer and the seller, minimizing the total costs of the seller and maximizing customer preferences. The aim of this study was to model the problems cited as multicommodity flow-based models. Mathematical optimization models have been proposed as well as some theoretical results developed. In the case of MTSP, the best proposed model was compared with the existing models of the literature, showing a better performance both on the number of instances solved at optimality, as in the linear programming relaxation value and execution time. It was shown that the value of linear relaxation in the proposed models corresponds to the difference between the number of tools and the machine capacity. Some metaheuristics based on proximity search and an exact enumerative method considering symmetry elimination were proposed. In the case of MTSPPP, the proposed model proved to be efficient in solving small and medium sized instances. Two metaheuristics with local search, BRKGA and adaptive BRKGA were also proposed for the PCVMPP, and computational results demonstrated good performance.

# Lista de Figuras

FIGURA 1.1 – Quantidade de publicações relacionadas com o tema PMTF por ano.	23
FIGURA 2.1 – Exemplo do subgrafo utilizado no Modelo CGL, considerando a ferramenta 5 e as tarefas 2 e 4.	55
FIGURA 3.1 – Grafo do Modelo FM-PMTF-1.	60
FIGURA 3.2 – Exemplo de uma solução (ótima) da instância do Exemplo 2.1.1 no grafo da Figura 3.1.	61
FIGURA 3.3 – Grafo do Modelo FM-PMTF-1A.	66
FIGURA 3.4 – Grafo do Modelo FM-PMTF-1B.	69
FIGURA 3.5 – Grafo do Modelo FM-PMTF-1C.	70
FIGURA 3.6 – Grafo do Modelo FM-PMTF-1D.	72
FIGURA 3.7 – Gráfico do tipo <i>performance profile</i> considerando os modelos FM-PMTF-1D, TD, LSS, e CGL.	88
FIGURA 4.1 – Esquema de classificação de matheurísticas (PUCHINGER; RAIDL, 2005).	90
FIGURA 4.2 – Estrutura básica do Método <i>Local Branching</i> (FISCHETTI; LODI, 2003).	93
FIGURA 4.3 – Exemplos de aplicação do Swap(1,1). Em 1) um exemplo de uma sequência de processamento de 7 tarefas. Em 2) e 3) sequências obtidas por Swap(1,1) a partir de 1).	101
FIGURA 4.4 – Diagrama de fluxo do BRKGA. Adaptado de Gonçalves e Resende (2011).	106
FIGURA 4.5 – Diagrama de fluxo do BRKGA adaptativo. Adaptado de Chaves <i>et al.</i> (2016), Chaves <i>et al.</i> (2018).	107
FIGURA 4.6 – Exemplo de uma iteração do algoritmo <i>Label Propagation</i> .	111

---

FIGURA 4.7 – Exemplo de uma solução codificada do PMTF. . . . .	112
FIGURA 4.8 – Exemplo de árvore <i>branch-and-bound</i> com 7 nós, sendo o nó 0 o nó correspondente a relaxação linear. . . . .	113
FIGURA 5.1 – Exemplo de uma solução codificada do PCVMP. . . . .	127
FIGURA 5.2 – Exemplos das três estruturas de vizinhança usadas no PCVMP. Em (a) um exemplo de uma rota de clientes. Em (b) uma rota obtida por 2-Opt. Em (c) uma rota obtida por Shift(1). Em (d) uma rota obtida por Swap(1,1). . . . .	128

# Lista de Tabelas

TABELA 2.1 – Um exemplo de uma instância do PMTF com 6 tarefas, 9 ferramentas e capacidade igual a 4 (TANG; DENARDO, 1988a). . . . .	29
TABELA 2.2 – Instância do Exemplo 2.1 apresentada na forma de uma matriz binária.	29
TABELA 2.3 – Uma solução ótima da instância do PMTF mostrada na Tabela 2.1.	29
TABELA 2.4 – Visão geral da literatura do PMTF uniforme. . . . .	40
TABELA 2.5 – Visão geral da literatura de variações do PMTF. . . . .	48
TABELA 2.6 – Exemplo de matriz P, com 0-bloco, associada ao Exemplo 2.1. . . .	54
TABELA 2.7 – Quadro resumo dos modelos de TD, LSS e CGL. . . . .	56
TABELA 3.1 – Quadro resumo dos modelos de fluxo <i>multicommodity</i> desenvolvidos nesta tese. . . . .	74
TABELA 3.2 – Resultados dos modelos de fluxo MF-PMTF-1, MF-PMTF-1A e MF-PMTF-1B para as instâncias introduzidas por Catanzaro <i>et al.</i> (2015). . . . .	81
TABELA 3.3 – Resultados dos modelos MF-PMTF-1C e MF-PMTF-1D de fluxo para as instâncias introduzidas por Catanzaro <i>et al.</i> (2015). . . . .	82
TABELA 3.4 – Resultados dos modelos de fluxo MF-PMTF-1, MF-PMTF-1A e MF-PMTF-1D para as instâncias introduzidas por Yanasse <i>et al.</i> (2009). . . . .	83
TABELA 3.5 – Resultados dos valores da relaxação linear para os Grupos A, B e C fornecidos por Yanasse <i>et al.</i> (2009). . . . .	84
TABELA 3.6 – Resultados dos valores da relaxação linear para os Grupos D e E fornecidos por Yanasse <i>et al.</i> (2009). . . . .	85
TABELA 3.7 – Resultados dos valores da relaxação linear para as instâncias introduzidas por Catanzaro <i>et al.</i> (2015). . . . .	85

TABELA 3.8 – Resultados para instâncias dos Grupos <i>A</i> , <i>B</i> , <i>C</i> , <i>D</i> , e <i>E</i> de Yanasse <i>et al.</i> (2009). . . . .	87
TABELA 3.9 – Resultados para instâncias dos Grupos <i>A</i> e <i>B</i> de Catanzaro <i>et al.</i> (2015). . . . .	88
TABELA 4.1 – Soluções médias encontradas pelas métricas 1 a 5 nas instâncias de Catanzaro <i>et al.</i> (2015). . . . .	99
TABELA 4.2 – Resultados obtidos pela Relaxação Lagrangeana para as instâncias de Catanzaro <i>et al.</i> (2015). . . . .	115
TABELA 4.3 – Resultados obtidos pela matheurística VNDLB para as instâncias de Catanzaro <i>et al.</i> (2015). . . . .	117
TABELA 4.4 – Resultados obtidos pela matheurística Busca por Proximidade para as instâncias de Catanzaro <i>et al.</i> (2015). . . . .	117
TABELA 4.5 – Resultados obtidos pela matheurística VNDLB para as instâncias das classes <i>A</i> , <i>B</i> , <i>C</i> , <i>D</i> e <i>E</i> de Yanasse <i>et al.</i> (2009). . . . .	119
TABELA 4.6 – Resultados obtidos pela matheurística VNDLB, o algoritmo A-BRKGA e o algoritmo ILS de Paiva e Carvalho (2017), considerando as instâncias de Catanzaro <i>et al.</i> (2015). . . . .	120
TABELA 4.7 – Resultados obtidos pela matheurística VNDLB, o algoritmo A-BRKGA e o algoritmo ILS de Paiva e Carvalho (2017), considerando as instâncias de Yanasse <i>et al.</i> (2009). . . . .	121
TABELA 4.8 – Resultados obtidos pelo método exato considerando as instâncias de Catanzaro <i>et al.</i> (2015). . . . .	122
TABELA 5.1 – Valores dos parâmetros utilizados no processo de ajuste do BRKGA+CS.	131
TABELA 5.2 – Parâmetros usados no BRKGA+CS e no A-BRKGA+CS. . . . .	131
TABELA 5.3 – As instâncias simétricas selecionadas no <i>TSPLIB</i> (REINELT, 2013), por tamanhos de instância. A instância <i>dj38</i> foi selecionada em <i>TSP Test Data</i> (REINELT, 2009). . . . .	132
TABELA 5.4 – Resultados do CPLEX, BRKGA+CS e A-BRKGA+CS para instâncias do conjunto A. . . . .	135
TABELA 5.5 – Resultados do CPLEX, BRKGA+CS e A-BRKGA+CS para instâncias do conjunto B. . . . .	136
TABELA 5.6 – Resultados do CPLEX, BRKGA+CS e A-BRKGA+CS para instâncias do conjunto C. . . . .	137

---

TABELA 5.7 – Resultado dos testes WSR e Friedman para os conjuntos A, B e C. . .	138
TABELA 5.8 – Resultados da relaxação linear do modelo PCVMP nos conjuntos A, B e C. . . . .	139
TABELA A.1 – Resultados dos modelos de fluxo para as instâncias introduzidas por Catanzaro <i>et al.</i> (2015) - Grupo A. . . . .	159
TABELA A.2 – Resultados dos modelos de fluxo para as instâncias introduzidas por Catanzaro <i>et al.</i> (2015) - Grupo B. . . . .	160
TABELA A.3 – Resultados dos modelos de fluxo para as instâncias introduzidas por Catanzaro <i>et al.</i> (2015) - Grupo C. . . . .	161
TABELA A.4 – Resultados dos modelos de fluxo para as instâncias introduzidas por Catanzaro <i>et al.</i> (2015) - Grupo D. . . . .	162

# Lista de Abreviaturas e Siglas

A-BRKGA	<i>Adaptative Biased Random-Key Genetic Algorithm</i>
A-BRKGA+CS	A-BRKGA com busca local
B	Modelo de Bard
BRKGA	<i>Biased Random-Key Genetic Algorithm</i>
BRKGA+CS	BRKGA com busca local
CGL	Modelo de Catanzaro, Gouveia e Labbé
FM-PMTF	Modelo de fluxo <i>multicommodity</i> para o PMTF
ILS	<i>Iterated Local Search</i>
KTNS	<i>Keep Tool Needed Soonest</i>
LB	<i>Local Branching</i>
LSS	Modelo de Laporte, Salazar-González e Semet
PCV	Problema do Caixeiro Viajante
PCVMPP	Problema do Caixeiro Viajante Multiproduto com Prêmios de Prioridade
PMTF	Problema de minimização de trocas de ferramentas
PS	<i>Proximity Search</i>
RF	<i>Relax-and-Fix</i>
TD	Modelo de Tang e Denardo
VND	<i>Variable Neighborhood Descent</i>
VNDLB	LB em um esquema VND

# Sumário

1	INTRODUÇÃO . . . . .	18
1.1	Modelos de fluxo multicommodity . . . . .	18
1.1.1	Problema de Minimização de Trocas de Ferramentas . . . . .	20
1.1.2	Problema do Caixeiro Viajante . . . . .	21
1.2	Objetivos . . . . .	22
1.3	Justificativa . . . . .	23
1.4	Delimitações da pesquisa . . . . .	24
1.5	Organização da tese . . . . .	24
2	O PROBLEMA DE MINIMIZAÇÃO DE TROCAS DE FERRAMENTAS	26
2.1	Definição do PMTF . . . . .	26
2.2	Revisão Bibliográfica . . . . .	29
2.2.1	PMTF uniforme . . . . .	30
2.2.2	Variações do PMTF . . . . .	41
2.3	Modelos da literatura . . . . .	47
2.3.1	Modelo de Tang e Denardo - TD . . . . .	47
2.3.2	Modelo de Bard - B . . . . .	50
2.3.3	Modelo de Laporte, Salazar-González e Semet - LSS . . . . .	51
2.3.4	Modelo de Catanzaro, Gouveia e Labbé - CGL . . . . .	53
2.4	Considerações finais . . . . .	56
3	MODELOS DE FLUXO MULTICOMMODITY PROPOSTOS PARA O PMTF . . . . .	58
3.1	Modelos de fluxo <i>multicommodity</i> para o PMTF . . . . .	59



---

3.1.1	Modelo FM-PMTF-1 . . . . .	59
3.1.2	Modelo FM-PMTF-1A . . . . .	65
3.1.3	Modelo FM-PMTF-1B . . . . .	67
3.1.4	Modelo FM-PMTF-1C . . . . .	69
3.1.5	Modelo FM-PMTF-1D . . . . .	72
<b>3.2</b>	<b>Relaxação Linear . . . . .</b>	<b>74</b>
<b>3.3</b>	<b>Testes computacionais . . . . .</b>	<b>78</b>
3.3.1	Performance dos modelos de fluxo <i>multicommodity</i> propostos para o PMTF . . . . .	79
3.3.2	Relaxação Linear dos modelos para o PMTF . . . . .	81
3.3.3	Análise Computacional do Modelo MF-PMTF-1D com os Modelos da Literatura . . . . .	84
<b>3.4</b>	<b>Considerações finais . . . . .</b>	<b>88</b>
<b>4</b>	<b>MÉTODOS EXATOS E HEURÍSTICOS APLICADOS AO PMTF . . . . .</b>	<b>89</b>
<b>4.1</b>	<b>Definição de Matheurísticas . . . . .</b>	<b>89</b>
<b>4.2</b>	<b>Descrição das metaheurísticas utilizadas . . . . .</b>	<b>90</b>
4.2.1	<i>Relax-and-fix</i> . . . . .	90
4.2.2	<i>Local Branching</i> . . . . .	91
4.2.3	Método de Descida em Vizinhança Variável . . . . .	94
<b>4.3</b>	<b>Busca por proximidade . . . . .</b>	<b>95</b>
<b>4.4</b>	<b>Métricas . . . . .</b>	<b>96</b>
<b>4.5</b>	<b>VNDLB . . . . .</b>	<b>99</b>
<b>4.6</b>	<b>Relaxação Lagrangeana . . . . .</b>	<b>101</b>
<b>4.7</b>	<b>BRKGA . . . . .</b>	<b>104</b>
<b>4.8</b>	<b>BRKGA Adaptativo (A-BRKGA) . . . . .</b>	<b>106</b>
4.8.1	Busca Local . . . . .	109
<b>4.9</b>	<b>A-BRKGA para o PMTF . . . . .</b>	<b>111</b>
<b>4.10</b>	<b>Método exato BB-PMTF . . . . .</b>	<b>112</b>
<b>4.11</b>	<b>Experimentos computacionais . . . . .</b>	<b>113</b>
4.11.1	Relaxação Lagrangeana . . . . .	114

---

4.11.2	Matheurísticas . . . . .	114
4.11.3	Método exato BB-PMTF . . . . .	118
4.12	Considerações finais . . . . .	118
5	O PROBLEMA DO CAIXEIRO VIAJANTE MULTIPRODUTO COM PRIORIDADES . . . . .	123
5.1	Modelo de fluxo <i>multicommodity</i> para o PCVMP . . . . .	124
5.2	BRKGA+CS e A-BRKGA+CS para o PCVMP . . . . .	127
5.2.1	Busca local para o PCVMP . . . . .	127
5.3	Resultados computacionais . . . . .	130
5.4	Considerações finais . . . . .	134
6	CONCLUSÕES . . . . .	140
6.1	Estudos futuros para o PMTF . . . . .	141
6.2	Estudos futuros para o PCVMP . . . . .	142
	REFERÊNCIAS . . . . .	143
	ANEXO A – RESULTADOS DO PMTF . . . . .	158
	ANEXO B – PUBLICAÇÕES . . . . .	163

# 1 Introdução

Neste capítulo fornece-se uma breve introdução aos principais tópicos abordados nesta tese. Na Seção 1.1, os modelos de fluxo *multicommodity* são brevemente introduzidos no contexto mais amplo da modelagem de redes. Na Subseção 1.1.1, uma breve introdução ao Problema de Minimização de Trocas de Ferramentas é realizada ressaltando a importância em se abordar o problema. Na Subseção 1.1.2, o Problema do Caixeiro Viajante é abordado considerando algumas lacunas na literatura. Os objetivos, as justificativas e as delimitações da pesquisa são apresentados nas Seções 1.2, 1.3 e 1.4, respectivamente. Finalmente, na Seção 1.5, a estrutura da presente tese é descrita.

## 1.1 Modelos de fluxo multicommodity

Durante a Guerra Fria (1947 – 1991)<sup>1</sup> os militares estadunidenses tiveram interesse no sistema ferroviário da URSS<sup>2</sup> que conectava a União Soviética Ocidental com a Europa Oriental, particularmente à Alemanha Oriental. Seu interesse estava em descobrir quais pontos na rede ferroviária eles deveriam bombardear a fim de impedir, com precisão, o transporte (de soldados, de armas, de suprimentos) entre a União Soviética e a Europa Oriental. Este problema foi formulado como um problema de fluxo em rede por Harris e Ross (1955), em que cada nó representava as estações e as arestas eram os trilhos que conectavam uma estação à outra. Cada aresta tinha uma capacidade não negativa que era a capacidade total de transporte de uma estação à outra. O problema de encontrar o número máximo de trens que podem viajar desde a origem até o destino é o problema de fluxo máximo. Entretanto, Harris e Ross queriam determinar a capacidade mínima (soma das capacidades das arestas) que pode ser removida, impedindo o movimento da origem até o destino. Este problema é o problema do corte mínimo (SCHRIJVER, 2002).

Ford e Fulkerson (1954) se interessaram pelo problema de Harris e Ross, e propuseram

---

<sup>1</sup>A Guerra Fria foi um conflito político-ideológico travado entre Estados Unidos e União Soviética, entre 1947 e 1991.

<sup>2</sup>As repúblicas Armênia, Azerbaijão, Bielo-Rússia, Cazaquistão, Estônia, Geórgia, Letônia, Lituânia, Moldávia, Quirguistão, Rússia, Tadjiquistão, Turcomenistão, Ucrânia e Uzbequistão, formaram a URSS (União das Repúblicas Socialistas Soviéticas) entre 30 de dezembro de 1922 a 26 de dezembro de 1991.

o algoritmo Ford-Fulkerson que determina o fluxo máximo ótimo em uma rede. Dantzig e Fulkerson (1957) mostraram que o problema de fluxo máximo é equivalente ao problema do corte mínimo, ou seja, em qualquer rede, o valor do fluxo máximo é igual à capacidade do corte mínimo. Este resultado é chamado Teorema do Fluxo Máximo e Corte Mínimo (do inglês “*Theorem Max-Flow Min Cut*”). Ford e Fulkerson continuaram estudando o problema (FORD; FULKERSON, 1955; FORD; FULKERSON, 1957; FULKERSON, 1961) e em 1962 publicaram o primeiro livro relacionado ao tema (FORD; FULKERSON, 1962).

Desde então, muita atenção foi dada a problemas envolvendo fluxo *multicommodity* com aplicação nas mais diversas áreas, como transporte, localização de facilidades, atribuição, entre outros. Este problema consiste em dado um grafo direcionado, um nó origem, um nó destino, e uma quantidade de *commodities* que devem atravessar essa rede, partindo da origem e chegando no destino, determinar o fluxo de cada tipo de *commodity* em cada arco da rede, atendendo a demanda preestabelecida, e respeitando as restrições de capacidade em cada arco e restrições de conservação de fluxo dos nós.

Muitas aplicações são encontradas na literatura. Bektas (2012) aborda o problema do caixeiro viajante com vários depósitos, em que o número de nós visitados por cada caixeiro é predeterminado dentro de um intervalo, desenvolvendo vários modelos baseados em fluxo *multicommodity*. O autor propôs vários métodos baseados em Decomposição de Benders e três dos métodos se mostraram alternativas viáveis para resolver problemas com instâncias de grande porte. Maharjan e Matis (2012) abordam o problema de atribuição de portão de embarque. O modelo proposto é resolvido via CPLEX 11.2 numa aplicação em um aeroporto real.

Akyuz *et al.* (2013) modelaram o problema de localização de facilidades capacitado e desenvolveram algumas heurísticas baseadas em *Beam Search* que apresentaram bons resultados quando comparados com os apresentados na literatura. Letchford e Salazar-González (2015) propuseram várias formulações fortes para o problema de roteamento de veículos capacitado e mostraram que suas relaxações são significativamente mais fortes que as conhecidas. Vários outros autores usam modelos baseados em fluxo *multicommodity*: problema de programação de horários (DORNELES *et al.*, 2017), problema de projeto de uma rede de serviços e localização de *hubs* (ROTHENBÄCHER *et al.*, 2016), problema de fluxo em rede *multicommodity* de custo fixo (MUNGUÍA *et al.*, 2017), problema de fluxo em rede *multicommodity* inteira com considerações de confiabilidade do nó (TADAYON; SMITH, 2014), lista de condutores com padrão de folga (MESQUITA *et al.*, 2015), planejamento da produção em uma indústria de bebidas (MALDONADO *et al.*, 2014), problema integrado de dimensionamento e sequenciamento de lotes (SILVA; RANGEL, 2016). Vários autores também tem dado atenção a estudos teóricos: Gendreau *et al.* (1992), Gondzio *et al.* (2016), Raack (2012), Burchett e Richard (2015).

### 1.1.1 Problema de Minimização de Trocas de Ferramentas

Com o crescente desenvolvimento da tecnologia e condições de mercado altamente competitivas, tornou-se necessário que os sistemas de fabricação tenham tempos de resposta rápidos e alta flexibilidade (GAMILA; MOTAVALLI, 2003). Em resposta à esta demanda, os Sistemas Flexíveis de Manufatura (SFM) ganharam atenção redobrada. Segundo Zeballos (2010), um SFM é uma rede de máquinas interligadas e controlada por um sistema de manuseio de materiais, em que um pequeno ou médio volume de tarefas pode ser fabricado, desde que as ferramentas necessárias para a produção estejam disponíveis e alocadas na máquina. O SFM tem a capacidade de produzir uma variedade de peças usando o mesmo sistema. No entanto, toda essa flexibilidade tem um custo: a necessidade do desenvolvimento de métodos eficientes e eficazes para o controle e planejamento da produção.

Como um SFM tem que atender requisitos de eficiência e flexibilidade, os problemas de planejamento e controle de produção são mais difíceis do que aqueles de sistemas de produção em massa ou sistemas de produção em lote. Cada estação de trabalho pode processar as várias operações usando muitas ferramentas disponíveis em suas caixas de ferramentas (JANG *et al.*, 1996).

De acordo com Stecke (1983), há cinco possíveis classificações para os problemas de planejamento de produção em um SFM:

1. Problema de seleção de tipo de peça (*“Part type selection problem”*): dado um conjunto de tarefas que possuem requisitos de produção comuns, determinar quais devem ser processadas imediatamente e simultaneamente;
2. Problema de agrupamento de máquinas (*“Machine grouping problem”*): consiste em particionar as máquinas em grupos de tal forma que as máquinas em um grupo específico sejam capazes de executar o mesmo conjunto de operações;
3. Problema da proporção do produto (*“Product ratio problem”*): determinar a quantidade a ser produzida dos produtos selecionados no problema 1;
4. Problema de alocação de recursos (*“Resource allocation problem”*): alocar uma quantidade fixa de recursos às atividades, a fim de minimizar o custo incorrido pela alocação;
5. Problema de carregamento (*“Loading problem”*): alocar as operações e ferramentas necessárias para a produção das tarefas selecionadas entre os grupos de máquinas satisfazendo restrições tecnológicas e de capacidade do SFM.

Este trabalho concentra-se em uma subclasse de problemas dentro da classe (5), que são os problemas de Minimização de Trocas de Ferramentas (PMTF) (do inglês “*Minimization of Tool Switching Problem*” - MTSP). Este problema também é encontrado na literatura como problema de sequenciamento de tarefas e troca de ferramentas (do inglês “*Job Sequencing and Tool Switching Problem*”). O problema consiste em determinar um plano de produção, ou seja, um sequenciamento das tarefas, em que cada tarefa requer um conjunto de ferramentas para ser processada, de tal modo que a quantidade total de trocas de ferramentas, entre as tarefas, seja minimizada. De acordo com Melnyk *et al.* (1989) trocas desnecessárias de ferramentas resultam na subutilização e um nível inaceitável de tempo de inatividade da máquina (AGAPIOU, 1991). Zhang e Hinduja (1995) defendem a necessidade de um sistema de gerenciamento de ferramentas mais sofisticado, já que o tempo dispendido nas trocas de ferramentas é superior a qualquer outro tempo de *setup* (HOP; NAGARUR, 2004).

O PMTF é classificado como NP-difícil, como mostrado por Tang e Denardo (1988a) e Crama *et al.* (1994). Devido a complexidade do problema, a maioria dos autores o aborda via métodos heurísticos de solução. Os poucos autores que desenvolveram modelos matemáticos não conseguiram resolver na otimalidade instâncias com mais de 30 tarefas.

Um dos primeiros e impactantes estudos sobre o PMTF foi produzido por Tang e Denardo (1988a). Eles propuseram um modelo de otimização linear inteiro para representar o problema, mas devido à solução da relaxação linear do modelo ser nula, os resultados computacionais não foram encorajadores. Laporte *et al.* (2004) também propuseram uma formulação inteira para o PMTF, fortaleceram a sua relaxação linear, mas novamente os resultados computacionais não foram encorajadores. Catanzaro *et al.* (2015) propuseram três formulações para o PMTF, mas devido a grande quantidade de variáveis e as restrições de eliminação de subrotas DFJ (DANTZIG *et al.*, 1954), os resultados computacionais deixaram a desejar.

Dentro deste contexto, neste trabalho tem-se como primeira questão de pesquisa: é possível desenvolver novas formulações matemáticas e novos métodos exatos para o PMTF que sejam mais eficientes que os encontrados na literatura?

Uma segunda questão de pesquisa: é possível determinar um subconjunto de variáveis no modelo desenvolvido de tal forma que ao integralizá-las consequentemente todas as outras variáveis assumirão valores inteiros?

### 1.1.2 Problema do Caixeiro Viajante

O problema do caixeiro viajante (PCV) consiste em determinar um ciclo hamiltoniano em um grafo, tal que o custo total deste ciclo seja minimizado. Um caixeiro viajante deseja

visitar um número  $n$  de cidades, a fim de vender seus produtos. Dado um conjunto de cidades e as distâncias  $d_{ij}$  entre as cidades  $i$  e  $j$ , o objetivo deste problema é determinar em qual ordem o caixeiro as percorrerá, a fim de visitar cada cidade uma única vez, percorrendo a menor distância possível.

De acordo com os estudos históricos de Bock (1963), o precursor mais direto do PCV data desde 1930 por Menger (MENGER, 1932), mas popularizou-se por Merrill Flood, que era um estudante de pós-graduação em Princeton em 1932, e posteriormente trabalhou na *Rand Corporation*, o centro intelectual de Pesquisa Operacional na época. O PCV se tornou um dos problemas de otimização combinatória mais conhecidos por surgir em diversas aplicações, assim como por pertencer à classe dos problemas  $NP$ -completos (PAPADIMITRIOU, 1977), ou seja, não se conhece um algoritmo determinístico de tempo polinomial para resolvê-lo. Outra razão para a rápida popularização do PCV pode ser atribuída a sua íntima conexão com alguns tópicos importantes em problemas combinatórios, como o problema de transporte (HITCHCOCK, 1941).

Ao longo dos anos, muitos problemas têm sido formulados como um PCV. Entretanto, a maioria destas formulações não considera o ponto de vista de ambos caixeiro e cliente. Essa questão surge em empresas de logística em que há conflitos entre custos operacionais, clientes com diferentes categorias de produtos e satisfação do cliente, esta, diretamente relacionada ao tempo de entrega. Torna-se então um desafio escolher entre minimizar os custos de viagem e garantir um grau de qualidade de serviço para todos os clientes. Este problema, que considera tanto clientes como caixeiro será denominado Problema do Caixeiro Viajante Multiproduto com Prioridades (PCVMP).

Dentro deste contexto, este trabalho tem como terceira questão de pesquisa: é possível desenvolver formulações matemáticas e métodos de solução que consideram custos operacionais e satisfação dos clientes?

## 1.2 Objetivos

Com o intuito de responder a estas duas questões de pesquisa o objetivo geral deste trabalho é propor novas formulações e novos métodos de solução para o PMTF e PCVMP. Como objetivos específicos tem-se:

- Desenvolver modelos baseados em fluxo *multicommodity* para o PMTF;
- Desenvolver métodos de solução baseados em busca por proximidade utilizando o modelo de fluxo proposto para o PMTF e algoritmo genético para o PCVMP.

### 1.3 Justificativa

O interesse no PMTF tem sido constante, com vários artigos recentes abordando o tema em questão, como Chaves *et al.* (2016), Beezão *et al.* (2017), Paiva e Carvalho (2017), Furrer e Mütze (2017), Farughi *et al.* (2017), Ahmadi *et al.* (2018), Amaya *et al.* (2019), Calmels (2019), Mecler *et al.* (2019), Silva *et al.* () (artigo desenvolvido a partir de resultados desta tese).

No gráfico exibido na Figura 1.1 apresenta-se a evolução das publicações relacionadas ao PMTF desde o artigo de Tang e Denardo (1988a) em 1988 até o ano de 2020 (com o artigo fruto desta tese), mostrando que o tema em questão vem sendo constantemente estudado. Estes trabalhos foram encontrados nas bases de dados do *Elsevier (Science Direct)* e *Google Scholar* utilizando as palavras-chave “*job sequencing and tool switching problem*”, “*minimization of tool switching problem*” e “*tool switching problem*”. Não foi considerada nenhuma variação relacionada ao tema.

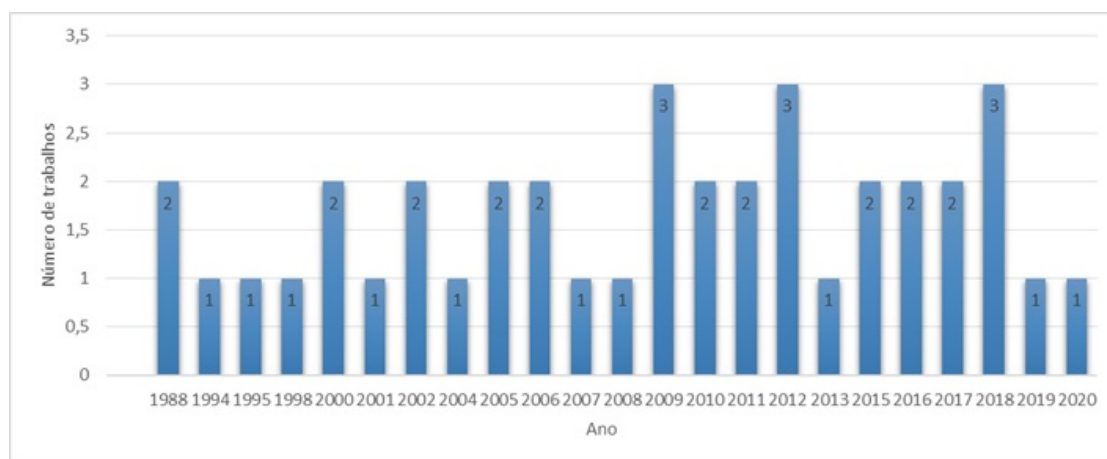


FIGURA 1.1 – Quantidade de publicações relacionadas com o tema PMTF por ano.

Com relação ao PCVMP, não foi encontrado nenhum artigo com a temática proposta, abordando tanto clientes quanto caixeiro, do ponto de vista mencionado acima.

A escolha dos dois problemas, o PMTF e o PCVMP, para estudos nesta tese se deve ao fato de ambos serem variações ou extensões do PCV. O PCVMP é o PCV com restrições adicionais. Se cada uma das tarefas requerer exatamente  $C$  ferramentas para serem produzidas no PMTF, então tem-se exatamente um PCV. Além disso, em casos gerais, a solução do PCV com uma dada métrica associada é um limitante superior para o PMTF (HERTZ *et al.*, 1998; TANG; DENARDO, 1988a).

Teve-se como objetivos desenvolver novos modelos de otimização matemática e métodos de solução com desempenho computacional melhores comparativamente aos propostos na literatura para os dois problemas combinatórios apresentados nas Subseções 1.1.1 e 1.1.2. Para alcançar este objetivo, uma possibilidade que foi explorada foram novas



formulações baseadas em fluxo *multicommodity*. A escolha de uma formulação baseada em fluxo se deve ao fato de que a matriz de restrições é totalmente unimodular consequentemente, a relaxação das restrições de balanço de fluxo pode fornecer soluções inteiras (e então a relaxação linear resolve a formulação inteira) (BAZARAA *et al.*, 2010; WOLSEY, 1998). Isto pontencialmente pode ser interessante na obtenção de soluções inteiras das variáveis de fluxo nas formulações dos PMTF e PCVMP. Cabe lembrar que formulações de fluxo para o PCV também já foram propostas na literatura para eliminar subrotas que podem ser geradas com as restrições de atribuição (veja, por exemplo, Ahuja *et al.* (1993)).

Neste trabalho desenvolveu-se cinco novas formulações matemáticas para o PMTF e uma para o PCVMPP, modelando-os através de fluxos *multicommodity* em digrafos valorados. Também se propôs um método de solução exato baseado em *branch-and-bound* e precedência no processo de produção das tarefas para o PMTF, e alguns métodos heurísticos para ambos problemas. O objetivo deste trabalho é desenvolver uma formulação cuja relaxação linear seja superior as dos modelos encontrados na literatura para o PMTF e desenvolver métodos eficientes de solução para o PMTF e o PCVMP.

## 1.4 Delimitações da pesquisa

Este trabalho tem como foco resolver o PMTF no contexto de uma única máquina, em que todas as ferramentas necessárias para a produção das tarefas estão sempre disponíveis, tem tamanhos uniformes, e não há nenhuma consideração quanto à vida útil ou manutenção destas ferramentas. Os tempos de trocas envolvidos são constantes, as trocas só podem ser realizadas uma por vez, não há precedência na produção das tarefas, e todos os valores envolvidos são determinísticos. O PCVMPP considera custos fixos e variáveis que dependem da rota e do produto a ser transportado na rota. Há prioridades que contabilizam as preferências dos clientes em receber sua encomenda numa dada ordem da sequência de entrega. Não há consideração sobre tempo de entrega, janela de tempo ou peso da carga. Todos os valores são determinísticos.

## 1.5 Organização da tese

Este texto está organizado em seis capítulos, sendo este o primeiro. No Capítulo 2 descreve-se formalmente o PMTF, realiza-se uma revisão bibliográfica abordando as principais formulações matemáticas encontradas na literatura, assim como métodos exatos e heurísticos de solução, e variações do PMTF. Os modelos de fluxo para o PMTF desenvolvidos nesta tese são apresentados no Capítulo 3, assim como um estudo quanto à

---

relaxação linear e testes computacionais. No Capítulo 4, os métodos de solução para o PMTF desenvolvidos e os seus testes computacionais são apresentados. No Capítulo 5 a formulação de fluxo *multicommodity* desenvolvida para o PCVMP é descrita e testes computacionais são realizados com o modelo e dois algoritmos do tipo BRKGA, sendo um deles, adaptativo. No Capítulo 6, as conclusões e perspectivas futuras são descritas. Por fim, nos Apêndice A e B, alguns resultados dos testes computacionais relativos ao PMTF são apresentados, e uma lista das publicações fruto deste trabalho é apresentada, respectivamente.

# 2 O Problema de minimização de trocas de ferramentas

Neste Capítulo, define-se formalmente o PMTF na Seção 2.1 e uma detalhada revisão bibliográfica é realizada na Seção 2.2 em que se aborda primeiramente a literatura sobre o PMTF uniforme, e em seguida as variações do PMTF. Por fim, na Seção 2.3 são descritos e analisados os principais modelos de otimização linear inteira para o PMTF.

## 2.1 Definição do PMTF

O PMTF consiste em, dado um conjunto de tarefas  $J = \{1, 2, \dots, N\}$ , e um conjunto de ferramentas  $T = \{1, 2, \dots, M\}$  sendo que cada tarefa requer um subconjunto de ferramentas  $T_j$ ,  $j \in J$ , ( $T_j \subset T$ ,  $\forall j \in J$ ) para ser processada sequencialmente em uma máquina, determinar uma sequência de processamento destas tarefas de tal forma que o número total de trocas de ferramentas seja mínimo. Considera-se que a máquina tenha uma capacidade limitada  $C$  para armazenamento das ferramentas, e que esta capacidade seja menor que a quantidade total de ferramentas necessária para processar todas as tarefas ( $C < M$ ), pois caso contrário o problema se torna trivial (basta manter todas as ferramentas na máquina). Desta forma, é necessário realizar trocas de ferramentas, ou seja, remover uma ferramenta da máquina e inserir uma nova em seu lugar, para se processar todas as tarefas. Este problema é classificado como NP-difícil (TANG; DENARDO, 1988a; CRAMA *et al.*, 1994).

Algumas suposições são feitas para melhor retratar o PMTF (WIDMER, 1991; ZHOU *et al.*, 2005):

1. Existe um conjunto de tarefas a serem processadas e cada tarefa requer um conjunto fixo específico de ferramentas;
2. Nenhuma tarefa requer um conjunto de ferramentas que exceda a capacidade  $C$  da máquina ( $|T_j| \leq C$ ,  $j \in J$ );

3. Todas as ferramentas estão sempre disponíveis;
4. Cada tarefa é executada em apenas uma máquina;
5. Todas as ferramentas são dedicadas a uma máquina;
6. Todo o planejamento é feito *offline*;
7. O sistema não admite preempção, ou seja, uma vez iniciado o processamento de uma tarefa ela deve ser terminada por completo;
8. Apenas uma troca de ferramenta é realizada por vez;
9. Cada ferramenta ocupa apenas um *slot* da máquina;
10. O tempo associado à inserção ou remoção (troca) de ferramentas é independente e constante;
11. Não há quebra, desgaste e nem manutenção de ferramentas;
12. As ferramentas não possuem *slots* específicos.

Tang e Denardo (1988a) observaram que o problema PMTF tem relação com o Problema do Caixeiro Viajante - PCV (do inglês “*Traveling Salesman Problem*”). Quando  $|T_j| = C, \forall j \in J$ , ou seja, o conjunto de ferramentas de cada tarefa tem cardinalidade igual a capacidade da máquina, então o PMTF corresponde ao PCV, uma vez que as trocas entre quaisquer tarefas são unicamente determinadas. Os autores ainda mostraram que o PMTF pode ser decomposto em dois subproblemas:

1. *Problema de Sequenciamento de Tarefas*: consiste em identificar uma sequência (ótima) de tarefas;
2. *Problema de Alocação de Ferramentas*: consiste em determinar, dado um sequenciamento de tarefas, quais ferramentas estarão na máquina durante o processamento de cada tarefa a fim de minimizar a troca de ferramentas.

Para a solução do problema de alocação de ferramentas, existe um algoritmo ótimo de tempo polinomial denominado KTNS (do inglês “*Keep Tool Needed Soonest*”), proposto por Tang e Denardo (1988a). Este algoritmo tem complexidade  $O(NM)$  (TANG; DENARDO, 1988a), e determina que se uma ferramenta for inserida na máquina, então as ferramentas que serão mantidas são aquelas que serão utilizadas o mais breve possível. Matematicamente, sejam  $N$  tarefas,  $M$  ferramentas, uma matriz  $A$ , de dimensões  $N \times M$ , cujas entradas  $A_{jt} = 1$ , se a ferramenta  $t$  é necessária para processar a tarefa  $j$ , e  $A_{jt} = 0$  caso contrário; vetores  $W_n$ , sendo  $W_{nt} = 1$  se a ferramenta  $t$  está na máquina no instante

$n$ , e  $W_{nt} = 0$  caso contrário, e um vetor  $F$ , em que  $F_t = 1$ , se a ferramenta  $t$  está na máquina no instante  $n$ . Dada uma sequência de tarefas  $x$  tal que  $x_{ik} = 1$  se a tarefa  $i$  é a  $k$ -ésima tarefa da sequência, e  $x_{ik} = 0$ , caso contrário, considere o conjunto:

$$T(t, n) = \{m - 1 : \exists j, m > n, A_{jt} = 1 \text{ e } x_{jm} = 1\} \cup \{N\}, \quad (2.1)$$

e o número

$$L(t, n) = \min\{m : m \in T(t, n)\}. \quad (2.2)$$

O conjunto  $T(t, n)$  é formado por todos instantes em que a ferramenta  $t$  é necessária, a partir do instante  $n$ , inclusive. Já o número inteiro  $L(t, n)$  é o primeiro instante, a partir de  $n$ , inclusive, em que a ferramenta  $t$  é requerida. Dadas estas definições, pode-se enunciar o Algoritmo KTNS:

---

**Algorithm 1** Algoritmo KTNS (TANG; DENARDO, 1988a)

---

- 1: Seja  $F_t = 1$  para  $C$  valores de  $t$  com  $L(t, 0)$  mínimo. Escolha arbitrariamente estes  $C$  valores de  $t$ .
  - 2: Seja  $F_t = 0$  para os restantes  $M - C$  valores de  $t$  e tome  $n = 1$ .
  - 3: Tome  $W_n = F$ . Pare se  $n = N$ .
  - 4: Se para cada  $t$ ,  $L(t, n) = n$  e  $F_t = 1$ , então seja  $n = n + 1$  e vá ao Passo 3.
  - 5: Escolha  $t$  em que  $L(t, n) = n$  e  $F_t = 0$ . Seja  $F_t = 1$ .
  - 6: Seja  $F_k = 0$  para  $k = \arg \max_n \{L(p, n) : J_p = 1\}$ . Vá para o Passo 4.
- 

**Exemplo 2.1.1.** *Considere a instância do PMTF reproduzida na Tabela 2.1 a seguir, utilizada em Tang e Denardo (1988a), com  $N = 6$ ,  $M = 9$  e  $C = 4$ . Na primeira linha desta Tabela, as tarefas que serão sequenciadas são identificadas com números. Em cada coluna, é apresentado o conjunto de ferramentas necessárias para produzir a tarefa correspondente. As mesmas informações dadas na Tabela 2.1 podem ser fornecidas usando uma matriz de tarefa-ferramenta binária em que cada linha corresponde a uma ferramenta e cada coluna corresponde a uma tarefa, conforme Tabela 2.2. Nesta matriz, um 1 na linha  $t$  e coluna  $j$  indica que para processar a tarefa  $j$ , precisa-se da ferramenta  $t$ , e 0 indica o contrário.*

*Na Tabela 2.3, mostra-se uma solução ótima da instância apresentada na Tabela 2.1. Na primeira linha, tem-se a ordem em que as tarefas são processadas. A sequência de processamento ótima apresentada é 4 – 2 – 5 – 3 – 6 – 1. As ferramentas em negrito são desnecessárias para processar a tarefa específica, mas mantê-las na máquina resultam no número mínimo de trocas de ferramentas induzidas pela sequência específica das tarefas. Esta solução resulta em um número total de trocas de ferramentas iguais a 7: 1 troca*

Tarefas	1	2	3	4	5	6
Ferramentas	1	1	2	1	3	1
	4	3	6	5	5	2
	8	5	7	7	8	4
	9		8	9		

TABELA 2.1 – Um exemplo de uma instância do PMTF com 6 tarefas, 9 ferramentas e capacidade igual a 4 (TANG; DENARDO, 1988a).

Tarefas	1	2	3	4	5	6
Ferramentas	1	1	0	1	0	1
	0	0	1	0	0	1
	0	1	0	0	1	0
	1	0	0	0	0	1
	0	1	0	1	1	0
	0	0	1	0	0	0
	0	0	1	1	0	0
	1	0	1	0	1	0
	1	0	0	1	0	0

TABELA 2.2 – Instância do Exemplo 2.1 apresentada na forma de uma matriz binária.

entre as tarefas 4 e 2; 1 troca entre as tarefas 2 e 5; 2 trocas entre as tarefas 5 e 3; 2 trocas entre as tarefas 3 e 6; e 1 troca entre as tarefas 6 e 1. As ferramentas sublinhadas na Tabela 2.3 são as ferramentas inseridas na máquina no lugar das demais que foram retiradas da máquina.

Tarefas	4	2	5	3	6	1
Ferramentas	1	1	<u>8</u>	8	<b>8</b>	8
	5	5	5	<u>6</u>	<u>1</u>	1
	7	<b>7</b>	<b>7</b>	7	<u>4</u>	4
	9	<u>3</u>	3	<u>2</u>	<u>2</u>	<u>9</u>
Número de trocas de ferramentas	0	1	1	2	2	1

TABELA 2.3 – Uma solução ótima da instância do PMTF mostrada na Tabela 2.1.

## 2.2 Revisão Bibliográfica

Há vários estudos que abordam o problema de minimização de trocas de ferramentas em uma única máquina, e suas variações, como máquinas paralelas e tamanhos diferentes das ferramentas. Na Subseção 2.2.1, revisam-se modelos, métodos exatos e heurísticos, considerando o caso mais simples, denominado uniforme, consistindo do PMTF delimitado

na Seção 1.4. Na Subseção 2.2.2 uma revisão sobre algumas extensões e variações existentes do problema é realizada, e por fim, situa-se na literatura os problemas envolvendo fluxos *multicommodity*.

### 2.2.1 PMTF uniforme

Um dos primeiros e impactantes estudos sobre o PMTF foi produzido por Tang e Denardo (1988a). Neste trabalho, eles mostraram que o problema é NP-difícil (assim como em Crama *et al.* (1994)) através de uma redução ao problema do Caixeiro Viajante. Os autores apresentaram um modelo de otimização linear inteira com variáveis binárias para o problema e tentaram resolvê-lo através dos códigos MPSX/MIP da IBM e PIPX (código desenvolvido por Crowder *et al.* (1983) para otimização linear binária) em um IBM 370. Devido ao fato deste modelo ter relaxação linear igual a zero (LAPORTE *et al.*, 2004) os resultados não se mostraram eficientes. Os autores também mostraram que o problema pode ser resolvido em duas decisões: determinar a ordem em que as tarefas devem ser processadas (sequenciamento de tarefas) e determinar as ferramentas a serem trocadas antes do processamento de cada tarefa (alocação de ferramentas). Um algoritmo KTNS (do inglês “*Keep Tool Needed Soonest*”) foi proposto e provado ser a solução ótima para o problema de alocação de ferramentas para uma sequência de tarefas fixada, em tempo polinomial. Com este algoritmo eles propuseram uma heurística (GP) (do inglês “*Greedy Perturbation*”) baseada em três passos: primeiro determina uma sequência de tarefas; utilizando o algoritmo KTNS determina-se a quantidade de troca de ferramentas nesta sequência, e por fim perturba-se a sequência de modo a obter o menor número de trocas de ferramentas. Para testar a heurística eles utilizaram instâncias com  $N=10, 20, 30, 40$  tarefas;  $M=10, 15, 25, 30$  ferramentas e  $C=4, 8, 10, 15$  correspondendo a capacidade da máquina. A conclusão foi que o procedimento GP é eficiente para as instâncias  $N=10, 20, 30$ , mas pouco eficiente quando  $N=40$ . O motivo apresentado para a pouca eficiência quando  $N$  grande se deve ao pouco tempo de CPU, não sendo suficiente para gerar boas soluções.

Bard (1988) propõe um modelo de otimização não linear inteiro com variáveis binárias para o PMTF. A função objetivo foi linearizada e vários códigos foram utilizados para a resolução do modelo linearizado, entre eles o código MPSX/MIP da IBM, em um IBM 3081. Os códigos, entretanto, não obtiveram bons resultados em tempos aceitáveis para uma instância testada pelo autor com  $N=10, M=9$ , e  $C=4$ . Ainda neste trabalho, uma heurística dual foi proposta. Ao dualizar as restrições de capacidade da máquina e as restrições de demanda das ferramentas através dos multiplicadores de Lagrange, dois subproblemas são criados, um resolvido pelo Algoritmo Húngaro e o outro por Programação Dinâmica. A heurística consiste basicamente em três passos iterativos: gerada uma

sequência de tarefas através da solução do modelo relaxado, determina-se uma solução factível, a partir da solução anterior, pelo algoritmo KTNS; perturba-se a solução factível para encontrar melhores soluções; e atualiza-se os multiplicadores para iterar o algoritmo, até que um limite no GAP seja estabelecido. Os testes foram realizados em instâncias com no máximo 25 tarefas e os resultados obtidos foram comparados com um procedimento enumerativo (incompleto), relatando em alguns casos os mesmos valores. O modelo linearizado de Bard é o mesmo modelo proposto por Tang e Denardo (1988a). O autor menciona duas aplicações do PMTF em indústrias e discute uma extensão do problema para várias máquinas, onde a sequência de tarefas permanece fixa e a mesma em todas as máquinas. Por fim, cita duas aplicações do problema em indústrias: um fabricante de equipamentos de aviação e comunicações militares, e um fabricante de eletrônicos.

Crama *et al.* (1994) e Oerlemans (1992) apresentam uma nova prova de que o problema é NP-difícil sempre que a capacidade da máquina for maior ou igual a 2. Eles também apresentam uma formulação inteira com variáveis binárias, baseada em subconjuntos máximos 0-blocos, ou seja, um subconjunto máximo de zeros consecutivos delimitados por 1's em uma linha de uma matriz de tarefa-ferramenta para o problema de alocação de ferramentas quando uma sequência de tarefas é fixa. Eles provam que esta formulação pode ser resolvida em tempo polinomial e fornece uma nova prova para a validade do algoritmo KTNS desenvolvida por Tang e Denardo (1988a). Os autores estabeleceram ligações entre o problema PMTF e outros problemas de otimização combinatória como o problema de permutação de matriz (MÖHRING, 1990), otimização com matriz de restrição gulosa (HOFFMAN *et al.*, 1985), problema de minimização de bloco (KOU, 1977), reconhecimento de intervalos em matrizes (FULKERSON; GROSS, 1965). Como o problema de alocação de ferramentas está bem resolvido pelo algoritmo KTNS, os autores focam no problema de sequenciamento de tarefas e propõem onze heurísticas divididas entre heurísticas de construção e heurísticas de refinamento. Dentre as heurísticas de construção estão quatro baseadas no Problema do Caixeiro Viajante (PCV): *Shortest edge*, *Nearest neighbor*, *Farthest insertion* e *B&B algorithms*; duas baseadas em “minimização de blocos”: *Nearest Neighbor Block Minimization* e *Farthest Insertion Block Minimization*; *Greedy Heuristics* e *Multiple-Start Greedy Heuristic*, e *Interval Heuristic*. Nas heurísticas de refinamento estão as heurísticas *Global 2-opt*, *Restricted 2-opt* e *Load-and-optimize strategy*. Vários testes computacionais foram realizados, e considerando apenas a qualidade da solução encontrada por cada heurística, os autores concluíram que *Multiple-Start Greedy* e *Global 2-opt* foram as melhores, sendo que *Simple Greedy* e *Farthest Insertion Block Minimization* são boas competidoras. Para instâncias densas, *Farthest Insertion* foi a melhor e a mais rápida.

Hertz *et al.* (1998) modelaram o problema de sequenciamento como um PCV e usaram o algoritmo KTNS para resolver o problema de alocação de ferramentas. Eles apresenta-



ram nove novas heurísticas para resolver o PCV resultante: *FI1*, *FI2*, *GENI*, *GENIUS*, *FI\**, *GENI\**, *GENIUS\**, *NN\** e *2-opt\**. As sete primeiras heurísticas são baseadas no PCV, utilizando cinco distâncias diferentes, sendo que duas delas foram desenvolvidas pelos autores. As duas novas distâncias levam em consideração, ao contrário das restantes, uma visão global da máquina, quanto às ferramentas, no exato momento de inserção da próxima tarefa a ser sequenciada. Para testar o efeito das distâncias nas heurísticas, extensivos testes computacionais foram realizados utilizando instâncias geradas de forma semelhante as de Crama *et al.* (1994). Os autores concluíram que as heurísticas *GENI* e *GENIUS* geraram melhores valores de solução do que os apresentados por Crama *et al.* (1994). A heurística *GENI* é uma heurística de construção para resolver o PCV proposta por Gendreau *et al.* (1992) e, a heurística *GENIUS* é a heurística *GENI* combinada com uma fase pós-otimização.

Privault e Finke (1995), Privault e Finke (2000) modelaram o problema PMTF como Problema de Localização de K-Servidores Móveis com Pedido em Massa (do inglês “*k-server problems with bulk requests*”), onde os *slots* de ferramentas na máquina representam os servidores e os conjuntos de ferramentas exigidos por uma tarefa correspondem a um pedido em massa. Os autores propuseram uma heurística adaptada do algoritmo de particionamento de McGeoch e Sleator (1991) para o Problema dos K-Servidores e concluíram que a heurística modificada fornecia melhores limites superiores quando comparada às heurísticas de agrupamento de tarefas Algo1 e Algo2. Essas heurísticas de agrupamento agrupam iterativamente tarefas que usam ferramentas comuns, criando novas tarefas fictícias que exigem um número total de ferramentas que não excedem a capacidade do magazine de ferramentas. As novas tarefas são sequenciadas de acordo com a heurística “*Shortest Edge*” (Algo1) ou a heurística “*Farthest Insertion*” (Algo2) e, posteriormente, uma heurística “*2-opt*” é aplicada para modificar a sequência obtida para melhorar a função objetivo (BENTLEY, 1992).

Djellab *et al.* (2000) reformularam o problema PMTF utilizando uma representação particular de hipergrafos denominada MGEP-Cap formulada em termos do problema MGEP (“*Minimizing the number of total weighted gap in edge-projection*”). Baseados nesta formulação, os autores prouseram uma heurística iterativa “*Iterative Best Insertion*” (IBI) que garante uma solução local para o PMTF, utilizando o algoritmo KTNS para determinar a quantidade de troca de ferramentas para cada sequência estabelecida pela heurística. Os autores concluíram que seus resultados melhoraram os resultados de (CRAMA *et al.*, 1994), em termos de tempo de execução computacional e qualidade da solução. Além disso, os autores afirmam que a heurística IBI não varia com a esparsidade da matriz tarefa-ferramenta.

Shirazi e Frizelle (2001) abordaram um estudo empírico do problema PMTF em sete indústrias que empregam altos processos de tecnologia e fabricação, as quais são indústrias

aeroespaciais, de ferramentas e fabricantes de máquinas CNC. Os autores comprovaram que o problema PMTF, antes apenas abordado na literatura, é real nas indústrias. Também observaram que apesar de todas as indústrias constatarem que trocas de ferramentas dispendem muito tempo e devem ser otimizadas, nenhuma utilizava alguma técnica para tal otimização, e sim um trabalhador experiente (como denominado pelos autores) para realizar as decisões de operações na máquina. O principal objetivo do trabalho foi comparar o comportamento e soluções obtidas pelas heurísticas da literatura com os métodos, se existentes, utilizados nestas indústrias. Para tanto, os autores utilizaram seis heurísticas: *Simple Greedy*, *Multiple Start Greedy*, *Best Position Insertion*, *Shortest Edge*, *Farthest Insertion with lower bounds*, e *Farthest Insertion with upper bounds*. Todas as heurísticas geraram soluções melhores que as utilizadas pelas indústrias, sendo que a heurística *Multiple Start Greedy* foi a mais robusta. Os resultados foram enviados às indústrias as quais implementaram as novas metodologias às suas produções. Por fim, os autores também concluíram que instâncias da literatura diferem das instâncias reais, sendo que no caso destas indústrias em particular, os conjuntos de dados e capacidades das máquinas são bem maiores que os da literatura e pedem que pesquisas futuras devem ser feitas de maneira resultados experimentais reflitam a realidade industrial.

Yanasse e Pinto (2002) propuseram um modelo baseado em fluxo em redes para o PMTF, e desenvolveram alguns pré processamentos para tentar diminuir o tamanho dos problemas, mas não realizam nenhum teste computacional. Em Yanasse e Lamosa (2005), Yanasse e Lamosa (2006), os autores modelaram o PMTF como um Problema Generalizado do Caixeiro Viajante (PGCV) e propuseram um método de solução para resolvê-lo, mas eles não realizaram nenhum estudo computacional.

Para superar as limitações do modelo de Tang e Denardo (1988a), Laporte *et al.* (2004) formularam um modelo de otimização inteiro baseado no PCV, inserindo uma tarefa fictícia, indicando o início e final do processamento das tarefas e modelando as restrições de eliminação do subsequência como em Dantzig *et al.* (1954). Os autores mostraram que sua nova formulação superou a proposta por Tang e Denardo (1988a), em relação aos valores de suas relaxações de programação linear, depois de mostrar que a relaxação de programação linear do modelo de Tang e Denardo (1988a) é igual a zero. Os autores propuseram várias desigualdades válidas para fortalecer seu modelo e dois algoritmos de enumeração exata para resolver o PMTF. O primeiro foi um algoritmo *Branch-and-cut*, no qual, no nó raiz, os autores usaram a heurística *GENIUS* com o algoritmo KTNS para construir uma solução viável. As restrições de eliminação de subsequência não foram consideradas no início. Elas foram introduzidos conforme necessário. Em cada nó da árvore de busca, as restrições de eliminação de subsequência mais violadas foram introduzidas. Quando nenhuma desigualdade violada pôde ser identificada, uma ampla ramificação foi realizada, ou seja, dada uma sequência de tarefas fixas, novos nós, correspondentes às

tarefas que não estão nessa sequência, são criadas. Testes computacionais mostraram que esse algoritmo *Branch-and-cut* foi capaz de resolver na otimalidade algumas instâncias com 9 tarefas. Além disso, os limitantes inferiores obtidos pelo modelo eram melhores do que os fornecidos pelo modelo de Tang e Denardo (1988a), mas ainda eram fracos. O segundo algoritmo era um *Branch-and-Bound* que enumerava todas as sequências possíveis das tarefas. Um limitante superior para o PMTF foi obtido por uma heurística gulosa. Os autores propuseram dois valores para os limitantes inferiores: um limite inferior simples, com base na sequência parcial dos trabalhos sequenciados, e o outro, resultado do Problema da Árvore Mínima para as tarefas ainda não sequenciadas. A regra de ramificação usa a ordem em que as tarefas aparecem no algoritmo guloso. Testes computacionais mostraram que esse algoritmo *Branch-and-Bound* foi capaz de resolver na otimalidade algumas instâncias com 25 tarefas.

Ghiani *et al.* (2007) demonstraram que o PMTF tem a seguinte propriedade: uma sequência de tarefas e sua sequência reversa tem a mesma quantidade de troca de ferramentas. Os autores utilizam esta propriedade no método *Branch-and-bound* proposto por Laporte *et al.* (2004), modificando a regra de ramificação para que o algoritmo considere uma sequência de tarefas apenas uma vez. Uma vantagem de seu método foi indicada no caso onde as instâncias do problema estão próximas das instâncias do PCV (ou seja, quanto maior a porcentagem de *slots* da magazine de ferramentas ocupados por tarefas, na média, mais próxima do PCV a instância do PMTF estava).

Yanasse (2009) propôs um novo limitante inferior ao PMTF com base na semelhança desse problema com o problema da minimização de pilhas abertas ((BECCENERI *et al.*, 2004), do inglês “*Minimization of Open Stacks Problem*” (MOSP)). O novo limitante inferior não foi superado pelos limitantes inferiores anteriores na literatura.

Yanasse *et al.* (2009) propuseram uma esquema enumerativo baseado em ordenamentos parciais para determinar a solução ótima do PMTF. Os autores propuseram a determinação de melhores limitantes inferiores utilizando subconjuntos de tarefas do conjunto original. O método proposto consiste em encontrar este subconjunto parcial de tarefas, e a ele ir acrescentando tarefas, de forma ordenada, de tal maneira que a quantidade mínima de trocas de ferramentas se mantenha ou aumente em pequeno número. O algoritmo KTNS é usado para encontrar as quantidades de trocas de ferramentas em cada subconjunto ordenado. O método foi testado em instâncias geradas aleatoriamente de acordo com os parâmetros indicados em Crama *et al.* (1994) e foi capaz de resolver 1,233 instâncias, contra 1,136 instâncias resolvidas por Crama *et al.* (1994). Os autores concluem afirmando que o método não teve um bom desempenho em instâncias onde a diferença entre o número máximo de ferramentas utilizadas pelas tarefas e a capacidade da máquina é relativamente grande.

Ghiani *et al.* (2010) formularam o PMTF como um problema de ciclo Hamiltoniano

de custo mínimo com função objetivo não linear. Os autores utilizaram a propriedade de simetria, demonstrada também em seu outro artigo para propor um algoritmo do tipo *branch-and-cut*, estabelecendo regras de dominância e usando técnicas de separação para detectar restrições de eliminação de subsequências violadas em sua formulação (GHIANI *et al.*, 2007). Testes computacionais foram realizadas com dois conjuntos de instâncias A e B. O conjunto A foi proposto em Laporte *et al.* (2004) e o conjunto B foi gerado de forma aleatória pelos autores, compreendendo problemas testes similares aos encontrados em indústrias, onde há maior saturação na caixa de ferramentas. O método proposto foi capaz de resolver instâncias de até 45 tarefas e 30 ferramentas ao passo que o método proposto por Laporte *et al.* (2004) resolveu instâncias com até 25 tarefas. Os autores concluem que esta superioridade na performance do método proposto é devido a redução do número de nós no processo de busca pela utilização da propriedade de simetria.

Yanasse (2011) demonstrou que para instâncias do PMTF que apresentam um grafo MOSP em que existem um vértice de corte (ou seja, a remoção deste vértice e todas as arestas a ele incidentes, divide o grafo em vários subgrafos conexos), quando se remove este vértice e todas as arestas incidentes a ele, pode-se construir uma solução ótima a partir de soluções ótimas dos problemas de minimização de trocas de ferramentas correspondentes às componentes conexas do grafo resultante. Em Yanasse (2012) o autor explora o resultado apresentado em Yanasse (2011) para apresentar um algoritmo polinomial para o caso especial que o MOSP correspondente ao PMTF é uma árvore. Relações entre PMTF e MOSP podem ser encontradas em Yanasse (1997a), Linhares e Yanasse (2002).

Al-Fawzan e Al-Sultan (2003) propuseram cinco diferentes versões de uma heurística Busca Tabu (GLOVER, 1986) para resolver o problema de sequenciamento, enquanto utilizaram o algoritmo KTNS para o problema de alocação de ferramentas. Os autores utilizaram estruturas de curto e longo prazo combinadas com esquemas probabilísticos e determinísticos para gerar as vizinhanças. Eles concluíram que os algoritmos propostos se mostraram promissores, sendo que os algoritmos de longo prazo demonstraram uma performance superior. Os autores concluem que os algoritmos podem ser estendidos para problemas de duas ou mais máquinas.

Zhou *et al.* (2005) desenvolvem um algoritmo do tipo *beam search* munido de duas funções simples para garantir a qualidade e rapidez das soluções. O algoritmo resolve o problema de sequenciamento de tarefas e o algoritmo KTNS é usado para determinar a quantidade mínima de trocas de ferramentas. Os autores propuseram um pré-processamento baseado em determinar tarefas cujo conjunto de ferramentas requeridas por uma é subconjunto do conjunto de ferramentas requeridas pela outra, e compararam seus resultados com os de Bard (1988). A conclusão a que chegaram é que seu algoritmo é mais eficiente tanto em tempo de execução computacional quanto em qualidade

da solução.

Salonen *et al.* (2006a) propuseram uma heurística *multistart*, combinando uma técnica de agrupamento de tarefas (SALONEN *et al.*, 2006b), o algoritmo *GENIUS* e o algoritmo KTNS. Os autores testaram seu método com instâncias do PMTF fornecidas por Crama *et al.* (1994) e Smed *et al.* (1999) e concluíram que tiveram um bom desempenho, considerando o *tradeoff* entre qualidade da solução e tempo de execução.

Amaya *et al.* (2008) propuseram um algoritmo de busca local, um algoritmo genético permutativo específico do problema, e um algoritmo memético que combina os dois métodos anteriores. Os autores mostraram que o algoritmo memético teve desempenho superior aos outros dois, e ao método *beam search* proposto por Zhou *et al.* (2005). Depois, Amaya *et al.* (2010) desenvolveram quatro métodos cooperativos, realizaram testes empíricos e constataram que o método cooperativo com topologia baseada em técnicas intermediárias heterogêneas tiveram desempenho melhor que o *beam search* (ZHOU *et al.*, 2005) e a busca tabu (AL-FAWZAN; AL-SULTAN, 2003). Mais tarde, Amaya *et al.* (2011) consideraram 36 modelos cooperativos meméticos, diferindo entre si na combinação particular de metaheurísticas atribuídas aos agentes e sua topologia de conexão. Modelos totalmente meméticos, ou seja, modelos cooperativos nos quais cada agente é dotado de um algoritmo memético (possivelmente diferente), foram os que forneceram os melhores resultados, superando significativamente o algoritmo proposto em Amaya *et al.* (2008).

Em Amaya *et al.* (2012) três diferentes técnicas de busca local (*Hill Climbing*, Busca Tabu, *Simulated Annealing*) foram incorporados em algoritmos evolutivos. Testes computacionais mostraram que o algoritmo memético incorporado à pesquisa *Hill Climbing* produziu os melhores resultados, apresentando melhor desempenho que seus constituintes autônomos, incluindo a heurística *Beam Search* definida em Zhou *et al.* (2005).

Amaya *et al.* (2013) desenvolveram um algoritmo memético baseado na técnica Entropia Cruzada, uma abordagem de Monte Carlo para otimização (RUBINSTEIN, 1999), que evolui uma população baseada em uma função de massa de probabilidade, que é então atualizada em cada geração. Os testes computacionais mostraram um desempenho aprimorado desse novo algoritmo em comparação com os algoritmos apresentados em Amaya *et al.* (2011).

Amaya *et al.* (2019) propuseram um modelo “*deep meta-cooperation*” para resolver o PMTF, ou seja, “modelos cooperativos nos quais pelo menos um de seus componentes é o próprio modelo cooperativo” (AMAYA *et al.*, 2019), e os resultados computacionais mostraram melhor desempenho em relação aos resultados alcançados por Amaya *et al.* (2013).

Senne e Yanasse (2009) desenvolveram três variações da metaheurística *Beam Search* baseadas em um esquema de enumeração que considera ordenamento parcial das tare-

fas. Todas as variações utilizam a busca em profundidade para ramificação da árvore de enumeração. Os autores geraram 1,350 instâncias conforme parâmetros adotados em Laporte *et al.* (2004), e concluíram que o algoritmo que mantém em cada nó da árvore de enumeração, apenas os 3 melhores ramos e escolhe aleatoriamente o próximo nó a ser ramificado, apresentou os melhores resultados dentre os métodos propostos.

Chaves *et al.* (2012) propuseram um método heurístico baseado em duas fases. Na primeira fase, uma solução é construída a partir de um grafo PMTF. Na segunda fase, a solução gerada anteriormente é melhorada através da aplicação da metaheurística Busca Local Iterativa (STÜTZLE, 1999). Um grafo PMTF é um grafo em que os vértices são as ferramentas e existe uma aresta se, e somente se, as duas ferramentas incidentes nessa aresta forem usadas juntas no processamento de alguma tarefa. Este grafo também é conhecido como grafo MOSP (YANASSE, 1997b). A solução encontrada por este método é utilizada como limitante superior inicial para o método enumerativo proposto em Yanasse *et al.* (2009). O método heurístico se mostrou eficiente, obtendo a solução ótima para a grande maioria dos exemplares testados.

Burger *et al.* (2015) consideraram o problema de sequenciamento de impressão de cores (CPSP, “*colour print scheduling problem*”), que surge durante o processo de impressão a cores de embrulhos de plástico ou alumínio, usados para embalar barras de chocolate, batatas frita e outros produtos. Dados que cada trabalho necessita de uma quantidade de cores, e a impressora tem uma capacidade limitada de cartuchos, entre a produção de um e outro trabalho, se há cores que serão utilizadas no trabalho seguinte que não estão no trabalho anterior, o processo de produção deve ser interrompido, os cartuchos com as cores que não serão utilizados devem ser lavados e enchidos com as cores a serem utilizadas. O problema consiste em minimizar a quantidade de cartuchos lavados na impressora. Os autores modelaram este problema como um PMTF. Eles propuseram e implementaram um método heurístico baseado em Salonen *et al.* (2006b). Instâncias aleatórias foram criadas e resolvidas com a heurística proposta. As instâncias também foram resolvidas usando CPLEX 11.0, com as formulações de Tang e Denardo (1988a) e Laporte *et al.* (2004). A heurística proposta produziu resultados relativamente bons, especialmente nos casos em que o número de ferramentas necessárias para processar um trabalho estava próximo da capacidade do magazine de ferramentas. Os autores também aplicaram a heurística proposta a um estudo de caso real de uma empresa de impressão localizada no Cabo Ocidental da África do Sul, obtendo uma melhoria, em média, de 61,6% no número de trocas de cartuchos de tinta.

Laporte *et al.* (2004) mostraram que a relaxação linear de sua formulação superou a relaxação linear da formulação proposta por Tang e Denardo (1988a). Segundo Catanzaro *et al.* (2015), a razão pela qual poucos autores propuseram métodos exatos para o PMTF deve-se ao fato de os limites inferiores obtidos pelas formulações propostas na literatura

serem fracos. Assim, Catanzaro *et al.* (2015) desenvolveram três formulações de otimização linear inteira para o PMTF para fortalecer os limites inferiores das formulações existentes na literatura. Eles mostraram que os limites inferiores obtidos pelas relaxções lineares de suas formulações eram pelo menos iguais aos obtidos por Laporte *et al.* (2004). A melhor formulação dentre as propostas consistiu em uma formulação de fluxo em arcos baseados nas propriedades estruturais do subproblema de alocação de ferramentas e 0-blocos. Apesar do fortalecimento da relaxação linear, o modelo proposto foi capaz de resolver na otimalidade, apenas dois conjuntos de instâncias, sendo eles:  $N=10$ ,  $M=10$ ,  $C=4$ , e  $N=10$ ,  $M=10$ ,  $C=5$ .

Chaves *et al.* (2016) desenvolveram um método híbrido combinando *Clustering Search* (CS) e *Biased Random Key Genetic Algorithm* (BRKGA). Esse método detecta regiões promissoras no espaço de busca, ou seja, regiões que poderiam conter boas soluções, e aplica-se uma heurística de busca local apenas nessas regiões. A heurística de busca local utilizada é a *Variable Neighborhood Descent* (VND). Os autores usaram 1,510 instâncias (SENNE; YANASSE, 2009; CRAMA *et al.*, 1994) para testar o método proposto, e encontrou solução ótima em 1,360 instâncias. Nas 150 instâncias restantes, o método encontrou melhores soluções em relação as encontradas na literatura.

Beezão (2016) estudou o PMTF como um Problema do Caixeiro Viajante Generalizado (YANASSE; LAMOSA, 2005; YANASSE; LAMOSA, 2006) e propôs quatro versões de um algoritmo de redução de domínio que foram: estratégia gulosa, estratégia reversa, estratégia de aproximação dos nós isolados e *Path relinking*. A autora concluiu que seu método é competitivo quando aplicado em instâncias de pequeno e médio portes.

Paiva e Carvalho (2017) introduziram uma nova representação baseada em grafo para o PMTF chamada “*Tool Graph*”, onde os nós representam as ferramentas, e existe uma aresta entre os nós  $i$  e  $j$  se há uma tarefa que necessita das ferramentas  $i$  e  $j$ . O peso de cada aresta corresponde a quantidade de tarefas que utilizam as ferramentas que formam a aresta. Os autores propuseram uma metaheurística *Iterated Local Search*, em que a solução inicial foi fornecida por uma heurística baseada no “*Tool Graph*” que analisa a relação entre as ferramentas, e propuseram um método de busca local baseado em agrupamento de 1-blocos, ou seja, um subconjunto máximo de 1's consecutivos delimitados por zeros em uma linha da matriz ferramenta-tarefa para o problema de alocação de ferramentas, quando uma sequência de tarefas está fixa. Os autores avaliaram a metaheurística proposta em 1,670 instâncias de diferentes autores (CRAMA *et al.*, 1994; YANASSE *et al.*, 2009; CATANZARO *et al.*, 2015), e concluíram que seu desempenho é superior à metaheurística proposta por Chaves *et al.* (2016).

Azevedo e Carvalho (2017) abordaram o PMTF modelado como PCV, adotando as cinco métricas de distâncias propostas na literatura (CRAMA *et al.*, 1994; HERTZ *et al.*, 1998). Uma discussão é realizada ao fato de somente métodos heurísticos serem utilizados

para a solução destes modelos. Os autores utilizaram o método exato Concorde (COOK, 2016), atual estado da arte para solução do PCV, em três conjuntos de instâncias, e concluíram que os modelos possuem baixa qualidade, já que nenhum deles conseguiu se aproximar razoavelmente dos resultados obtidos por (PAIVA; CARVALHO, 2017), mesmo para instâncias consideradas mais fáceis, ou de dimensões reduzidas.

Ahmadi *et al.* (2018) modelaram o problema de sequenciamento de tarefas por um PCV de segunda ordem (JÄGER; MOLITOR, 2008) chamado 2-JSeP, e mostraram que esta versão induzida do problema é NP-difícil. Em um PCV de segunda ordem, os custos não dependem dos arcos, mas de cada sequência de três vértices consecutivos na rota. Os autores utilizaram a solução obtida pelo 2-JSeP para semear um algoritmo genético combinado com aprendizado de máquina (*Dynamic Q-learning-based Genetic Algorithm (DQGA)*) e compararam seus resultados com os de Paiva e Carvalho (2017). O algoritmo proposto teve um desempenho superior em relação ao de Paiva e Carvalho (2017) tanto em tempo de execução, quanto em qualidade de solução.

Em sua dissertação de Mestrado, Dehaybe (2018), abordou o PMTF e propôs uma metaheurística hibridizando Colônia de Formigas (DORIGO, 1992) com uma nova busca local nomeada *2.75-opt*, que combina *2-opt*, dois movimentos simples de *3-opt* e um *job swap* (BENTLEY, 1992). A técnica proposta teve um desempenho superior em relação as técnicas propostas por Ahmadi *et al.* (2018) e Paiva e Carvalho (2017) em muitas instâncias, sendo capaz de gerar novas melhores soluções para algumas instâncias.

Mecler *et al.* (2019) propuseram uma busca genética híbrida (do inglês *hybrid genetic searches*, HGS), normalmente utilizada em problemas de roteamento, para o PMTF. Um HGS consiste de um algoritmo genético com busca local e gestão da diversidade populacional. Os autores consideraram um objetivo secundário para controlar a exploração de soluções com mesmo valor objetivo, fazendo com que pequenos 0-blocos guiem a busca por melhores soluções. Os resultados dos testes computacionais mostraram que o método proposto teve uma melhor performance em relação aos métodos já existentes, tanto em qualidade de solução quanto em tempo de execução. Por fim, os autores propuseram três novos conjuntos de instâncias, com o número de tarefas variando de 50 a 70, e o número de ferramentas variando de 75 a 105.

Todas as referências do PMTF uniforme estão resumidas na Tabela 2.4, cuja coluna Categoria indica se na referência correspondente, o PMTF foi abordado via modelo de otimização matemática (modelo), métodos heurísticos e/ou metaheurísticos (heurística), ou estudo teórico (teoria). Já a coluna Técnica especifica detalhes da categoria do método de solução abordado.



Referência	Categoria	Técnica
Bard (1988)	modelo, heurística	modelo não linear, relaxação lagrangeana
Tang e Denardo (1988a)	modelo, heurística	modelo inteiro misto, perturbação gulosa
Crama <i>et al.</i> (1994)	heurística	baseadas no PCV
Privault e Finke (1995)	heurística	baseadas no PCV
Hertz <i>et al.</i> (1998)	heurística	baseadas no PCV
Djellab <i>et al.</i> (2000)	heurística	hipergrafo
Privault e Finke (2000)	heurística	particionamento
Shirazi e Frizelle (2001)	heurística	aplicação em indústrias
Yanasse e Pinto (2002)	modelo	modelo inteiro
Al-Fawzan e Al-Sultan (2003)	heurística	Busca Tabu
Laporte <i>et al.</i> (2004)	modelo, exato	modelo inteiro, “ <i>branch-and-bound</i> ” e “ <i>branch-and-cut</i> ”
Yanasse e Lamosa (2005)	modelo	PCV generalizado
Zhou <i>et al.</i> (2005)	heurística	“ <i>Beam Search</i> ”
Salonen <i>et al.</i> (2006a)	heurística	heurística de melhoria
Yanasse e Lamosa (2006)	heurística	utilizando o PCV generalizado
Ghiani <i>et al.</i> (2007)	exato	“ <i>branch-and-bound</i> ”
Amaya <i>et al.</i> (2008)	heurística	algoritmo memético
Senne e Yanasse (2009)	heurística	“ <i>Beam Search</i> ”
Yanasse (2009)	limitante	baseado no MOSP
Yanasse <i>et al.</i> (2009)	heurística	beam search
Amaya <i>et al.</i> (2010)	heurística	método cooperativo
Ghiani <i>et al.</i> (2010)	modelo, exato	modelo não linear, “ <i>branch-and-cut</i> ”
Amaya <i>et al.</i> (2011)	heurística	modelo cooperativo memético
Yanasse (2011)	teoria	relação com MOSP
Amaya <i>et al.</i> (2012)	heurística	algoritmo memético com busca local
Chaves <i>et al.</i> (2012)	heurística	ILS
Yanasse (2012)	exato	MOSP é uma árvore
Amaya <i>et al.</i> (2013)	heurística	entropia cruzada
Burger <i>et al.</i> (2015)	heurística	agrupamento
Catanzaro <i>et al.</i> (2015)	modelo	modelo linear inteiro
Beezão (2016)	heurística	redução de domínio do PCV generalizado
Chaves <i>et al.</i> (2016)	heurística	BRKGA
Paiva e Carvalho (2017)	heurística	ILS
Ahmadi <i>et al.</i> (2018)	heurística	algoritmo genético com aprendizado de máquina
Amaya <i>et al.</i> (2019)	heurística	modelo “ <i>deep meta-cooperation</i> ”
Mecler <i>et al.</i> (2019)	heurística	busca genética híbrida
Silva <i>et al.</i> ()	modelo	fluxo <i>multicommodity</i>

TABELA 2.4 – Visão geral da literatura do PMTF uniforme.

### 2.2.2 Variações do PMTF

Nos parágrafos subsequentes discute-se estudos feitos com variantes do PMTF, por exemplo, considerando máquinas paralelas e tamanhos não uniformes das ferramentas.

Tang e Denardo (1988b) abordaram o problema de trocas de ferramentas num contexto em que várias ferramentas podem ser trocadas de uma só vez. Neste contexto, o objetivo é minimizar a quantidade total de instantes em que trocas são realizadas, já que os autores consideram (e se torna mais apropriado) que o tempo de troca é constante e independente da quantidade de trocas realizadas. É mostrado que este problema generaliza o problema de *bin packing*. Os autores propuseram um método do tipo *branch-and-bound* e estenderam o problema para o caso de várias máquinas.

Widmer (1991) abordaram o problema num ambiente *job shop* com o objetivo de minimizar o *makespan*, o número de trocas de ferramentas, o número de atrasos e o número de *oversteppings*, ou seja, a quantidade de máquinas que não estão hábeis para completar seu processo de produção no período. Os autores propuseram um método baseado na metaheurística Busca Tabu e afirmaram que seus resultados são satisfatórios.

Privault e Finke (1995) consideraram que os tempos de trocas entre as ferramentas não são iguais. Com o intuito de minimizar a soma dos tempos de trocas de ferramentas para uma dada sequência de tarefas, os autores definiram pesos entre as ferramentas  $i$  e  $j$  como sendo o tempo de troca de remover a ferramenta  $i$  e inserir a  $j$ , e resolveram um problema de encontrar o fluxo máximo de custo mínimo em um grafo acíclico. Os autores mostraram que este problema pode ser resolvido em  $C^3N^2$  operações. Para o caso em que os tempos de trocas são iguais, os autores mostraram que este método é alternativo ao algoritmo KTNS quando  $M$  é muito maior que  $N$ . Em um segundo momento de seu trabalho, várias heurísticas foram propostas para resolver o problema de sequenciamento de tarefas, sendo que a heurística “*Super Task*”, que consiste em agrupar tarefas que usam ferramentas em comum, e uma adaptação do algoritmo *online* de particionamento (MCGEOCH; SLEATOR, 1991) obtiveram os melhores resultados.

Gómez e Lorena (1998) apresentaram um problema de sequenciamento de tarefas e carregamento de ferramentas em uma única máquina. Os autores consideraram datas de vencimentos das tarefas, períodos de turnos de produção e capacidade da máquina quanto às ferramentas. Em sua função objetivo, foram considerados *makespan*, atraso, tempo de *setup* (tempo de troca das ferramentas) e tempo ocioso dos turnos. O método de solução proposto foi composto por duas fases: na primeira, uma sequência inicial foi gerada usando um algoritmo para identificação de grupos que considera a capacidade da máquina, denominado AMIG (GOMEZ, 1996); em seguida, na segunda fase, a solução inicial foi melhorada via metaheurística Busca Tabu, gerando novas soluções que condizem com as estratégias determinadas pelos pesos nas parcelas da função objetivo. Os autores

concluíram que este método pode ser aplicado para resolver o PMTF desde que os pesos não relacionados às trocas de ferramentas na função objetivo se mantenham constantes com os valores correspondentes aos de uma solução denominada solução não tendenciosa (ou seja, a melhor solução obtida considerando pesos proporcionais).

Rupe e Kuo (1997) abordaram o problema PMTF considerando que tarefas podem requerer mais ferramentas que a capacidade da máquina, e que tarefas podem ser produzidas em duas ou mais partes. Os autores propuseram um modelo de otimização não linear e um algoritmo GTNS (do inglês, “*get tool needed soonest*”), similar ao algoritmo KTNS, considerando uma dada sequência fixa de tarefas. Quando a sequência de tarefas não é conhecida *a priori*, os autores propuseram uma heurística similar à proposta por Crama *et al.* (1994).

Khan *et al.* (2000) abordaram o caso de duas máquinas paralelas e idênticas, com o objetivo de minimizar o *makespan*. Os autores consideraram que o tempo de reposição da ferramenta da caixa de ferramenta para o *slot* da máquina e vice-versa é constante, mas o tempo total de troca da ferramenta varia de ferramenta para ferramenta. Um procedimento heurístico foi proposto e aplicado a problemas reais industriais.

Matzliach e Tzur (2000) provaram que o problema de alocação de ferramentas com tamanhos não uniformes de ferramentas é NP-completo desconsiderando a localização física das ferramentas na máquina. Os autores abordaram neste contexto, o problema de gerenciamento de armazenamento de itens com tamanhos diferentes, em dois armazéns, em que a capacidade do armazém representa a capacidade da magazine de ferramentas. Um item é solicitado ao armazém mais próximo, que no entanto não é capaz de armazenar todos os itens que podem ser solicitados. A sequência dos itens solicitados é conhecida e o custo de transferência de um item depende do seu tamanho. O objetivo é minimizar o custo total de transferência. Os autores propuseram duas heurísticas construtivas que obtiveram bons resultados.

Fathi e Barnette (2002) consideraram o problema de trocas de ferramentas em máquinas paralelas idênticas, com o objetivo de minimizar o *makespan*. As autoras propuseram três heurísticas para sua solução: busca local, processamento de lista (“*list-processing*”, heurística comumente usada no contexto de máquinas paralelas) e heurística construtiva. Experimentos computacionais limitados foram realizados, e concluíram que a primeira e terceira heurísticas tiveram um desempenho melhor, especialmente em instâncias em que havia alguma estrutura aparente na matriz de tarefa-ferramenta.

Song e Hwang (2002) abordaram o problema de minimizar a frequência de movimentos do transportador de ferramentas. As ferramentas que são necessárias, mas não estão disponíveis na caixa de ferramentas, são transferidas por um transportador automático de ferramentas, como um veículo a motor, um veículo guiado automático ou um robô. Os

autores afirmaram que o tempo de transportar ferramentas tem um impacto significativo no tempo total de processamento, demonstraram que o algoritmo KTNS não garante otimalidade para este problema, e propuseram uma política de ferramentas baseada no trabalho de Rupe e Kuo (1997).

Ghrayeb *et al.* (2003) consideraram o problema de sequenciar circuitos impressos (PCPs) e fizeram uma analogia ao PMTF, sendo as tarefas, os circuitos impressos, e os componentes eletrônicos como ferramentas. Os autores propuseram uma formulação matemática baseada em otimização inteira com variáveis binárias e uma heurística de construção. Testes computacionais foram realizados, e o desempenho da heurística foi comparado com o método proposto por Al-Fawzan e Al-Sultan (2003), demonstrando sua superioridade.

Tzur e Altman (2004) abordaram o problema de trocas de ferramentas *offline* em que cada ferramenta pode ter um tamanho diferente. Os autores propuseram uma formulação inteira para o problema e uma adaptação da política KTNS, denominada KSTNS (“*Keep smaller tools needed soonest*”), que não necessariamente é ótima, em que as ferramentas que serão mantidas na máquina tem relação ao uso mais rápido e aos seus tamanhos. Como o problema é complexo e a formulação não foi capaz de resolver instâncias grandes, foi proposta uma heurística *Alladin* e testada em instâncias utilizadas por Hertz *et al.* (1998), comprovando uma melhor performance quando comparada com heurísticas modificadas da literatura.

Van Hop (2005) considerou além de ferramentas com tamanhos não uniformes, capacidade relaxada da máquina, ou seja, podem haver tarefas que precisam de mais ferramentas que a máquina suporta. O autor considerou trocas simultâneas de ferramentas em seu estudo assim como divisão parcial e completa das ferramentas das tarefas, e propôs um modelo que maximiza as trocas simultâneas de ferramentas e minimiza as trocas totais de ferramentas em uma única abordagem objetiva. O autor também propôs uma heurística construtiva que teve bom desempenho.

Raduly-Baka *et al.* (2005) abordaram o mesmo problema que Tzur e Altman (2004), e propuseram um algoritmo baseado nos algoritmos propostos por Djellab *et al.* (2000), Hirvikorpi *et al.* (2006) e Tzur e Altman (2004). Os autores compararam os resultados da heurística proposta com a heurística *Alladin* (TZUR; ALTMAN, 2004), apresentando um bom desempenho tanto em qualidade de solução quanto em tempo computacional.

Hirvikorpi *et al.* (2006) estenderam os trabalhos de Matzliach (1998) e Matzliach e Tzur (2000) considerando custos de inserção, remoção e movimento de ferramentas com tamanhos diferentes. Os autores consideraram que cada tarefa requer apenas uma ferramenta, a ordem de processamento das tarefas é fixa, e propuseram um modelo de programação inteira e três heurísticas, no contexto de montagem de placa de circuito impresso (*PCB*

*Assembly - Printed circuit board assembly*). Os autores também estenderam os estudos ao caso em que várias ferramentas são requeridas por tarefa, e propuseram duas heurísticas para resolver o problema.

Karakayali e Azizoglu (2006) abordaram o problema de minimizar o tempo total de fluxo de todas as tarefas, em que o tempo de fluxo de uma tarefa é dado pela diferença entre o tempo de conclusão de produção e o tempo de liberação para o chão de fábrica. Os autores mostraram que este problema é NP-difícil, que o algoritmo KTNS é válido e desenvolveram um método exato *branch-and-bound* baseado em precedência. A precedência é estabelecida afirmando que existe um sequenciamento ótimo em que a tarefa  $i$  precede a  $j$  se o tempo de processamento de  $i$  for menor que  $j$ , e se o conjunto de ferramentas requeridas por  $i$  for subconjunto das ferramentas requeridas por  $j$ . Os autores propuseram vários limitantes superiores e inferiores, e através de testes feitos com instâncias geradas aleatoriamente por distribuições uniformes, mostraram que seu método pode resolver satisfatoriamente problemas de tamanhos moderados.

Crama *et al.* (2007) mostraram que o problema de minimizar a quantidade total de trocas de ferramentas com tamanhos não uniformes de ferramentas, considerando uma dada sequência fixa de tarefas é fortemente NP-completo, e pode ser resolvido em tempo polinomial quando a capacidade da máquina também é conhecida.

Konak e Kulturel-Konak (2007) propuseram um método Colônia de Formiga para minimizar o número de instantes de troca de ferramentas (MTSIP, do inglês “*Minimum number of tool switching instants problem*”). O problema surge quando pode haver trocas simultâneas de ferramentas ou quando o tempo de troca das ferramentas é dependente do número de troca de ferramentas. Os autores mostraram que o método é promissor sendo que na grande maioria dos problemas testes, a solução ótima foi encontrada. Depois, Konak *et al.* (2008) desenvolveram dois algoritmos Busca Tabu, um utilizando técnica de desempate (“*tie breaking*”) e o outro, uma codificação alternativa. Testes computacionais mostraram que os algoritmos encontraram soluções de alta qualidade em tempo razoável.

Karzan e Azizoglu (2008) abordaram o PMTF minimizando o número de movimentos do transportador de ferramentas. As autoras propuseram vários valores de limitantes inferiores e superiores, um algoritmo *branch-and-bound* e um algoritmo *Beam Search*, utilizando os limitantes propostos. O Algoritmo *branch-and-bound* foi capaz de resolver algumas instâncias com 25 tarefas e 25 ferramentas. O algoritmo *Beam Search* produziu resultados muito próximos dos ótimos em tempos de CPU muito pequenos. Por fim, as autoras estenderam o problema considerando também minimizar o tempo total de fluxo, que inclui tempos de processamento, transporte e troca.

Zeballos (2010) considerou uma variação do problema PMTF. Os autores abordaram limitações no número de ferramentas, vida útil das ferramentas, capacidade das máquinas,

máquinas paralelas, e tempo de processamento dependente. Além disso, eles abordaram o problema de forma integrada, considerando o planejamento e alocação de ferramentas, a atribuição de máquinas, o roteamento e as decisões de tempo das tarefas. Os autores propuseram uma abordagem via Programação por Restrições (“*Constraint Programming*”) e uma estratégia de busca denominada GLMT (“*Guided load of machines and tools*”), que demonstrou um bom desempenho computacional para problemas de tamanho pequeno e médio, obtendo em alguns casos a solução ótima. Para problemas grandes o método conseguiu encontrar soluções factíveis em curtos tempos de processamento.

Sarmadi e Gholami (2011) abordaram o PMTF no contexto de máquinas idênticas, considerando duas ou mais máquinas, em que cada máquina tem uma capacidade diferente de ferramentas. Os autores propuseram um modelo de otimização não linear inteiro misto e conseguiram resolver algumas instâncias de até 15 tarefas, 10 ferramentas e 3 máquinas (CALMELS, 2019).

Adjashvili *et al.* (2015) abordaram o “problema de minimização da parada da máquina” (do inglês, *machine stop minimization problem*). Uma máquina é parada quando há violação de *setup*, ou seja, sendo que há troca da ferramenta  $i$  para  $j$ , em um determinado *slot*, o tempo total ocioso deste *slot* antes de processar a ferramenta  $j$  é menor que o tempo de *setup* da ferramenta  $i$ . Os autores mostraram que o problema é NP-difícil e o aplicaram em uma indústria de correspondência.

Raduly-Baka e Nevalainen (2015) abordaram o problema modular de troca de ferramenta. Uma única máquina é equipada com um compartimento de ferramentas *online* que consiste em vários módulos de alimentadores de ferramentas que podem ser substituídas. Os módulos podem conter várias ferramentas necessárias para produção das tarefas. Além dos módulos *online*, há um conjunto de módulos *offline* que podem ser alterados na máquina durante uma mudança de produção de tarefa. As ferramentas entre tarefas podem ser trocadas individualmente ou substituindo um módulo inteiro contendo várias ferramentas. O problema consiste em alimentar ferramentas nos módulos, para que o carregamento de módulos e ferramentas seja minimizado. As ferramentas tem tamanhos uniformes e custos de carregamento unitários. Os autores mostraram que o problema é NP-difícil e, no caso em que o número de módulos e capacidade são fixos, o problema pode ser resolvido em tempo polinomial.

Beezão *et al.* (2017) abordaram o problema de trocas de ferramentas no contexto de máquinas paralelas idênticas, com tempo de processamento. Foram propostas duas formulações matemáticas para o problema, com o objetivo de minimizar o *makespan*, e uma metaheurística *Adaptative Large Neighborhood Search* (ALNS) para resolvê-lo. Resultados computacionais mostraram que os modelos tem desempenho limitado a instâncias com 15 tarefas, e que a metaheurística teve melhores resultados que os apresentados por Fathi e Barnette (2002). Diessen (2019) estendeu o problema abordado por Beezão *et al.*

(2017) considerando restrições de precedência, conjuntos de ferramentas com tamanhos não uniformes, uma função multi-objetivo e momentos de liberação das operações. O autor propôs uma formulação para o problema e duas heurísticas.

Baykasoglu e Ozsoydan (2017) abordaram o PMTF conjuntamente com o problema de indexação do trocador automático de ferramentas (ATC) que consiste em alocar as ferramentas às posições, *slots* das máquinas. Dereli e Filiz (2000) afirmaram que a determinação das melhores posições na máquina são cruciais para reduzir o tempo de não utilização da máquina. Os autores propuseram um algoritmo *Simulated Annealing* para o problema integrado.

Furrer e Mütze (2017) consideraram o problema PMTF com tempo de *setup* e tempo de processamento. Eles definiram um novo problema “*lex-minimization of stops and tool switches*”, em que uma “parada” no processamento da máquina acontece quando dadas duas tarefas  $i$  e  $j$  (sequenciadas nesta ordem), a soma dos tempos de processamento de todas as tarefas compreendidas entre  $i$  e  $j$  é estritamente menor que o tempo de *setup* de  $j$ . Os autores propuseram um método *branch-and-bound* baseado em estrutura algorítmica que pode ser ajustado para o PMTF original e suas variações. O método foi testado em instâncias reais de Adjashvili *et al.* (2015) e em instâncias geradas aleatoriamente. Instâncias muito maiores que as estudadas até então na literatura foram geradas, com  $N=1000$  e  $C=1000$ , e todas foram resolvidas otimamente em poucos segundos.

Farughi *et al.* (2017) consideraram o problema de trocas de ferramentas com limitações da vida útil e falha das ferramentas. Os autores estudaram o impacto da confiabilidade das ferramentas na minimização do número de trocas de ferramentas e propuseram um modelo matemático para o problema. Devido à complexidade do problema, instâncias de tamanhos médios e grandes não foram resolvidas em um tempo razoável. Os autores desenvolveram um algoritmo genético e utilizaram o algoritmo KTNS para obter a solução do problema.

Schwerdfeger e Boysen (2017) aplicaram o PMTF em um sistema de separação de pedidos em que um guindaste realoca continuamente as unidades de manutenção de estoque (do inglês, “*stock keeping units*” - SKUs), em que as tarefas são os pedidos e as ferramentas são os SKUs. Os autores utilizaram como objetivo minimizar o número máximo de trocas entre quaisquer pares de pedidos sucessivos (min-max). Também mostraram que o problema é NP-difícil e desenvolveram um algoritmo KSNS (do inglês *Keep SKU Needed Soonest*) baseado no algoritmo KTNS. Os autores utilizaram o modelo de Tang e Denardo (1988a), propuseram um método heurístico e um método baseado em *branch-and-bound* para solução do problema, e mostraram que o novo objetivo min-max supera o objetivo tradicional de soma mínima na maioria dos casos.

Dadashi *et al.* (2016) estudaram o PMTF considerando a vida útil das ferramentas

e tempos diferentes de troca para cada ferramenta. Os autores propuseram um modelo inteiro misto para o problema e desenvolveram um algoritmo genético para sua resolução. O modelo foi resolvido pelo método *Branch-and-bound* no *software* Lingo 8.0 apenas para pequenas instâncias.

Özpeynirci *et al.* (2016) abordaram o PMTF em máquinas paralelas com o objetivo de minimizar o *makespan*. Os autores propuseram dois modelos de otimização inteira mista, mas não foram capazes de resolver instâncias com mais de 15 tarefas. Os autores também propuseram um algoritmo Busca Tabu mas não compararam com outros métodos pois o problema, segundo os autores, ainda não tinha sido abordado na literatura. Gökgür *et al.* (2018) propuseram três modelos baseados em programação por restrição para este problema, e os testes computacionais mostraram que um dos modelos propostos é superior que a Busca Tabu proposta por Özpeynirci *et al.* (2016).

Calmels (2019) elaborou uma revisão da literatura considerando PMTF e suas variações.

Todas as referências das variações do PMTF estão resumidas na Tabela 2.5, cuja coluna Variação indica qual especificidade foi abordada na referência correspondente, e a coluna Categoria indica se o PMTF foi abordado via modelo de otimização matemática (modelo), métodos heurísticos e/ou metaheurísticos (heurística), ou estudo teórico (teoria).

## 2.3 Modelos da literatura

Nesta seção apresentam-se os principais modelos da literatura para o PMTF uniforme e analisam-se alguns de seus aspectos teóricos.

### 2.3.1 Modelo de Tang e Denardo - TD

O primeiro modelo proposto para o PMTF é devido a Tang e Denardo (1988a). Seja  $u_{jk}$  a variável de decisão igual a 1 se a tarefa  $j \in J$  é processada na posição  $k \in K$ , e 0, caso contrário. Seja  $v_{kt}$  a variável de decisão igual a 1 se a ferramenta  $t \in T$  está na máquina processando alguma tarefa na posição  $k \in K$ , e 0, caso contrário. Seja  $w_{kt}$  igual a 1 se a ferramenta  $t \in T$  está na máquina processando alguma tarefa na posição  $k \in K$ , mas não está na máquina durante o processamento da tarefa da posição  $k - 1$ , e 0, caso contrário, onde  $K := \{1, \dots, N\}$ . Considere  $J_t$  o conjunto de tarefas que utilizam a ferramenta  $t \in T$ . O modelo TD é dado por:



Referência	Variação	Categoria
Tang e Denardo (1988b)	instante de troca	exato
Widmer (1991)	<i>job shop</i>	heurística
Privault e Finke (1995)	tempos de trocas desiguais	heurística
Gómez e Lorena (1998)	<i>makespan</i> , atraso, <i>setup</i> , tempo ocioso	heurística
Rupe e Kuo (1997)	tarefas excedem capacidade	modelo, heurística
Khan <i>et al.</i> (2000)	máquinas paralelas e idênticas	heurística
Matzliach e Tzur (2000)	tamanho não uniforme das ferramentas	heurística
Fathi e Barnette (2002)	máquinas paralelas e idênticas	heurística
Song e Hwang (2002)	movimentos do transportador de ferramentas	heurística
Ghrayeb <i>et al.</i> (2003)	contexto de PCPs	modelo, heurística
Tzur e Altman (2004)	tamanho não uniforme das ferramentas	modelo, heurística
Van Hop (2005)	tamanho não uniforme das ferramentas, capacidade relaxada da máquina	modelo, heurística
Raduly-Baka <i>et al.</i> (2005)	tamanho não uniforme das ferramentas	heurística
Hirvikorpi <i>et al.</i> (2006)	tamanho não uniforme das ferramentas	modelo, heurística
Karakayali e Azizoglu (2006)	tempo total de fluxo de todas as tarefas	exato
Crama <i>et al.</i> (2007)	tamanho não uniforme das ferramentas	teoria
Konak e Kulturel-Konak (2007)	instante de troca	heurística
Konak <i>et al.</i> (2008)	instante de troca	heurística
Karzan e Azizoglu (2008)	movimentos do transportador de ferramentas	exato, heurística
Zeballos (2010)	vida útil, máquinas paralelas, abordagem integrada	heurística
Sarmadi e Gholami (2011)	máquinas com capacidades diferentes	modelo
Adjashvili <i>et al.</i> (2015)	violação de <i>setup</i>	heurística
Raduly-Baka e Nevalainen (2015)	trocas de módulos de ferramentas	teoria
Beezão <i>et al.</i> (2017)	máquinas paralelas idênticas	modelo, heurística
Baykasoglu e Ozsoydan (2017)	abordagem integrada	heurística
Furrer e Mütze (2017)	<i>setup</i> e tempo de processamento	exato
Farughi <i>et al.</i> (2017)	vida útil e falhas	modelo, heurística
Schwerdfeger e Boysen (2017)	objetivo min-max	exato, heurística
Dadashi <i>et al.</i> (2016)	vida útil	modelo, heurística
Özpeynirci <i>et al.</i> (2016)	máquinas paralelas	modelo, heurística
Calmels (2019)	revisão da literatura	

TABELA 2.5 – Visão geral da literatura de variações do PMTF.

$$\text{Min } Z_{TD} = \sum_{k \in K \setminus \{1\}} \sum_{t \in T} w_{kt} \quad (2.3)$$

sujeito a:

$$\sum_{j \in J} u_{jk} = 1, \quad k \in K, \quad (2.4)$$

$$\sum_{k \in K} u_{jk} = 1, \quad j \in J, \quad (2.5)$$

$$\sum_{j \in J_t} u_{jk} \leq v_{kt} \quad k \in K, t \in T, \quad (2.6)$$

$$\sum_{t \in T} v_{kt} \leq C, \quad k \in K, \quad (2.7)$$

$$v_{kt} - v_{(k-1)t} \leq w_{kt}, \quad k \in K \setminus \{1\}, t \in T, \quad (2.8)$$

$$u_{jk} \in \{0, 1\}, \quad k \in K, j \in J, \quad (2.9)$$

$$v_{kt} \in \{0, 1\}, \quad k \in K, t \in T, \quad (2.10)$$

$$w_{kt} \in \{0, 1\}, \quad k \in K, t \in T. \quad (2.11)$$

A função objetivo (2.3) minimiza o número total de trocas de ferramentas. As restrições (2.4) e (2.5) garantem que toda tarefa será processada em uma, e apenas uma posição. As restrições (2.6) asseguram que uma tarefa será processada na posição  $k$  somente se as ferramentas necessárias para sua produção estiverem na máquina. As restrições (2.7) são restrições de capacidade, garantindo que nunca haja mais que  $C$  ferramentas na máquina. As restrições (2.8) garantem que uma troca da ferramenta  $t$  deve acontecer sempre que ela estiver na posição  $k$  mas não na posição  $k - 1$ . As restrições (2.9) - (2.11) são restrições de domínio de variáveis.

O número de variáveis deste modelo é  $N^2 + 2MN$  e o número de restrições é  $3N + 2MN - M$ .

Laporte *et al.* (2004) mostraram que a relaxação linear do Modelo TD é zero, conforme Proposição 2.3.1 abaixo.

**Proposição 2.3.1.** (LAPORTE *et al.*, 2004) *A relaxação linear do Modelo TD é zero.*

*Demonstração.* Observe que tomando  $u_{jk} = \frac{1}{N}$ ,  $\forall j \in J, k \in K$ ,  $v_{kt} = \frac{|J_t|}{N}$ ,  $\forall k \in K, t \in T$  e  $w_{kt} = 0$ ,  $\forall k \in K, t \in T$ , obtém-se uma solução factível para a relaxação linear com  $Z_{TD} = 0$ .

Como  $0 \leq w_{kt} \leq 1$ , segue que  $Z_{TD} \geq 0$ .

Portanto, como o problema TD é de minimização, o resultado segue.  $\square$

### 2.3.2 Modelo de Bard - B

Seja  $x_{ik}$  uma variável de decisão igual a 1 se, e somente se, a tarefa  $i \in J$  é processada na posição  $k \in K$ . Seja  $y_{tk}$  uma variável de decisão igual a 1 se, e somente se, a ferramenta  $t \in T$  está na máquina durante o processamento de alguma tarefa na posição  $k \in K$ . Bard (1988) propôs o seguinte modelo B:

$$(B) \quad \text{Min } Z_B = \sum_{t=1}^M \sum_{k=1}^N y_{tk}(1 - y_{t(k-1)}) \quad (2.12)$$

sujeito a:

$$\sum_{k=1}^N x_{ik} = 1, \quad i \in J, \quad (2.13)$$

$$\sum_{i=1}^N x_{ik} = 1, \quad k \in K, \quad (2.14)$$

$$\sum_{t=1}^M y_{tk} \leq C, \quad k \in K, \quad (2.15)$$

$$A_{tj}x_{jk} \leq y_{tk} \quad j \in J, k \in K, t \in T, \quad (2.16)$$

$$x_{ik}, y_{tk}, \in \{0, 1\}, \quad k \in K, i \in J, t \in T. \quad (2.17)$$

em que  $y_{t0}$  são as condições iniciais dadas e será admitido que  $y_{t0} = 1$ , para todo  $t \in T$ . A função objetivo (2.12) minimiza o número total de trocas de ferramentas. As restrições (2.13) e (2.14) garantem que cada tarefa é designada a exatamente uma posição na sequência de processamento. As restrições (2.15) asseguram que não terá mais que  $C$  ferramentas no magazine em qualquer posição de processamento. As restrições (2.16) garantem que se a tarefa  $j$  requer a ferramenta  $t$  e é processada na posição  $k$ , então a ferramenta  $t$  deve estar na máquina no instante  $k$ . As restrições (2.17) são restrições de integralidade em todas as variáveis de decisão. Este modelo tem  $N^2 + MN$  variáveis e  $N^2M + 3N$  restrições.

A função objetivo do Modelo B simplesmente conta o número de trocas e pode ser linearizado de várias maneiras. Uma dessas maneiras é introduzir as variáveis  $z_{tk} \in \{0, 1\}$ , adicionar as restrições  $y_{tk} - y_{t(k-1)} \leq z_{tk}$ ,  $t \in T, k \in K$  e mudar a função objetivo (2.12) por  $\sum_{t=1}^M \sum_{k=1}^N z_{tk}$ . Este novo modelo (linearizado) tem  $N^2 + 2MN$  variáveis e  $2N^2M + 3N$

restrições.

O modelo linearizado é exatamente o mesmo modelo proposto por Tang e Denardo (1988a), em que as restrições (2.6) são as restrições (2.16) mais restritas.

### 2.3.3 Modelo de Laporte, Salazar-González e Semet - LSS

Para superar as limitações do modelo TD, Laporte *et al.* (2004) propuseram um modelo de otimização linear inteiro baseado no Problema do Caixeiro Viajante (PCV). Para isto, introduziram uma tarefa *dummy* 0, que não requer nenhuma ferramenta, representando o início e fim do processamento das tarefas. Defina  $J_0 = J \cup \{0\}$ , e  $T_i$  o conjunto das ferramentas utilizadas para processar a tarefa  $i \in J$ . Seja  $x_{ij}$  uma variável de decisão igual a 1 se a tarefa  $i$  é imediatamente seguida pela tarefa  $j$ ,  $i, j \in J_0$ ,  $i \neq j$ , e 0, caso contrário. Seja  $y_{it}$  a variável de decisão igual a 1 se a ferramenta  $t$  está na máquina enquanto a tarefa  $i$  está sendo processada,  $i \in J$ ,  $t \in T$ , e 0, caso contrário. Seja  $z_{it}$  uma variável de decisão igual a 1 se a ferramenta  $t$  é inserida na máquina no começo do processamento da tarefa  $i$ ,  $i \in J$ ,  $t \in T$ , e 0, caso contrário. O Modelo LSS é dado por:

$$\text{Min } Z_{LSS} = \sum_{i \in J} \sum_{t \in T_i} z_{it} \quad (2.18)$$

sujeito a:

$$\sum_{j \in J_0 \setminus \{i\}} x_{ij} = 1, \quad i \in J_0, \quad (2.19)$$

$$\sum_{i \in J_0 \setminus \{j\}} x_{ij} = 1, \quad j \in J_0, \quad (2.20)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad S \subset J_0, \quad 2 \leq |S| \leq N - 1, \quad (2.21)$$

$$\sum_{t \in T} y_{jt} \leq C, \quad j \in J, \quad (2.22)$$

$$x_{ij} + y_{jt} - y_{it} \leq z_{jt} + 1, \quad i \in J_0, \quad j \in J, \quad t \in T, \quad (2.23)$$

$$y_{it} = 1, \quad i \in J, \quad t \in T_i, \quad (2.24)$$

$$z_{it} = 0, \quad i \in J, \quad t \in T \setminus T_i, \quad (2.25)$$

$$x_{ij} \in \{0, 1\}, \quad i, j \in J_0, \quad (2.26)$$

$$y_{it} \in \{0, 1\}, \quad i \in J, \quad t \in T, \quad (2.27)$$

$$z_{it} \in \{0, 1\}, \quad i \in J, \quad t \in T. \quad (2.28)$$

Neste modelo, a função objetivo (2.18) minimiza a quantidade de trocas de ferramentas. As restrições (2.19) - (2.20) garantem que toda tarefa seja sequenciada em uma única posição. As restrições (2.21) são restrições de eliminação de subsequência propostas por Dantzig *et al.* (1954). As restrições (2.22) são as restrições de capacidade da máquina, enquanto as restrições (2.23) são restrições de ligação, que impedem que uma tarefa  $j$  seja processada depois da tarefa  $i$ , e requer a ferramenta  $t$ , ao mesmo tempo, sendo que a ferramenta  $t$  não está na máquina para processar a tarefa  $i$  e nem foi inserida para processar a tarefa  $j$ . As restrições (2.24) garantem que as ferramentas serão utilizadas sempre que requeridas por uma tarefa. As restrições (2.25) asseguram que nenhuma ferramenta é inserida na máquina sem ser requerida. As restrições (2.26) - (2.28) são as restrições de domínio das variáveis.

Segundo Laporte *et al.* (2004) o Modelo LSS domina o Modelo TD, já que a relaxação linear de TD é sempre zero e a relaxação linear de LSS pode ser positiva, como mostrado por um exemplo em seu artigo. Este exemplo é o mesmo exibido no Exemplo 2.1.1. A relaxação linear do modelo TD é nula, enquanto o valor da relaxação linear do modelo LSS é igual a 2.13.

Burger *et al.* (2015) aplicaram os dois Modelos TD e LSS em várias instâncias e concluíram que o Modelo TD tem um desempenho melhor para instâncias pequenas, enquanto o Modelo LSS é superior para instâncias de grande porte.

Nos testes computacionais desenvolvidos nesta tese relativos ao Modelo LSS, as restrições de eliminação de subsequências (2.21) que são de ordem exponencial, foram substituídas por restrições de ordem polinomiais. Várias restrições poderiam ser utilizadas, como as propostas por Desrochers e Laporte (1991), Miller *et al.* (1960), Sherali e Driscoll (2002) e Wong (1980). Outras restrições podem ser encontradas em Öncan *et al.* (2009). Testes computacionais preliminares realizados no modelo LSS, nesta tese, mostraram que as restrições propostas por Desrochers e Laporte (1991) (restrições DL) tiveram os melhores resultados. As restrições DL são formuladas pelas restrições (2.29)-(2.32).

$$u_i - u_j + Nx_{ij} + (N - 2)x_{ji} \leq N - 1, \quad i, j = 1, \dots, N, i \neq j, \quad (2.29)$$

$$1 + (N - 2)x_{i0} + \sum_{j=1}^N x_{ji} \leq u_i, \quad i = 1, \dots, N, \quad (2.30)$$

$$u_i \leq N - (N - 2)x_{0i} - \sum_{j=1}^N x_{ij}, \quad i = 1, \dots, N, \quad (2.31)$$

$$u_i \in \mathbb{Z}, \quad i = 1, \dots, N. \quad (2.32)$$

O número de variáveis do modelo LSS é  $(N + 1)^2 + N(2M + 1)$  enquanto o número de

restrições é  $N^2(M + 1) + N(2M + 5) + 2$  (considerando o modelo LSS com as restrições DL).

### 2.3.4 Modelo de Catanzaro, Gouveia e Labbé - CGL

Segundo Catanzaro *et al.* (2015), o motivo de poucos autores abordarem métodos exatos para o PMTF, se deve ao fato dos limitantes inferiores obtidos pelas formulações propostas na literatura serem fracos. Laporte *et al.* (2004) mostraram que a relaxação linear do modelo LSS domina a relaxação linear do modelo TD proposto por Tang e Denardo (1988a). Diante disto, os autores propõem três novos modelos de otimização inteira a fim de fortalecer os limitantes inferiores da literatura. O modelo aqui apresentado é o que melhor forneceu limitantes inferiores para o PMTF, segundo os autores Catanzaro *et al.* (2015).

Os autores propuseram um modelo de otimização baseado em fluxo em arcos (“*arc-flow integer linear programming formulation*”) e utilizaram a noção de 0-bloco.

**Definição 2.3.1. (0-bloco)** *Seja  $P$  uma matriz de dimensão  $M \times N$ , cujas entradas  $p_{ij}$  recebe 1 se a ferramenta  $t$  é requerida pela tarefa  $j$ , e 0 caso contrário. Um 0-bloco de  $P$  é um conjunto  $\{(t, j), (t, j + 1), \dots, (t, j + k)\}$  tal que:*

- (1)  $1 < j \leq j + k < N$ ;
- (2)  $p_{tj} = p_{t,j+1} = \dots = p_{t,j+k} = 0$ ;
- (3)  $p_{t,j-1} = p_{t,j+k+1} = 1$ .

Pela definição, um 0-bloco representa um conjunto de zeros consecutivos em uma linha delimitado por 1's, ou ainda, uma sequência de tarefas em que as tarefas  $i$  e  $j$  necessitam da ferramenta  $t$ , mas as tarefas entre  $i$  e  $j$  não utilizam  $t$  em seu processamento. Um exemplo de 0-bloco é apresentado na Tabela 2.6, e corresponde ao conjunto  $\{(4, 2), (4, 3), (4, 4), (4, 5)\}$ .

Ferramentas	Tarefas					
	1	2	3	4	5	6
1	1	1	0	1	0	1
2	0	0	1	0	0	1
3	0	1	0	0	1	0
4	1	0	0	0	0	1
5	0	1	0	1	1	0
6	0	0	1	0	0	0
7	0	0	1	1	0	0
8	1	0	1	0	1	0
9	1	0	0	1	0	0

TABELA 2.6 – Exemplo de matriz P, com 0-bloco, associada ao Exemplo 2.1.

Os autores definem seu modelo de otimização minimizando a quantidade de 0-blocos em cada linha, minimizando assim, a quantidade total de trocas de ferramentas. Para tanto, seja  $G = (V, A)$  um grafo completo em que seus vértices são as tarefas e qualquer aresta  $(k, l)$  pertencente a  $A$  corresponde a sequenciar a tarefa  $l$  imediatamente após a tarefa  $k$ . Defina  $G_{ij}^t = (V_{ij}^t, A_{ij}^t)$  o subgrafo direcionado induzido de  $G$ , em que  $V_{ij}^t = J_t^c \cup \{i, j\}$ , com  $i \neq j$  e  $t \in T_i \cap T_j$ , e o conjunto  $A_{ij}^t$ , um subconjunto de  $A$ , definido pelos arcos formados pelos elementos de  $V_{ij}^t$  a menos do arco  $(i, j)$ ; dos arcos convergentes a  $j$  e dos arcos divergentes de  $i$ . Assim, um caminho em  $G_{ij}^t$  pode ser associado a um 0-bloco. Na Figura 2.1 o subgrafo  $G_{24}^5$  associado a instância do Exemplo 2.1 é ilustrado. O conjunto de vértices é definido por  $V_{24}^5 = J_5^c \cup \{2, 4\} = \{1, 2, 3, 4, 6\}$  e o conjunto de arestas  $A_{24}^5 = \{(1, 3), (1, 4), (1, 6), (2, 1), (2, 3), (2, 6), (3, 1), (3, 4), (3, 6), (6, 1), (6, 3), (6, 4)\}$ . Observe que qualquer caminho que inicie em 2 e termina em 4 determina um 0-bloco, como por exemplo o caminho  $2 - 3 - 6 - 4$ .

Sejam as variáveis de decisão  $x_{ij}$  que é igual a 1 se a tarefa  $j \in J_0$  segue imediatamente a tarefa  $i \in J_0$ , e 0, caso contrário. Seja  $f_{ij,kl}^t$  a variável de decisão igual a 1 se a tarefa  $l$  segue imediatamente a tarefa  $k$  em um caminho de  $G_{ij}^t$ , e 0, caso contrário. O modelo CGL é dado por:

$$\text{Min } Z_{CGL} = \sum_{j \in J} |T_j| x_{0j} + \sum_{\substack{i, j \in J: \\ i \neq j \\ t \in T_j \setminus T_i}} \left( x_{ij} - \sum_{\substack{k \in J: \\ k \neq j \\ t \in T_k}} f_{kj,ij}^t \right) \quad (2.33)$$

sujeito a:

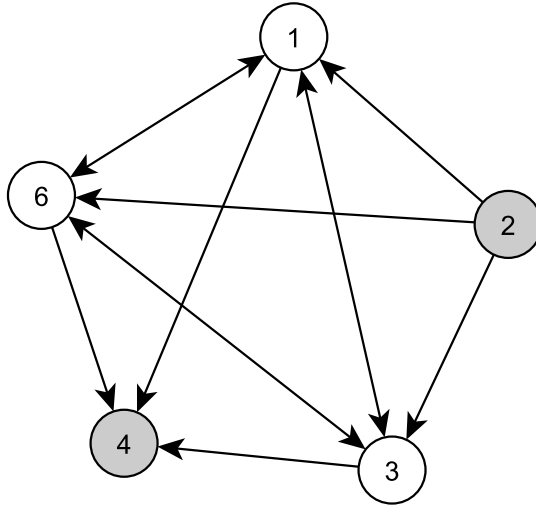


FIGURA 2.1 – Exemplo do subgrafo utilizado no Modelo CGL, considerando a ferramenta 5 e as tarefas 2 e 4.

$$\sum_{j \in J_0^i} x_{ij} = 1, \quad i \in J_0, \quad (2.34)$$

$$\sum_{i \in J_0^j} x_{ij} = 1, \quad j \in J_0, \quad (2.35)$$

$$\sum_{\substack{i,j \in S: \\ i \neq j}} x_{ij} \leq |S| - 1, \quad S \subset J_0, \quad 2 \leq |S| \leq N - 1, \quad (2.36)$$

$$\sum_{\substack{l \in V_{ij}^t: \\ l \neq k}} f_{ij,kl}^t - \sum_{\substack{l \in V_{ij}^t: \\ l \neq k}} f_{ij,lk}^t \geq 0, \quad i, j \in J : i \neq j,$$

$$t \in T_i \cap T_j, \quad k \in J_t^c, \quad (2.37)$$

$$\sum_{i \in J_t} f_{il,kl}^t \leq x_{kl}, \quad t \in T, \quad k \in J_t^c, \quad l \in J_t, \quad (2.38)$$

$$\sum_{j \in J_t} f_{kj,kl}^t \leq x_{kl}, \quad t \in T, \quad k \in J_t, \quad l \in J_t^c, \quad (2.39)$$

$$\sum_{\substack{i,j \in J_t: \\ i \neq j}} f_{ij,kl}^t \leq x_{kl}, \quad t \in T, \quad k, l \in J_t^c : k \neq l, \quad (2.40)$$

$$\sum_{\substack{t \in T: \\ t \in T_k \\ t \notin T_l}} \sum_{\substack{j \in J_t: \\ j \neq k}} f_{kj,kl}^t + \sum_{\substack{t \in T: \\ t \notin T_k \\ t \notin T_l}} \sum_{\substack{i, j \in J_t: \\ i \neq j}} f_{ij,kl}^t \leq (C - |T_l|)x_{kl}, \quad k, l \in J : k \neq l, \quad (2.41)$$

$$x_{ij} \in \{0, 1\}, \quad i, j \in J_0 : i \neq j, \quad (2.42)$$

$$f_{ij,kl}^t \geq 0, \quad t \in T, \quad i, j \in J : i \neq j, \\ k, l \in V_{ij}^t : k \neq l, \\ \{k, l\} \neq \{i, j\}. \quad (2.43)$$



A função objetivo (2.33) minimiza a quantidade de trocas de ferramentas. As restrições (2.34) - (2.36) são equivalentes às restrições (2.19) - (2.21) do modelo LSS; as restrições (2.37) permitem um caminho (0-bloco) em  $V_{ij}^t$  se tal caminho for escolhido como solução ótima do problema; as restrições (2.38) - (2.40) asseguram que a ferramenta  $t$  permaneça na máquina quando as tarefas  $k$  e  $l$  forem processadas em sequência e pertencerem a um 0-bloco. As restrições (2.41) correspondem às restrições de capacidade, e as restrições (2.42) e (2.43) são restrições de domínio das variáveis.

Este modelo tem  $N(2N + 1) + N(N - 1) \sum_{t \in T_i \cap T_j} (|J_t^c| + 2)$  variáveis enquanto o número de restrições é  $N^2(M + 2) + 5N - \sum_{t=1}^M (|J_t^c| + |J_t|^2) + \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N \sum_{t \in T_i \cap T_j} |J_t^c| + 2$ .

Os autores mostraram que seu modelo obtêm limitantes inferiores melhores que os da literatura. Entretanto, devido ao grande número de variáveis e restrições, seu modelo apresentou gaps muito altos quando executado com um tempo limite de 3 horas.

## 2.4 Considerações finais

Neste Capítulo, apresentou-se formalmente o PMTF e uma detalhada revisão da literatura foi realizada, abordando tanto modelos de otimização quanto métodos de solução exatos e heurísticos. Na Tabela 2.7 apresentam-se as quantidades de restrições e variáveis binárias para cada um dos modelos TD, LSS, e CGL.

	Restrições	Variáveis
TD	$3N + 2MN - M$	$N^2 + 2MN$
LSS	$N^2(M + 1) + N(2M + 5) + 2$	$(N + 1)^2 + N(2M + 1)$
CGL	$N^2(M + 2) + 5N - \sum_{t=1}^M ( J_t^c  +  J_t ^2) +$ $+ \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N \sum_{t \in T_i \cap T_j}  J_t^c  + 2$	$N(2N + 1) + N(N - 1) \sum_{t \in T_i \cap T_j} ( J_t^c  + 2)$

TABELA 2.7 – Quadro resumo dos modelos de TD, LSS e CGL.

O Modelo TD, apesar de ter poucas restrições e poucas variáveis, tem valor de relaxação linear nula, o que dificulta a resolução de métodos de enumeração implícita. Já o Modelo LSS, baseado no modelo do PCV, domina a relaxação linear do modelo LSS, uma vez que pode obter limitantes inferiores maiores que zero. Por fim, o Modelo CGL, apesar de teoricamente sempre obter valores de relaxação linear melhores ou iguais aos obtidos pelo Modelo LSS, apresenta uma grande quantidade de restrições e variáveis.

Os modelos de fluxo *multicommodity* desenvolvidos nesta tese, tem poucas restrições

e poucas variáveis, e o valor de relaxação linear é sempre igual a  $(M - C)$ , independente da instância a ser resolvida. Estes fatos, contribuíram para uma melhor performance destes modelos desenvolvidos frente aos modelos TD, LSS e CGL da literatura, tanto em quantidade de instâncias resolvidas na otimalidade quanto em tempo de execução, conforme pode ser visto da Seção 3.3.

# 3 Modelos de fluxo multicommodity propostos para o PMTF

Neste capítulo, os modelos desenvolvidos nesta tese para o PMTF, baseados em fluxo *multicommodity*, são apresentados. Também faz-se algumas considerações teóricas sobre os mesmos, analisa-se algumas relaxações de conjuntos de variáveis, e demonstra-se que o valor da relaxação linear de todos os modelos desenvolvidos é igual a diferença entre a quantidade de ferramentas e a capacidade da máquina.

Uma propriedade intrínseca aos problemas PMTF diz respeito à desigualdade triangular, ou seja, dadas quaisquer três tarefas  $i_1$ ,  $i_2$  e  $i_3$ , o número de trocas de ferramentas de  $i_1$  para  $i_3$  é sempre menor ou igual ao número de trocas de  $i_1$  para  $i_2$  somado ao número de trocas de  $i_2$  para  $i_3$ , conforme a Proposição 3.0.1.

**Proposição 3.0.1** (Desigualdade Triangular (YANASSE *et al.*, 2009)). *Para quaisquer instância e solução factível do PMTF, vale a desigualdade triangular em relação à troca de ferramentas.*

Outra propriedade do PMTF diz respeito ao seu limitante inferior natural cujo valor é  $M - C$ , conforme Proposição 3.0.2.

**Proposição 3.0.2** (Limitante inferior natural (LAPORTE *et al.*, 2004)). *O limitante inferior natural para o PMTF é  $M - C$ .*

*Demonstração.* De fato, sabe-se que há  $M$  ferramentas e que supostamente serão todas utilizadas no processamento das tarefas. Tem-se ainda que  $C$  ferramentas iniciam o processo de produção. Logo,  $M - C$  ferramentas ficarão de fora da máquina, no início. Como todas elas devem entrar no processo de produção em algum momento, pelo menos  $M - C$  ferramentas serão trocadas por estas que inicialmente ficaram de fora. Logo, haverá no mínimo,  $M - C$  trocas.  $\square$

As proposições 3.0.3 e 3.0.4 foram desenvolvidas nesta tese, e diz respeito a uma tarefa que não compartilha ferramentas com outras tarefas, e tarefas que podem ser substituídas

por uma tarefa equivalente, respectivamente.

**Proposição 3.0.3** (Tarefa isolada). *Se há uma tarefa  $i$  tal que  $T_i \cap T_j = \emptyset$ , para todo  $j = 1, 2, \dots, N$ ,  $j \neq i$ , então esta tarefa só pode ser processada na primeira ou última posição de uma solução.*

*Demonstração.* Considere  $s$  a sequência de processamento das tarefas desconsiderando a tarefa  $i$ . Como  $T_i \cap T_j = \emptyset$ , para todo  $j = 1, 2, \dots, N$ ,  $j \neq i$ , ao acrescentar  $i$  à qualquer posição de  $s$ , o valor objetivo de  $s$  seria acrescido em  $2|T_i|$ , a menos da primeira e última posições que seria acrescido em  $|T_i|$ . Logo, esta tarefa  $i$  só pode ser sequenciada na primeira ou última posição. □

**Proposição 3.0.4** (Pré-processamento). *Se existem  $p$  tarefas tais que  $|\bigcup_{j=1}^p T_j| \leq C$ , então não há trocas de ferramentas entre elas, e podem ser processados sequencialmente.*

*Demonstração.* Basta considerar uma tarefa fictícia  $i_p$  tal que  $T_{i_p} = |\bigcup_{j=1}^p T_j|$ . □

O PMTF tem propriedade de simetria, ou seja, toda sequência e sua sequência reversa tem a mesma quantidade de trocas de ferramentas (GHIANI *et al.*, 2007).

## 3.1 Modelos de fluxo *multicommodity* para o PMTF

### 3.1.1 Modelo FM-PMTF-1

O primeiro modelo de fluxo desenvolvido, denominado FM-PMTF-1 é apresentado a seguir. Considere um grafo  $G(V, A)$  em que o conjunto  $V$  é composto por  $N + 2$  nós, numerados de 0 a  $N + 1$ , em que o nó 0 é o nó origem e o nó  $N$  é o nó destino ( $V = \{0, 1, \dots, N + 1\}$ ). O conjunto de arcos  $A$  deste grafo é composto pelos arcos da forma  $(i, i + 1)$  para  $i = 0, 1, \dots, N - 1$ ;  $(i, N + 1)$  para  $i = 0, 1, \dots, N - 1$  e  $(N + 1, i)$  para  $i = 1, 2, \dots, N$ . Há um limite superior  $C$  no fluxo dos arcos  $(i, i + 1)$  para  $i = 0, 1, \dots, N - 1$ . O custo unitário de transportar qualquer *commodity* nos arcos  $(i, N + 1)$  para  $i = 1, 2, \dots, N - 1$  é 1, e o custo em todos os outros arcos é nulo. Uma ilustração do grafo  $G(V, A)$  é apresentado na Figura 3.1.

Considere um problema de fluxo *multicommodity* em  $G(V, A)$  em que na origem há  $M$  tipos de *commodities* a serem transportadas a um custo mínimo até o destino, sendo que há apenas uma unidade de cada *commodity*. Admita que o fluxo de cada *commodity* deva ser inteiro, ou seja, há interesse somente nas soluções em que cada *commodity* usa um caminho simples para transportar uma unidade do nó 0 para o nó  $N$ .

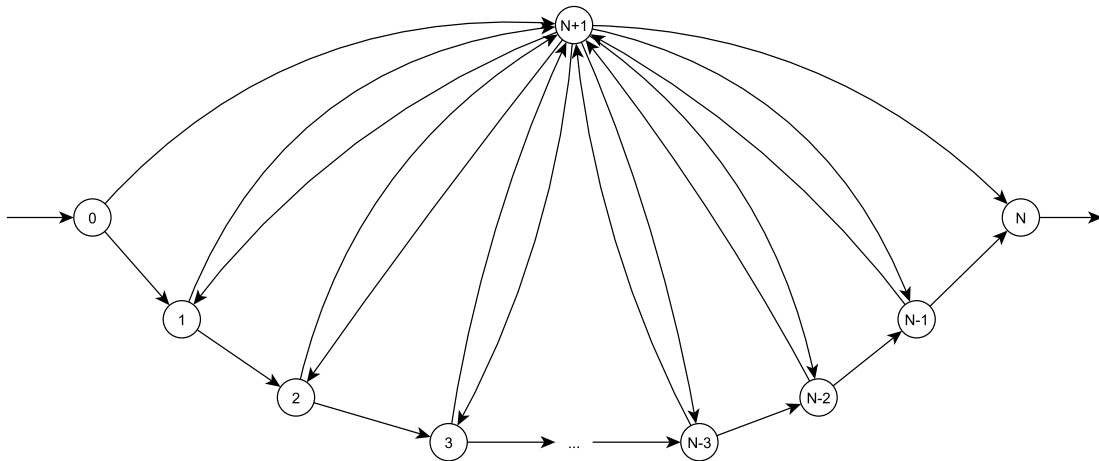


FIGURA 3.1 – Grafo do Modelo FM-PMTF-1.

Cada *commodity* pode ser interpretada como uma ferramenta. Assim, pode-se associar uma solução de fluxo *multicommodity* atravessando os arcos  $(i, i + 1)$ , para  $i = 0, 1, \dots, N - 1$ , como as ferramentas que estarão na máquina num determinado período de processamento de uma tarefa.

Admite-se, sem perda de generalidade, que a máquina sempre estará em sua capacidade completa, ou seja, sempre haverá  $C$  ferramentas nela, pois não há custos para ferramentas ociosas. Com isto, pode-se impor um limitante inferior de  $C$  nos arcos  $(i, i + 1)$  para  $i = 0, 1, \dots, N - 1$ .

Uma solução do PMTF pode ser obtida determinando-se o fluxo *multicommodity* de custo mínimo no grafo  $G(V, A)$ . Observe que cada nó  $(1, 2, \dots, N)$  representa a posição (ordem) de processamento de uma tarefa em uma determinada sequência, e que uma unidade de fluxo da *commodity*  $t$  no arco  $(i, i + 1)$  representa a ferramenta  $t$  que será utilizada, se necessária, no processamento da tarefa que foi sequenciada na posição  $i + 1$ . De maneira semelhante, uma unidade de fluxo da *commodity*  $t$  no arco  $(i, N + 1)$  significa que esta ferramenta deixou a máquina depois do processamento da tarefa na posição  $i$ , e se ela atravessar o arco  $(N + 1, i)$ , ela entrará na máquina antes do processamento da tarefa na posição  $i + 1$ . Um exemplo de uma solução neste grafo é apresentado no Exemplo 3.1.1.

**Exemplo 3.1.1.** *O grafo da Figura 3.2 representa a solução ótima da instância apresentada no Exemplo 2.1.1. As ferramentas em vermelho configuram as trocas de ferramentas. As ferramentas em qualquer arco  $(i, i + 1)$ ,  $i = 0, \dots, 5$ , são as ferramentas que estão na máquina no momento do processamento da tarefa da posição  $i + 1$ . Por exemplo, as tarefas 8, 5, 7 e 3, no arco  $(2, 3)$  são as ferramentas que estão na máquina para processar a tarefa 5.*

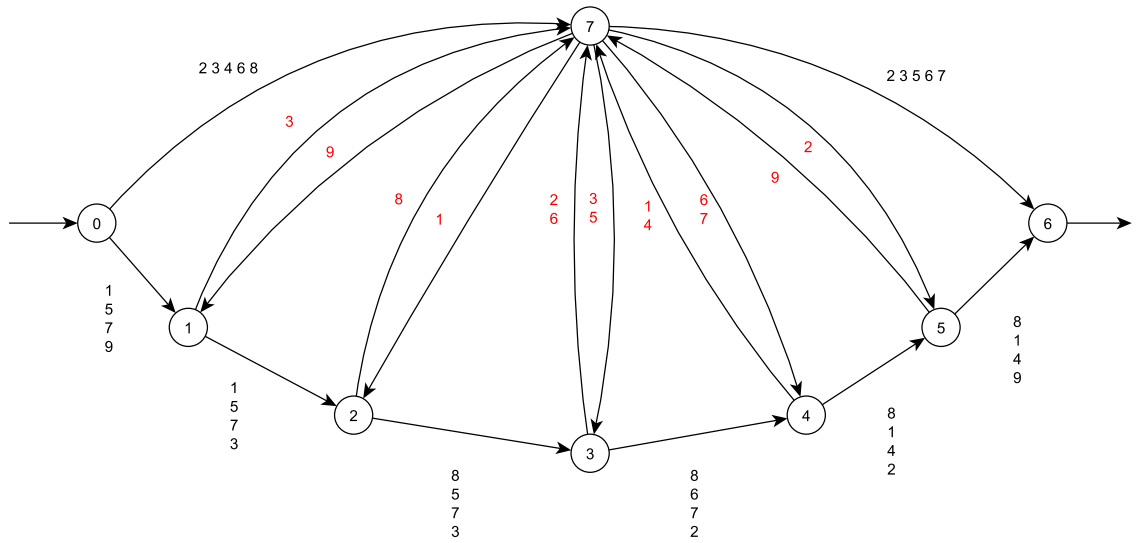


FIGURA 3.2 – Exemplo de uma solução (ótima) da instância do Exemplo 2.1.1 no grafo da Figura 3.1.

Formula-se o PMTF como um problema de fluxo *multicommodity*. Para tanto, considere  $x_{ik}$  uma variável de decisão binária igual a 1 se a tarefa  $i \in J$  é a  $k$ -ésima tarefa sequenciada,  $k \in J$ , e 0, caso contrário. Seja  $y_{ikt}$  uma variável de decisão binária igual a 1, se há fluxo da *commodity*  $t \in T$  no arco  $(i, k) \in A$ , e 0, caso contrário. Admite-se que  $|T_i| \leq C$ ,  $i = 1, 2, \dots, N$ .

Um primeiro modelo de otimização para o PMTF denominado Modelo FM-PMTF-1 (problema de minimização de trocas de ferramentas modelado como fluxo *multicommodity*) é dado pelas expressões (3.1) - (3.15).

$$\text{Min } Z_0 = \sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(N+1)t} \tag{3.1}$$

sujeito a:

$$\sum_{k=1}^N x_{ik} = 1, \quad i \in J, \tag{3.2}$$

$$\sum_{i=1}^N x_{ik} = 1, \quad k \in J, \tag{3.3}$$

$$y_{0it} + y_{0(N+1)t} = 1, \quad t \in T, \tag{3.4}$$

$$y_{(i-1)it} + y_{(N+1)it} - y_{i(N+1)t} - y_{i(i+1)t} = 0, \quad t \in T, i \in J \setminus \{N\}, \quad (3.5)$$

$$y_{(N-1)Nt} + y_{(N+1)Nt} = 1, \quad t \in T, \quad (3.6)$$

$$\sum_{i=0}^{N-1} (y_{i(N+1)t} - y_{(N+1)(i+1)t}) = 0, \quad t \in T, \quad (3.7)$$

$$y_{i(N+1)t} + y_{(N+1)it} \leq 1, \quad i \in J \setminus \{N\}, t \in T, \quad (3.8)$$

$$x_{ik} \leq y_{(k-1)kt}, \quad i \in J, k \in J, t \in T_i \quad (3.9)$$

$$\sum_{t=1}^M y_{(k-1)kt} = C, \quad k \in J, \quad (3.10)$$

$$y_{0(N+1)t} \leq \sum_{i=1}^{N-1} y_{(N+1)it}, \quad t \in T, \quad (3.11)$$

$$x_{ik} \in \{0, 1\}, \quad i, k \in J, \quad (3.12)$$

$$y_{i(i+1)t} \in \{0, 1\}, \quad i = 0, \dots, N-1, t \in T, \quad (3.13)$$

$$y_{i(N+1)t} \in \{0, 1\}, \quad i = 0, \dots, N-1, t \in T, \quad (3.14)$$

$$y_{(N+1)it} \in \{0, 1\}, \quad i \in J, t \in T. \quad (3.15)$$

A função objetivo (3.1) minimiza o custo dos fluxos nas arestas  $(i, N + 1)$ ,  $i = 1, 2, \dots, N - 1$ . Na solução ótima, os fluxos unitários nestes arcos correspondem às ferramentas que deixam a máquina, isto é, as ferramentas que são substituídas por outras. As restrições (3.2) e (3.3) impõem que todas as tarefas sejam sequenciadas em uma e apenas uma ordem. As restrições (3.4) conservam o fluxo no nó 0, as restrições (3.5) conservam o fluxo nos nós  $1, 2, \dots, N - 1$ , as restrições (3.6) conservam o fluxo no nó  $N$ , e as restrições (3.7) conservam o fluxo no nó  $N + 1$ . As restrições (3.8) asseguram que uma ferramenta  $t$  não entra e sai da máquina (no mesmo nó) no momento de processar alguma tarefa na posição  $i$ . As restrições (3.9) impõem que se a tarefa  $i$  é a  $k$ -ésima tarefa a ser processada então todas as ferramentas necessárias para processar a tarefa  $i$  devem estar na máquina, ou seja, atravessam o arco  $(k - 1, k)$ . As restrições (3.10) exigem que sempre haja  $C$  ferramentas na máquina no processamento de qualquer tarefa. As restrições (3.11) garantem que toda ferramenta que não inicia na máquina, deve retornar à ela em algum momento para processar a(s) tarefa(s) que necessitam dela. Esta restrição é redundante mas fortalece a relaxação linear do modelo. Por fim, as restrições (3.12) - (3.15) são restrições de integralidade de todas as variáveis de decisão.

O número total de restrições no Modelo FM-PMTF-1 é  $3N + 2M + 2MN + N^2 \sum_i |T_i|$  e o número total de variáveis binárias é  $N^2 + 3MN$ .

**Teorema 3.1.1.** *O Modelo FM-PMTF-1 não admite subsequências.*

*Demonstração.* Suponha que exista uma solução factível do modelo FM-PMTF-1 com subsequência.

Dada uma ferramenta  $t$ , a única possibilidade de subsequência é da forma

$$y_{(N+1)it} = y_{i(i+1)t} = y_{(i+1)(i+2)t} = \dots = y_{(i+k-1)(i+k)t} = y_{(i+k)(N+1)t} = 1, \quad (3.16)$$

para algum  $k \in \mathbb{N}$ , algum  $i \in \{1, \dots, N-1\}$ , e

$$y_{(i-1)it} = 0, \quad (3.17)$$

$$y_{j(N+1)t} = 0, \quad j = 0, \dots, i-1, \quad (3.18)$$

$$y_{(N+1)jt} = 0, \quad j = 1, \dots, i-1, \quad (3.19)$$

$$y_{(N+1)jt} = 0, \quad j = i+k+1, \dots, N, \quad (3.20)$$

$$y_{j(N+1)t} = 0, \quad j = i+k+1, \dots, N-1. \quad (3.21)$$

Como, por (3.18),  $y_{0(N+1)t} = 0$ , então, por (3.4), segue que  $y_{01t} = 1$ .

Agora, por (3.18), (3.19) e  $y_{01t} = 1$ , tem-se que  $y_{j(j+1)t} = 1$ ,  $j = 1, \dots, i-1$ .

Resumindo, tem-se

$$y_{(i-1)it} = y_{i(i+1)t} = 1, \quad (3.22)$$

$$y_{(N+1)it} = y_{(i+1)(N+1)t} = 1. \quad (3.23)$$

Agora, considerando (3.5), tem-se que  $y_{i(N+1)t} = 1$ , o que é um absurdo pois  $y_{(N+1)it} = 1$  e (3.8).

Logo, não há subsequências no Modelo FM-PMTF-1. □

Visando responder a segunda questão de pesquisa, tem-se o Teorema 3.1.2 e a Conjectura 3.1.1.

Denominando por LP-1, o Modelo FM-PMTF-1 depois de relaxar as restrições de integralidade (3.14) e (3.15), pode-se derivar o seguinte resultado:

**Teorema 3.1.2.** *Existe uma solução ótima do Modelo LP-1 que é inteira.*



*Demonstração.* Seja  $(x^*, y^*)$  a solução ótima do Modelo LP-1 e  $v^*$  o correspondente valor ótimo da função objetivo. Considere qualquer  $i$ ,  $i = 0, 1, \dots, N - 1$ . Pelas restrições (3.13), para cada  $i$  considerado, e para cada  $t$ ,  $t = 1, 2, \dots, M$ , uma e apenas uma das quatro possibilidades pode acontecer:

1.  $y_{(i-1)it}^* = y_{i(i+1)t}^* = 0$ ;
2.  $y_{(i-1)it}^* = y_{i(i+1)t}^* = 1$ ;
3.  $y_{(i-1)it}^* = 1$  e  $y_{i(i+1)t}^* = 0$ ;
4.  $y_{(i-1)it}^* = 0$  e  $y_{i(i+1)t}^* = 1$ .

Caso (1). Se  $y_{(i-1)it}^* = y_{i(i+1)t}^* = 0$ , deseja-se mostrar que  $y_{i(N+1)t}^* = y_{(N+1)it}^* = 0$ .

Suponha, sem perda de generalidade que

$$y_{(N+1)it}^* \neq 0.$$

Assim, pelas restrições (3.5), tem-se que

$$y_{i(N+1)t}^* = y_{(N+1)it}^* \neq 0.$$

Como  $y_{i(N+1)t}^* = y_{(N+1)it}^*$ , construa uma outra solução  $(x^*, z)$  factível para o Modelo LP-1 tomando  $z = y^*$  sendo que  $z_{i(N+1)t} = z_{(N+1)it} = 0$ . Uma vez que o vetor  $y^*$  é uma solução factível (no caso, ótima), e sendo  $z_{i(N+1)t} = z_{(N+1)it}$ , o vetor  $z$  também é solução pois satisfaz as restrições do modelo. Entretanto, o valor da função objetivo correspondente para  $z$  é  $v^* - y_{i(N+1)t}^* < v^*$ , o que contradiz a hipótese de que  $(x^*, y^*)$  é solução ótima.

Portanto,  $y_{i(N+1)t}^* = y_{(N+1)it}^* = 0$ .

Caso (2). Se  $y_{(i-1)it}^* = y_{i(i+1)t}^* = 1$ , deseja-se mostrar que  $y_{i(N+1)t}^* = y_{(N+1)it}^* = 0$ .

A demonstração é análoga ao Caso (1).

Caso (3). Se  $y_{(i-1)it}^* = 1$  e  $y_{i(i+1)t}^* = 0$ , quer-se mostrar que  $y_{i(N+1)t}^* = 1$  e  $y_{(N+1)it}^* = 0$ .

Utilizando as restrições (3.5) e o fato de que  $y_{(i-1)it}^* = 1$  e  $y_{i(i+1)t}^* = 0$ , tem-se que

$$1 + y_{(N+1)it}^* - y_{i(N+1)t}^* = 0,$$

ou ainda,

$$y_{i(N+1)t}^* = 1 + y_{(N+1)it}^*,$$

de onde conclui-se que  $y_{i(N+1)t}^* = 1$  e  $y_{(N+1)it}^* = 0$ , uma vez que  $0 \leq y_{(N+1)it} \leq 1$  e  $0 \leq y_{i(N+1)t} \leq 1$ .

Caso (4). Se  $y_{(i-1)it}^* = 0$  e  $y_{i(i+1)t}^* = 1$ , quer-se mostrar que  $y_{i(N+1)t}^* = 0$  e  $y_{(N+1)it}^* = 1$ .

A demonstração é análoga ao Caso (3).

Pelos casos acima, pode-se concluir que  $y_{i(N+1)t}^*$  e  $y_{(N+1)it}^*$  são valores inteiros, para todos  $i$  e  $t$ , e portanto, segue que a solução ótima  $(x^*, y^*)$  também é inteira com valor objetivo  $v^*$  inteiro. □

O número total de restrições do Modelo LP-1 é o mesmo que o Modelo FM-PMTF-1, mas a quantidade de variáveis binárias se reduz a  $N^2 + MN$  e agora há  $2MN$  variáveis contínuas.

Denominando por Modelo LP-2, o Modelo LP-1 depois de relaxar as restrições de integralidade (3.13), conjectura-se o seguinte resultado:

**Conjectura 3.1.1.** *Existe uma solução ótima do Modelo LP-2 que é inteira.*

### 3.1.2 Modelo FM-PMTF-1A

A primeira variação do Modelo FM-PMTF-1, denominada Modelo FM-PMTF-1A, consiste em especificar as ferramentas que saem da máquina. Esta variação poderia ser utilizada num contexto múltiplas máquinas também. Estas ferramentas podem sair da máquina e retornar posteriormente para continuar o processamento de outras tarefas, ou então sair definitivamente da máquina. As ferramentas que voltam à máquina correspondem ao fluxo das *commodities* nos arcos  $(i, N + 2)$ ,  $i = 1, \dots, N - 2$ , e as ferramentas que saem definitivo correspondem ao fluxo nos arcos  $(i, N + 1)$ ,  $i \in J \setminus \{N\}$ . O grafo resultante é apresentado na Figura 3.3 e a formulação é dada pelas expressões (3.24)-(3.40).

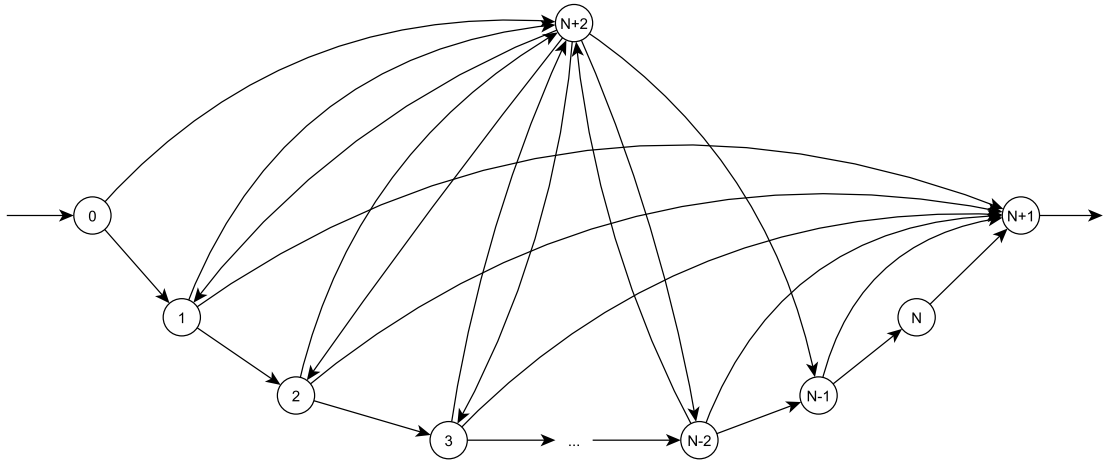


FIGURA 3.3 – Grafo do Modelo FM-PMTF-1A.

O Modelo FM-PMTF-1A é dado por:

$$\text{Min } Z_1 = \sum_{t=1}^M \sum_{i=1}^{N-2} y_{i(N+2)t} + \sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(N+1)t} \quad (3.24)$$

sujeito a:

$$\sum_{k=1}^N x_{ik} = 1, \quad i \in J, \quad (3.25)$$

$$\sum_{i=1}^N x_{ik} = 1, \quad k \in J, \quad (3.26)$$

$$y_{i(N+1)t} \leq 1 - y_{j(N+2)t}, \quad \begin{aligned} i &= 1, \dots, N-2, \\ j &= i, \dots, N-2, t \in T, \end{aligned} \quad (3.27)$$

$$x_{ik} \leq y_{(k-1)kt}, \quad i \in J, k \in J, t \in T_i, \quad (3.28)$$

$$y_{01t} + y_{0(N+2)t} = 1, \quad t \in T, \quad (3.29)$$

$$y_{(i-1)it} + y_{(N+2)it} - y_{i(N+2)t} - y_{i(N+1)t} - y_{i(i+1)t} = 0, \quad t \in T, i = 1, \dots, N-2, \quad (3.30)$$

$$y_{(N-1)Nt} - y_{N(N+1)t} = 0, \quad t \in T, \quad (3.31)$$

$$\sum_{i=1}^N y_{i(N+1)t} = 1, \quad t \in T, \quad (3.32)$$

$$y_{(N-2)(N-1)t} + y_{(N+2)(N-1)t} - y_{(N-1)(N+1)t} - y_{(N-1)Nt} = 0, \quad t \in T, \quad (3.33)$$

$$\sum_{i=0}^{N-2} y_{i(N+2)t} - \sum_{i=1}^{N-1} y_{(N+2)it} = 0, \quad t \in T, \quad (3.34)$$

$$\sum_{t=1}^M y_{(k-1)kt} = C, \quad k \in J, \quad (3.35)$$

$$x_{ik} \in \{0, 1\}, \quad i, k = 1, \dots, N, \quad (3.36)$$

$$y_{i(i+1)t} \in \{0, 1\}, \quad i = 0, \dots, N, \quad t \in T, \quad (3.37)$$

$$y_{i(N+1)t} \in \{0, 1\}, \quad i \in J \setminus \{N\}, \quad t \in T, \quad (3.38)$$

$$y_{i(N+2)t} \in \{0, 1\}, \quad i = 0, \dots, N-2, \quad t \in T, \quad (3.39)$$

$$y_{(N+2)it} \in \{0, 1\}, \quad i \in J \setminus \{N\}, \quad t \in T. \quad (3.40)$$

A função objetivo (3.24) minimiza a quantidade de trocas de ferramentas. As restrições (3.25) e (3.26) garantem que toda tarefa seja sequenciada em uma única posição. As restrições (3.27) são restrições de eliminação de subsequências relativas ao fluxo das ferramentas. As restrições (3.28) impõem que se a tarefa  $i$  é a  $k$ -ésima tarefa a ser processada então todas as ferramentas necessárias para processá-la devem estar na máquina. As restrições (3.29) - (3.34) são restrições de conservação de fluxo. As restrições (3.35) são restrições de capacidade, e as restrições (3.36) - (3.40) são domínio das variáveis.

A quantidade de restrições do Modelo FM-PMTF-1A é  $3N + 4M - \frac{MN}{2} + N^2(\sum_i |T_i| + \frac{M}{2})$  e a quantidade de variáveis binárias é dada por  $N^2 + 4MN - 2M$ .

### 3.1.3 Modelo FM-PMTF-1B

O terceiro modelo, denominado FM-PMTF-1B, foi desenvolvido com o intuito de além de especificar o destino das ferramentas que deixam a máquina, levar ao nó  $N + 2$  apenas as ferramentas que realmente deixam a máquina e retornam em algum momento. Assim, as ferramentas saem do nó 0 e já são encaminhadas à algum nó, conforme requisitadas durante o sequenciamento. De modo geral, ao invés de ter fluxo na aresta  $(0, N + 2)$ , haverá fluxo nas arestas  $(0, i)$ ,  $i = 2, \dots, N - 1$ . O grafo do Modelo FM-PMTF-1B é apresentado na Figura 3.4 e sua formulação é dada pelas expressões (3.41)-(3.60).

$$\text{Min } Z_2 = \sum_{t=1}^M \sum_{i=1}^{N-2} y_{i(N+2)t} + \sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(N+1)t} \quad (3.41)$$

sujeito a:

$$\sum_{k=1}^N x_{ik} = 1, \quad i \in J, \quad (3.42)$$

$$\sum_{i=1}^N x_{ik} = 1, \quad k \in J, \quad (3.43)$$

$$\sum_{i=1}^{N-1} y_{0it} = 1, \quad t \in T, \quad (3.44)$$

$$y_{01t} - y_{1(N+2)t} - y_{1(N+1)t} - y_{12t} = 0, \quad t \in T, \quad (3.45)$$

$$y_{(i-1)it} + y_{0it} + y_{(N+2)it} - y_{i(N+2)t} - y_{i(N+1)t} - y_{i(i+1)t} = 0, \quad t \in T, \quad i = 2, \dots, N-2, \quad (3.46)$$

$$y_{(N-2)(N-1)t} + y_{0(N-1)t} + y_{(N+2)(N-1)t} - y_{(N-1)(N+1)t} - y_{(N-1)Nt} = 0, \quad t \in T, \quad (3.47)$$

$$y_{(N-1)Nt} - y_{N(N+1)t} = 0, \quad t \in T, \quad (3.48)$$

$$\sum_{i=1}^N y_{i(N+1)t} = 1, \quad t \in T, \quad (3.49)$$

$$\sum_{i=1}^{N-2} y_{i(N+2)t} - \sum_{i=2}^{N-1} y_{(N+2)it} = 0, \quad t \in T, \quad (3.50)$$

$$x_{ik} \leq y_{(k-1)kt}, \quad i \in J, \quad k \in J, \quad t \in T_i, \quad (3.51)$$

$$\sum_{t=1}^M y_{(k-1)kt} = C, \quad k = 2, \dots, N, \quad (3.52)$$

$$y_{i(N+1)t} \leq 1 - y_{j(N+2)t}, \quad i = 1, \dots, N - 2, \quad j = i, \dots, N - 2, \quad t \in T, \quad (3.53)$$

$$y_{0it} \leq 1 - y_{j(N+2)t}, \quad i = 2, \dots, N - 1, \quad j = 1, \dots, i - 1, \quad t \in T, \quad (3.54)$$

$$x_{ik} \in \{0, 1\}, \quad i, k = 1, \dots, N, \quad (3.55)$$

$$y_{i(i+1)t} \in \{0, 1\}, \quad i = 0, \dots, N, \quad t \in T, \quad (3.56)$$

$$y_{i(N+1)t} \in \{0, 1\}, \quad i \in J \setminus \{N\}, \quad t \in T, \quad (3.57)$$

$$y_{i(N+2)t} \in \{0, 1\}, \quad i = 1, \dots, N - 2, \quad t \in T, \quad (3.58)$$

$$y_{(N+2)it} \in \{0, 1\}, \quad i = 2, \dots, N - 1, \quad t \in T. \quad (3.59)$$

$$y_{0it} \in \{0, 1\}, \quad i = 2, \dots, N - 1. \quad (3.60)$$

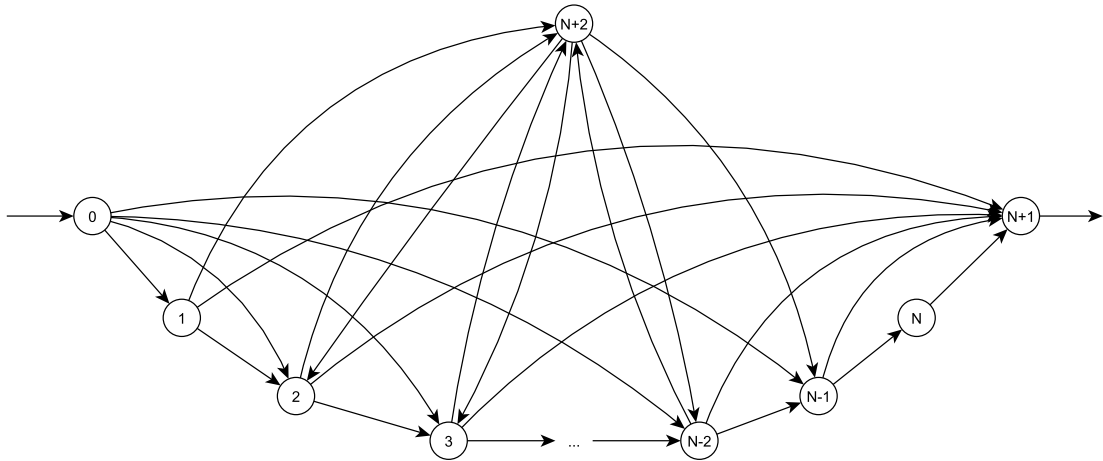


FIGURA 3.4 – Grafo do Modelo FM-PMTF-1B.

A função objetivo (3.41) minimiza a quantidade de trocas de ferramentas. As restrições (3.42) e (3.43) garantem que toda tarefa seja sequenciada em uma única posição. As restrições (3.44) - (3.50) são restrições de conservação de fluxo. As restrições (3.51) impõem que se a tarefa  $i$  é a  $k$ -ésima tarefa a ser processada então todas as ferramentas necessárias para processá-la devem estar na máquina. As restrições (3.52) são restrições de capacidade. As restrições (3.53) e (3.54) são restrições de eliminação de subsequências, e as restrições (3.55) - (3.60) são restrições de domínio das variáveis.

O Modelo FM-PMTF-1B tem  $3N + 5M - 2MN + N^2(\sum_i |T_i| + M) - 1$  restrições e apresenta  $N^2 + 4MN + 4M + N - 2$  variáveis binárias.

### 3.1.4 Modelo FM-PMTF-1C

O quarto modelo desenvolvido, denominado Modelo FM-PMTF-1C, difere do Modelo FM-PMTF-1B na questão de especificar quando uma ferramenta sai da máquina, mas

volta em algum momento. Cada uma destas ferramentas sai para um nó específico e pode retornar em qualquer outra posição subsequente em relação a que foi trocada. De modo geral, subdivide-se o nó  $(N + 2)$  em  $N - 2$  novos nós, conforme o grafo apresentado na Figura 3.5. Cada um destes novos nós representa as ferramentas que saíram em uma dada ordem, por exemplo, o nó  $(N + 3)$  corresponde às ferramentas que porventura saíram depois do processamento da tarefa na posição 2. Este modelo é dado pelas expressões (3.61)-(3.78).

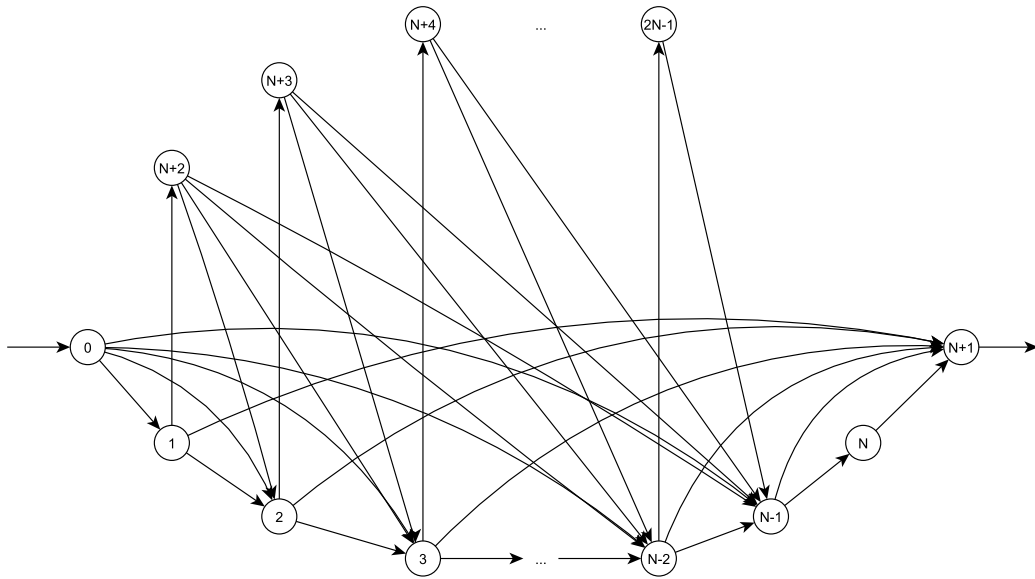


FIGURA 3.5 – Grafo do Modelo FM-PMTF-1C.

$$\text{Min } Z_3 = \sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(N+1)t} + \sum_{t=1}^M \sum_{i=1}^{N-2} y_{i(N+1+i)t} \quad (3.61)$$

sujeito a:

$$\sum_{k=1}^N x_{ik} = 1, \quad i \in J, \quad (3.62)$$

$$\sum_{i=1}^N x_{ik} = 1, \quad k \in J, \quad (3.63)$$

$$\sum_{i=1}^{N-1} y_{0it} = 1, \quad t \in T, \quad (3.64)$$

$$\sum_{i \in \{0, N-2\}} y_{i(N-1)t} + \sum_{j=N+2}^{2N-1} y_{j(N-1)t} - \sum_{i \in \{N, N+1\}} y_{(N-1)it} = 0, \quad t \in T, \quad (3.65)$$

$$y_{(i-1)it} + y_{0it} + \sum_{j=N+2}^{N+i} y_{jit} - y_{i(N+1+i)t} - y_{i(N+1)t} - y_{i(i+1)t} = 0, \quad t \in T, \quad i = 2, \dots, N-2, \quad (3.66)$$

$$y_{01t} - y_{1(N+2)t} - y_{1(N+1)t} - y_{12t} = 0, \quad t \in T, \quad (3.67)$$

$$y_{(N-1)Nt} - y_{N(N+1)t} = 0, \quad t \in T, \quad (3.68)$$

$$\sum_{i=1}^N y_{i(N+1)t} = 1, \quad t \in T, \quad (3.69)$$

$$y_{(i-N-1)it} - \sum_{j=i-N}^{N-1} y_{ijt} = 0, \quad t \in T, \quad i = N+2, \dots, 2N-1, \quad (3.70)$$

$$x_{ik} \leq y_{(k-1)kt}, \quad i \in J, \quad k \in J, \quad t \in T_i, \quad (3.71)$$

$$\sum_{t=1}^M y_{(k-1)kt} = C, \quad k = 2, \dots, N, \quad (3.72)$$

$$x_{ik} \in \{0, 1\}, \quad i, k = 1, \dots, N, \quad (3.73)$$

$$y_{i(i+1)t} \in \{0, 1\}, \quad i = 0, \dots, N, \quad t \in T, \quad (3.74)$$

$$y_{i(N+1)t} \in \{0, 1\}, \quad i \in J \setminus \{N\}, \quad t \in T, \quad (3.75)$$

$$y_{i(N+1+i)t} \in \{0, 1\}, \quad i = 1, \dots, N-2, \quad t \in T, \quad (3.76)$$

$$y_{(N+1+i)(j+1)t} \in \{0, 1\}, \quad i = 1, \dots, N-2, \quad j = i, \dots, N-2, \quad t \in T, \quad (3.77)$$

$$y_{0it} \in \{0, 1\}, \quad i = 2, \dots, N-2. \quad (3.78)$$

A função objetivo (3.61) minimiza a quantidade de trocas de ferramentas. As restrições



(3.62) e (3.63) garantem que toda tarefa seja sequenciada em uma única posição. As restrições (3.64) - (3.70) são restrições de conservação de fluxo. As restrições (3.71) impõem que se a tarefa  $i$  é a  $k$ -ésima tarefa a ser processada então todas as ferramentas necessárias para processá-la devem estar na máquina. As restrições (3.72) são restrições de capacidade, e as restrições (3.73) - (3.78) são restrições de domínio das variáveis. Observe que este modelo não há subsequências.

Este modelo apresenta  $3N + 2MN + N^2 \sum_i |S_i| - 1$  restrições e  $N^2 + \frac{M}{2}(N^2 + 3N - 2)$  variáveis binárias.

### 3.1.5 Modelo FM-PMTF-1D

A última variação do Modelo FM-PMTF-1, denominada Modelo FM-PMTF-1D, consiste em não especificar as ferramentas que saem da máquina, apenas contabilizar as trocas. Para isto, considera-se que há  $N$  cópias de cada uma das ferramentas no nó 0. As ferramentas que deixam a máquina correspondem ao fluxo das *commodities* nos arcos  $(i, N + 1)$ ,  $i \in J \setminus \{N\}$ , sem especificar se voltam ou não à máquina. Toda vez que uma ferramenta sai da máquina, mas é necessária para produzir alguma tarefa posteriormente, uma cópia desta ferramenta entra novamente na máquina para processar esta tarefa. As trocas são contabilizadas pelo fluxo nos arcos  $(i, N + 1)$ ,  $i \in J \setminus \{N\}$ , uma vez que se houver fluxo de uma ferramenta  $t_1$  em um arco  $(i, N + 1)$ , então haverá fluxo de uma ferramenta  $t_2$  no arco  $(0, i)$ . O grafo resultante é apresentado na Figura 3.6 e sua formulação é dada pelas expressões (3.79)-(3.94).

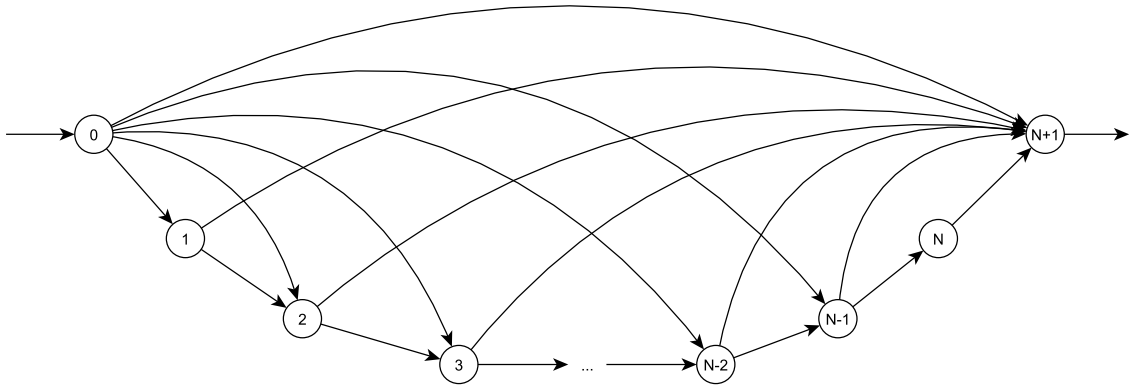


FIGURA 3.6 – Grafo do Modelo FM-PMTF-1D.

$$\text{Min } Z_4 = \sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(N+1)t} \tag{3.79}$$

sujeito a:

$$\sum_{k=1}^N x_{ik} = 1, \quad i \in J, \quad (3.80)$$

$$\sum_{i=1}^N x_{ik} = 1, \quad k \in J, \quad (3.81)$$

$$\sum_{i=1}^{N-1} y_{0it} + y_{0(N+1)t} = N, \quad t \in T, \quad (3.82)$$

$$y_{0it} - y_{1(N+1)t} - y_{12t} = 0, \quad t \in T, \quad (3.83)$$

$$y_{(i-1)it} + y_{0it} - y_{i(i+1)t} - y_{i(N+1)t} = 0, \quad t \in T, \quad i = 2, \dots, N-1, \quad (3.84)$$

$$y_{(N-1)Nt} - y_{N(N+1)t} = 0, \quad t \in T, \quad (3.85)$$

$$\sum_{i=0}^N y_{i(N+1)t} = N, \quad t \in T, \quad (3.86)$$

$$x_{ik} \leq y_{(k-1)kt}, \quad i \in J, \quad k \in J, \quad t \in T_i, \quad (3.87)$$

$$\sum_{t=1}^M y_{(k-1)kt} = C, \quad k = 2, \dots, N, \quad (3.88)$$

$$\sum_{t=1}^M y_{0it} \leq 2C, \quad (3.89)$$

$$x_{ik} \in \{0, 1\}, \quad i, k = 1, \dots, N, \quad (3.90)$$

$$y_{(i-1)it} \in \{0, 1\}, \quad i = 1, \dots, N+1, \quad t \in T, \quad (3.91)$$

$$y_{i(N+1)t} \in \{0, 1\}, \quad i \in J \setminus \{N\}, \quad t \in T, \quad (3.92)$$

$$y_{0it} \in \{0, 1\}, \quad i \in J \setminus \{N\}, \quad (3.93)$$

$$y_{0(N+1)t} \in \{1, \dots, N-1\}, \quad t \in T. \quad (3.94)$$

A função objetivo (3.79) minimiza a quantidade de trocas de ferramentas. As restrições (3.80) e (3.81) garantem que toda tarefa seja sequenciada em uma única posição. As restrições (3.82) - (3.86) são restrições de conservação de fluxo. As restrições (3.87) impõem que se a tarefa  $i$  é a  $k$ -ésima tarefa a ser processada então todas as ferramentas necessárias para processá-la devem estar na máquina. As restrições (3.88) e (3.89) são restrições de capacidade, e as restrições (3.90) - (3.94) são restrições de domínio das variáveis. Observe que este modelo não há subsequências, e as restrições (3.89) são necessárias para garantir factibilidade.

Este modelo apresenta  $3N + 2M + MN + N^2 \sum_i |S_i|$  restrições,  $N^2 + 2MN + N - 1$  variáveis binárias, e  $M$  variáveis inteiras.

Na Tabela 3.1 apresentam-se as quantidades de restrições e variáveis binárias para cada um dos modelos de fluxo *multicommodity* desenvolvidos nesta tese.

	Restrições	Variáveis
FM-PMTF-1	$3N + 2M + 2MN + N^2 \sum_i  T_i $	$N^2 + 3MN$
FM-PMTF-1A	$3N + 4M - \frac{MN}{2} + N^2(\sum_i  T_i  + \frac{M}{2})$	$N^2 + 4MN - 2M$
FM-PMTF-1B	$3N + 5M - 2MN + N^2(\sum_i  T_i  + M) - 1$	$N^2 + 4MN + 4M + N - 2$
FM-PMTF-1C	$3N + 2MN + N^2 \sum_i  T_i  - 1$	$N^2 + \frac{M}{2}(N^2 + 3N - 2)$
FM-PMTF-1D	$3N + 2M + MN + N^2 \sum_i  T_i $	$N^2 + 2MN + N - 1$

TABELA 3.1 – Quadro resumo dos modelos de fluxo *multicommodity* desenvolvidos nesta tese.

### 3.2 Relaxação Linear

Uma relaxação linear (RL) forte do modelo de otimização linear inteiro, ou seja, valor ótimo da RL é mais próximo do valor ótimo do modelo original, impacta fortemente no desempenho computacional de métodos exatos enumerativos, como o algoritmo *branch-and-bound*. Se a relaxação linear é fraca, como é o caso da formulação TD, pode ser necessária uma enumeração implícita quase completa da árvore de solução (LAPORTE *et al.*, 2004). Nesta seção uma análise das relaxações lineares dos modelos propostos é realizada.

**Teorema 3.2.1.** *A relaxação linear do Modelo FM-PMTF-1 é pelo menos igual a  $(M - C)$ .*

*Demonstração.* Considerando as restrições (3.4) e somando-as em todo  $t \in T$ , tem-se que:

$$\sum_{t=1}^M y_{01t} + \sum_{t=1}^M y_{0(N+1)t} = M. \quad (3.95)$$

Pelas restrições (3.10), tomando  $k = 1$  tem-se:

$$\sum_{t=1}^M y_{01t} = C. \quad (3.96)$$

Juntando as relações (3.95) e (3.96) segue que:

$$\sum_{t=1}^M y_{0(N+1)t} = M - C. \quad (3.97)$$

Tomando a restrição (3.11), somando-as em todo  $t \in T$ , e considerando a relação

(3.97) tem-se que:

$$\sum_{t=1}^M \sum_{i=1}^{N-1} y_{(N+1)it} \geq M - C. \quad (3.98)$$

Agora, levando em conta as restrições (3.5), somando-as em todo  $t \in T$  e  $i \in J \setminus \{N\}$ , tem-se:

$$\sum_{t=1}^M \sum_{i=1}^{N-1} (y_{(i-1)it} + y_{(N+1)it} - y_{i(N+1)t} - y_{i(i+1)t}) = 0, \quad (3.99)$$

$$\sum_{t=1}^M y_{01t} + \sum_{t=1}^M \sum_{i=2}^{N-1} y_{(i-1)it} + \sum_{t=1}^M \sum_{i=1}^{N-1} y_{(N+1)it} - \sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(N+1)t} - \sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(i+1)t} = 0, \quad (3.100)$$

Observe que

$$\sum_{t=1}^M \sum_{i=2}^{N-1} y_{(i-1)it} = \sum_{t=1}^M \sum_{i=1}^{N-2} y_{i(i+1)t}. \quad (3.101)$$

Substituindo a relação (3.101) na igualdade (3.100), tem-se

$$\sum_{t=1}^M y_{01t} + \sum_{t=1}^M \sum_{i=1}^{N-2} y_{i(i+1)t} + \sum_{t=1}^M \sum_{i=1}^{N-1} y_{(N+1)it} - \sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(N+1)t} - \sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(i+1)t} = 0, \quad (3.102)$$

Somando e subtraindo o termo  $\sum_{t=1}^M y_{(N-1)Nt}$  na igualdade (3.102) tem-se

$$\sum_{t=1}^M y_{01t} - \sum_{t=1}^M y_{(N-1)Nt} + \sum_{t=1}^M \sum_{i=1}^{N-1} y_{(N+1)it} - \sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(N+1)t} = 0. \quad (3.103)$$

Utilizando as restrições (3.10) com  $i = 1$  e  $i = N$ , e substituindo-as na relação (3.103) tem-se

$$C - C + \sum_{t=1}^M \sum_{i=1}^{N-1} y_{(N+1)it} - \sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(N+1)t} = 0, \quad (3.104)$$

$$\sum_{t=1}^M \sum_{i=1}^{N-1} y_{(N+1)it} = \sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(N+1)t}. \quad (3.105)$$

Portanto, unindo as relações (3.98) e (3.105), tem-se que

$$\sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(N+1)t} \geq M - C, \quad (3.106)$$

de onde o resultado segue. □

**Teorema 3.2.2.** *A relaxação linear dos Modelos FM-PMTF-1A, FM-PMTF-1B e FM-PMTF-1C é pelo menos M-C.*

*Demonstração.* A restrição (3.32) é comum a todos os três modelos. Desta forma, somando esta igualdade em todo  $t \in T$ , e abrindo o somatório, tem-se que

$$\sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(N+1)t} + \sum_{t=1}^M y_{N(N+1)t} = M. \quad (3.107)$$

Agora, como

$$y_{N(N+1)t} = y_{(N-1)Nt}, \quad t \in T, \quad (3.108)$$

e

$$\sum_{t=1}^M y_{(k-1)kt} = C, \quad k = 2, \dots, N, \quad (3.109)$$

tomando  $k = N$  em (3.109), e substituindo na relação (3.107) o resultado segue. □

**Teorema 3.2.3.** *A relaxação linear do Modelo FM-PMTF-1D é pelo menos M-C.*

*Demonstração.* Considerando a igualdade (3.86) e somando-a para todo  $t \in T$ , temos

$$\sum_{t=1}^M \sum_{i=0}^N y_{i(N+1)t} = MN, \quad (3.110)$$

e abrindo o somatório para  $i = 0$  e  $i = N$ , temos

$$\sum_{t=1}^M y_{0(N+1)t} + \sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(N+1)t} + \sum_{t=1}^M y_{N(N+1)t} = MN. \quad (3.111)$$

Considerando a igualdade (3.82) e somando-a para todo  $t \in T$ , temos

$$\sum_{t=1}^M \sum_{i=1}^{N-1} y_{0it} + \sum_{t=1}^M y_{0(N+1)t} = MN, \quad (3.112)$$

e, então,

$$\sum_{t=1}^M y_{0(N+1)t} = MN - \sum_{t=1}^M \sum_{i=1}^{N-1} y_{0it}. \quad (3.113)$$

Tomando  $i = N + 1$  na restrição (3.88), temos

$$\sum_{t=1}^M y_{N(N+1)t} = C. \quad (3.114)$$

Substituindo as relações (3.113) e (3.114) na igualdade (3.111) segue que

$$MN - \sum_{t=1}^M \sum_{i=1}^{N-1} y_{0it} + C + \sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(N+1)t} = MN, \quad (3.115)$$

e, então,

$$C + \sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(N+1)t} = \sum_{t=1}^M \sum_{i=1}^{N-1} y_{0it}. \quad (3.116)$$

Sabemos da restrição (3.94) que

$$y_{0(N+1)t} \leq N - 1. \quad (3.117)$$

Assim, somando a relação (3.117) para todo  $t \in T$ , temos

$$\sum_{t=1}^M y_{0(N+1)t} \leq MN - M. \quad (3.118)$$

Substituindo a relação (3.113) na igualdade (3.118), temos

$$\sum_{t=1}^M \sum_{i=1}^{N-1} y_{0it} \geq M. \quad (3.119)$$

Finalmente, substituindo a relação (3.119) na igualdade (3.116) temos

$$\sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(N+1)t} \geq M - C, \quad (3.120)$$

de onde o resultado segue.

□

### 3.3 Testes computacionais

Nesta seção os resultados dos estudos computacionais desenvolvidos para o PMTF são apresentados. O objetivo do primeiro estudo é testar a performance dos vários modelos propostos, gerados a partir do Modelo MF-PMTF-1. O segundo estudo teve o objetivo de testar a qualidade dos limitantes obtidos através da Relaxação Linear do melhor modelo de fluxo proposto com os modelos da literatura, assim como suas performances.

Todos os modelos utilizados nos testes desta seção foram implementados na sintaxe do AMPL (FOURER *et al.*, 2002) e o *solver* CPLEX 12.6.3 (ILOG..., 2006) foi usado para resolvê-los em um computador equipado com um processador Intel® Core™ i7 de 3.6 GHz, com 16 GB de RAM em um ambiente Windows 10. Testes iniciais com versões posteriores do CPLEX foram realizados, mas em todos eles observou-se um desempenho inferior. As

instâncias foram resolvidas pelo método “*branch-and-cut*” com a configuração padrão do CPLEX, considerando o parâmetro `branch = 1` (ou seja, a direção de ramificação é restrita ao valor mais alto em cada nó).

Todas as instâncias utilizadas podem ser encontradas em (<https://sites.google.com/site/antoniochaves/publications/data>). Cada conjunto de dados contém  $i$  instâncias aleatórias com o mesmo número de tarefas, ferramentas e capacidade da máquina. Os valores de  $i$  são apresentados nas Tabelas 3.8 e 3.9.

Yanasse *et al.* (2009) propuseram 1,350 instâncias divididas em 5 grupos ( $A$ ,  $B$ ,  $C$ ,  $D$  e  $E$ ) de acordo com as configurações de  $N$ ,  $M$  e  $C$ . Nas Tabelas 3.5, 3.6 e 3.8, resumem-se os resultados obtidos para esse conjunto de instâncias. Catanzaro *et al.* (2015) propuseram 160 instâncias divididas em 4 grupos ( $A$ ,  $B$ ,  $C$  e  $D$ ) de acordo com as configurações de  $N$ ,  $M$  e  $C$ . Nas Tabelas 3.7 e 3.9, resumem-se os resultados obtidos para esse conjunto de instâncias.

Restrições adicionais podem ser adicionadas aos modelos de fluxo *multicommodity* para reduzir o conjunto viável, mas garantindo que a solução obtida seja ótima. Com base na propriedade de simetria (GHIANI *et al.*, 2007; YANASSE *et al.*, 2009) do PMTF, tem-se o seguinte resultado:

**Proposição 3.3.1** (Eliminação de simetria). *Dado qualquer tarefa  $p$ , existe uma solução ótima na qual  $p$  é processada em uma das primeiras  $\lceil N/2 \rceil$  posições.*

A proposição 3.3.1 é válida para qualquer tarefa  $p$ . A restrição 3.121 impõe que essa tarefa  $p$  precise ser processada em uma das primeiras  $\lceil N/2 \rceil$  posições. Nos testes computacionais executados nesta tese, escolheu-se  $p$  tal que  $p = \arg \max_{j \in J} |T_j|$ .

$$\sum_{k=1}^{\lceil N/2 \rceil} x_{pk} = 1. \tag{3.121}$$

### 3.3.1 Performance dos modelos de fluxo *multicommodity* propostos para o PMTF

Este estudo teve por objetivo avaliar o desempenho dos modelos de fluxo *multicommodity* desenvolvidos utilizando o CPLEX 12.6.3 (ILOG..., 2006) para a solução das instâncias envolvidas. Para isto, utilizou-se as instâncias propostas por Catanzaro *et al.* (2015) e Yanasse *et al.* (2009). Nas Tabelas 3.2 e 3.3 são apresentados os resultados obtidos para as instâncias de Catanzaro *et al.* (2015) e na Tabela 3.4 são apresentados os resultados obtidos para as instâncias de Yanasse *et al.* (2009). O número de tarefas ( $N$ ), o número de ferramentas ( $M$ ), a capacidade da máquina ( $C$ ) e o número de instâncias ( $i$ ) em cada um



dos grupos (Grupo) das instâncias usadas nos testes são apresentados no início da Tabela. Para cada um dos modelos de fluxo, em cada um dos grupos de instâncias, apresentam-se o valor de solução médio (S), a quantidade de instâncias resolvidas na otimalidade (O) dentro do tempo limite predefinido de 3600 segundos, e o tempo médio de execução das instâncias resolvidas na otimalidade (T).

Considerando as instâncias de Catanzaro *et al.* (2015), todas as instâncias do grupo A, por serem triviais, foram resolvidas na otimalidade considerando todos os modelos de fluxo. Quanto ao tempo computacional, o modelo MF-PMTF-1A apresentou um melhor desempenho, com tempo médio melhor nas duas primeiras classes de instâncias. Nas duas últimas classes, o modelo MF-PMTF-1D apresentou melhores tempos, empatando na quarta classe com o tempo médio fornecido pelo modelo MF-PMTF-1B. No grupo B, o CPLEX foi capaz de resolver 29, 37, 35, 34 e 37 instâncias na otimalidade considerando os modelos MF-PMTF-1, MF-PMTF-1A, MF-PMTF-1B, MF-PMTF-1C e MF-PMTF-1D, respectivamente. O tempo computacional dos modelos MF-PMTF-1A e MF-PMTF-1D, que foram os com melhor desempenho neste grupo, tiveram resultados diferentes, sendo que em todas as classes, o primeiro modelo teve tempo melhor. O modelo MF-PMTF-1A utilizou 88,55%, 73,15%, 49,44% e 18,43% dos tempos médios do modelo MF-PMTF-1D nas classes A, B, C e D, respectivamente.

O CPLEX não foi capaz de resolver nenhuma instância de Catanzaro *et al.* (2015) com nenhum dos modelos de fluxo propostos nos grupos C e D. Entretanto, considerando o grupo C, o CPLEX retornou melhores limitantes superiores em 13, 9, 4, 0 e 14 instâncias, enquanto no grupo D, o CPLEX retornou os melhores limitantes superiores em 10, 4, 0, 5 e 21 instâncias nos modelos MF-PMTF-1, MF-PMTF-1A, MF-PMTF-1B, MF-PMTF-1C e MF-PMTF-1D, respectivamente (conforme resultados no Anexo A).

Como os modelos MF-PMTF-1, MF-PMTF-1A e MF-PMTF-1D apresentaram melhores resultados nos testes computacionais considerando as instâncias fornecidas por Catanzaro *et al.* (2015), apenas estes modelos foram utilizados nos testes computacionais nas instâncias fornecidas por Yanasse *et al.* (2009).

Considerando as instâncias de Yanasse *et al.* (2009), todas as instâncias dos grupos A e B, por serem triviais, foram resolvidas na otimalidade considerando todos os modelos de fluxo. Quanto ao tempo computacional, o modelo MF-PMTF-1A apresentou um melhor desempenho, com tempo médio melhor em 7 classes de instâncias, enquanto o modelo MF-PMTF-1D teve melhor desempenho em 2 classes de instâncias, no grupo A. Já no grupo B, o resultado foi ao contrário, sendo que o modelo MF-PMTF-1D teve melhor desempenho em 7 classes de instâncias contra 2 do modelo MF-PMTF-1A. Há que se observar entretanto, que os tempos computacionais destes dois modelos foram muito próximos.

CPLEX foi capaz de resolver 198, 236 e 242 instâncias do grupo C na otimalidade considerando os modelos MF-PMTF-1, MF-PMTF-1A e MF-PMTF-1D, respectivamente. Já nos grupos D e E, o CPLEX foi capaz de resolver 96, 110 e 108 instâncias na otimalidade (grupo D), e 70, 73 e 73 instâncias na otimalidade (grupo E) considerando os modelos MF-PMTF-1, MF-PMTF-1A e MF-PMTF-1D, respectivamente.

De todas as instâncias fornecidas por Yanasse *et al.* (2009), CPLEX foi capaz de resolver 1034 (76,59%), 1089 (80,67%) e 1093 (80,96%) instâncias na otimalidade considerando os modelos MF-PMTF-1, MF-PMTF-1A e MF-PMTF-1D, respectivamente.

De modo geral, o modelo MF-PMTF-1D teve melhor desempenho em relação aos outros modelos de fluxo. Este resultado se deve ao seu menor número de variáveis e restrições.

TABELA 3.2 – Resultados dos modelos de fluxo MF-PMTF-1, MF-PMTF-1A e MF-PMTF-1B para as instâncias introduzidas por Catanzaro *et al.* (2015).

Grupo	$N$	$M$	$C$	$i$	MF-PMTF-1			MF-PMTF-1A			MF-PMTF-1B		
					S	O	T	S	O	T	S	O	T
A	10	10	4	10	<b>8,50</b>	<b>10</b>	5,37	<b>8,50</b>	<b>10</b>	<b>1,61</b>	<b>8,50</b>	<b>10</b>	2,64
	10	10	5	10	<b>5,80</b>	<b>10</b>	5,88	<b>5,80</b>	<b>10</b>	0,68	<b>5,80</b>	<b>10</b>	1,25
	10	10	6	10	<b>4,10</b>	<b>10</b>	2,56	<b>4,10</b>	<b>10</b>	0,23	<b>4,10</b>	<b>10</b>	0,25
	10	10	7	10	<b>3,00</b>	<b>10</b>	0,14	<b>3,00</b>	<b>10</b>	0,09	<b>3,00</b>	<b>10</b>	0,07
B	15	20	6	10	20,60	7	1241,95	<b>20,50</b>	9	1299,73	20,60	6	1013,22
	15	20	8	10	<b>13,70</b>	6	1524,39	<b>13,70</b>	<b>9</b>	<b>798,83</b>	<b>13,70</b>	<b>9</b>	1192,90
	15	20	10	10	<b>9,70</b>	6	41,43	<b>9,70</b>	9	223,54	<b>9,70</b>	<b>10</b>	<b>371,08</b>
	15	20	12	10	<b>7,20</b>	<b>10</b>	3,82	<b>7,20</b>	<b>10</b>	1,27	<b>7,20</b>	<b>10</b>	6,91
C	30	40	15	10	97,10	0	-	97,50	0	-	98,50	0	-
	30	40	17	10	77,40	0	-	77,40	0	-	78,00	0	-
	30	40	20	10	56,90	0	-	57,80	0	-	57,50	0	-
	30	40	25	10	<b>32,70</b>	0	-	32,80	0	-	34,40	0	-
D	40	60	20	10	215,40	0	-	221,90	0	-	231,50	0	-
	40	60	22	10	187,00	0	-	189,90	0	-	193,00	0	-
	40	60	25	10	<b>151,80</b>	0	-	156,70	0	-	161,60	0	-
	40	60	30	10	114,60	0	-	117,00	0	-	129,60	0	-

### 3.3.2 Relaxação Linear dos modelos para o PMTF

Foram realizados experimentos computacionais, comparando a relaxação linear dos modelos de fluxo desenvolvidos, cujos valores são todos iguais a  $M - C$ , com as relaxações lineares dos modelos TD, LSS e CGL, sugeridos em (TANG; DENARDO, 1988a; LAPORTE *et al.*, 2004; CATANZARO *et al.*, 2015), respectivamente. O modelo proposto por Bard (1988) não foi considerado nos testes, pois seu modelo linearizado é exatamente o TD. Considerou-se os conjuntos de dados descritos em Yanasse *et al.* (2009) e Catanzaro *et al.* (2015).

Nas Tabelas 3.5, 3.6 e 3.7, apresentam-se o número de tarefas ( $N$ ), o número de ferramentas ( $M$ ), a capacidade da máquina ( $C$ ) e o número de instâncias ( $i$ ) em cada

TABELA 3.3 – Resultados dos modelos MF-PMTF-1C e MF-PMTF-1D de fluxo para as instâncias introduzidas por Catanzaro *et al.* (2015).

Grupo	$N$	$M$	$C$	$i$	MF-PMTF-1C			MF-PMTF-1D		
					S	O	T	S	O	T
A	10	10	4	10	<b>8,50</b>	<b>10</b>	3,44	<b>8,50</b>	<b>10</b>	3,33
	10	10	5	10	<b>5,80</b>	<b>10</b>	1,11	<b>5,80</b>	<b>10</b>	1,38
	10	10	6	10	<b>4,10</b>	<b>10</b>	0,25	<b>4,10</b>	<b>10</b>	0,19
	10	10	7	10	<b>3,00</b>	<b>10</b>	0,08	<b>3,00</b>	<b>10</b>	0,07
B	15	20	6	10	<b>20,50</b>	6	1742,65	<b>20,50</b>	<b>10</b>	<b>1467,82</b>
	15	20	8	10	<b>13,70</b>	8	1516,80	<b>13,70</b>	8	1092,00
	15	20	10	10	<b>9,70</b>	<b>10</b>	381,26	<b>9,70</b>	9	452,18
	15	20	12	10	<b>7,20</b>	<b>10</b>	7,67	<b>7,20</b>	<b>10</b>	6,89
C	30	40	15	10	101,60	0	-	<b>96,60</b>	0	-
	30	40	17	10	80,50	0	-	<b>76,70</b>	0	-
	30	40	20	10	60,90	0	-	<b>55,00</b>	0	-
	30	40	25	10	35,20	0	-	33,00	0	-
D	40	60	20	10	375,83	0	-	<b>212,40</b>	0	-
	40	60	22	10	310,00	0	-	<b>184,60</b>	0	-
	40	60	25	10	255,40	0	-	156,30	0	-
	40	60	30	10	355,60	0	-	<b>111,00</b>	0	-

um dos grupos (Grupo) das instâncias usadas nos testes. Para cada um dos modelos, apresenta-se o valor objetivo médio da relaxação linear associada (RL). Considerou-se que toda inserção de uma ferramenta em um magazine de ferramentas é uma troca, como em Catanzaro *et al.* (2015) e, no início das operações na máquina, haverá  $C$  ferramentas na máquina. Portanto, o valor de  $C$  foi adicionado aos valores da relaxação linear dos modelos TD, LSS e MF-PMTF-1D. Considerou-se também subtrair o valor  $C$  nos valores da relaxação linear do modelo CGL, mas, em alguns casos, o valor resultante teria sido negativo, como no exemplo  $B4 - 1$  de Catanzaro *et al.* (2015), cujo valor de RL é 5 e o valor de  $C$  é 12.

Os resultados dos testes indicam que quanto maior a capacidade da máquina, para valores fixos de  $N$  e  $M$ , mais o valor da relaxação linear do modelo LSS se aproxima do valor da relaxação linear do modelo TD, tornando-se igual em alguns casos e, quando a capacidade da máquina é apertada, a relaxação linear do modelo LSS é bem mais forte.

O modelo CGL é baseado no modelo LSS, portanto, apresenta o mesmo comportamento. No entanto, quando a capacidade da máquina se reduz, o valor da relaxação linear é mais forte. Quando a capacidade da máquina aumenta, os valores da relaxação linear são baixos, piores que os valores da relaxação linear obtidos pelo modelo TD em 23 (Tabela 3.5, 3.6) e 6 (Tabela 3.7) conjuntos de instâncias. Estes resultados diferem dos resultados apresentados no trabalho de Catanzaro *et al.* (2015). O sinal “-” usado nas Tabelas indica que o CPLEX não conseguiu resolver o modelo relaxado dentro do tempo predefinido de 3.600 segundos.

Os resultados mostram que, em geral, o modelo MF-PMTF-1D forneceu os melhores limitantes inferiores. Dos 66 conjuntos de instâncias testados, o modelo MF-PMTF-1D

TABELA 3.4 – Resultados dos modelos de fluxo MF-PMTF-1, MF-PMTF-1A e MF-PMTF-1D para as instâncias introduzidas por Yanasse *et al.* (2009).

Grupo	N	M	C	i	MF-PMTF-1			MF-PMTF-1A			MF-PMTF-1D		
					S	O	T	S	O	T	S	O	T
A	8	15	5	10	<b>12,00</b>	<b>10</b>	1,57	<b>12,00</b>	<b>10</b>	<b>0,71</b>	<b>12,00</b>	<b>10</b>	0,99
	8	15	10	30	<b>6,83</b>	<b>30</b>	1,95	<b>6,83</b>	<b>30</b>	<b>0,69</b>	<b>6,83</b>	<b>30</b>	0,82
	8	20	5	10	<b>16,80</b>	<b>10</b>	3,29	<b>16,80</b>	<b>10</b>	<b>1,02</b>	<b>16,80</b>	<b>10</b>	1,48
	8	20	10	30	<b>13,07</b>	<b>30</b>	3,22	<b>13,07</b>	<b>30</b>	<b>1,56</b>	<b>13,07</b>	<b>30</b>	1,74
	8	20	15	60	<b>7,08</b>	<b>60</b>	3,18	<b>7,08</b>	<b>60</b>	1,03	<b>7,08</b>	<b>60</b>	<b>1,02</b>
	8	25	5	10	<b>20,20</b>	<b>10</b>	0,85	<b>20,20</b>	<b>10</b>	<b>0,46</b>	<b>20,20</b>	<b>10</b>	0,54
	8	25	10	30	<b>18,20</b>	<b>30</b>	4,63	<b>18,20</b>	<b>30</b>	<b>2,30</b>	<b>18,20</b>	<b>30</b>	3,01
	8	25	15	60	<b>12,95</b>	<b>60</b>	9,25	<b>12,95</b>	<b>60</b>	<b>2,36</b>	<b>12,95</b>	<b>60</b>	2,55
8	25	20	100	<b>6,61</b>	<b>100</b>	10,16	<b>6,61</b>	<b>100</b>	1,28	6,61	<b>100</b>	<b>1,21</b>	
B	9	15	5	10	<b>12,20</b>	<b>10</b>	5,86	<b>12,20</b>	<b>10</b>	<b>2,15</b>	<b>12,20</b>	<b>10</b>	3,11
	9	15	10	30	<b>7,37</b>	<b>30</b>	9,78	<b>7,37</b>	<b>30</b>	2,05	<b>7,37</b>	<b>30</b>	<b>1,82</b>
	9	20	5	10	<b>17,40</b>	<b>10</b>	7,22	<b>17,40</b>	<b>10</b>	2,80	<b>17,40</b>	<b>10</b>	<b>2,69</b>
	9	20	10	30	<b>14,17</b>	<b>30</b>	6,35	<b>14,17</b>	<b>30</b>	4,77	<b>14,17</b>	<b>30</b>	<b>4,31</b>
	9	20	15	60	<b>7,60</b>	<b>60</b>	25,21	<b>7,60</b>	<b>60</b>	<b>2,93</b>	<b>7,60</b>	<b>60</b>	2,47
	9	25	5	10	<b>20,40</b>	<b>10</b>	5,92	<b>20,40</b>	<b>10</b>	<b>1,06</b>	<b>20,40</b>	<b>10</b>	1,22
	9	25	10	30	<b>18,77</b>	<b>30</b>	8,57	<b>18,77</b>	<b>30</b>	6,14	<b>18,77</b>	<b>30</b>	<b>5,57</b>
	9	25	15	50	<b>14,74</b>	<b>50</b>	32,01	<b>14,74</b>	<b>50</b>	6,30	<b>14,74</b>	<b>50</b>	<b>5,78</b>
9	25	20	100	<b>7,19</b>	<b>100</b>	37,00	<b>7,19</b>	<b>100</b>	3,08	<b>7,19</b>	<b>100</b>	<b>2,50</b>	
C	15	15	5	10	<b>16,60</b>	8	916,67	<b>16,60</b>	8	632,63	<b>16,60</b>	<b>9</b>	<b>697,26</b>
	15	15	10	30	9,83	28	810,32	<b>9,80</b>	<b>29</b>	<b>462,36</b>	<b>9,80</b>	28	380,14
	15	20	5	10	<b>20,60</b>	<b>9</b>	1758,54	<b>20,60</b>	<b>9</b>	<b>867,65</b>	<b>20,60</b>	<b>9</b>	1092,87
	15	20	10	30	18,47	17	679,65	<b>18,40</b>	19	708,07	18,43	<b>20</b>	<b>735,65</b>
	15	20	15	60	10,55	35	802,10	10,55	45	526,93	<b>10,52</b>	<b>48</b>	<b>754,72</b>
	15	25	5	10	<b>27,50</b>	1	1233,86	27,70	<b>4</b>	<b>1307,37</b>	<b>27,50</b>	3	1258,49
	15	25	10	30	25,33	13	308,51	25,27	15	693,22	<b>25,23</b>	<b>15</b>	<b>620,30</b>
	15	25	15	60	19,27	27	1016,14	19,22	29	968,15	<b>19,17</b>	<b>30</b>	<b>977,98</b>
15	25	20	100	9,72	60	545,99	<b>9,71</b>	78	570,31	<b>9,71</b>	<b>80</b>	<b>565,50</b>	
D	20	15	5	10	21,70	0	-	21,70	0	-	<b>21,60</b>	0	-
	20	15	10	20	8,40	6	194,86	<b>8,35</b>	<b>11</b>	<b>343,81</b>	8,40	<b>11</b>	681,81
	20	20	5	10	<b>25,30</b>	0	-	25,40	0	-	25,40	0	-
	20	20	10	10	<b>10,70</b>	<b>6</b>	803,21	<b>10,70</b>	<b>6</b>	<b>31,51</b>	<b>10,70</b>	<b>6</b>	541,66
	20	20	15	30	6,73	10	0,34	<b>6,70</b>	<b>12</b>	<b>72,10</b>	6,73	11	245,65
	20	25	5	10	<b>31,20</b>	0	-	31,30	0	-	31,40	0	-
	20	25	10	10	15,70	5	421,63	15,60	<b>6</b>	<b>491,60</b>	<b>15,50</b>	5	99,07
	20	25	15	40	22,40	<b>10</b>	6,39	22,73	<b>10</b>	1,15	<b>22,38</b>	<b>10</b>	<b>0,99</b>
	20	25	20	40	6,20	25	74,11	6,20	<b>26</b>	60,85	<b>6,15</b>	<b>26</b>	<b>91,82</b>
	25	15	10	10	6,10	2	8,68	6,00	<b>6</b>	1435,93	<b>5,90</b>	<b>6</b>	<b>996,28</b>
	25	20	10	10	12,30	<b>1</b>	119,20	12,30	<b>1</b>	104,83	<b>12,20</b>	<b>1</b>	<b>80,49</b>
	25	20	15	10	8,20	0	-	7,90	0	-	<b>7,70</b>	0	-
	25	25	10	10	17,90	<b>2</b>	1780,40	<b>17,70</b>	1	3427,69	18,00	<b>2</b>	<b>795,24</b>
	25	25	15	10	<b>10,00</b>	<b>10</b>	56,06	<b>10,00</b>	<b>10</b>	14,91	<b>10,00</b>	<b>10</b>	<b>14,60</b>
25	25	20	30	5,60	19	61,07	5,57	<b>21</b>	18,07	<b>5,53</b>	<b>21</b>	<b>204,81</b>	
E	10	10	4	10	<b>9,50</b>	<b>10</b>	5,31	<b>9,50</b>	<b>10</b>	<b>3,06</b>	<b>9,50</b>	<b>10</b>	3,34
	10	10	5	10	<b>6,20</b>	<b>10</b>	9,92	<b>6,20</b>	<b>10</b>	<b>1,55</b>	<b>6,20</b>	<b>10</b>	1,79
	10	10	6	10	<b>4,30</b>	<b>10</b>	3,91	<b>4,30</b>	<b>10</b>	<b>0,23</b>	<b>4,30</b>	<b>10</b>	0,38
	10	10	7	10	<b>3,00</b>	<b>10</b>	0,24	<b>3,00</b>	<b>10</b>	0,10	<b>3,00</b>	<b>10</b>	<b>0,07</b>
	15	20	6	10	<b>21,50</b>	7	911,26	<b>21,50</b>	8	<b>884,18</b>	<b>21,50</b>	8	901,99
	15	20	8	10	<b>14,30</b>	6	1105,62	<b>14,30</b>	8	<b>577,74</b>	<b>14,30</b>	7	768,90
	15	20	10	10	<b>10,30</b>	<b>9</b>	246,10	<b>10,30</b>	<b>9</b>	<b>41,68</b>	<b>10,30</b>	<b>9</b>	363,83
15	20	12	10	<b>8,20</b>	8	3,83	<b>8,20</b>	8	1,12	<b>8,20</b>	<b>9</b>	<b>180,81</b>	

obteve melhores resultados em 62 conjuntos de instâncias. O modelo CGL obteve os melhores resultados em 4 conjuntos de instâncias, devido ao pré-processamento utilizado para definição das variáveis de decisão, beneficiando-se quando a capacidade é reduzida.

TABELA 3.5 – Resultados dos valores da relaxação linear para os Grupos *A*, *B* e *C* fornecidos por Yanasse *et al.* (2009).

Grupo	<i>N</i>	<i>M</i>	<i>C</i>	<i>i</i>	MF-PMTF	TD	LSS	CGL
					RL	RL	RL	RL
<i>A</i>	8	15	5	10	<b>15,00</b>	5,00	6,59	13,74
	8	15	10	30	<b>15,00</b>	10,00	10,20	9,89
	8	20	5	10	<b>20,00</b>	5,00	7,16	19,02
	8	20	10	30	<b>20,00</b>	10,00	10,71	16,42
	8	20	15	60	<b>20,00</b>	15,00	15,25	12,22
	8	25	5	10	<b>25,00</b>	5,00	7,21	23,03
	8	25	10	30	<b>25,00</b>	10,00	11,10	22,02
	8	25	15	60	<b>25,00</b>	15,00	15,42	17,83
	8	25	20	100	<b>25,00</b>	20,00	20,06	14,03
<i>B</i>	9	15	5	10	<b>15,00</b>	5,00	6,19	12,95
	9	15	10	30	<b>15,00</b>	10,00	10,45	9,69
	9	20	5	10	<b>20,00</b>	5,00	7,09	19,03
	9	20	10	30	<b>20,00</b>	10,00	10,91	16,25
	9	20	15	60	<b>20,00</b>	15,00	15,24	11,96
	9	25	5	10	<b>25,00</b>	5,00	6,99	22,42
	9	25	10	30	<b>25,00</b>	10,00	10,88	21,59
	9	25	15	60	<b>25,00</b>	15,00	15,60	18,84
	9	25	20	100	<b>25,00</b>	20,00	20,10	13,56
<i>C</i>	15	15	5	10	<b>15,00</b>	5,00	6,43	14,88
	15	15	10	30	<b>15,00</b>	10,00	10,66	8,63
	15	20	5	10	<b>20,00</b>	5,00	6,45	18,19
	15	20	10	30	<b>20,00</b>	10,00	11,01	15,00
	15	20	15	60	<b>20,00</b>	15,00	15,40	10,82
	15	25	5	10	25,00	5,00	7,30	<b>27,00</b>
	15	25	10	30	<b>25,00</b>	10,00	11,01	20,80
	15	25	15	60	<b>25,00</b>	15,00	15,59	15,64
	15	25	20	100	<b>25,00</b>	20,00	20,25	11,17

### 3.3.3 Análise Computacional do Modelo MF-PMTF-1D com os Modelos da Literatura

O objetivo deste estudo foi verificar a performance do Modelo MF-PMTF-1D com os modelos propostos na literatura, quanto a tempo de execução e qualidade da solução. Nas Tabelas 3.8 e 3.9, apresentam-se o número de tarefas (*N*), o número de ferramentas (*M*), a capacidade da máquina (*C*) e o número de instâncias (*i*) em cada um dos grupos (Grupo) de instâncias usadas nos testes. Para cada um dos modelos, apresentam-se o número de instâncias resolvidas até a otimalidade dentro do prazo preestabelecido, 3.600 segundos (O), e o tempo médio das instâncias resolvidas até a otimalidade (T).

Os Grupos *A* e *B* fornecidos por Yanasse *et al.* (2009) são triviais e o CPLEX resolveu todas as instâncias em todos os quatro modelos, exceto o modelo CGL que não resolveu uma instância do Grupo *B*. Do total de 18 conjuntos de instâncias desses dois grupos, o modelo MF-PMTF-1D apresentou melhor tempo médio de execução em 13 conjuntos e,

TABELA 3.6 – Resultados dos valores da relaxação linear para os Grupos *D* e *E* fornecidos por Yanasse *et al.* (2009).

Grupo	<i>N</i>	<i>M</i>	<i>C</i>	<i>i</i>	MF-PMTF	TD	LSS	CGL
					RL	RL	RL	RL
<i>D</i>	20	15	5	10	15,00	5,00	6,65	<b>17,15</b>
	20	15	10	20	<b>15,00</b>	10,00	10,08	3,49
	20	20	5	10	<b>20,00</b>	5,00	6,43	19,44
	20	20	10	10	<b>20,00</b>	10,00	10,00	3,55
	20	20	15	30	<b>20,00</b>	15,00	15,00	3,93
	20	25	5	10	25,00	5,00	6,65	<b>25,09</b>
	20	25	10	10	<b>25,00</b>	10,00	10,00	6,30
	20	25	15	40	<b>25,00</b>	15,00	15,96	16,36
	20	25	20	40	<b>25,00</b>	20,00	20,00	4,96
	25	15	10	10	<b>15,00</b>	10,00	10,00	2,10
	25	20	10	10	<b>20,00</b>	10,00	10,00	2,45
	25	20	15	10	<b>20,00</b>	15,00	15,00	2,55
	25	25	10	10	<b>25,00</b>	10,00	10,00	4,45
	25	25	15	10	<b>25,00</b>	15,00	15,00	3,25
	25	25	20	30	<b>25,00</b>	20,00	20,00	3,98
<i>E</i>	10	10	4	10	10,00	4,00	5,58	<b>11,41</b>
	10	10	5	10	<b>10,00</b>	5,00	5,00	5,46
	10	10	6	10	<b>10,00</b>	6,00	6,00	2,95
	10	10	7	10	<b>10,00</b>	7,00	7,00	2,58
	15	20	6	10	<b>20,00</b>	6,00	7,46	18,34
	15	20	8	10	<b>20,00</b>	8,00	8,00	6,64
	15	20	10	10	<b>20,00</b>	10,00	10,00	5,23
	15	20	12	10	<b>20,00</b>	12,00	12,00	4,70

TABELA 3.7 – Resultados dos valores da relaxação linear para as instâncias introduzidas por Catanzaro *et al.* (2015).

Grupo	<i>N</i>	<i>M</i>	<i>C</i>	<i>i</i>	MF-PMTF	TD	LSS	CGL
					RL	RL	RL	RL
<i>A</i>	10	10	4	10	<b>10,00</b>	4,00	4,89	9,02
	10	10	5	10	<b>10,00</b>	5,00	5,00	4,32
	10	10	6	10	<b>10,00</b>	6,00	6,00	2,92
	10	10	7	10	<b>10,00</b>	7,00	7,00	2,88
<i>B</i>	15	20	6	10	<b>20,00</b>	6,00	7,28	17,65
	15	20	8	10	<b>20,00</b>	8,00	8,00	5,37
	15	20	10	10	<b>20,00</b>	10,00	10,00	3,75
	15	20	12	10	<b>19,20</b>	12,00	12,00	3,75
<i>C</i>	30	40	15	10	<b>40,00</b>	15,00	17,10	-
	30	40	17	10	<b>40,00</b>	17,00	17,00	-
	30	40	20	10	<b>40,00</b>	20,00	20,00	-
	30	40	25	10	<b>40,00</b>	25,00	25,00	-
<i>D</i>	40	60	20	10	<b>60,00</b>	20,00	23,36	-
	40	60	22	10	<b>60,00</b>	22,00	22,00	-
	40	60	25	10	<b>60,00</b>	25,00	25,00	-
	40	60	30	10	<b>60,00</b>	30,00	30,00	-

nos demais, o tempo médio de execução foi próximo aos outros modelos que obtiveram melhor tempo médio de execução.

Na Tabela 3.8, mostra-se que, de 340 instâncias, o CPLEX conseguiu resolver 242 (71,18%) instâncias na otimalidade para o Grupo C usando o modelo MF-PMTF-1D, enquanto usando os modelos TD, LSS e CGL, conseguiu resolver 168 (49,41%), 3 (0,88%) e 25 (7,35%) instâncias na otimalidade, respectivamente. Na classe ( $N = 15$ ,  $M = 25$ ,  $C = 5$ ), o modelo CGL teve mais instâncias resolvidas do que o modelo MF-PMTF-1D, pois o modelo CGL teve melhores limitantes inferiores para as instâncias dessa classe (de acordo com a Tabela 3.5).

Considerando o Grupo D, o conjunto que contém as maiores instâncias fornecidas por Yanasse *et al.* (2009), um total de 260 instâncias, o CPLEX conseguiu resolver 108 (41,53%), 13 (5,00%), 0 (0,00%) e 1 (0,38%) instâncias na otimalidade usando os modelos MF-PMTF-1D, TD, LSS e CGL, respectivamente.

Para o Grupo E na Tabela 3.8, de 80 instâncias, o CPLEX conseguiu resolver 73 (91,25%) instâncias na otimalidade usando o modelo MF-PMTF-1D, enquanto que usando os modelos TD, LSS e CGL, conseguiu solucionar apenas 60 (75,00%), 41 (51,25%) e 33 (41,25%) instâncias na otimalidade, respectivamente.

Os conjuntos de instâncias fornecidos por Catanzaro *et al.* (2015) produziram resultados semelhantes aos fornecidos por Yanasse *et al.* (2009), conforme mostrado na Tabela 3.9. Das 40 instâncias do Grupo A, o CPLEX conseguiu resolver 40 (100,00%) instâncias na otimalidade usando o modelo MF-PMTF-1D, enquanto usando os modelos TD, LSS e CGL, conseguiu resolver 40 (100,00%), 40 (100,00%) e 32 (80,00%) instâncias na otimalidade, respectivamente. No entanto, no Grupo B, de 40 instâncias, o CPLEX conseguiu resolver 37 (92,50%) instâncias na otimalidade usando o modelo MF-PMTF-1D, enquanto que, usando os modelos TD, LSS e CGL, conseguiu resolver apenas 25 (62,50%), 0 (0,00%) e 2 (5,00%) instâncias na otimalidade, respectivamente.

O CPLEX não foi capaz de resolver na otimalidade nenhuma instância dos Grupos C e D de Catanzaro *et al.* (2015) dentro do prazo preestabelecido considerando qualquer um dos modelos.

Do total de 1510 instâncias testadas, o CPLEX foi capaz de resolver 1170 (77,48%), 911 (60,33%), 714 (47,28%) e 728 (48,21%) instâncias na otimalidade considerando os modelos MF-PMTF-1D, TD, LSS e CGL, respectivamente. O CPLEX resolveu 194 instâncias a mais com o modelo de fluxo desenvolvido em relação ao modelo TD, que era o melhor modelo encontrado na literatura, considerando os testes aqui realizados.

Em todos os conjuntos de instâncias, o modelo MF-PMTF-1D obteve melhores resultados quando a capacidade da máquina aumenta, tanto no número de instâncias resolvidas na otimalidade quanto no tempo médio de solução destas instâncias. Este comportamento

fica mais evidente no gráfico tipo *performance profile* (DOLAN; MORÉ, 2002) apresentado na Figura 3.7. Este gráfico mostra a porção do conjunto de instâncias,  $\rho(\tau)$ , que pode ser resolvido por cada um dos modelos, dado um parâmetro  $\tau$ , considerando o *solver* CPLEX.

TABELA 3.8 – Resultados para instâncias dos Grupos *A*, *B*, *C*, *D*, e *E* de Yanasse *et al.* (2009).

Grupo	<i>N</i>	<i>M</i>	<i>C</i>	<i>i</i>	MF-PMTF		TD		LSS		CGL	
					<i>O</i>	<i>T</i>	<i>O</i>	<i>T</i>	<i>O</i>	<i>T</i>	<i>O</i>	<i>T</i>
<i>A</i>	8	15	5	10	<b>10</b>	<b>0,99</b>	<b>10</b>	1,16	<b>10</b>	2,74	<b>10</b>	<b>0,99</b>
	8	15	10	30	<b>30</b>	<b>0,82</b>	<b>30</b>	0,84	<b>30</b>	6,70	<b>30</b>	15,71
	8	20	5	10	<b>10</b>	1,48	<b>10</b>	1,91	<b>10</b>	4,09	<b>10</b>	<b>0,35</b>
	8	20	10	30	<b>30</b>	1,74	<b>30</b>	<b>1,26</b>	<b>30</b>	10,80	<b>30</b>	11,91
	8	20	15	60	<b>60</b>	<b>1,02</b>	<b>60</b>	2,03	<b>60</b>	8,85	<b>60</b>	46,17
	8	25	5	10	<b>10</b>	0,54	<b>10</b>	2,85	<b>10</b>	5,32	<b>10</b>	<b>0,25</b>
	8	25	10	30	<b>30</b>	3,01	<b>30</b>	<b>2,18</b>	<b>30</b>	15,31	<b>30</b>	6,31
	8	25	15	60	<b>60</b>	2,55	<b>60</b>	<b>2,46</b>	<b>60</b>	15,23	<b>60</b>	50,93
	8	25	20	100	<b>100</b>	<b>1,21</b>	<b>100</b>	3,22	<b>100</b>	11,41	<b>100</b>	86,27
<i>B</i>	9	15	5	10	<b>10</b>	3,11	<b>10</b>	<b>2,14</b>	<b>10</b>	12,15	<b>10</b>	4,47
	9	15	10	30	<b>30</b>	<b>1,82</b>	<b>30</b>	3,04	<b>30</b>	43,55	<b>30</b>	198,17
	9	20	5	10	<b>10</b>	2,69	<b>10</b>	2,95	<b>10</b>	15,48	<b>10</b>	<b>1,47</b>
	9	20	10	30	<b>30</b>	<b>4,31</b>	<b>30</b>	7,74	<b>30</b>	73,53	<b>30</b>	145,69
	9	20	15	60	<b>60</b>	<b>2,47</b>	<b>60</b>	8,75	<b>60</b>	63,42	<b>60</b>	532,38
	9	25	5	10	<b>10</b>	<b>1,22</b>	<b>10</b>	5,16	<b>10</b>	31,11	<b>10</b>	1,68
	9	25	10	30	<b>30</b>	<b>5,57</b>	<b>30</b>	6,73	<b>30</b>	115,65	<b>30</b>	62,43
	9	25	15	60	<b>60</b>	<b>5,78</b>	<b>60</b>	18,80	<b>60</b>	113,12	<b>60</b>	412,17
	9	25	20	100	<b>100</b>	<b>2,50</b>	<b>100</b>	11,20	<b>100</b>	90,03	99	1087,51
<i>C</i>	15	15	5	10	<b>9</b>	905,82	8	1225,59	1	960,24	2	491,55
	15	15	10	30	27	642,24	<b>28</b>	840,36	0	-	1	573,47
	15	20	5	10	<b>9</b>	1113,47	7	1363,13	0	-	4	1882,45
	15	20	10	30	<b>17</b>	480,36	12	1383,68	0	-	2	2101,17
	15	20	15	60	<b>43</b>	725,83	35	1275,42	0	-	4	1779,36
	15	25	5	10	4	1979,25	1	794,52	1	182,76	<b>8</b>	657,53
	15	25	10	30	<b>16</b>	816,96	8	699,41	1	2379,97	2	1376,73
	15	25	15	60	<b>30</b>	886,45	19	1637,43	0	-	2	1952,06
	15	25	20	100	<b>68</b>	534,49	50	1212,19	0	-	0	-
<i>D</i>	20	15	5	10	0		0		0	-	1	185,74
	20	15	10	20	10	567,36	3	1686,84	0	-	0	-
	20	20	5	10	0		0		0	-	0	-
	20	20	10	10	6	541,66	0		0	-	0	-
	20	20	15	30	11	245,65	0		0	-	0	-
	20	25	5	10	0		0		0	-	0	-
	20	25	10	10	5	99,07	3	512,74	0	-	0	-
	20	25	15	40	10	0,99	0		0	-	0	-
	20	25	20	40	26	91,82	3	1300,58	0	-	0	-
	25	15	10	10	6	996,28	2	36,89	0	-	0	-
	25	20	10	10	1	80,49	0		0	-	0	-
	25	20	15	10	0		0		0	-	0	-
	25	25	10	10	2	795,24	1	426,92	0	-	0	-
	25	25	15	10	10	14,60	0		0	-	0	-
	25	25	20	30	21	204,81	1	3217,48	0	-	0	-
<i>E</i>	10	10	4	10	<b>10</b>	3,34	<b>10</b>	4,44	<b>10</b>	6,62	<b>10</b>	<b>2,43</b>
	10	10	5	10	<b>10</b>	<b>1,79</b>	<b>10</b>	3,07	<b>10</b>	69,31	<b>10</b>	263,78
	10	10	6	10	<b>10</b>	<b>0,38</b>	<b>10</b>	1,99	<b>10</b>	156,11	8	2328,55
	10	10	7	10	<b>10</b>	<b>0,07</b>	<b>10</b>	1,06	<b>10</b>	271,27	2	3037,55
	15	20	6	10	<b>8</b>	901,99	6	1154,33	1	43,42	3	1779,86
	15	20	8	10	<b>7</b>	768,90	4	838,22	0	-	0	-
	15	20	10	10	<b>9</b>	363,83	6	154,15	0	-	0	-
	15	20	12	10	<b>9</b>	180,81	4	1134,79	0	-	0	-



TABELA 3.9 – Resultados para instâncias dos Grupos *A* e *B* de Catanzaro *et al.* (2015).

Grupo	<i>N</i>	<i>M</i>	<i>C</i>	<i>i</i>	MF-PMTF		TD		LSS		CGL	
					<i>O</i>	<i>T</i>	<i>O</i>	<i>T</i>	<i>O</i>	<i>T</i>	<i>O</i>	<i>T</i>
<i>A</i>	10	10	4	10	<b>10</b>	<b>2,15</b>	<b>10</b>	3,41	<b>10</b>	19,23	<b>10</b>	15,15
	10	10	5	10	<b>10</b>	<b>0,95</b>	<b>10</b>	2,64	<b>10</b>	75,57	<b>10</b>	390,34
	10	10	6	10	<b>10</b>	<b>0,21</b>	<b>10</b>	1,39	<b>10</b>	152,39	<b>10</b>	2207,79
	10	10	7	10	<b>10</b>	<b>0,07</b>	<b>10</b>	1,12	<b>10</b>	233,29	2	2881,08
<i>B</i>	15	20	6	10	<b>10</b>	1467,82	6	1376,85	0	-	2	1073,07
	15	20	8	10	<b>8</b>	1092,00	8	1665,75	0	-	0	-
	15	20	10	10	<b>9</b>	452,18	6	698,36	0	-	0	-
	15	20	12	10	<b>10</b>	6,89	5	793,61	0	-	0	-



FIGURA 3.7 – Gráfico do tipo *performance profile* considerando os modelos FM-PMTF-1D, TD, LSS, e CGL.

### 3.4 Considerações finais

Neste capítulo, vários modelos de fluxo *multicommodity* foram desenvolvidos para o PMTF e vários testes computacionais foram realizados com estes modelos e os modelos da literatura. Os vários modelos de fluxo desenvolvidos foram analisados, demonstrando melhor resultado aquele que possui menos variáveis e menos restrições, ou seja, o modelo FM-PMTF-1D. Quando comparado aos outros modelos da literatura, TD, LSS e CGL, sua performance também foi superior tanto em quantidade de instâncias resolvidas na otimalidade quanto em tempo de execução. Observou-se também que o valor da relaxação linear dos modelos de fluxo, é em geral, melhor que os valores das relaxações lineares dos modelos da literatura.

# 4 Métodos exatos e heurísticos aplicados ao PMTF

Neste capítulo, apresentam-se as matheurísticas desenvolvidas para solucionar o PMTF baseadas nos modelos desenvolvidos: FM-PMTF-1, FM-PMTF-1A, FM-PMTF-1B, FM-PMTF-1C e FM-PMTF-1D.

Inicialmente uma breve definição de matheurística é apresentada na Seção 4.1. Nas Seções seguintes realiza-se uma descrição das metaheurísticas utilizadas na matheurística proposta: *Relax-and-Fix*, *Local Branching*, Método de Descida em Vizinhança Variável (do inglês *Variable Neighborhood Descent*, VND) e Busca por Proximidade (do inglês *Proximity Search*, PS). Nas Seções 4.5 a 4.10 são apresentados os métodos desenvolvidos. Por fim, na Seção 4.11, são apresentados os resultados obtidos pelos testes computacionais.

## 4.1 Definição de Matheurísticas

Matheurísticas são heurísticas híbridas que integram métodos heurísticos (ou metaheurísticas) com técnicas de otimização matemática. A principal motivação para este tipo de hibridização é explorar características complementares de diferentes estratégias de otimização, ou seja, os métodos híbridos se beneficiam da sinergia entre as técnicas hibridizadas. A chave para alcançar o melhor desempenho na solução de muitos problemas de otimização difíceis está na escolha de uma combinação adequada de conceitos algorítmicos complementares (BLUM *et al.*, 2011; BOSCHETTI *et al.*, 2009).

Há duas maneiras usuais de hibridizar técnicas heurísticas e de otimização matemática: combinação colaborativa e combinação integrativa (PUCHINGER; RAIDL, 2005). Em um método colaborativo os algoritmos trocam informações entre si, sendo executados sequencialmente, de forma intercalada ou em paralelo. Já em um método integrativo, um dos algoritmos é subordinado ao outro, se tornando um novo método. Na Figura 4.1 pode-se ver um esquema de classificação de matheurísticas. A proposta desta seção se baseia na combinação integrativa, inserindo elementos de otimização matemática em metaheurísticas.

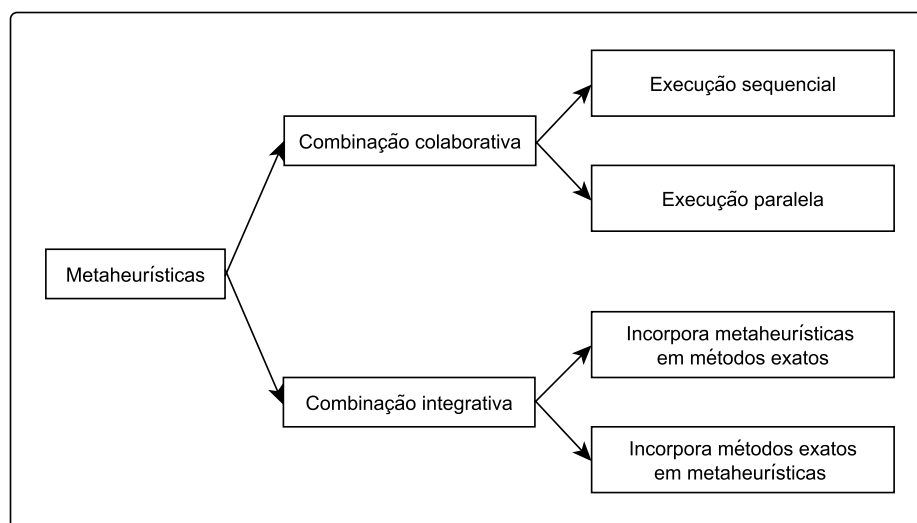


FIGURA 4.1 – Esquema de classificação de metaheurísticas (PUCHINGER; RAIDL, 2005).

## 4.2 Descrição das metaheurísticas utilizadas

Nesta seção apresenta-se brevemente as metaheurísticas utilizadas na metaheurística proposta na seção 4.5.

### 4.2.1 *Relax-and-fix*

A heurística *Relax-and-Fix* (R&F) é descrita em Wolsey (1998) e consiste em decompor o problema original em vários subproblemas menores disjuntos, sendo que a solução do último subproblema é um limitante superior para o problema original (problema de minimização).

Esta heurística tem sido amplamente aplicada em diversos tipos de problemas de otimização combinatória, de forma pura ou híbrida, entre eles os problemas de planejamento da produção: planejamento da produção de refrigerantes (FERREIRA *et al.*, 2010), dimensionamento e programação de lotes em uma fábrica para compostos de ração animal (TOSO *et al.*, 2009), dimensionamento e programação de lotes em uma indústria de fundição (ARAUJO *et al.*, 2008). Outros autores também utilizam esta abordagem em seus trabalhos: Kelly e Mann (2004), Albareda-Sambola *et al.* (2013), Baena *et al.* (2015), Cunha *et al.* (2019).

Considere  $Q$  o conjunto de variáveis binárias de um problema de otimização inteiro misto  $P$ . Este conjunto pode ser particionado em  $r$  subconjuntos  $Q_i$ ,  $i = 1, 2, \dots, r$ , de tal forma que  $Q_1 \cup Q_2 \cup \dots \cup Q_r = Q$  e  $Q_i \cap Q_j = \emptyset$ ,  $i = j = 1, 2, \dots, r$ ,  $i \neq j$ . O método R&F procede de forma iterativa. Primeiro, todas as variáveis pertencentes aos conjuntos

$Q_i, i = 2, 3, \dots, r$  são relaxadas linearmente, e a integralidade das variáveis pertencentes ao conjunto  $Q_1$  é mantida. Desta forma, obtém-se o subproblema  $P_1$ . Ao resolver  $P_1$ , por algum método de resolução específico, obtém-se uma solução cujas variáveis em  $Q_1$  tem valor inteiro. No próximo passo, esta solução inteira obtida no passo anterior é fixada, as variáveis pertencentes aos conjuntos  $Q_i, i = 3, 4, \dots, r$  são relaxadas linearmente e a integralidade das variáveis em  $Q_2$  é mantida. Assim, tem-se o subproblema  $P_2$ . Ao resolver  $P_2$  obtém-se valores inteiros nas variáveis pertencentes a  $Q_2$ . O método continua até o conjunto  $Q_r$  ser integralizado ou uma solução inteira ser obtida. O pseudocódigo da heurística R&F é apresentado no Algoritmo 2.

O número  $r$  de partições determina a quantidade de subproblemas a serem resolvidos e a quantidade de iterações da heurística R&F. A forma de particionar as variáveis determina o sucesso deste método, pois influencia diretamente na dificuldade em resolver os subproblemas (WOLSEY, 1998).

---

**Algorithm 2** Pseudocódigo da Heurística *Relax-and-Fix*

---

- 1: Defina as partições  $Q_i, i = 1, \dots, r$ .
  - 2: Relaxe as variáveis dos conjuntos  $Q_i, i = 1, \dots, r$ .
  - 3:  $i = 0$ . ▷ contador de iterações
  - 4:  $T_{max}$ . ▷ tempo máximo para resolução de cada subproblema
  - 5:  $T_{max}$ . ▷ tempo limite para resolução do problema
  - 6: **enquanto** ( $i < r$ ) e ( $tempo \leq T_{max}$ ) **faça**
  - 7:      $i = i + 1$ .
  - 8:     Integralize as variáveis do conjunto  $Q_i$ .
  - 9:     Resolva o subproblema  $P_i$  com tempo limite  $T_{max}$ .
  - 10:     Fixe os valores das variáveis no conjunto  $Q_1 \cup \dots \cup Q_i$ .
  - 11:     Se uma solução inteira ou infactível for obtida então PARE.
  - 12: **fim enquanto**
- 

### 4.2.2 *Local Branching*

*Local Branching* é um método exato para resolver problemas de programação inteira mista (MIP) e foi proposto por Fischetti e Lodi (2003). Trata-se de um mecanismo de fixação flexível de variáveis cuja estratégia para obter a solução ótima é através de buscas locais nas vizinhanças das soluções factíveis obtidas. Estas buscas locais são realizadas em vizinhanças determinadas por cortes no modelo inteiro, chamados cortes *local branching*.

Considere um modelo MIP genérico:

$$(P) \text{ Min } c^T x \tag{4.1}$$

sujeito a:

$$Ax \geq b, \tag{4.2}$$

$$x_j \in \{0, 1\}, \quad \forall j \in \mathcal{B} \neq \emptyset, \tag{4.3}$$

$$x_j \geq 0, \text{ inteiro}, \quad \forall j \in \mathcal{G}, \tag{4.4}$$

$$x_j \geq 0, \quad \forall j \in \mathcal{C}, \tag{4.5}$$

em que  $\mathcal{B}$  corresponde ao conjunto de índices das variáveis binárias, e  $\mathcal{G}$  e  $\mathcal{C}$  correspondem aos índices das variáveis inteiras e contínuas, respectivamente. O conjunto  $\mathcal{B}$  é sempre não vazio e admite-se que ao fixar as variáveis binárias simplifica-se fortemente o problema resultante (FISCHETTI; LODI, 2003).

Dada uma solução factível  $\bar{x}$ , denominada *solução referência*, define-se como *suporte binário* de  $\bar{x}$ , o conjunto  $\bar{S} = \{j \in \mathcal{B} \mid \bar{x}_j = 1\}$ , ou seja, conjunto formado pelas variáveis com valor 1. Para um dado valor inteiro  $k$  defini-se uma vizinhança  $k$ -OPT  $\mathcal{N}(\bar{x}, k)$  de  $\bar{x}$  como o conjunto de soluções factíveis de (P) satisfazendo a restrição adicional *local branching*:

$$\Delta(x, \bar{x}) = \sum_{j \in \bar{S}} (1 - x_j) + \sum_{j \in \mathcal{B} \setminus \bar{S}} x_j \leq k, \tag{4.6}$$

cujo lado esquerdo contabiliza a quantidade de variáveis que terão seu valor alterado de 1 para 0, ou de 0 para 1, respectivamente. A restrição (4.6) impõe uma distância de Hamming de  $k$  unidades sobre os vizinhos factíveis de  $\bar{x}$ . A distância de Hamming entre dois vetores  $x$  e  $\bar{x}$ ,  $D(x, \bar{x})$ , pode ser definida de acordo com a equação (4.7):

$$D(x, \bar{x}) = |j \in N; x_j \neq \bar{x}_j|, \tag{4.7}$$

em que  $|\cdot|$  representa a cardinalidade do conjunto citado, e  $N$  o conjunto dos índices das variáveis. A ideia do método é utilizar as restrições de *local branching* (4.6) dentro de um método enumerativo para (P), em que cada subproblema pode ser resolvido por um resolvidor exato de modelos MIP, como, por exemplo, CPLEX, Gurobi, Xpress, dentre outros. Dada uma solução factível  $\bar{x}$ , o espaço de solução associado ao nó corrente pode ser particionado através das duas disjunções  $\Delta(x, \bar{x}) \leq k$  e  $\Delta(x, \bar{x}) \geq k + 1$ .

Na Figura 4.2 apresenta-se a ideia básica do método, em que os triângulos representam as subárvores exploradas pelo resolvedor exato de modelos de PIM. Admite-se que há uma solução factível  $\bar{x}^1$  no nó 0, que pode ser obtida por alguma heurística de construção de soluções, alguma metaheurística ou executando o resolvedor até obter uma solução factível (a primeira solução factível, ou a solução obtida depois de um pequeno tempo fixado de execução). No nó 2, busca-se uma solução melhor na vizinhança  $\mathcal{N}(\bar{x}^1, k)$ , ou seja, acrescenta-se a restrição  $\Delta(x, \bar{x}^1) \leq k$  a (P) e resolve-se o subproblema resultante. A solução  $\bar{x}^2$  do subproblema é melhor que a solução  $\bar{x}^1$ , e então esta se torna a solução incumbente. O esquema é reaplicado ao nó 3 cuja exploração da vizinhança  $\mathcal{N}(\bar{x}^2, k) \setminus \mathcal{N}(\bar{x}^1, k)$ , ou seja, a adição dos cortes  $\Delta(x, \bar{x}^1) \geq k + 1$  e  $\Delta(x, \bar{x}^2) \leq k$ , produz a solução  $\bar{x}^3$ , que se torna a solução incumbente. Neste exemplo, o nó 4 produz um subproblema que não retorna uma solução melhor. Neste caso, a adição do corte  $\Delta(\bar{x}^3, k) \geq k + 1$  atinge o nó 5 que passa a ser explorado pelo resolvedor. A vantagem deste método é a atualização frequente da solução incumbente, ou seja, espera-se encontrar rapidamente soluções melhores até o ponto em que *local branching* não pode mais ser aplicado (nó 5, no exemplo), onde o resolvedor conclui a enumeração.

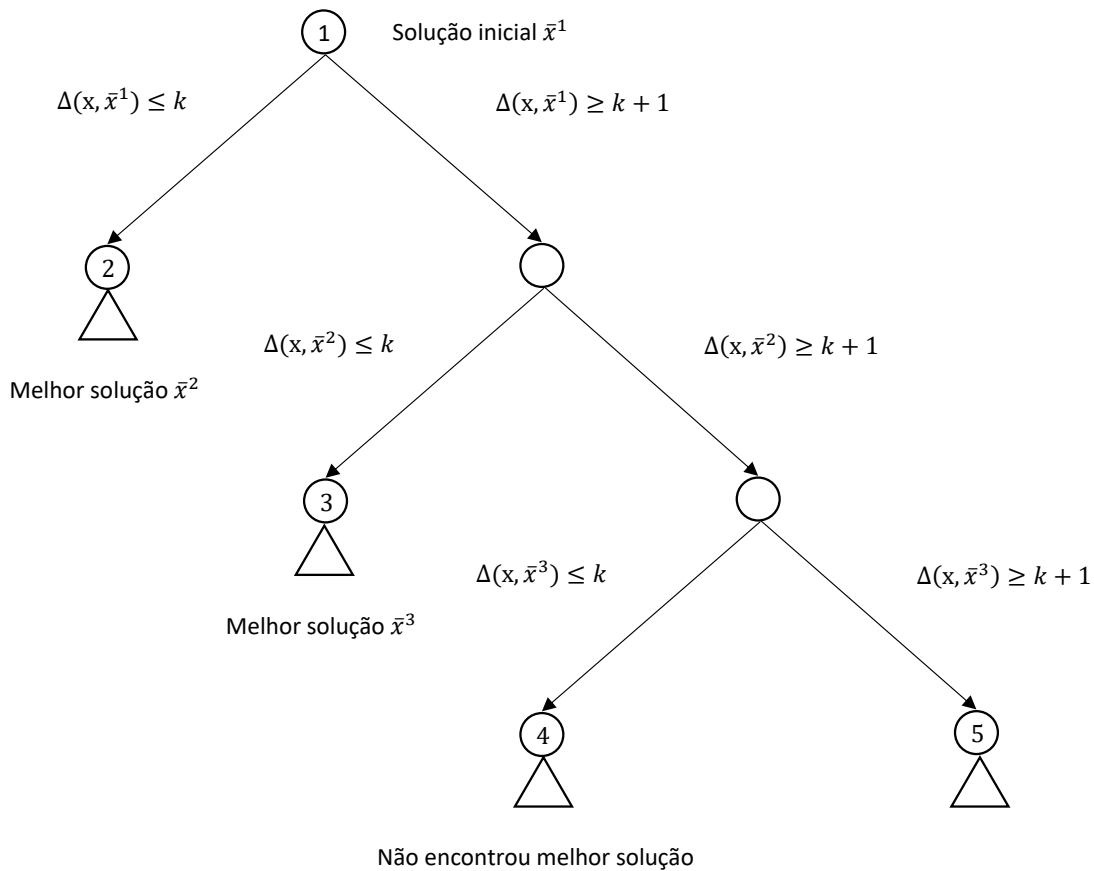


FIGURA 4.2 – Estrutura básica do Método *Local Branching* (FISCHETTI; LODI, 2003).

A escolha do valor do parâmetro  $k$  é muito importante, pois ele define o tamanho

da vizinhança a ser percorrida na busca local. Este parâmetro, geralmente é fixo, e deve ser suficientemente pequeno para que os subproblemas sejam resolvidos em tempo computacional aceitável, mas grande o suficiente para conter soluções melhores que a atual. De acordo com Fischetti e Lodi (2003) bons valores para  $k$  se encontram no intervalo  $[10, 20]$ .

### 4.2.3 Método de Descida em Vizinhança Variável

O Método de Descida em Vizinhança Variável (do inglês *Variable Neighborhood Descent*, VND) é uma metaheurística proposta por Mladenović e Hansen (1997), Mladenović e Hansen (1999) e consiste em explorar o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança. O método sempre retorna à primeira estrutura quando uma melhor solução é encontrada. A ideia é encontrar uma solução que seja ótima para todas as estruturas de vizinhança consideradas.

Seja  $f$  a função a ser minimizada e  $N^{(k)}(s)$  a  $k$ -ésima estrutura de vizinhança de uma solução  $s$ . O Algoritmo 3 apresenta o pseudocódigo da Metaheurística VND.

A solução inicial  $s_0$ , a quantidade de estruturas de vizinhança, e as inicializações da solução corrente e do tipo de estrutura de vizinhança são definidos nos passos 1 a 4, respectivamente. Uma busca é realizada em todas as vizinhanças até que seja obtida a melhor solução para todas as estruturas (passos 5 a 12). Em cada estrutura de vizinhança  $k$  (linha 5), encontra-se o melhor vizinho (linha 6). Sempre que uma melhor solução é encontrada (linha 7), retorna-se à primeira estrutura de vizinhança (linha 9) atualizando-se a solução corrente (linha 8), garantindo que a solução final obtida seja a melhor para todas as estruturas de vizinhança. Quando uma melhor solução não é encontrada usando a estrutura de vizinhança de  $k$ , a estrutura muda para a vizinhança  $k + 1$  (linha 10). A solução encontrada é retornada na linha 13.

---

#### Algorithm 3 Pseudocódigo da Metaheurística VND

---

```

1:  $s_0$                                 ▷ Solução inicial
2:  $r$                                     ▷ Quantidade de estruturas de vizinhança
3:  $s = s_0$                                 ▷ Solução corrente
4:  $k = 1$                                 ▷ Tipo de estrutura de vizinhança
5: enquanto ( $k < r$ ) faça
6:   Encontre o melhor vizinho  $s'$  na vizinhança  $N^{(k)}(s)$ ;
7:   se  $f(s') < f(s)$  então
8:      $s = s'$ .
9:      $k = 1$ .
10:  senão  $k = k + 1$ 
11:  fim se
12: fim enquanto
13: retorne  $s$ 

```

---

### 4.3 Busca por proximidade

Fischetti e Monaci (2014) propuseram uma metaheurística denominada Busca por Proximidade (BP, do inglês *Proximity Search*). Esta metaheurística utiliza a própria formulação do problema a ser resolvido para melhorar uma solução factível inicial, e pertence à *Large Neighborhood Search* (LNS) (PISINGER; ROPKE, 2010). Os métodos de busca baseados em LNS exploram vizinhanças complexas por meio de heurísticas.

Considere o problema de programação inteiro misto, possivelmente não linear, MIP, dado pelas expressões (4.8)-(4.10).

$$(MIP) \text{ Min } f(x) \tag{4.8}$$

sujeito a:

$$g(x) \leq 0, \tag{4.9}$$

$$x_j \in \{0, 1\}, \quad \forall j \in J, \tag{4.10}$$

em que  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , e  $J \subseteq N := \{1, \dots, n\}$ ,  $J \neq \emptyset$ , o conjunto dos índices das variáveis binárias.

A metaheurística BP inicia com uma solução factível  $\tilde{x}$ , e adiciona a restrição de corte 4.11

$$f(x) \leq f(\tilde{x}) - \theta \tag{4.11}$$

ao MIP, sendo  $\theta > 0$  uma dada tolerância de corte. Neste momento, pode-se modificar a função objetivo a fim de encontrar heurísticamente, melhores soluções factíveis. Uma opção natural é usar uma função de proximidade que penalize uma solução  $x$  de acordo com sua distância de  $\tilde{x}$ , por exemplo, tomando a distância de Hamming

$$\Delta(x, \tilde{x}) = \sum_{j \in J: \tilde{x}_j=0} x_j + \sum_{j \in J: \tilde{x}_j=1} (1 - x_j). \tag{4.12}$$

A ideia deste método é buscar na vizinhança de uma dada solução,  $\tilde{x}$ , uma nova solução, que tenha valor objetivo no máximo igual a  $f(\tilde{x}) - \theta$ .

O Algoritmo 4 apresenta o pseudocódigo da Metaheurística BP. No passo 1, uma solução inicial  $\tilde{x}$  é definida. Esta solução pode ser encontrada por alguma heurística



de construção, metaheurística, ou até mesmo executando um solver na formulação até encontrar a primeira solução factível ou retornar a solução depois de um tempo limite de execução.

No passo 3, a tolerância de corte  $\theta$  é definida e uma nova restrição de corte é adicionada ao MIP, válida para todas as soluções com valores melhores que o de  $\tilde{x}$ .

No passo 4, a função objetivo  $f(x)$  é substituída pela distância de Hamming entre  $x$  e  $\tilde{x}$ , calculada de acordo com (4.12).

No passo 5 usa-se um solver MIP para encontrar uma nova solução  $x^*$  com  $f(x^*) \leq f(\tilde{x}) - \theta$ , considerando alguma condição de parada (tempo de execução, por exemplo).

Nos passos 6 a 9, a nova solução  $x^*$ , se pôde ser encontrada, é possivelmente melhorada resolvendo um problema inteiro misto em que todas as variáveis com valores binários em  $x^*$  são fixadas em  $x$ , a fim de encontrar a melhor solução na vizinhança induzida por  $\Delta(x, x^*) = 0$ .

No passo 10, o centro da função de distância  $\Delta$  é atualizado caso  $x^*$  tenha sido encontrado, fazendo  $\tilde{x} = x^*$ , e a tolerância de corte  $\theta$  pode ser modificada. O valor de  $\theta$  será modificado, caso não seja possível encontrar  $x^*$ . Neste caso,  $\theta$  terá seu valor diminuído.

---

**Algorithm 4** Pseudocódigo da Metaheurística BP (FISCHETTI; MONACI, 2014)

---

- 1:  $\tilde{x}$  ▷ Solução inicial
  - 2: **enquanto** (critério de parada) **faça**
  - 3:     Adicione a restrição de corte  $f(x) \leq f(\tilde{x}) - \theta$  ao modelo;
  - 4:     Troque  $f(x)$  pela função objetivo de proximidade  $\Delta(x, \tilde{x})$ .
  - 5:     Resolva o modelo resultante até uma condição de parada ser atingida, e seja  $x^*$  a melhor solução factível.
  - 6:     **se** ( $x^*$  pode ser encontrado) **então**
  - 7:         Refina  $x^*$  resolvendo o problema convexo:
  - 8:          $x^* := \arg \min \{f(x) : g(x) \leq 0, x_j = x^*, \forall j \in J\}$ .
  - 9:     **fim se**
  - 10:     Atualize  $\Delta(x, \tilde{x})$  com  $\tilde{x} = x^*$  e/ou  $\theta$ .
  - 11: **fim enquanto**
  - 12: **retorne**  $x^*$ .
- 

## 4.4 Métricas

O PMTF pode ser aproximado por um PCV, considerando um grafo  $G(V, A)$ , em que  $V$  é o conjunto de tarefas,  $A$  é o conjunto de todos os pares de tarefas, e o custo  $l_{ij}$  de cada aresta  $(i, j) \in A$  é um valor subestimado do número de trocas de ferramentas entre as tarefas  $i$  e  $j$  quando estas tarefas são processadas sequencialmente (CRAMA *et al.*, 1994).

Hertz *et al.* (1998) mostraram que o PMTF é equivalente ao PCV apenas quando cada

tarefa requer exatamente  $C$  ferramentas. Como  $|T_i| = C$ ,  $\forall i \in T$ , o número exato de trocas de ferramentas entre as tarefas  $i$  e  $j$  é dado pela expressão (4.13), que corresponde a subtrair da capacidade, o número de ferramentas comuns entre as tarefas  $i$  e  $j$ .

$$l_{ij}^1 = C - |T_i \cap T_j|. \quad (4.13)$$

Quando nem todas as tarefas requerem exatamente  $C$  ferramentas, a expressão (4.13) se torna um limitante superior ao número de trocas de ferramentas entre as tarefas  $i$  e  $j$ .

A expressão (4.14) também é um limitante superior para o número de trocas de ferramentas entre duas tarefas e é dada pela diferença entre a união e a interseção dos conjuntos de ferramentas necessárias para processar duas tarefas diferentes.

$$l_{ij}^2 = |T_i \cup T_j| - |T_i \cap T_j|. \quad (4.14)$$

A expressão (4.15), proposta por Crama *et al.* (1994), é um limitante inferior para o número de trocas de ferramentas entre duas tarefas. Este limitante é mais forte que o limitante  $|T_i \cup T_j| - C$  proposto por Tang e Denardo (1988a) uma vez que nunca é negativo.

$$l_{ij}^3 = \max\{0, |T_i \cup T_j| - C\}. \quad (4.15)$$

As três métricas,  $l^1$ ,  $l^2$  e  $l^3$ , apenas consideram as interações entre duas tarefas processadas sequencialmente e não consideram as ferramentas que já estavam na máquina ou as que serão inseridas. Assim, Hertz *et al.* (1998) propuseram duas novas métricas  $l^4$  e  $l^5$ , aprimorando as métricas  $l^3$  e  $l^2$ , respectivamente. Para tanto, define-se a função  $\Lambda(i, j)$  abaixo:

$$\Lambda(i, j) = \sum_{k \in T_i \cup T_j} |C_{ij}^k|, \quad (4.16)$$

em que  $C_{ij}^k = \{p \in J_k | p \neq i, p \neq j\}$ ,  $i, j \in J$ ,  $k \in T_i \cup T_j$ , ou seja, os números das tarefas, a menos das tarefas  $i$  e  $j$ , que requerem a ferramenta  $k \in T_i \cup T_j$ . Tomando  $\theta \in [0, 1]$ , a métrica  $l^4$  é dada pela expressão (4.17).

$$l_{ij}^4 = \max \left\{ 0, |T_i \cup T_j| - \left[ \theta \frac{\Lambda(i, j)}{(N-2)|T_i \cup T_j|} \right] C \right\} \quad (4.17)$$

Observe que  $|C_{ij}^k| \leq N-2$ , pois  $i, j \notin C_{ij}^k$  e, assim,  $\sum_{k \in T_i \cup T_j} |C_{ij}^k| \leq |T_i \cup T_j|(N-2)$ .

Logo,  $\Lambda(i, j) \leq |T_i \cup T_j|(N-2)$  e, portanto, como  $\theta \in [0, 1]$ ,  $\theta \frac{\Lambda(i, j)}{(N-2)|T_i \cup T_j|} \leq 1$ , ou seja, subtrai-se um valor menor que  $C$  de  $|T_i \cup T_j|$ .

A última métrica,  $l^5$ , é dada pela expressão (4.18). O fator  $\frac{C+1}{C}$  pode variar no intervalo  $(1, 2]$ , dando um peso maior ao termo  $|T_i \cup T_j|$  caso a capacidade seja pequena, ou seja, se mais trocas forem prováveis. O último fator é semelhante ao usado na métrica  $l^4$  e assume valor pelo menos igual a 1, se tornando maior quando as ferramentas de  $|T_i \cup T_j|$  forem raramente utilizadas. O valor 0.5 é usado para evitar divisão por 0 quando  $\Lambda(i, j) = 0$ .

$$l_{ij}^5 = \left( \left[ \frac{C+1}{C} \right] |T_i \cup T_j| - |T_i \cap T_j| \right) \left[ \frac{(N-2)|T_i \cup T_j|}{\max\{\Lambda(i, j), 0.5\}} \right] \quad (4.18)$$

A fim de testar quais das métricas fornecem os melhores valores de limitantes superiores para o PMTF, nesta tese usou-se o conjunto de instâncias de Catanzaro *et al.* (2015). O PCV correspondente foi resolvido pelo CPLEX 12.6.3 usando-se as restrições de eliminação de subsequência propostas por Desrochers e Laporte (1991). O algoritmo KTNS foi utilizado para encontrar o número de trocas correspondente.

Na Tabela 4.1 mostra-se as instâncias testadas divididas em conjuntos contendo 10 exemplares cada. A quantidade de tarefas, a quantidade de ferramentas e a capacidade da máquina são mostradas nas colunas N, M e C, respectivamente. Em cada coluna Métrica  $i$ ,  $i = 1, \dots, 5$  mostra-se as soluções médias relativas a cada conjunto de instâncias considerando as métricas  $l^i$ ,  $i = 1, \dots, 5$ , respectivamente. Por fim, a linha “Total Melhor” retorna a quantidade de vezes em que cada métrica apresentou resultados melhores.

Analisando a Tabela 4.1, vê-se uma clara superioridade das métricas 4 e 5 em relação às outras métricas. Dentre estas duas métricas, a métrica 5 teve um resultado melhor em 8 conjuntos, contra 7 da métrica 4, sendo a escolhida para ser utilizada nos demais testes.

Conjuntos	N	M	C	Métrica 1	Métrica 2	Métrica 3	Métrica 4	Métrica 5
A1	10	10	4	9.80	9.50	10.30	9.50	<b>9.40</b>
A2	10	10	5	6.50	6.70	7.90	<b>6.30</b>	6.50
A3	10	10	6	<b>4.40</b>	4.70	5.90	4.70	4.50
A4	10	10	7	<b>3.00</b>	<b>3.00</b>	4.20	<b>3.00</b>	<b>3.00</b>
B1	15	20	6	23.70	22.90	25.80	<b>22.80</b>	23.50
B2	15	20	8	16.60	<b>15.80</b>	20.40	15.90	16.00
B3	15	20	10	12.30	11.40	17.00	11.50	<b>11.20</b>
B4	15	20	12	8.70	<b>7.80</b>	11.10	7.90	<b>7.80</b>
C1	30	40	15	100.00	96.90	106.90	<b>95.70</b>	96.50
C2	30	40	17	83.00	79.80	95.90	<b>78.10</b>	79.00
C3	30	40	20	61.40	60.40	82.10	<b>58.90</b>	60.00
C4	30	40	25	39.00	37.80	50.90	36.50	<b>36.30</b>
D1	40	60	20	205.00	<b>199.50</b>	211.00	199.90	200.70
D2	40	60	22	178.40	175.60	194.90	175.60	<b>175.40</b>
D3	40	60	25	148.10	146.40	182.50	146.10	<b>145.60</b>
D4	40	60	30	110.60	108.50	144.10	<b>107.90</b>	<b>107.90</b>
Total Melhor				2	4	0	7	8

TABELA 4.1 – Soluções médias encontradas pelas métricas 1 a 5 nas instâncias de Cantanzaro *et al.* (2015).

## 4.5 VNDLB

A matheurística proposta, denominada VNDLB, é uma combinação do VND com a ideia utilizada no *Local Branching*. O objetivo é utilizar a própria formulação como busca local.

Para um número inteiro  $r$ , a vizinhança  $r$ -flip  $NB_r(\bar{x})$  é definida como

$$NB_r(\bar{x}) = \{x \in \{0, 1\}^n \mid |D(x, \bar{x})| \leq r\}, \quad (4.19)$$

ou seja,  $NB_r(\bar{x})$  é o conjunto de soluções obtidas a partir de  $\bar{x}$  que diferem em até  $r$  posições (GAO *et al.*, ; UMETANI *et al.*, 2018; YAGIURA; IBARAKI, 2001). Observe que a vizinhança  $NB_r(\bar{x})$  é a mesma obtida pela restrição de *local branching* (4.6).

Yagiura e Ibaraki (2001) observaram que o tempo computacional para pesquisar  $NB_r$  aumenta exponencialmente a medida que  $r$  aumenta, pois  $|NB_r| = O(n^r)$ . Assim, torna-se inviável pesquisar  $NB_r$  quando  $n$  e o tamanho  $r$  da vizinhança são grandes.

A matheurística VNDLB baseia-se em resolver um problema de otimização combinatoria, usando sua própria formulação matemática como estrutura de vizinhança, ao adicionar a restrição *local branching*, uma vez que esta corresponde a pesquisar a vizinhança  $NB_r(\bar{x})$ , para o mesmo valor de  $r$ . Ao invés de realizar a busca local (exaustiva)

em uma dada solução factível, VNDLB determina qual movimento (se existir, de acordo com a quantidade possível de trocas,  $r$ ) deve ser realizado nesta solução factível para encontrar uma melhor solução.

O pseudocódigo da metaheurística VNDLB é apresentado no Algoritmo 5. Por ser uma heurística de melhoria, ela necessita de uma solução inicial. Esta solução inicial (linha 2) pode ser obtida, por exemplo, resolvendo-se o PCV correspondente usando a métrica  $l^5$  dada na seção 4.4, ou então resolvendo-se a heurística R&F apresentada na seção 4.2.1. Em seguida, aplica-se o movimento Swap(1,1) nesta solução inicial obtendo a sequência  $s$  com número de trocas de ferramentas  $f$ , calculado pelo algoritmo KTNS (linhas 3 e 4). Na linha 5 inicializa-se  $k = 2$ , a distância de Hamming. Nas linhas 6 a 16 executa-se a busca local proposta dentro de um método VND, resolvendo o modelo MF-PMTF com a restrição de *local branching* dentro do tempo limite de execução preestabelecido. Se um valor melhor for encontrado, atualiza-se os valores correntes  $s$  e  $f$  (linhas 10 e 11), e volta-se à primeira estrutura de vizinhança  $k = 2$  (linha 12). Se uma solução melhor não pôde ser encontrada, então muda-se a estrutura de vizinhança, aumentando a distância de Hamming (linha 14).

---

**Algorithm 5** Pseudocódigo da Metaheurística VNDLB
 

---

```

1: Resultado: sequência factível  $s_{best}$  com número de troca de ferramentas correspon-
   dente  $f_{best}$ .
2:  $s_0$  ▷ sequência inicial de tarefas
3:  $s$  ▷ sequência de tarefas obtida aplicando Swap(1,1)
4:  $f$  ▷ quantidade de trocas de ferramentas da sequência  $s$ 
5:  $k = 2$ ;
6: enquanto  $k \leq 12$  faça
7:   Resolva o modelo MF-PMTF dentro do tempo limite estabelecido com restrição
   de local branching considerando a sequência  $s$ ;
8:   Seja  $s_{best}$  a sequência obtida no passo anterior com valor objetivo  $f_{best}$ ;
9:   se  $f_{best} < f$  então
10:      $s = s_{best}$ ;
11:      $f = f_{best}$ ;
12:      $k = 2$ ;
13:   senão
14:      $k = k + 2$ ;
15:   fim se
16: fim enquanto

```

---

O movimento Swap(1,1) consiste em alterar a ordem de produção da tarefa  $i$  com a ordem de produção da tarefa  $j$  na sequência  $s$ . Exemplos de aplicação do movimento Swap(1,1) são mostrados na Figura 4.3. Em 1) há um exemplo de uma sequência de processamento de 7 tarefas cuja ordem é 1 – 2 – 3 – 4 – 5 – 6 – 7. Em 2) há a sequência 1 – 2 – 6 – 4 – 5 – 3 – 7 obtida aplicando Swap(1,1) às tarefas 3 e 6. Em 3) há a sequência 1 – 2 – 4 – 3 – 5 – 6 – 7 obtida aplicando Swap(1,1) às tarefas 3 and 4.

Por motivo de tempo computacional limitado, o movimento Swap(1,1) não foi aplicado à todas as posições da sequência. A tarefa na posição  $k$ , foi trocada com as tarefas das posições  $k + 1, \dots, N$ . O objetivo em utilizar este movimento é acelerar o tempo computacional da VNDLB. Observe que VNDLB pode ser executada sem este movimento.

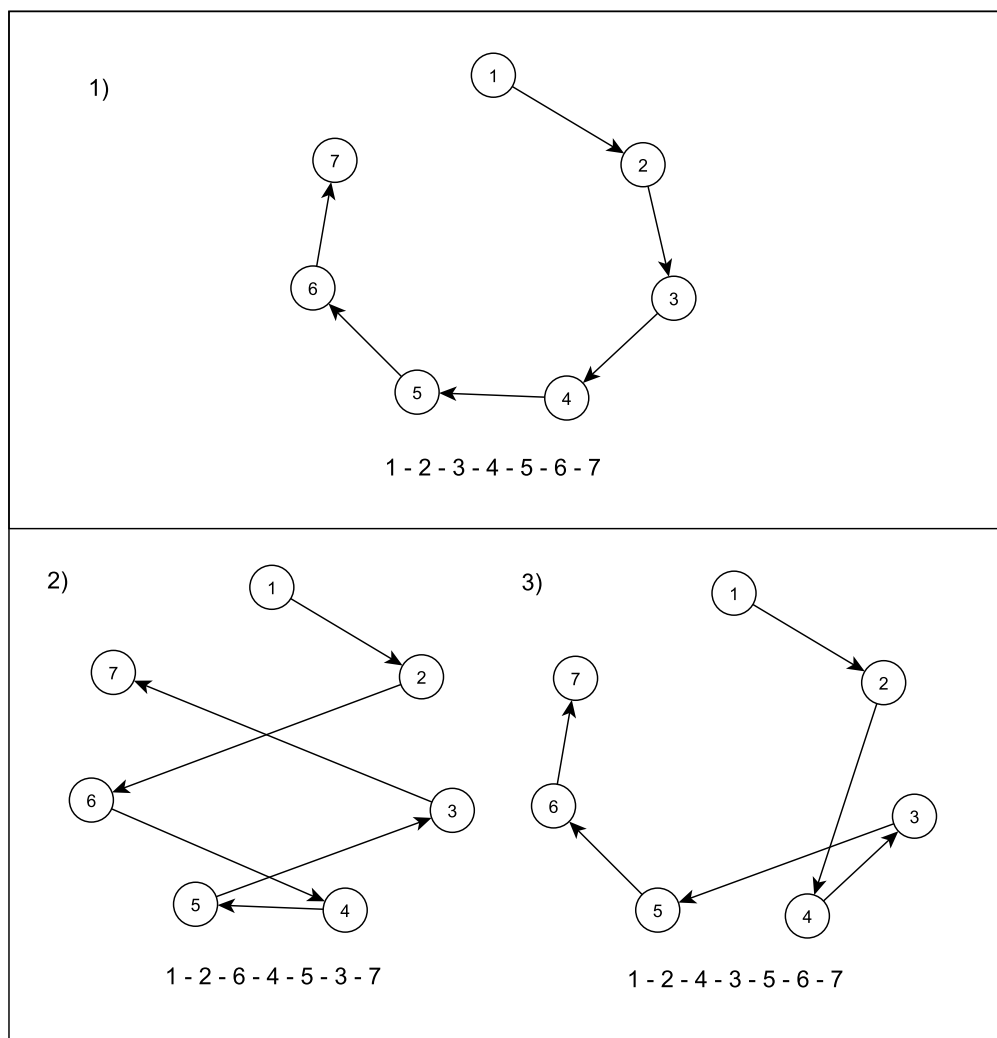


FIGURA 4.3 – Exemplos de aplicação do Swap(1,1). Em 1) um exemplo de uma sequência de processamento de 7 tarefas. Em 2) e 3) sequências obtidas por Swap(1,1) a partir de 1).

## 4.6 Relaxação Lagrangeana

O método de solução via relaxação Lagrangeana foi introduzida por Held e Karp (1970), Held e Karp (1971). Esta técnica tem sido largamente estudada e aplicada na literatura, como nos trabalhos de Geoffrion (1974), Wolsey (1998), Guignard (2003), e Fukuda (2007).

O modelo a ser relaxado nesta tese via multiplicadores de Lagrange é o modelo MF-PMTF-1D que foi o que obteve melhores resultados, conforme resultados da Seção 3.3.3. Observando a estrutura do modelo proposto, constata-se que o conjunto de restrições (3.87) é o único elo de ligação entre as variáveis de decisão  $x$  e  $y$ , e que se estas restrições não existissem, o problema poderia ser decomposto em dois subproblemas: um problema de designação e um problema de fluxo em rede. Assim, dualizando as restrições (3.87) através dos multiplicadores de Lagrange  $\lambda_{ijt}$ ,  $i, j \in J$ ,  $t \in T_i$ , o subproblema lagrangeano é dado pelas expressões (4.20)-(4.34) a seguir:

$$L(\lambda) = \text{Min } Z_4 = \sum_{t=1}^M \sum_{i=1}^{N-1} y_{i(N+1)t} + \sum_{i=1}^N \sum_{k=1}^N \sum_{t \in T_i} \lambda_{ikt} (x_{ik} - y_{(k-1)kt}) \quad (4.20)$$

sujeito a:

$$\sum_{k=1}^N x_{ik} = 1, \quad i = 1, \dots, N, \quad (4.21)$$

$$\sum_{i=1}^N x_{ik} = 1, \quad k = 1, \dots, N, \quad (4.22)$$

$$\sum_{i=1}^{N-1} y_{0it} + y_{0(N+1)t} = N, \quad t = 1, \dots, M, \quad (4.23)$$

$$y_{01t} - y_{1(N+1)t} - y_{12t} = 0, \quad t = 1, \dots, M, \quad (4.24)$$

$$y_{(i-1)it} + y_{0it} - y_{i(i+1)t} - y_{i(N+1)t} = 0, \quad t = 1, \dots, M, \quad i = 2, \dots, N-1, \quad (4.25)$$

$$y_{(N-1)Nt} - y_{N(N+1)t} = 0, \quad t = 1, \dots, M, \quad (4.26)$$

$$\sum_{i=0}^N y_{i(N+1)t} = N, \quad t = 1, \dots, M, \quad (4.27)$$

$$\sum_{t=1}^M y_{(k-1)kt} = C, \quad k = 2, \dots, N, \quad (4.28)$$

$$\sum_{t=1}^M y_{01t} \leq 2C, \quad (4.29)$$

$$x_{ik} \in \{0, 1\}, \quad i, k = 1, \dots, N, \quad (4.30)$$

$$y_{(i-1)it} \in \{0, 1\}, \quad i = 1, \dots, N+1, \quad t \in T, \quad (4.31)$$

$$y_{i(N+1)t} \in \{0, 1\}, \quad i = 1, \dots, N-1, \quad t \in T, \quad (4.32)$$

$$y_{0it} \in \{0, 1\}, \quad i = 1, \dots, N-1, \quad (4.33)$$

$$y_{0(N+1)t} \in \{1, \dots, N-1\}, \quad t = 1, \dots, M. \quad (4.34)$$

Denote por  $Z_{RL}(\lambda)$  o valor ótimo do problema (4.20) - (4.34) para um dado  $\lambda$ .

**Teorema 4.6.1.** *Se  $\lambda_{ikt} = 0, \forall i, k, t \in T_i$ , então  $Z_{RL}(0) = M - C$ .*

*Demonstração.* Pelos mesmos argumentos utilizados na demonstração do Teorema 3.2.3, conclui-se que  $Z_{RL}(0) \geq M - C$ .

Agora, construa  $x$  e  $y$  da seguinte maneira:

- Para  $i = 1, \dots, N, k = 1, \dots, N$ , se  $i = k$ , então seja  $x_{ik} = 1$ ; caso contrário,  $x_{ik} = 0$ .

Para  $t = 1, 2, \dots, C$ , seja  $y_{(N-1)Nt} = y_{N(N+1)t} = 1$ ;

Restaram  $M - C$  ferramentas. Escolha alguns  $i, i = 1, \dots, k$ , nesta ordem, com  $k \leq N - 1$ , tais que  $y_{i(N+1)t} = 1, t = C + 1, \dots, M$ , em que cada  $t$  corresponde a um único  $i$ .

Construa o fluxo nos arcos  $(i, i + 1)$  e  $(0, i)$ , de acordo com a saída das ferramentas  $t, t = C + 1, \dots, M$ :

$y_{01t} = 1, t = C + 1, \dots, 2C$  e as ferramentas  $t$  que atravessam o arco  $(1, N + 1)$ .

$y_{12t} = 1$ , para todas as ferramentas que atravessam o arco  $(0, 1)$  menos as que atravessam o arco  $(1, N + 1)$ .

$y_{23t} = 1$ , para todas as ferramentas que atravessam o arco  $(1, 2)$  menos as que atravessam o arco  $(2, N + 1)$ . Neste momento, pode-se atravessar ferramentas no arco  $(0, 2)$  correspondentes as ferramentas que estão atravessando algum arco  $(i, N + 1)$  e que ainda não foram inseridas na máquina, de acordo com a capacidade residual deste arco.

Este processo é repetido até que não haja mais ferramentas a ser inserida, as quais deixarão a máquina. Neste momento, começa-se a inserir as ferramentas que estão no arco  $(N - 1, N)$ .

Em todos os outros casos, tome  $y_{ijt} = 0$ .

Claramente,  $x$  e  $y$  são soluções factíveis pois satisfazem as restrições (4.21) - (4.34), e assim,  $Z_{RL} = M - C$ .

Como o problema é de minimização, tem-se que  $Z_{RL}(0) \geq M - C$  e uma solução factível com valor  $M - C$  foi construída, segue que  $Z_{RL}(0) = M - C$ .

□

O algoritmo usado para resolver o problema dual correspondente à relaxação lagrangeana do Modelo MF-PMTF-1D, é o algoritmo subgradiente surrogate que foi proposto por Bragin *et al.* (2014). Este método pode levar a melhorias computacionais significativas em



comparação com o algoritmo de subgradiente tradicional, e diminui o efeito zigzague dos limitantes inferiores (problema de minimização) (BRAGIN *et al.*, 2014).

No Algoritmo 6 apresenta-se o pseudocódigo do algoritmo de subgradiente surrogate proposto para resolver o problema dual Lagrangeano associado ao Modelo MF-PMTF-1D. O multiplicador  $\lambda_{ijt}$  na iteração  $l$  será denotado por  $\lambda_{ijt}^l$ .

Os parâmetros  $U$  e  $r$  são utilizados para atualizar o tamanho do passo e são escolhidos através de testes computacionais (linhas 1 e 2). Nas linhas 3 e 4, os limitantes superior  $UB$  e inferior  $LB$  iniciais são declarados. Na linha 5 é fixado o critério de parada do algoritmo, que corresponde à quantidade máxima de iterações permitida  $max\_iter$ , e na linha 6, declara-se o tamanho do passo  $c^l$  na iteração  $l$ . Nas linhas 7 e 8, o subproblema lagrangeano é resolvido com os multiplicadores de Lagrange iguais a 0, e  $LB$  recebe o valor ótimo. Nas linhas 9 a 30 o algoritmo subgradiente surrogate é aplicado.

Iterativamente, até atingir a quantidade máxima de iterações, o subproblema lagrangeano é resolvido fixando  $\lambda$  aos seus valores correntes, e os parâmetros  $Lag$  e  $Ntrocas$  recebem o valor ótimo e a quantidade de trocas de ferramentas corresponde ao aplicar o algoritmo KTNS na solução  $x^l$  (linhas 10 e 11). Nas linhas 12 a 17, os limitantes inferior e superior são atualizados. Os subgradientes são calculados na linha 19, e o tamanho do passo é atualizado na linha 26 utilizando a atualização dos parâmetros necessários para seu cálculo na linha 24. Isto diminui a oscilação dos valores do limitante inferior. Caso seja a primeira iteração do algoritmo, o tamanho do passo é inicializado conforme a equação na linha 21. Finalmente, na linha 29 os multiplicadores de Lagrange são atualizados.

## 4.7 BRKGA

O algoritmo genético de chaves aleatórias viciadas (BRKGA, do inglês “*Biased Random-Key Genetic Algorithm*”) foi proposto por Gonçalves e Resende (2011). Ele evolui uma população de chaves aleatórias, em que cada vetor de  $n$  chaves aleatórias ( $n$ , o tamanho do vetor) representa uma solução para um problema de otimização combinatória. Uma chave aleatória é um número real gerado aleatoriamente no intervalo contínuo  $[0, 1)$ . Um vetor de chaves aleatórias é decodificado em uma solução viável e um valor objetivo (ou *fitness*) através de um algoritmo determinístico chamado *decoder*, que depende do problema a ser resolvido.

Uma população de  $p$  indivíduos é construída. A cada geração, a população é dividida em dois conjuntos, de acordo com o *fitness* de cada indivíduo. Os  $p_e$  melhores indivíduos, em relação ao seus *fitness*, são identificados como conjunto elite e o restante da população como conjunto de não-elite. Todos os indivíduos da elite são copiados, sem alteração, para a próxima geração e  $p_m$  indivíduos mutantes são introduzidos. Um mutante é um vetor de

---

**Algorithm 6** Pseudocódigo do algoritmo subgradiente surrogate

---

- 1: Fixe um valor para  $U$ , parâmetro usado para atualizar o tamanho do passo;
  - 2: Escolha um valor para  $r$ , outro parâmetro usado para atualizar o tamanho do passo;
  - 3: Seja  $UB$  um limitante superior para o problema (4.20) – (4.34);
  - 4: Seja  $LB$  um limitante inferior para o problema (4.20) – (4.34);
  - 5: Seja  $max\_iter$  um limitante para a quantidade de iterações;
  - 6: Seja  $c^l$  o tamanho do passo na iteração  $l$ ;
  - 7: Inicialize  $l = 0$  e  $\lambda_{ijt}^l = 0, \forall i, j \in J, t \in T_i$ ;
  - 8: Resolva o subproblema (4.20)–(4.34) com  $\lambda_{ijt} = \lambda_{ijt}^l, \forall i, j, t$ , e seja  $x_{ik}^l$  e  $y_{ijt}^l$  os minimizadores,  $LB$  o valor ótimo, e  $UB$  a quantidade de trocas de ferramentas corresponde, ao aplicar KTNS na solução  $x_{ik}^l$ ;
  - 9: **enquanto**  $l \leq max\_iter$  **faça**
  - 10:     Resolva o subproblema (4.20)–(4.34) com  $\lambda_{ijt} = \lambda_{ijt}^l, \forall i, j, t$ , e seja  $x_{ik}^l$  e  $y_{ijt}^l$  os minimizadores, e  $Lag$  o valor ótimo;
  - 11:     Aplique KTNS na sequência obtida a partir de  $x^l$  e seja  $Ntrocas$  a quantidade de trocas de ferramentas correspondente;
  - 12:     **se**  $UB > Ntrocas$  **então**
  - 13:          $UB = Ntrocas$ ;
  - 14:     **fim se**
  - 15:     **se**  $Lag < LB$  **então**
  - 16:          $LB = Lag$ ;
  - 17:     **fim se**
  - 18:     Calcule os subgradientes:
  - 19:         
$$v_{ijt}^l = x_{ik}^l - y_{(k-1)kt}^l, i, j \in J, t \in T_i.$$
  - 20:     **se**  $l < 2$  **então**
  - 21:         
$$c^0 = \frac{UB - L(\lambda^0)}{\|v^l\|^2};$$
  - 22:     **senão**
  - 23:         Calcule
  - 24:         
$$\alpha^l = 1 - \frac{1}{Ul^p}, p = 1 - \frac{1}{r},$$
  - 25:         e atualize o tamanho do passo
  - 26:         
$$c^l = \alpha^l \frac{c^{l-1} \|v^{l-1}\|}{\|v^l\|}.$$
  - 27:     **fim se**
  - 28:     Atualize os multiplicadores de Lagrange:
  - 29:         
$$\lambda_{ijt}^{l+1} = max(\lambda_{ijt}^l + c^k v_{ijt}^l, 0), i, j \in J, t \in T_i.$$
  - 30: **fim enquanto**
-

chaves aleatórias geradas da mesma maneira que todos os indivíduos da população inicial. Os  $p - p_e - p_m$  indivíduos restantes são gerados pela combinação de pares de soluções escolhidas aleatoriamente, uma do conjunto elite e a outra do conjunto não-elite (SPEARS *et al.*, 1991).

Na Figura 4.4, um diagrama de fluxo do BRKGA é ilustrado. Neste diagrama, o retângulo cinza representa a função do decodificador. Este é o único componente do BRKGA que depende do problema que está sendo resolvido.

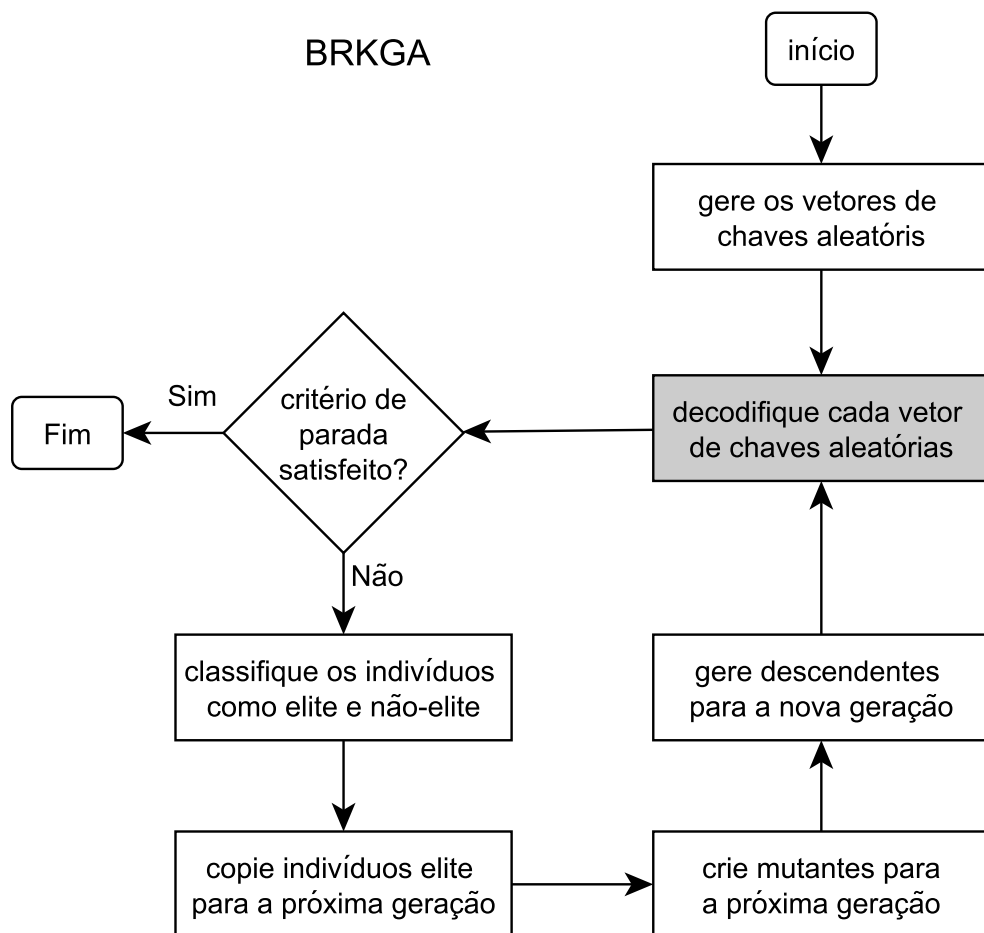


FIGURA 4.4 – Diagrama de fluxo do BRKGA. Adaptado de Gonçalves e Resende (2011).

## 4.8 BRKGA Adaptativo (A-BRKGA)

O algoritmo genético de chaves aleatórias viciadas adaptativo (A-BRKGA) é uma metaheurística proposta por Chaves *et al.* (2018). Essa metaheurística ajusta on-line os parâmetros para que os processos de intensificação e diversificação sejam equilibrados. Na Figura 4.5, o processo evolutivo do A-BRKGA é ilustrado.

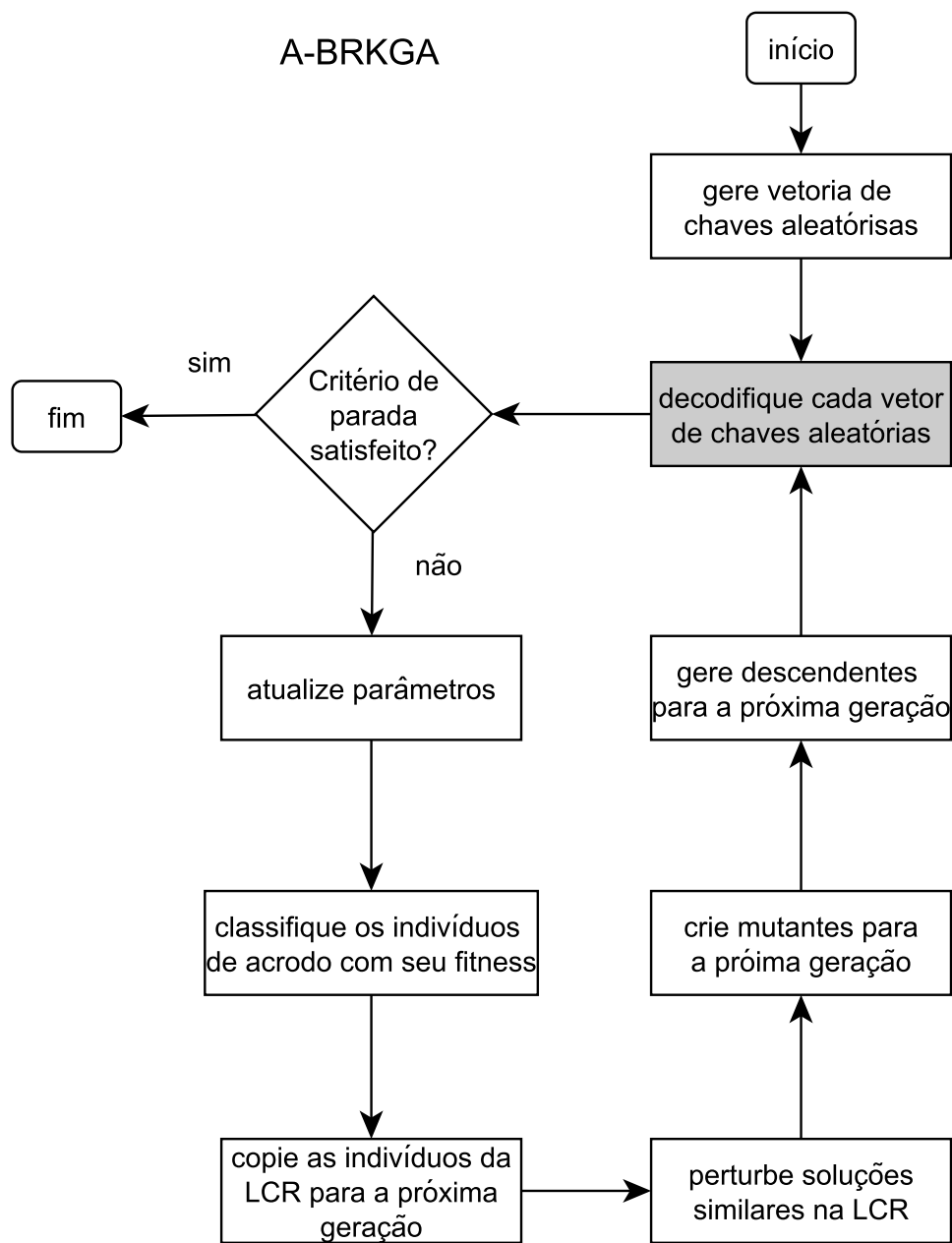


FIGURA 4.5 – Diagrama de fluxo do BRKGA adaptativo. Adaptado de Chaves *et al.* (2016), Chaves *et al.* (2018).

No A-BRKGA, uma solução de um problema de otimização combinatória também é representada por um vetor de  $n$  chaves aleatórias mais duas chaves aleatórias extras nas posições  $n + 1$  e  $n + 2$ . Elas são usadas para calcular os parâmetros autoadaptativos.

Ao denotar os tamanhos máximo e mínimo da população como  $p_{max}$  e  $p_{min}$ , respectivamente, o número máximo de gerações ( $max_{gen}$ ), que também é o critério de parada, é

dado pela equação:

$$max_{gen} = \gamma^{(\log_{\gamma}^{p_{min}} - p_{max})}. \quad (4.35)$$

A equação (4.35) é motivada pela expressão usada no *Simulated Annealing* (INGBER, 1993) para calcular o número máximo de iterações necessárias para uma determinada temperatura. Os parâmetros  $p_{max}$  e  $p_{min}$  são definidos em 1000 e 100, respectivamente. Esses valores são recomendados na literatura (GONÇALVES; RESENDE, 2018; PRASETYO *et al.*, 2015). O parâmetro  $\gamma$  é selecionado no conjunto  $\{0.999, 0.998, 0.997\}$  e depende do tempo de execução disponível.

Em cada geração, a população é dividida em dois grupos: o primeiro grupo (LRC, lista restritiva de cromossomos) consiste nos melhores indivíduos avaliados pelo *fitness* e, o segundo grupo composto pelos indivíduos da população que não estão na LRC. A LRC é composta por indivíduos cujo *fitness* é menor que um limite  $f_e$  (problema de minimização), fornecido por uma combinação convexa dos valores de *fitness* dos melhores ( $f_{min}$ ) e piores ( $f_{max}$ ) indivíduos da população atual:

$$f_e = \alpha f_{max} + (1 - \alpha) f_{min}, \quad (4.36)$$

com  $\alpha \in [0, 1]$ . O número de indivíduos na LRC é limitado por  $p_e = p * \kappa_e$ , em que  $\kappa_e$  é dado pela Equação (4.37). Na Equação (4.37),  $g_k$  é o número da geração atual. De acordo com a literatura, o número de indivíduos elite deve variar entre 10% e 25% da população (GONÇALVES; RESENDE, 2018; PRASETYO *et al.*, 2015). O número de indivíduos elite é menor nas gerações iniciais, quando a qualidade média dos indivíduos é baixa (quanto ao *fitness*) e esse número aumenta nas gerações subsequentes.

$$\kappa_e = 0.10 + \frac{g_k}{max_{gen}} * (0.25 - 0.10). \quad (4.37)$$

O parâmetro  $\alpha$  em (4.36) é modificado a cada iteração, de acordo com a Equação (4.38), o qual inicia com o valor 0.3 e termina com o valor 0.1. O valor  $f_e$  diminui durante a evolução da população devido à tendência de achatamento do *fitness* dos indivíduos após um certo número de gerações.

$$\alpha = 0.10 + \left(1 - \frac{g_k}{max_{gen}}\right) * (0.30 - 0.10). \quad (4.38)$$

Os indivíduos na LRC com o mesmo *fitness* são perturbados por um parâmetro adaptativo  $\beta$ , que é a probabilidade de trocar aleatoriamente dois valores de chaves aleatórias. Este parâmetro é atualizado de acordo com a Equação (4.39), em que  $x_{n+2}$  é a chave

aleatória na posição  $n + 2$  do indivíduo  $x$  na geração atual. Essa perturbação mantém a diversidade da população e evita a convergência prematura.

$$\beta = 0.001 + x_{n+2} * (0.10 - 0.001). \quad (4.39)$$

Todos os indivíduos na LRC na população da geração  $k$  são copiados para a população da geração  $k + 1$ . Em cada geração,  $p_m = p * \kappa_m$  indivíduos são gerados como na população inicial e inseridos na população da geração  $k + 1$ , em que  $\kappa_m$  é dado pela Equação (4.40). Esses indivíduos são chamados de mutantes. Com base na literatura, o número de mutantes deve variar entre 5% e 20% da população (GONÇALVES; RESENDE, 2018; PRASETYO *et al.*, 2015). À medida que população evolui, o número de mutantes diminui, visando maior intensificação na busca pela solução.

$$\kappa_m = 0.05 + \left(1 - \frac{g_k}{max_{gen}}\right) * (0.20 - 0.05). \quad (4.40)$$

Para completar a população, são necessários  $p - |LRC| - p_m$  indivíduos adicionais. Esses indivíduos são obtidos combinando pares de indivíduos da população atual, usando o *crossover* uniforme parametrizado (SPEARS *et al.*, 1991). Um número  $r_j \in [0, 1]$  é gerado para cada gene de cada indivíduo. Se  $r_j \leq \rho_e$ , esse gene herda o alelo do pai da LRC. Caso contrário, ele herda o alelo do outro pai. O parâmetro  $\rho_e$  é autoadaptativo e é dado pela equação:

$$\rho_e = 0.65 + y_{n+1} * (0.80 - 0.65), \quad (4.41)$$

em que  $y_{n+1}$  é a chave aleatória na posição  $n + 1$  do indivíduo  $y$  que não está na LRC. A probabilidade de aceitar um alelo da elite pertence ao intervalo [65%, 80%, conforme recomendado na literatura (CHAVES *et al.*, 2018).

Todas os intervalos dos parâmetros utilizadas no A-BRKGA foram baseadas em estudos na literatura e em testes empíricos. Eles foram definidos com o objetivo de desenvolver um algoritmo com código fácil de utilizar e que seja eficiente para solucionar diferentes problemas de otimização combinatória. Na subseção 4.8.1, uma heurística de busca local integrada ao BRKGA e A-BRKGA é apresentada.

### 4.8.1 Busca Local

Chaves *et al.* (2018) mostraram que o uso de heurísticas de busca local pode melhorar o desempenho do BRKGA e do A-BRKGA. No entanto, para evitar um aumento significativo no tempo de execução, essas heurísticas são aplicadas apenas em regiões con-

sideradas promissoras pelo algoritmo. Para encontrar essas regiões promissoras, o método de agrupamento *Label Propagation* (LP) (RAGHAVAN *et al.*, 2007) é usado para identificar comunidades com um grande número de indivíduos semelhantes na LRC. Como a LRC contém os melhores indivíduos avaliados pelo *fitness*, essas comunidades são consideradas regiões promissoras.

O nível de similaridade,  $r$ , entre dois indivíduos  $x$  e  $y$  é calculado pelo coeficiente de correlação de Pearson (RODGERS; NICEWANDER, 1988) e é dado pela Equação (4.42), em que  $x$  e  $y$  são os vetores de chaves aleatórias e  $\bar{x}$  e  $\bar{y}$  são as médias aritméticas de  $x$  e  $y$ , respectivamente. Um parâmetro adaptativo  $\sigma$  é definido pela Equação (4.43) e usado para representar o grau de similaridade durante o processo de busca. Se  $r \geq \sigma$ , os indivíduos  $x$  e  $y$  são semelhantes. O parâmetro  $\sigma$  começa em 0.3 e aumenta durante a evolução da população, uma vez que a diversidade diminui ao longo do processo de evolução.

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}. \quad (4.42)$$

$$\sigma = 0.30 + \frac{g^k}{max_{gen}} * (0.70 - 0.30). \quad (4.43)$$

Inicialmente, um grafo  $G(V, A)$  é construído, em que cada indivíduo da LRC é representado por um nó  $v \in V$ . Existe um arco  $a_{uv} \in A$  apenas se os indivíduos  $u$  e  $v$  tiverem um nível de similaridade maior que  $\sigma$ . Então, para cada nó, um número diferente é atribuído, chamado *label*. Para cada nó  $v$  do grafo, o LP busca o *label* com a maior frequência entre seus vizinhos. Se ocorrer um empate, um deles é escolhido aleatoriamente. O *label* de  $v$  é então substituído pelo *label* desse vizinho com a frequência mais alta. O algoritmo termina quando o *label* de cada nó no grafo é o mais comum de sua vizinhança. Por fim, todos os nós com o mesmo *label* são agrupados em uma comunidade (cluster). A Figura 4.6 ilustra um exemplo do LP aplicado a uma LRC com 9 indivíduos. Cada aresta neste grafo é incidente em nós com indivíduos semelhantes. Começando com o grafo da Figura 4.6A, o LP termina com o grafo da Figura 4.6 D. A maneira como os *labels* dos nós são atualizados é mostrada uma de cada vez. No final, dois clusters com *labels* 2 e 6 são obtidos.

O indivíduo em um cluster com o melhor valor de *fitness* e que ainda não foi explorado pela pesquisa local específica é escolhido como a solução representativa desse cluster. Esse indivíduo é decodificado em uma solução do problema e a busca local é aplicada a essa solução. A solução ótima local encontrada não retorna à população atual, pois isso exigiria um processo de recodificação. Se esta solução for melhor que a melhor solução conhecida

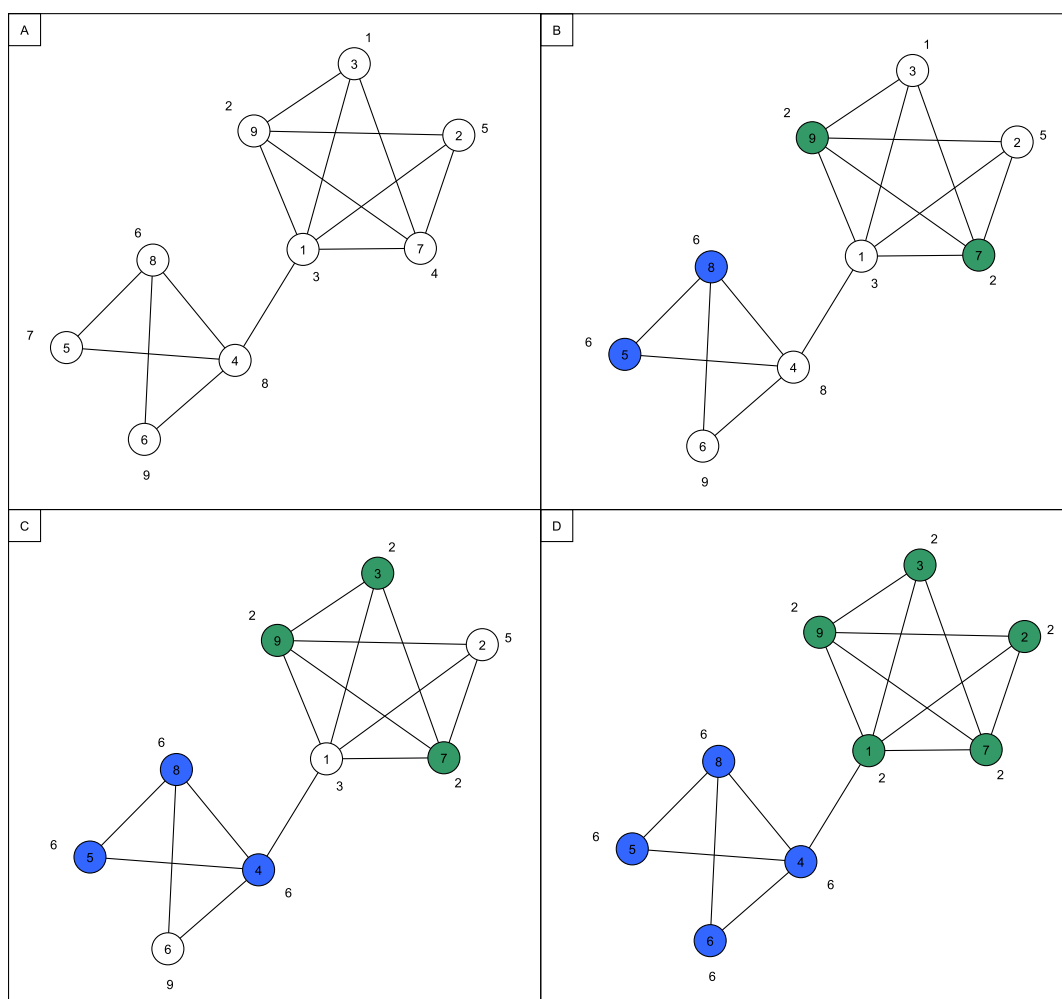


FIGURA 4.6 – Exemplo de uma iteração do algoritmo *Label Propagation*.

até o momento, a melhor solução atual será atualizada. Como essa solução não é devolvida à população atual, ela não interfere no processo evolutivo do BRKGA ou do A-BRKGA.

Os métodos BRKGA e A-BRKGA com essas heurísticas de busca local são denominados BRKGA+CS e A-BRKGA+CS, respectivamente (CHAVES *et al.*, 2016; CHAVES *et al.*, 2018).

## 4.9 A-BRKGA para o PMTF

Uma solução do PMTF pode ser codificada como um vetor  $S$  de  $n + 2$  chaves aleatórias no A-BRKGA, em que  $n$  é o número de tarefas, e as posições  $n + 1$  e  $n + 2$  são usadas para calcular o valor dos parâmetros  $\rho_e$  e  $\beta$ , respectivamente. A decodificação de  $S$  em uma solução  $S'$  do PMTF é realizada classificando as tarefas em ordem crescente de seus valores correspondentes de chaves aleatórias. Na Figura 4.7, um exemplo do processo de decodificação para o PMTF com sete tarefas é mostrado.



O *fitness* da solução  $S'$  é calculado a partir da política KTNS.

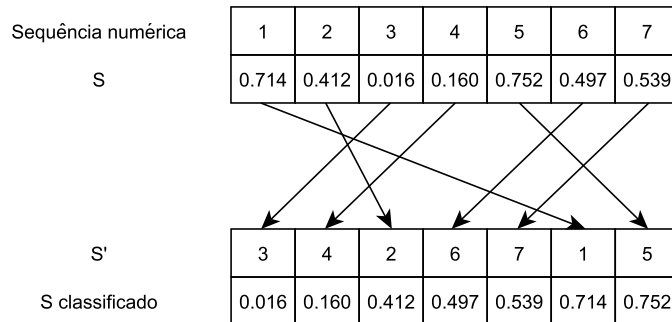


FIGURA 4.7 – Exemplo de uma solução codificada do PMTF.

## 4.10 Método exato BB-PMTF

Um método exato do tipo *branch-and-bound*, denominado BB-PMTF, foi desenvolvido para resolver o PMTF. O método baseia-se na propriedade de simetria em que toda sequência de tarefas e sua sequência reversa tem a mesma quantidade de trocas de ferramentas. Baseado nesta propriedade, tem-se o seguinte resultado imediato:

**Propriedade 4.10.1.** *Dada qualquer tarefa  $p$ , existe uma solução ótima na qual  $p$  é processada em uma das  $\lceil N/2 \rceil$  primeiras posições.*

O modelo utilizado foi o modelo MF-PMTF-1, uma vez que derivamos algumas desigualdades válidas para este modelo. A Figura 4.8 ilustra um exemplo da aplicação do método proposto com 7 nós. Para iniciar o método, uma solução factível com respectivo valor objetivo (limitante superior) é requerida. Suponha que esta solução seja dada pela sequência de tarefas  $(i_1, i_2, \dots, i_n)$  com valor objetivo UB. Pela Propriedade 4.10.1, existe uma solução ótima em que a tarefa  $i_1$  seja processada em uma das  $\lceil N/2 \rceil$  primeiras posições. Assim, adicionando esta restrição, o espaço de busca se reduz pela metade, uma vez que toda sequência reversa é eliminada. O nó 0 corresponde à relaxação linear e o nó 1 corresponde ao problema original adicionado da restrição acima. Cada nível da árvore corresponde à uma tarefa da sequência dada, e cada nó filho corresponde a requerer que a tarefa  $i_s$  seja sequenciada em uma das  $\lceil N/2 \rceil$  primeiras posições ou então em uma das  $N - \lceil N/2 \rceil - 1$  últimas posições. O menor valor obtido da resolução de todos os subproblemas de uma dado nível é um limitante inferior (LB) para o PMTF. Então, o método prossegue resolvendo todos os subproblemas de um nível, atualizando os valores de LB e UB, quando possível. Há um tempo limite para resolução de cada subproblema, e todo o método tem um tempo limite para execução. Quando atinge este tempo, o algoritmo termina e devolve os valores obtidos.

As variáveis  $x$  e  $y$  são relaxadas linearmente, e em cada nível da árvore, as variáveis  $x_{ik}$  correspondentes às tarefas daquele nível e dos níveis anteriores são integralizadas para a solução do subproblema. Caso este procedimento não seja realizado, o valor do limitante inferior obtido é  $M - C$ , uma vez que estaria apenas dividindo a relaxação linear em vários subproblemas.

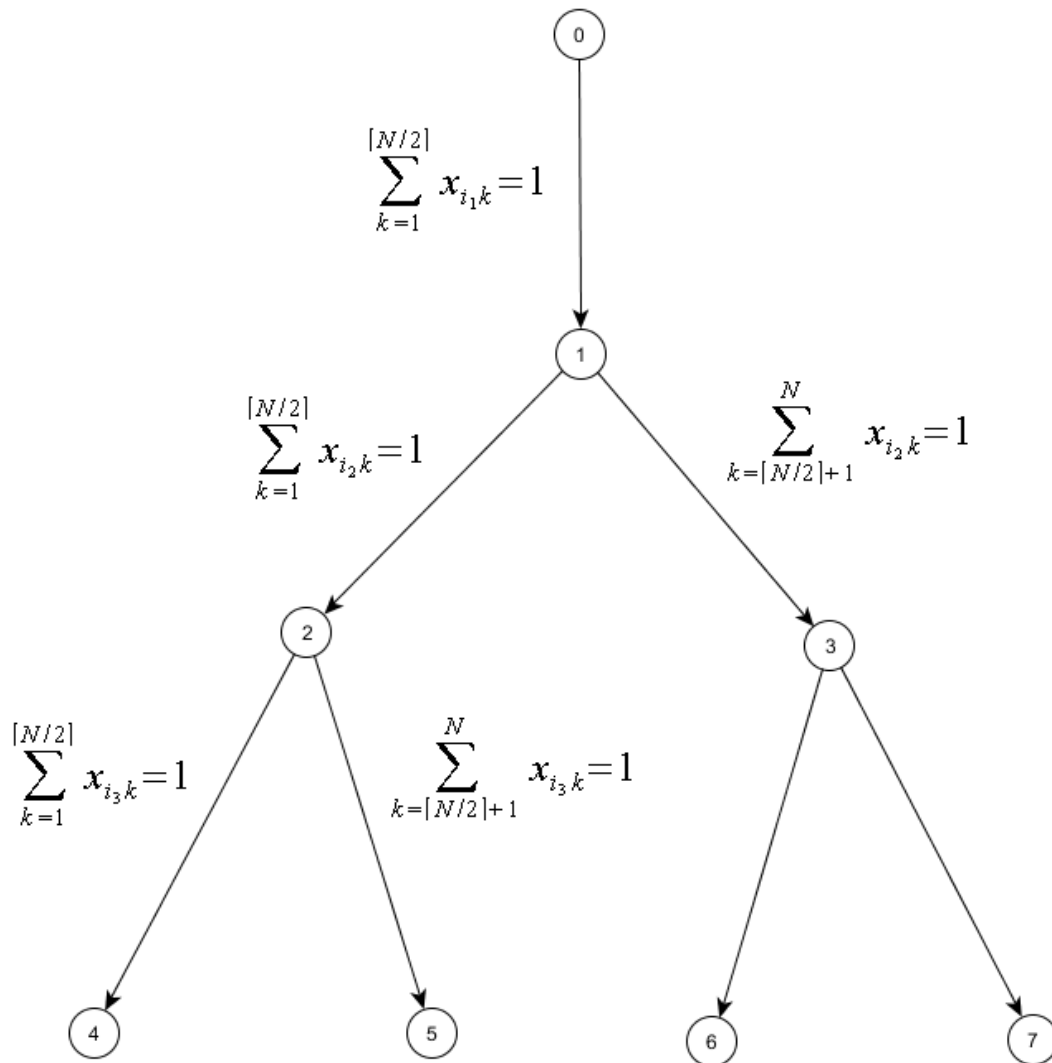


FIGURA 4.8 – Exemplo de árvore *branch-and-bound* com 7 nós, sendo o nó 0 o nó correspondente a relaxação linear.

## 4.11 Experimentos computacionais

Nesta seção os resultados dos estudos computacionais desenvolvidos para o PMTF são apresentados. O objetivo do primeiro estudo é comparar os limitantes obtidos pela Relaxação Lagrangeana. O segundo estudo teve como objetivo estudar a performance das

matheurísticas desenvolvidas. Por fim, um terceiro estudo teve como objetivo avaliar o desempenho do método exato BB-PMTF proposto.

Todas as instâncias utilizadas podem ser encontradas em (<https://sites.google.com/site/antoniochaves/publications/data>). Cada conjunto de dados contém  $i$  instâncias aleatórias com o mesmo número de tarefas, ferramentas e capacidade da máquina.

### 4.11.1 Relaxação Lagrangeana

Este estudo teve por objetivo avaliar o desempenho do Algoritmo de Subgradiente Surrogate na obtenção de limitantes inferiores e superiores para o PMTF como apresentado na Seção 4.6. O algoritmo foi escrito na sintaxe do AMPL (FOURER *et al.*, 2002) e utilizou-se o *solver* CPLEX 12.6.3 (ILOG..., 2006) na resolução dos subproblemas lagrangeanos. Os códigos foram compilados e executados em um computador com processador Intel® Core™ i7 com 3.1 GHz e com 8 GB de memória RAM, sob a plataforma Windows 10.

Na Tabela 4.2 apresentam-se os resultados para as instâncias analisadas. As instâncias utilizadas neste estudo foram as propostas por Catanzaro *et al.* (2015). A coluna UB apresenta os valores médios dos melhores limitantes superiores obtidos, a coluna Tempo(s) informa o tempo médio de execução do Algoritmo de Subgradiente Surrogate, em segundos, em cada uma das classes de instâncias, e a coluna FM-PMTF-1D apresenta os limitantes superiores fornecidos pelo modelo FM-PMTF-1D, quando não o valor ótimo, no tempo limite de 3.600 segundos. O algoritmo foi rodado por 2.000 iterações, em cada instância, utilizando  $U = 40$ , e  $r = 0, 15$ . Estes valores foram determinados através de testes computacionais preliminares. O tempo total de execução de cada instância foi limitado a 3.600 segundos.

Analisando a Tabela apresentada, e comparando com os resultados obtidos pelos modelos, nas seções anteriores, vê-se que os valores encontrados pela Relaxação Lagrangeana foram superiores às soluções ótimas, ou seja, encontrou-se soluções de qualidade inferior às dos modelos. Apenas na última classe do primeiro grupo, as soluções ótimas foram atingidas. Os limitantes inferiores obtidos foram iguais aos limitantes inferiores obtidos pela relaxação linear dos modelos propostos. Os melhores valores de limitantes inferiores foram obtidos quando  $\lambda = 0$ .

### 4.11.2 Matheurísticas

Este estudo teve por objetivo avaliar o desempenho das matheurísticas propostas: Busca por Proximidade e VNDLB, na obtenção de limitantes superiores para o PMTF como apresentado nas Subseções 4.3 e 4.5, respectivamente. As duas matheurísticas foram

TABELA 4.2 – Resultados obtidos pela Relaxação Lagrangeana para as instâncias de Catanzaro *et al.* (2015).

Grupo	$N$	$M$	$C$	UB	Tempo(s)	FM-PMTF-1D
A	10	10	4	9,00	224,99	8,50
	10	10	5	5,90	194,56	5,80
	10	10	6	4,30	207,74	4,10
	10	10	7	3,00	202,22	3,00
B	15	20	6	26,10	216,79	20,50
	15	20	8	17,00	192,46	13,70
	15	20	10	11,60	203,22	9,70
	15	20	12	7,60	254,06	7,20
C	30	40	15	116,80	447,59	96,60
	30	40	17	95,70	446,00	76,70
	30	40	20	70,00	439,60	55,00
	30	40	25	40,60	425,73	33,00
D	40	60	20	243,20	613,13	212,40
	40	60	22	211,10	592,69	184,60
	40	60	25	175,60	589,12	156,30
	40	60	30	126,70	580,88	111,00

escritas na sintaxe do AMPL (FOURER *et al.*, 2002) e utilizou-se o *solver* CPLEX 12.6.3 (ILOG..., 2006) na resolução dos subproblemas envolvidos. Os códigos foram compilados e executados em um computador com processador Intel® Core™ i7 com 3.2 GHz e com 16 GB de memória RAM, sob a plataforma Windows 10.

Através de testes computacionais preliminares, o modelo de fluxo escolhido para a utilização nas matheurísticas foi o modelo MF-PMTF-1A.

Restrições adicionais podem ser adicionadas ao modelo MF-PMTF-1A para reduzir o espaço de busca, garantindo que a solução não seja afetada. A igualdade (4.44) impõe que uma ferramenta  $t$  pode deixar permanentemente a máquina apenas depois de ser usada pelo menos a quantidade de vezes que é necessária, ou seja,  $|J_t|$  vezes.

$$y_{k(N+1)t} = 0, \quad t = 1, 2, \dots, M, \quad k = 1, \dots, |J_t| - 1. \quad (4.44)$$

Na matheurística Busca por Proximidade, o critério de parada final foi  $\theta$  atingir o valor 3 (tentando provar otimalidade) ou tempo de execução total igual a 3.600. O tempo limite de execução de cada subproblema foi de 60 segundos.

Nas Tabelas 4.3 e 4.4 apresentam-se os resultados das matheurísticas VNDLB e Busca por Proximidade, respectivamente, para as instâncias propostas por Catanzaro *et al.* (2015). Como a matheurística Busca por Proximidade apresentou resultado inferior em relação a VNDLB no conjunto de instâncias de Catanzaro *et al.* (2015), para as instâncias propostas por Yanasse *et al.* (2009) apenas a matheurística VNDLB foi executada e os resultados obtidos por ela são apresentados na Tabela 4.5. Nas Tabelas, apresentam-se o

número de tarefas ( $N$ ), o número de ferramentas ( $M$ ), a capacidade da máquina ( $C$ ) e o número de instâncias ( $i$ ) em cada um dos grupos (Grupo) das instâncias usadas nos testes. Para cada uma das matheurísticas, considerando as instâncias propostas por Catanzaro *et al.* (2015), iniciou-se com a solução fornecida pela solução do PCV apresentado na Seção 4.4 utilizando a métrica (4.18), e com a solução inicial dada pela solução do *Relax-and-Fix* (RF) utilizando  $r = 2$ . Para cada grupo de instâncias e cada método que forneceu a solução inicial, apresentam-se o valor inicial médio (SI), a solução final média (S), o tempo médio para resolução das  $i$  instâncias consideradas (T), e o GAP em relação a solução dada pelo modelo FM-PMTF-1D no tempo limite de 3.600 segundos, em cada classe. O  $GAP_i^j$  é calculado dividindo a solução obtida pelo modelo FM-PMTF-1D pela solução da matheurística  $j$  considerando a solução inicial dada pelo método  $i$ , subtraída de 1, e o resultado é multiplicado por 100, em que  $j = V$  (VNDLB) ou B (Busca por Proximidade), e  $i = P$  (PCV) ou R (RF). Se para dados  $i$  e  $j$ ,  $GAP_i^j > 0$ , então a matheurística  $j$  obteve uma solução média melhor, que é equivalente ao valor deste  $GAP_i^j$ , em relação ao modelo. Se  $GAP_i^j < 0$ , então a matheurística  $j$  obteve uma solução média pior em relação ao modelo. Por fim, se  $GAP_i^j = 0$ , então ambas matheurística  $j$  e modelo obtiveram as mesmas soluções médias. Para as instâncias propostas por Yanasse *et al.* (2009) apresentam-se apenas os resultados considerando a solução do PCV.

Analisando as Tabelas 4.3 e 4.4, percebe-se que a solução inicial influencia o resultado obtido pelas matheurísticas, tanto em qualidade de solução quanto em tempo computacional. Quanto melhor a qualidade da solução inicial ambas matheurísticas produziram limitantes superiores melhores. Entretanto, a matheurística VNDLB partindo da solução inicial fornecida pela solução do PCV correspondente obteve os melhores resultados, tanto em qualidade de solução quanto em tempo de execução. Isto se deve ao fato de os subproblemas na matheurística Busca por Proximidade, se tornarem mais difíceis quando o valor da função objetivo,  $f(x)$ , e/ou o valor de  $\theta$  é diminuído. Como os subproblemas são limitados a 60 segundos de execução, algumas vezes os subproblemas nem encontraram soluções factíveis. Comparando os valores dos GAPs, a matheurística VNDLB encontrou soluções melhores para todas as classes dos grupos C e D, alcançando uma melhora de 16,90%, por exemplo, na classe D3.

Na Tabela 4.5 os resultados para as instâncias de Yanasse *et al.* (2009) são apresentados. Observa-se que em todos os conjuntos de instâncias, a matheurística VNDLB melhorou os resultados obtidos do PCV, e que esta melhoria foi em média 12.84%. Observa-se que o tempo de execução foi maior quando os exemplares foram maiores, uma vez que os subproblemas se tornaram mais difíceis, como é o caso do Grupo D.

Nas Tabelas 4.6 e 4.7 apresentam-se as soluções produzidas para as instâncias de Catanzaro *et al.* (2015) e Yanasse *et al.* (2009), respectivamente, pela VNDLB, os limitantes superiores fornecidos pelo A-BRKGGA, e a solução obtida pela heurística ILS de Paiva e

Carvalho (2017). Observa-se que quanto maior a capacidade da máquina, melhores são os resultados da VNDLB comparados aos da A-BRKGA e ILS, sendo iguais em vários conjuntos de instâncias. Quanto as instâncias de Catanzaro *et al.* (2015), no primeiro grupo, VNDLB, A-BRKGA e ILS produziram as mesmas soluções, a menos de uma instância da primeira classe que VNDLB produziu uma troca a mais. No segundo grupo, A-BRKGA e ILS tiveram resultados iguais, sendo que VNDLB forneceu resultados muito próximos aos obtidos pelos outros dois métodos. Nos dois últimos grupos, VNDLB e A-BRKGA produziram resultados piores em relação ao ILS. Já quanto as instâncias de Yanasse *et al.* (2009), VNDLB forneceu resultados muito próximo em relação aos de A-BRKGA e ILS, sendo que quando não foram iguais, houve uma diferença de no máximo uma unidade. Observa-se que no conjunto N=15, M=20 e C=12, do Grupo E, A-BRKGA foi o que forneceu os melhores resultados. VNDLB não foi competitiva com A-BRKGA e ILS.

TABELA 4.3 – Resultados obtidos pela matheurística VNDLB para as instâncias de Catanzaro *et al.* (2015).

Grupo	N	M	C	i	PCV				RF			
					SI	S	T	GAP <sub>P</sub> <sup>V</sup>	SI	S	T	GAP <sub>P</sub> <sup>V</sup>
A	10	10	4	10	9,40	<b>8,60</b>	2,62	-1,16%	14,40	<b>8,60</b>	3,52	-1,16%
	10	10	5	10	6,50	<b>5,80</b>	3,20	0,00%	9,80	<b>5,80</b>	3,78	0,00%
	10	10	6	10	4,40	<b>4,10</b>	3,29	0,00%	5,60	<b>4,10</b>	3,63	0,00%
	10	10	7	10	3,00	<b>3,00</b>	4,19	0,00%	3,60	<b>3,00</b>	4,54	0,00%
B	15	20	6	10	23,50	<b>21,20</b>	46,93	-3,30%	33,60	21,80	40,14	-5,96%
	15	20	8	10	16,00	<b>14,30</b>	73,90	-4,20%	22,50	14,40	74,94	-4,86%
	15	20	10	10	11,20	<b>9,80</b>	67,83	-1,02%	16,00	9,90	68,38	-2,02%
	15	20	12	10	7,80	<b>7,20</b>	34,25	0,00%	11,00	<b>7,20</b>	37,14	0,00%
C	30	40	15	10	96,50	<b>89,80</b>	551,60	7,57%	132,70	91,10	627,80	6,04%
	30	40	17	10	79,00	<b>71,60</b>	880,22	7,12%	109,50	74,00	940,20	3,65%
	30	40	20	10	60,00	<b>51,50</b>	1065,58	6,80%	80,50	53,40	1179,49	3,00%
	30	40	25	10	36,30	<b>30,00</b>	1323,73	10,00%	50,00	30,70	1463,93	7,49%
D	40	60	20	10	200,70	<b>189,80</b>	1223,08	11,91%	265,90	195,30	3069,93	8,76%
	40	60	22	10	175,40	<b>164,90</b>	2485,04	11,95%	236,10	171,00	2034,87	7,95%
	40	60	25	10	145,60	<b>133,70</b>	1276,67	16,90%	196,10	136,90	2817,97	14,17%
	40	60	30	10	107,90	<b>97,80</b>	1271,63	13,50%	145,10	100,60	1708,60	10,34%

TABELA 4.4 – Resultados obtidos pela matheurística Busca por Proximidade para as instâncias de Catanzaro *et al.* (2015).

Grupo	N	M	C	i	PCV				RF			
					SI	S	T	GAP <sub>P</sub> <sup>B</sup>	SI	S	T	GAP <sub>P</sub> <sup>B</sup>
A	10	10	4	10	9,40	<b>8,50</b>	1,38	0,00%	14,40	<b>8,50</b>	1,71	0,00%
	10	10	5	10	6,50	<b>5,80</b>	0,88	0,00%	9,80	<b>5,80</b>	1,17	0,00%
	10	10	6	10	4,40	<b>4,10</b>	0,41	0,00%	5,60	<b>4,10</b>	0,44	0,00%
	10	10	7	10	3,00	<b>3,00</b>	0,09	0,00%	3,60	<b>3,00</b>	0,28	0,00%
B	15	20	6	10	23,50	<b>21,30</b>	191,22	-3,76%	33,60	21,5	179,86	-4,65%
	15	20	8	10	16,00	14,1	147,35	-2,84%	22,50	<b>14,00</b>	144,35	-2,14%
	15	20	10	10	11,20	<b>9,80</b>	35,56	-1,02%	16,00	<b>9,80</b>	37,95	-1,02%
	15	20	12	10	7,80	<b>7,20</b>	0,43	0,00%	11,00	<b>7,2</b>	1,88	0,00%
C	30	40	15	10	96,50	<b>91,00</b>	258,22	6,15%	132,70	94,90	466,60	1,79%
	30	40	17	10	79,00	<b>74,70</b>	252,12	2,68%	109,50	77,00	421,94	-0,39%
	30	40	20	10	60,00	<b>54,80</b>	252,71	0,36%	80,50	55,10	398,73	-0,18%
	30	40	25	10	36,30	<b>31,60</b>	239,29	4,43%	50,00	32,20	318,40	2,48%
D	40	60	20	10	200,70	<b>199,10</b>	231,90	6,68%	265,90	223,10	1293,28	-4,80%
	40	60	22	10	175,40	<b>174,00</b>	254,31	6,09%	236,10	196,40	1223,88	-6,01%
	40	60	25	10	145,60	<b>143,20</b>	248,13	9,15%	196,10	162,30	958,06	-3,70%
	40	60	30	10	107,90	<b>105,50</b>	234,04	5,21%	145,10	118,30	795,75	-6,17%

### 4.11.3 Método exato BB-PMTF

Este estudo teve por objetivo avaliar o desempenho do método exato BB-PMTF proposto na Subseção 4.10. O método foi escrito na sintaxe do AMPL (FOURER *et al.*, 2002) e utilizou-se o *solver* CPLEX 12.6.3 (ILOG..., 2006) na resolução dos subproblemas envolvidos. O código foi compilado e executado em um computador com processador Intel® Core™ i7 com 3.2 GHz e com 16 GB de memória RAM, sob a plataforma Windows 10. As instâncias utilizadas nos testes foram as propostas por Catanzaro *et al.* (2015). Para a solução referência e o correspondente valor de UB inicial, usou-se a solução do A-BRKGA proposto na Subseção 4.9.

Na Tabela 4.8 apresentam-se os resultados obtidos pelo BB-PMTF. Nela, apresentam-se o número de tarefas ( $N$ ), o número de ferramentas ( $M$ ), a capacidade da máquina ( $C$ ) e o número de instâncias ( $i$ ) em cada um dos grupos (Grupo) das instâncias usadas nos testes. Os limitantes superiores médios ( $UB$ ), os limitantes inferiores médios ( $LB$ ), o tempo computacionais médio utilizado ( $Tempo$ ), limitado a uma hora, e a quantidade de nós ( $Nós$ ) ramificados também são apresentados. Dos 16 conjuntos de instâncias testados, o método conseguiu resolver todas as instâncias de 5 conjuntos. Na classe  $N=15$ ,  $M=20$  e  $C=8$ , o método exato resolveu 1 instâncias na otimalidade, e na classe  $N=15$ ,  $M=20$  e  $C=10$  o método resolveu 7 instâncias na otimalidade. Observe que a quantidade de nós ramificadas é baixa e que o método conseguiu melhorar os valores dos limitantes inferiores de  $M - C$ , fornecendo melhores valores quando a capacidade da máquina é pequena.

## 4.12 Considerações finais

Neste capítulo, várias abordagens de solução foram desenvolvidas para o PMTF e vários testes computacionais foram realizados com estas abordagens. Algumas matheurísticas foram desenvolvidas, mas seus desempenhos, tanto em qualidade de solução quanto em tempo de execução, foram inferiores ao atual estado da arte. Dentre os métodos de solução desenvolvidos: VNDLB, Busca por Proximidade, método exato BB-PMTF, e Relaxação Lagrangeana, o que teve melhor desempenho foi o método exato, uma vez que a solução inicial utilizada foi dada pelo A-BRKGA que tem boa qualidade. Considerando as matheurísticas desenvolvidas, a que teve melhor desempenho foi a VNDLB com solução inicial dada pela solução do PCV correspondente.

TABELA 4.5 – Resultados obtidos pela matheurística VNDLB para as instâncias das classes *A*, *B*, *C*, *D* e *E* de Yanasse *et al.* (2009).

Grupo	<i>N</i>	<i>M</i>	<i>C</i>	<i>i</i>	SI	S	T	GAP <sub>P</sub> <sup>V</sup>
<i>A</i>	8	15	5	10	13,10	12,00	2,02	0,00%
	8	15	10	30	7,33	6,83	3,32	0,00%
	8	20	5	10	19,20	16,90	36,97	-0,59%
	8	20	10	30	13,57	13,07	17,40	0,00%
	8	20	15	60	7,73	7,08	14,39	0,00%
	8	25	5	10	23,20	20,20	31,57	0,00%
	8	25	10	30	19,00	18,20	23,29	0,00%
	8	25	15	60	13,70	12,95	22,66	0,00%
	8	25	20	100	7,35	6,62	28,29	-0,15%
<i>B</i>	9	15	5	10	13,20	12,20	3,62	0,00%
	9	15	10	30	7,97	7,37	4,48	0,00%
	9	20	5	10	18,90	17,50	47,10	-0,57%
	9	20	10	30	15,07	14,17	25,45	0,00%
	9	20	15	60	8,43	7,63	21,00	-0,44%
	9	25	5	10	25,20	20,40	49,88	0,00%
	9	25	10	30	19,73	18,87	37,14	-0,51%
	9	25	15	50	15,72	14,86	38,05	-0,81%
	9	25	20	100	7,94	7,19	40,20	0,00%
<i>C</i>	15	15	5	10	18,60	17,00	20,35	-2,35%
	15	15	10	30	11,37	9,97	29,22	-1,67%
	15	20	5	10	22,30	21,00	62,56	-1,90%
	15	20	10	30	20,37	18,73	73,57	-1,62%
	15	20	15	60	12,78	10,98	117,33	-4,22%
	15	25	5	10	30,50	28,00	175,35	-1,79%
	15	25	10	30	27,73	25,67	191,59	-1,70%
	15	25	15	60	21,67	19,62	257,37	-2,28%
	15	25	20	100	11,90	10,02	326,43	-3,09%
<i>D</i>	20	15	5	10	24,40	21,60	64,25	0,00%
	20	15	10	20	10,90	8,75	77,28	-4,00%
	20	20	5	10	27,50	25,30	157,89	0,40%
	20	20	10	10	12,50	11,00	174,24	-2,73%
	20	20	15	30	8,80	6,93	220,69	-2,93%
	20	25	5	10	33,90	31,40	317,86	0,00%
	20	25	10	10	17,60	15,80	358,67	-1,90%
	20	25	15	40	24,80	22,10	443,89	1,27%
	20	25	20	40	8,18	6,48	522,31	-5,02%
	25	15	10	10	8,50	6,40	632,41	-7,81%
	25	20	10	10	15,80	13,00	806,13	-6,15%
	25	20	15	10	11,40	8,50	819,67	-9,41%
	25	25	10	10	21,10	17,50	1026,69	2,86%
	25	25	15	10	11,40	10,00	821,05	0,00%
	25	25	20	30	7,47	5,87	1039,30	-5,74%
<i>E</i>	10	10	4	10	10,30	9,50	3,13	0,00%
	10	10	5	10	7,20	6,20	3,92	0,00%
	10	10	6	10	4,80	4,30	3,81	0,00%
	10	10	7	10	3,10	3,00	4,81	0,00%
	15	20	6	10	24,50	21,90	45,42	-2,28%
	15	20	8	10	16,40	14,70	61,99	-2,72%
	15	20	10	10	11,30	10,40	42,49	-0,96%
	15	20	12	10	9,00	8,20	49,91	0,00%



TABELA 4.6 – Resultados obtidos pela matheurística VNDLB, o algoritmo A-BRKGA e o algoritmo ILS de Paiva e Carvalho (2017), considerando as instâncias de Catanzaro *et al.* (2015).

Grupo	$N$	$M$	$C$	VNDLB	A-BRKGA	ILS
$A$	10	10	4	12,60	<b>12,50</b>	<b>12,50</b>
	10	10	5	<b>10,80</b>	<b>10,80</b>	<b>10,80</b>
	10	10	6	<b>10,10</b>	<b>10,10</b>	<b>10,10</b>
	10	10	7	<b>10,00</b>	<b>10,00</b>	<b>10,00</b>
$B$	15	20	6	27,20	<b>26,50</b>	<b>26,50</b>
	15	20	8	22,30	<b>21,70</b>	<b>21,70</b>
	15	20	10	19,80	<b>19,70</b>	<b>19,70</b>
	15	20	12	<b>19,20</b>	<b>19,20</b>	<b>19,20</b>
$C$	30	40	15	104,80	100,70	<b>98,90</b>
	30	40	17	88,60	83,50	<b>82,50</b>
	30	40	20	71,50	67,50	<b>66,60</b>
	30	40	25	55,00	51,50	<b>51,30</b>
$D$	40	60	20	209,80	203,90	<b>198,00</b>
	40	60	22	186,90	179,70	<b>173,60</b>
	40	60	25	158,70	150,40	<b>146,20</b>
	40	60	30	127,80	118,40	<b>115,20</b>

TABELA 4.7 – Resultados obtidos pela matheurística VNDLB, o algoritmo A-BRKGA e o algoritmo ILS de Paiva e Carvalho (2017), considerando as instâncias de Yanasse *et al.* (2009).

Grupo	$N$	$M$	$C$	VNDLB	A-BRKGA	ILS
A	8	15	5	<b>17,00</b>	<b>17,00</b>	<b>17,00</b>
	8	15	10	<b>16,83</b>	<b>16,83</b>	<b>16,83</b>
	8	20	5	21,90	<b>21,80</b>	<b>21,80</b>
	8	20	10	<b>23,07</b>	<b>23,07</b>	<b>23,07</b>
	8	20	15	<b>22,08</b>	<b>22,08</b>	<b>22,08</b>
	8	25	5	25,20	<b>25,10</b>	<b>25,10</b>
	8	25	10	<b>28,20</b>	<b>28,20</b>	<b>28,20</b>
	8	25	15	<b>27,95</b>	<b>27,95</b>	<b>27,95</b>
B	9	15	5	<b>17,20</b>	<b>17,20</b>	<b>17,20</b>
	9	15	10	<b>17,37</b>	<b>17,37</b>	<b>17,37</b>
	9	20	5	22,50	<b>22,40</b>	<b>22,40</b>
	9	20	10	<b>24,17</b>	<b>24,17</b>	<b>24,17</b>
	9	20	15	22,63	<b>22,60</b>	<b>22,60</b>
	9	25	5	<b>25,40</b>	<b>25,40</b>	<b>25,40</b>
	9	25	10	28,87	<b>28,77</b>	<b>28,77</b>
	9	25	15	29,86	<b>29,74</b>	<b>29,74</b>
C	15	15	5	22,00	<b>21,60</b>	<b>21,60</b>
	15	15	10	19,97	<b>19,80</b>	<b>19,80</b>
	15	20	5	26,00	<b>25,60</b>	<b>25,60</b>
	15	20	10	28,73	<b>28,33</b>	<b>28,33</b>
	15	20	15	25,98	<b>25,52</b>	<b>25,52</b>
	15	25	5	33,00	<b>32,50</b>	<b>32,50</b>
	15	25	10	35,67	<b>35,07</b>	<b>35,07</b>
	15	25	15	34,62	<b>34,07</b>	<b>34,07</b>
D	20	15	5	26,60	<b>25,90</b>	<b>25,90</b>
	20	15	10	18,75	18,25	<b>18,20</b>
	20	20	5	30,30	29,30	<b>29,20</b>
	20	20	10	21,00	<b>20,60</b>	<b>20,60</b>
	20	20	15	21,93	<b>21,67</b>	<b>21,67</b>
	20	25	5	36,40	<b>35,10</b>	<b>35,10</b>
	20	25	10	25,80	<b>25,40</b>	<b>25,40</b>
	20	25	15	37,10	36,30	<b>36,25</b>
	20	25	20	26,48	<b>26,15</b>	<b>26,15</b>
	25	15	10	16,40	<b>15,90</b>	<b>15,90</b>
	25	20	10	23,00	<b>21,60</b>	<b>21,60</b>
	25	20	15	23,50	<b>22,60</b>	<b>22,60</b>
	25	25	10	27,50	<b>26,60</b>	<b>26,60</b>
	25	25	15	<b>25,00</b>	<b>25,00</b>	<b>25,00</b>
E	10	10	4	<b>13,50</b>	<b>13,50</b>	<b>13,50</b>
	10	10	5	<b>11,20</b>	<b>11,20</b>	<b>11,20</b>
	10	10	6	<b>10,30</b>	<b>10,30</b>	<b>10,30</b>
	10	10	7	<b>10,00</b>	<b>10,00</b>	<b>10,00</b>
	15	20	6	27,90	<b>27,40</b>	<b>27,40</b>
	15	20	8	22,70	<b>22,20</b>	<b>22,20</b>
	15	20	10	20,40	<b>20,30</b>	<b>20,30</b>
	15	20	12	20,20	<b>20,18</b>	20,20

TABELA 4.8 – Resultados obtidos pelo método exato considerando as instâncias de Catanzaro *et al.* (2015).

Grupo	$N$	$M$	$C$	UB	LB	Tempo	Nós
$A$	10	10	4	8,50	8,50	12,12	52,00
	10	10	5	5,80	5,80	7,00	29,90
	10	10	6	4,10	4,10	1,68	11,50
	10	10	7	3,00	3,00	0,03	1,00
$B$	15	20	6	20,50	15,78	3600,00	328,20
	15	20	8	13,70	11,50	3240,02	188,30
	15	20	10	9,70	9,22	1407,54	60,60
	15	20	12	7,20	7,20	0,09	1,00
$C$	30	40	15	85,70	25,67	3600,00	78,20
	30	40	17	66,50	23,13	3600,00	95,60
	30	40	20	47,50	20,13	3600,00	209,40
	30	40	25	26,50	15,00	3600,00	821,60
$D$	40	60	20	185,33	43,49	3600,00	76,89
	40	60	22	157,70	38,34	3600,00	82,80
	40	60	25	125,40	35,25	3600,00	94,40
	40	60	30	88,40	30,00	3600,00	176,40

# 5 O Problema do Caixeiro Viajante Multiproduto com Prioridades

O Problema do Caixeiro Viajante (PCV) é um problema NP-difícil bem estudado e conhecido em otimização em rede (APPLEGATE *et al.*, 2007; DANTZIG *et al.*, 1954; FLOOD, 1956; LAWLER *et al.*, 1991). Consiste em determinar um circuito Hamiltoniano de custo mínimo, visitando todos os clientes apenas uma vez e retornando ao ponto de partida. O PCV surge em muitas aplicações, como a perfuração de placas de circuito impresso (“*drilling of printed circuit boards*”) (GRÖTSCHEL *et al.*, 1991), o problema de coleta de pedidos nos depósitos (“*the order-picking problem in depots*”) (RATLIFF; ROSENTHAL, 1983), o problema de roteamento de ônibus escolar (“*the school bus routing problem*”) (ANGEL *et al.*, 1972), o problema defensor-atacante-defensor (“*the defender-attacker-defender problem*”) (LOZANO *et al.*, 2017) e na logística marítima (“*the maritime logistics*”) (MALAGUTI *et al.*, 2018).

O PCV e a maioria de suas variações são orientadas por custos e, na literatura, quase não se vê estudos considerando demandas diferentes. Uma variação que considera demandas diferentes é o problema do caixeiro viajante multiproduto (PCVM) (SARUBBI, 2003; SARUBBI; LUNA, 2007; SARUBBI *et al.*, 2007). No PCVM, os tipos e quantidades de produtos que passam por uma rota que conecta dois clientes são considerados no custo total. Na prática, os clientes com demandas maiores ou produtos mais preciosos ou de alto risco devem ser atendidos com maior prioridade. Por exemplo, materiais sensíveis podem exigir estrutura de transporte especial, produtos perecíveis devem ser refrigerados, ambos levando a custos de transporte mais altos. Sarubbi (2003) e Sarubbi *et al.* (2008) consideraram custos variáveis para cada tipo de produto em cada rota entre dois clientes e propuseram um modelo de fluxo em rede que foi resolvido por relaxação lagrangeana e um método *branch-and-cut*.

Outra variação do PCV considera prêmios de prioridade para os clientes ao longo do percurso. Esses prêmios são coletados pelo caixeiro de acordo com a ordem de visita de cada cliente na rota estabelecida. Nesta variação, a qualidade do atendimento ao cliente e as prioridades de entrega são consideradas, uma vez que as preferências de

cada cliente em termos de ordem da sequência de entrega devem ser quantificadas e otimizadas. Pureza *et al.* (2018) modelaram esse problema como um caso especial de um problema de atribuição quadrática (KOOPMANS; BECKMANN, 1957), chamado de Problema do Caixeiro Viajante com Prêmios de Prioridade (PCVPP) e o solucionou por uma metaheurística Busca Tabu (GLOVER, 1986). Silva (2017), Silva *et al.* (2018) utilizaram o PCVPP e o Problema do Passeio Lucrativo (do inglês “*Profitable Tour Problem*”, PTP) (ARCHETTI *et al.*, 2014; BALAS, 1989) para formular o problema de elaboração de itinerários turísticos considerando como custos variáveis, o perfil dos visitantes, o planejamento da viagem, as preferências dos visitantes e atrações turísticas. Os autores consideraram um estudo de caso na cidade de Belém, no estado do Pará, resolvendo o problema por uma metaheurística Busca Tabu.

Além dessas situações, as empresas de logística enfrentam conflitos entre custos operacionais, clientes com diferentes categorias de produtos e satisfação do cliente, esta, diretamente relacionada ao tempo de entrega. Torna-se então um desafio escolher entre minimizar os custos de viagem e garantir um grau de qualidade de serviço para todos os clientes. Para obter soluções que equilibram custos operacionais e satisfação do cliente, um novo modelo que combina o PCVPP com o PCVM é proposto.

Neste estudo, o PCV do ponto de vista do cliente e do vendedor é examinado. Considerou-se como objetivo, minimizar os custos totais e, ao mesmo tempo, satisfazer a preferência dos clientes, maximizando as prioridades. O modelo é baseado no problema de atribuição e no problema de fluxo em rede. As *commodities* que fluem na rede representam os produtos. Esse problema será denominado Problema do caixeiro viajante multiproduto com prioridades (PCVMP).

## 5.1 Modelo de fluxo *multicommodity* para o PCVMP

Seja  $N$  o número de clientes a ser atendido, incluindo o depósito. Considere um grafo  $G(V, A)$ , em que  $V$  é um conjunto composto por  $N$  nós, numerados de 1 a  $N$ , e  $A$  o conjunto de arcos. Por conveniência, o nó 1 é o depósito e os outros nós representam os clientes a serem atendidos. O conjunto de arcos  $A$  neste grafo representa os possíveis caminhos entre os clientes. Por simplicidade, é considerado que o produto  $l$ , uma commodity, será entregue ao cliente  $l$ . Também é assumido que cada cliente pode solicitar apenas um tipo de produto, ou seja, o mesmo nó/cliente não pode solicitar produtos diferentes. Denote  $c_{ij}$  como o custo fixo no arco  $(i, j)$ ,  $q_l$  como a demanda exigida pelo cliente  $l$  do produto  $l$ ,  $d_{ijl}$  como o custo variável para transportar o produto  $l$  pelo arco  $(i, j)$  e  $p_{ki}$  o valor da prioridade coletada quando o cliente  $i$  é visitado na ordem  $k$  da rota. Assume-se, sem perda de generalidade, que  $q_l > 0$  e inteiro para todo  $l$ ,  $l = 1, \dots, N$ .

Seja  $x_{ki}$  uma variável de decisão binária igual a 1, se o cliente  $i$  for visitado na  $k$ -ésima posição da rota, e 0, caso contrário. Da mesma forma, seja  $y_{ij}$  uma variável de decisão binária igual a 1, se o cliente  $j$  for visitado imediatamente após o cliente  $i$ , e 0, caso contrário. Por fim, seja  $f_{ijl}$  uma variável de decisão real correspondente à quantidade do produto  $l$  que é transportado entre os cliente  $i$  e  $j$ , ou seja, a quantidade de fluxo do produto  $l$  no arco  $(i, j)$ . Observe que, em uma solução ideal, devido a restrições (5.7), (5.8) e (5.9) do modelo proposto, os valores das variáveis  $f_{ijl}$  serão inteiros desde que  $q_l$  seja inteiro para todos  $l$ ,  $l = 1, \dots, N$ . Portanto, uma formulação matemática para o problema PCVMP é dada pelas expressões (5.1)-(5.12) a seguir:

$$\text{Max } Z = \sum_{k=2}^N \sum_{i=2}^N p_{ki} x_{ki} - \sum_{i=1}^N \sum_{\substack{j=1 \\ i \neq j}}^N (c_{ij} y_{ij} + \sum_{l=2}^N d_{ijl} f_{ijl}) \quad (5.1)$$

sujeito a:

$$\sum_{i=1}^N x_{ki} = 1, \quad k = 1, 2, \dots, N \quad (5.2)$$

$$\sum_{k=1}^N x_{ki} = 1, \quad i = 1, 2, \dots, N \quad (5.3)$$

$$\sum_{j=1}^N y_{ij} = 1, \quad i = 1, 2, \dots, N \quad (5.4)$$

$$\sum_{i=1}^N y_{ij} = 1, \quad j = 1, 2, \dots, N \quad (5.5)$$

$$x_{(k-1)i} + x_{kj} - y_{ij} \leq 1, \quad i \neq j = 1, 2, \dots, N, \quad k = 2, 3, \dots, N \quad (5.6)$$

$$\sum_{j=2}^N f_{1jl} = q_l, \quad l = 2, 3, \dots, N \quad (5.7)$$

$$\sum_{\substack{i=1 \\ i \neq l}}^N f_{iil} = q_l, \quad l = 2, 3, \dots, N \quad (5.8)$$

$$\sum_{\substack{h=1 \\ i \neq l, h \neq i}}^N f_{hil} - \sum_{\substack{j=1 \\ i \neq l, i \neq j}}^N f_{ijl} = 0, \quad l, i = 2, 3, \dots, N \quad (5.9)$$

$$f_{ijl} \leq q_l y_{ij}, \quad i \neq j = 1, 2, \dots, N, \quad l = 2, 3, \dots, N \quad (5.10)$$

$$f_{ijl} \geq 0, \quad i, j = 1, 2, \dots, N, \quad l = 2, 3, \dots, N, \quad (5.11)$$

$$x_{ki}, y_{ij} \in \{0, 1\}, \quad k, i, j = 1, 2, \dots, N, \quad i \neq j. \quad (5.12)$$

A função objetivo (5.1) maximiza as prioridades enquanto minimiza os custos totais fixos e os custos variáveis. As restrições (5.2) e (5.3) impõem que cada cliente deve ser visitado apenas uma vez. As restrições (5.4) e (5.5) são restrições de atribuição. As restrições (5.6) vinculam as variáveis  $x$  e  $y$ . Elas garantem que, se o cliente  $i$  for visitado na posição  $(k - 1)$  e o cliente  $j$  for visitado na posição  $k$ , então o arco  $(i, j)$  deve ser usado. As restrições (5.7) garantem que todos os produtos saiam do depósito com suas respectivas demandas, e as restrições (5.8) garantem que todos os produtos cheguem aos seus destinos, levando em consideração a demanda necessária. As restrições (5.9) impõem a conservação do fluxo em qualquer nó que não seja o destino final dos produtos. As restrições (5.10) vinculam as variáveis  $f$  e  $y$ . Elas garantem que nenhum fluxo seja permitido em um arco  $(i, j)$ , a menos que seja usado. Finalmente, as restrições (5.11) e (5.12) definem o domínio das variáveis de decisão. A primeira posição na rota é definida como 1, ou seja,  $x_{11} = 1$ , pois, por conveniência, o nó 1 é definido como depósito.

**Proposição 5.1.1.** *O Modelo PCVMP não admite subrotas.*

*Demonstração.* As restrições de eliminação de subrotas são implícitas, mas são devidas às restrições (5.4), (5.5), (5.7)-(5.12). De fato, o nó 1 é o nó de fornecimento para todos os produtos  $l$ ,  $l = 2, 3, \dots, N$ . Assim, em uma solução viável, deve haver um caminho do nó 1 para o nó  $l$ ,  $l = 2, 3, \dots, N$ , em que a quantidade  $q_l$  do produto  $l$  deve fluir para satisfazer a demanda do nó  $l$ ,  $l = 2, 3, \dots, N$ , respectivamente. Em cada nó neste caminho, a conservação do fluxo tem que ser respeitada, imposta pelas restrições (5.9). De acordo com (5.10) e (5.12), o  $y_{ij}$  correspondente aos arcos nesses caminhos deve ter o valor 1. Suponha que exista uma subrota na solução. Existem duas possibilidades:

- (i) o nó 1 não pertence a esta subrota;
- (ii) o nó 1 pertence a esta subrota.

O caso (i) não pode acontecer, pois não há caminho do nó 1 para nenhum dos nós pertencentes a subrota. Portanto, as demandas desses nós não podem ser atendidas. No caso (ii), há pelo menos um nó, digamos  $t$ , que não pertence a subrota. Como deve haver um caminho do nó 1 para o nó  $t$  para satisfazer a demanda desse nó, tem-se duas possibilidades: esse caminho do nó 1 para o nó  $t$  contém apenas nós que não pertencem a subrota, ou esse caminho contém outros nós da subrota antes de atingir o nó  $t$ . O primeiro caso não pode ocorrer devido à restrição (5.4) que impõe que haja apenas um único arco com fluxo saindo do nó 1. Portanto, não há como ter dois arcos com fluxos positivos deixando o nó 1, um para suprir a demanda do nó  $t$  e o outro para suprir as demandas dos nós pertencentes a subrota. O segundo caso também não pode acontecer devido às restrições (5.4). Considere que o nó  $s$  pertença a subrota, onde o fluxo para o nó  $t$  deixa a subrota. Novamente, não há como ter dois arcos com fluxos positivos saindo do nó  $s$ , um para suprir a demanda do nó  $t$  e o outro para suprir as demandas dos nós

pertencentes a subrota. □

## 5.2 BRKGA+CS e A-BRKGA+CS para o PCVMP

Uma solução do PCVMP pode ser codificada como um vetor  $S$  de  $n$  chaves aleatórias no BRKGA+CS e um vetor de  $n + 2$  chaves aleatórias no A-BRKGA+CS, onde  $n$  é o número de clientes a serem visitados mais o depósito, e as posições  $n + 1$  e  $n + 2$  são usadas para calcular o valor dos parâmetros  $\rho_e$  e  $\beta$ , respectivamente, conforme apresentado nas Seções 4.7 e 4.8. A decodificação de  $S$  em uma solução  $S'$  do PCVMP é realizada classificando os clientes em ordem crescente de seus valores correspondentes de chaves aleatórias. A primeira posição em  $S$  é fixada em 1, representando o depósito, cujo caixeiro inicia e termina sua viagem. Na Figura 5.1, um exemplo do processo de decodificação para o PCVMP com cinco clientes é mostrado, em que o retângulo cinza representa o depósito com seu respectivo valor de chave aleatória.

O *fitness* da solução  $S'$  é calculado pelo valor correspondente da função objetivo (5.1), considerando custos fixos, variáveis e prioridades.

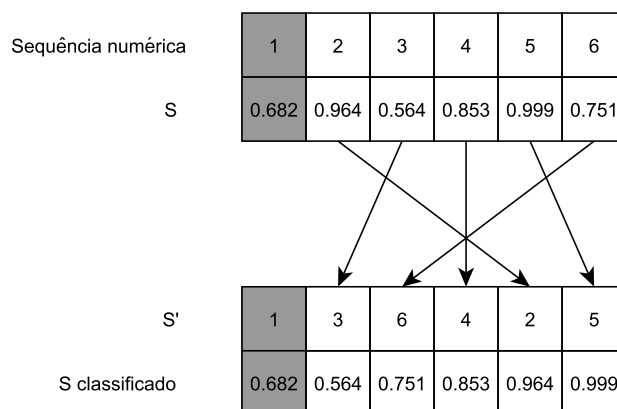


FIGURA 5.1 – Exemplo de uma solução codificada do PCVMP.

### 5.2.1 Busca local para o PCVMP

A heurística de busca local proposta é um algoritmo baseado na Busca Local Iterativa (ILS, “*Iterated Local Search*”) (STÜTZLE, 1999). O ILS combina uma fase de busca local com uma fase de perturbação. Especificamente, o VND (seção 4.2.3) foi usado na fase de busca local do ILS e Swap(1,1) foi usado na fase de perturbação.

O VND usa três estruturas de vizinhança:

- 1)  $N^{(1)}$  - 2-Opt: Reverte a ordem de visita dos clientes entre as posições  $i$  e  $j$ , inclusive,



na Solução  $S'$ ;

2)  $N^{(2)}$  - Shift(1): Transfira o cliente  $i$  da sua posição atual para a posição  $k$ , na Solução  $S'$ ;

3)  $N^{(3)}$  - Swap(1,1): Altere a posição de visitação do cliente  $i$  com a posição de visitação do cliente  $j$  na Solução  $S'$ .

Exemplos dessas vizinhanças são mostrados na Figura 5.2. Em (a), apresenta-se um exemplo de rota com 6 clientes cuja rota é  $1 - 2 - 3 - 4 - 5 - 6$ . Em (b), apresenta-se a rota  $1 - 5 - 4 - 3 - 2 - 6$  obtida aplicando 2-Opt nas posições 2 e 5. Em (c), apresenta-se a rota  $1 - 3 - 4 - 5 - 2 - 6$  obtida aplicando Shift(1) ao cliente 2 na posição 5. Em (d), apresenta-se a rota  $1 - 5 - 3 - 4 - 2 - 6$  obtida pela aplicação de Swap(1,1) aos clientes 2 e 5.

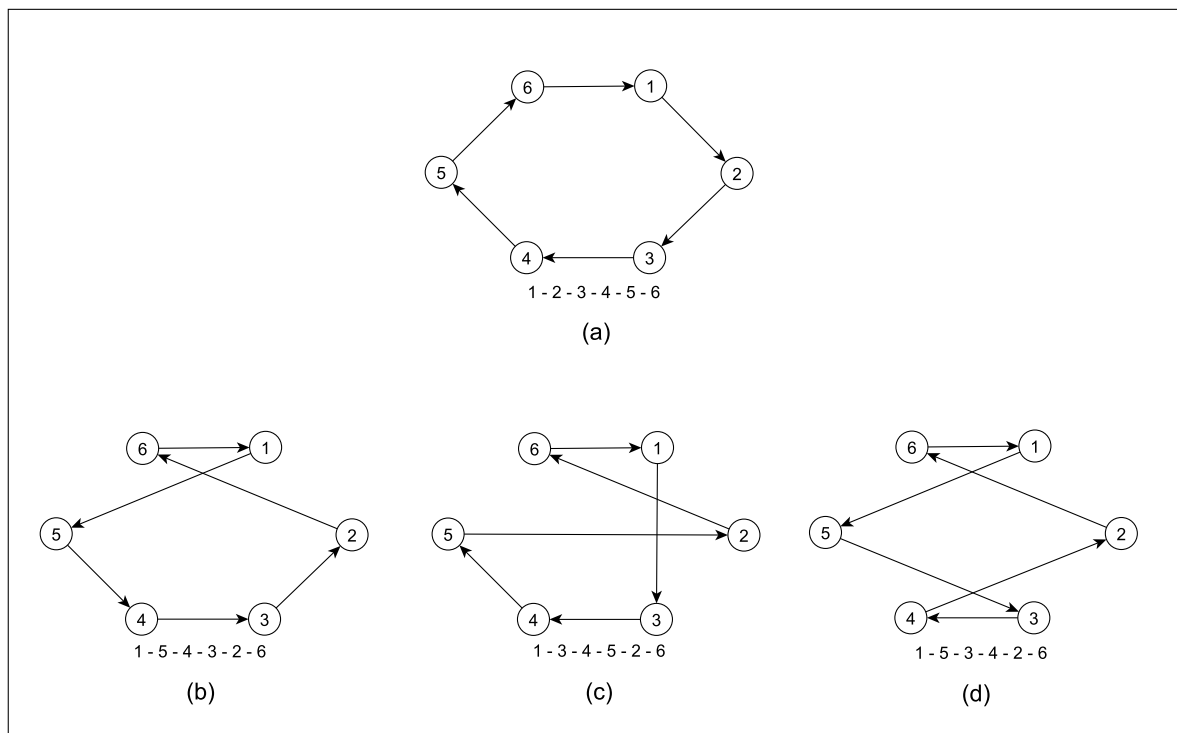


FIGURA 5.2 – Exemplos das três estruturas de vizinhança usadas no PCVMP. Em (a) um exemplo de uma rota de clientes. Em (b) uma rota obtida por 2-Opt. Em (c) uma rota obtida por Shift(1). Em (d) uma rota obtida por Swap(1,1).

Seja  $S'$  uma solução representativa decodificada, isto é, uma solução fornecida pelo processo de agrupamento e decodificada pela função *decoder*. Para cada uma dessas soluções, os Algoritmos 7 e 8 são aplicados, os quais são os procedimentos ILS e VND, respectivamente.

No Algoritmo 7, a solução inicial  $s_0 = S'$  é obtida usando o procedimento de busca local em uma região promissora determinada conforme descrito na subseção 4.8.1.  $s_0$  é a solução representativa desta região promissora. O VND é aplicado para obter uma

**Algorithm 7** Pseudocódigo do algoritmo ILS

```

1: Solução inicial  $s_0$                                 ▷ solução decodificada representativa da região
2:  $s \leftarrow \text{VND}(s_0)$ 
3:  $Iter$                                                 ▷ iteração corrente
4:  $Iter_{max} \leftarrow 3$                                 ▷ número máximo de iterações
5:  $j \leftarrow 1$                                         ▷ quantidade de perturbações realizadas
6: enquanto ( $Iter < Iter_{max}$ ) faça
7:    $Iter \leftarrow Iter + 1$ 
8:   enquanto ( $j \leq 0.1 * N$ ) faça
9:      $s' \leftarrow \text{Swap}(1, 1)$ 
10:     $j \leftarrow j + 1$ 
11:   fim enquanto
12:    $s'' \leftarrow \text{VND}(s')$ 
13:   se ( $f(s'') > f(s)$ ) então
14:      $s \leftarrow s''$ 
15:   fim se
16: fim enquanto
17: Return  $s$ 

```

**Algorithm 8** Pseudocódigo do algoritmo VND

```

1: Solução inicial  $s_0$ 
2: Número de estruturas de vizinhanças  $n_r$ 
3:  $s \leftarrow s_0$                                     ▷ iteração corrente
4:  $k \leftarrow 1$                                         ▷ tipo da estrutura de vizinhança
5: enquanto ( $k \leq n_r$ ) faça
6:   Encontre o melhor vizinho  $s' \in N^{(k)}$ .
7:   se ( $f(s') > f(s)$ ) então
8:      $s \leftarrow s'$ 
9:      $k \leftarrow 1$ 
10:  senão
11:     $k \leftarrow k + 1$ 
12:  fim se
13: fim enquanto
14: Return  $s$ 

```

solução melhor que será a solução de referência (linha 2). Nas linhas 6 a 16, a busca local VND é aplicada; nas linhas 8 a 11,  $0.1 * N$  perturbações são aplicadas na solução atual por Swap(1,1) antes de aplicar o VND. Se uma solução melhor for encontrada, a solução atual será atualizada (linhas 13 e 14). Este procedimento é aplicado iterativamente até que o número máximo de iterações seja alcançado.

No Algoritmo 8, a solução inicial  $s_0$  é obtida pelo procedimento baseado no ILS (Algoritmo 7, linhas 2 e 12). O número de estruturas de vizinhança  $n_r$  foi definido como 3, que são 2-opt, Shift (1) e Swap(1,1). O número de iterações dos algoritmos (linha 4) foi fixado em 3 devido ao tempo de execução disponível. Uma busca é realizada em todos os vizinhos até que seja obtida a melhor solução para todas as estruturas (linhas 5 a 11). Em cada estrutura de vizinhança  $k$  (linha 5), o melhor vizinho é determinado (linha 6). Sempre que uma solução melhor é encontrada, retorna-se à primeira estrutura de vizinhança (linha 9), garantindo que a solução final obtida seja a melhor para todas as estruturas de vizinhança. Quando uma solução melhor não é encontrada usando a estrutura de vizinhança de  $k$  (linhas 10 e 11), a estrutura muda para  $k + 1$ .

### 5.3 Resultados computacionais

Os algoritmos BRKGA+CS e A-BRKGA+CS foram codificados em C++ e os testes computacionais foram realizados em um computador equipado com processador Intel® Core™ i7 com 3,6 GHz e 16GB de memória RAM em um ambiente Windows 10. O *solver* ILOG CPLEX 12.6.3.0 (ILOG..., 2006) foi usado para resolver o modelo PCVMP. As duas versões do BRKGA foram executadas usando 20 sementes diferentes.

O ajuste dos parâmetros do BRKGA+CS foi realizado pelo Índice de Desvio Relativo (RDI) (KIM, 1993). O  $RDI_i$  de uma solução  $i$  com o valor objetivo  $S_i$  é definido como:

$$RDI_i = \frac{S_B - S_i}{S_B - S_W}, \quad (5.13)$$

em que  $S_B$  é o melhor valor obtido pelas metaheurísticas e  $S_W$  é o pior valor obtido pelas metaheurísticas. Diferentes combinações de parâmetros foram testadas em um subconjunto de instâncias e a configuração com o melhor RDI foi escolhida para os testes computacionais do BRKGA+CS.

Os conjuntos de parâmetros testados são mostrados na Tabela 5.1 (GONÇALVES; RESENDE, 2015). Os parâmetros encontrados são dados na Tabela 5.2, juntamente com o intervalo dos parâmetros usados no A-BRKGA+CS. As configurações de BRKGA+CS foram mantidas constantes em todas as instâncias, enquanto os parâmetros de A-BRKGA+CS variaram dentro desses intervalos, de acordo com a teoria apresentada na subseção

4.8.

TABELA 5.1 – Valores dos parâmetros utilizados no processo de ajuste do BRKGA+CS.

Parameter	Significado	Valores
$p$	tamanho da população	1000
$p_e$	tamanho da população de elite	0.10, 0.15, 0.20, 0.25, 0.30
$p_m$	tamanho da população mutante	0.15, 0.20, 0.25, 0.30
$\rho_e$	probabilidade de herança de alelos da elite	0.70, 0.75, 0.80, 0.85

TABELA 5.2 – Parâmetros usados no BRKGA+CS e no A-BRKGA+CS.

Parâmetro	BRKGA+CS	A-BRKGA+CS
	Valor	Intervalo de valores
$p$	1000	[100, 1000]
$Gen$	300	271
$p_e$	0.15	[0.10, 0.25]
$p_m$	0.20	[0.05, 0.20]
$\rho_e$	0.70	[0.65, 0.80]

As instâncias usadas nos testes foram geradas a partir da biblioteca *TSPLib* (REINELT, 2013) e foram baseadas na tese de Sarubbi (SARUBBI, 2008). Cada cliente tem uma demanda inteira que variou entre 1 e  $\tau$  (demanda máxima), na qual  $\tau$  variou entre 5 e 20 unidades. O depósito, ou nó inicial, sempre tem demanda de 1 unidade. Observe que as demandas podem ser qualquer número real positivo. Nesse caso, o valor das variáveis  $f_{ijl}$  em uma solução ótima será um número real. O produto  $\gamma_l * c_{ij}$  representa o custo de transportar o produto  $l$  pelo arco  $(i, j)$ . Quanto melhores as condições de transporte de um arco  $(i, j)$ , menor o valor do parâmetro correspondente. Este parâmetro  $\gamma_l$  variou de 0,5% a 2%. Cada cliente também tem um valor  $\eta_l$  correspondente. Isso corresponde ao valor relativo do produto e, quanto mais valioso for o produto, maior será o valor deste parâmetro. Os parâmetros  $\eta_l$  foram gerados aleatoriamente entre 0.5 e 1.5. As prioridades foram gerados aleatoriamente entre 0 e 100, considerando apenas valores inteiros (SILVA, 2017). Todos esses parâmetros foram gerados usando uma distribuição uniforme. No A-BRKGA+CS, o parâmetro  $\gamma$  foi fixado em 0.999 uma vez que o tempo de execução implicado por ele foi suficiente para encontrar soluções satisfatórias.

Todas as instâncias satisfazem a desigualdade triangular em relação aos custos fixos. Cada instância tem seu número de clientes especificado em seu nome. Por exemplo, a Instância **burma14** tem 13 clientes para atender e o depósito. Na Tabela 5.3, todas as instâncias usadas nos testes computacionais são apresentadas divididas pela quantidade de clientes.

Três conjuntos de instâncias foram gerados, diferindo entre si pelo parâmetro  $\theta$  multiplicado pelas prioridades:

TABELA 5.3 – As instâncias simétricas selecionadas no *TSPLIB* (REINELT, 2013), por tamanhos de instância. A instância dj38 foi selecionada em *TSP Test Data* (REINELT, 2009).

1-50 clientes	51-100 clientes	101-150 clientes
burma14	eil51	eil101
ulysses16	berlin52	lin105
gr17	brazil58	pr107
gr21	st70	gr120
ulysses22	eil76	pr124
gr24	pr76	bier127
fri26	gr96	ch130
bays29	rat99	
dj38	kroA100	
dantzig42	kroB100	
att48		
gr48		
hk48		

- Conjunto A:  $\theta = \bar{c}$ ,
- Conjunto B:  $\theta = \frac{\bar{c}}{4(N-1)\bar{p}} \left( 2N + \sum_{l=2}^N z_l \right)$ ,
- Conjunto C:  $\theta = 1$ ,

em que  $\bar{c} = \frac{1}{N(N-1)} \sum_{i=1}^N \sum_{\substack{j=1 \\ i \neq j}}^N c_{ij}$ ,  $\bar{p} = \frac{1}{(N-1)^2} \sum_{k=2}^N \sum_{i=2}^N p_{ki}$ , e  $z_l = \gamma_l \eta_l q_l$ .

Este parâmetro  $\theta$  serve apenas para gerar instâncias diferentes. O conjunto A torna o PCVMP mais próximo do problema de atribuição, pois as prioridades são muito maiores que os custos fixos. O conjunto B é um equilíbrio entre o problema de atribuição e o problema do caixeiro viajante, tentando zerar o valor da função objetivo, e o conjunto C torna o PCVMP mais próximo do problema do caixeiro viajante, pois as prioridades são muito menores que os custos fixos.

Nas tabelas 5.4, 5.5 e 5.6 são apresentados os resultados para o CPLEX, A-BRKGA+CS e BRKGA+CS para os conjuntos A, B e C, respectivamente. As entradas nas tabelas são as instâncias; a solução obtida pelo CPLEX ( $sol_C$ ) dentro do prazo de 10,800 segundos, a melhor solução ( $S^*$ ), a solução média ( $S$ ) considerando 20 execuções, o tempo para encontrar a melhor solução ( $T^*$ ), o tempo médio de execução ( $T$ ) em segundos, o desvio ( $Dev$ ) definido como  $100 \times (S - S^*) / S^*$  e o GAP fornecido pelo CPLEX definido como  $(|limite superior - limite inferior| / |limite inferior|)$ . As entradas “-” na coluna  $sol_C$  significam que o CPLEX não conseguiu encontrar uma solução viável dentro do tempo predefinido.

Na Tabela 5.8 são apresentados os resultados para a relaxação linear comparada com as

soluções das metaheurísticas propostas e do CPLEX. Os gaps foram calculados conforme a Equação (5.14), em que  $S_{RL}$  é a solução da relaxação linear,  $S_C$  é a solução obtida pelo CPLEX,  $S_A$  é a solução obtida pelo A-BRKGA+CS e  $S_B$  é a solução obtida pelo BRKGA+CS. Na coluna Tempo(s) são apresentados os tempos usados pelo CPLEX para resolver a relaxação linear do modelo PCVMP nas instâncias dos conjuntos A, B e C.

$$GAP_{RL}^i = \frac{|S_{RL} - S_i|}{|S_{RL}|}, \quad i = A, B, C. \quad (5.14)$$

O CPLEX não encontrou uma solução factível para todas as instâncias dentro do tempo predefinido. Para o conjunto A, o CPLEX conseguiu resolver 8 instâncias na otimalidade dentro do tempo limite predefinido e 6 instâncias com GAPS menores que 4%. Porém, não encontrou solução factível em 14 das 30 instâncias. Para os conjuntos B e C, o CPLEX resolveu apenas duas instâncias na otimalidade e não encontrou soluções viáveis em 13 instâncias.

Quando as prioridades são muito maiores que os custos fixos (Conjunto A), os custos do problema de atribuição dominaram a função objetivo geral e o CPLEX foi capaz de resolver o modelo PCVMP em mais instâncias na otimalidade. O problema de atribuição possui a propriedade de integralidade, ou seja, a solução da relaxação linear é a solução ótima do problema, o que pode explicar o menor tempo de execução dessas instâncias em comparação com as instâncias dos conjuntos B e C. Como consequência, a relaxação linear do modelo PCVMP se aproximou da solução ótima. Na Tabela 5.8, os valores de  $GAP_{RL}^C$  mostram a distância relativa entre a relaxação linear e a solução ótima (ou melhor solução fornecida pelo CPLEX). Seis instâncias tiveram  $GAP_{RL}^C$  menor que 4.5%, sem contar as instâncias **eil51** e **berlin52**, em que as soluções encontradas pelo CPLEX dentro do tempo limite não foram muito boas, e outras instâncias em que o CPLEX não encontrou uma solução factível.

Nos Conjuntos B e C, a solução da relaxação linear não apresentou bons resultados, produzindo gaps em torno de 100% e, em alguns casos, gaps ainda maiores, como nos casos das instâncias **gr48**, **berlin52** e **brazil58**, com gaps de 3.742, 34%, 1.883, 20% e 1.157, 06%, respectivamente, no conjunto C.

Nas Tabelas 5.4, 5.5 e 5.6, pode ser observado que o A-BRKGA+CS teve desempenho pior que o BRKGA+CS para as instâncias do Conjunto A. A primeira metaheurística foi melhor em 30% das instâncias, enquanto a segunda foi melhor em 43%. Ambas as metaheurísticas encontraram as mesmas soluções em 26% das instâncias. No conjunto B, o A-BRKGA+CS foi melhor em 46% das instâncias, enquanto o BRKGA+CS foi melhor em 23%. No conjunto C, o A-BRKGA+CS foi melhor em 33% das instâncias, enquanto o BRKGA+CS foi melhor em 20%. Observe que, quando as penalidades nas prioridades diminuem, o número de instâncias nas quais as metaheurísticas retornam a mesma solução

umenta de 26% em A para 30% e 46% em B e C.

Para comparar os dois conjuntos de soluções fornecidos pelo A-BRKGA+CS e BRKGA+CS para cada conjunto e analisar se há uma diferença significativa entre eles, o teste de postos sinalizados de Wilcoxon (WSR) (WILCOXON, 1945) e o teste de Friedman (FRIEDMAN, 1940) foram aplicados. Os resultados dos testes são mostrados na Tabela 5.7, em que  $Z$  é a soma dos postos sinalizados do teste WSR, Qui-quadrado é a estatística do teste e  $df$  é o grau de liberdade do teste de Friedman. Pela tabela, conclui-se que os ranks de A-BRKGA+CS e BRKGA+CS são estatisticamente equivalentes (teste WSR) e as distribuições dos *scores* para os métodos comparados são iguais ( $p$ -valor  $> 0.001$  - teste de Friedman) para todos os conjuntos, em um nível de significância de 0.05.

Nas Tabelas 5.4, 5.5 e 5.6, observa-se que o modelo PCVMP proposto encontrou soluções melhores que as metaheurísticas propostas para três instâncias do conjunto A: `dantzig42`, `gr48` e `hk48`. Os gaps das soluções obtidas pelo A-BRKGA+CS, para essas instâncias, são 0.76%, 1.51% e 1.54%, respectivamente; e, para as soluções obtidas pelo BRKGA+CS, 0.77%, 1.69% e 1.54%, respectivamente. As soluções obtidas pelos modelos apresentaram gaps de 0.52%, 1.22% e 1.32%. Por outro lado, o modelo encontrou soluções piores para duas instâncias, `eil51` e `berlin52`, com gaps de 44.99% e 55.86%, respectivamente; enquanto A-BRKGA+CS obteve soluções com gaps de 1.97% e 2.69%, e BRKGA+CS obteve soluções com gaps de 1.96% e 2.44%. Para todas as outras instâncias nos Conjuntos B e C, o modelo proposto teve desempenho pior que os métodos heurísticos propostos.

## 5.4 Considerações finais

Neste capítulo, apresentamos uma nova variante do Problema do Caixeiro Viajante, que considera custos variáveis e prioridades, além de custos fixos. Um modelo matemático foi proposto para representar o problema, usando um modelo de fluxo multicomoditie. Dois métodos foram propostos para encontrar soluções factíveis para esse problema. O primeiro consiste em um algoritmo genético viciado de chaves aleatórias (BRKGA) com um método de detecção de comunidades “*Label propagation*” (LP). Para a busca local, foi proposto um método baseado no “*Iterated Local Search*” (ILS) combinado com um “*Variable Neighborhood Descent*” (VND). O segundo método consiste em um BRKGA adaptativo com o mesmo método de detecção de comunidades e processo de busca local.

Para os experimentos computacionais, 90 instâncias diferentes foram geradas com base nas instâncias da biblioteca *TSPLib*, *TSP Test Data* e na tese (SARUBBI, 2008). Foram geradas três penalidades para as prioridades, resultando em três conjuntos de instâncias: O conjunto A que consiste em penalizar as prioridades pela média dos custos fixos; no

TABELA 5.4 – Resultados do CPLEX, BRKGA+CS e A-BRKGA+CS para instâncias do conjunto A.

Instância	CPLEX				A-BRKGA+CS				BRKGA+CS					
	sol <sub>C</sub>	T*	T	GAP	S*	S	T*	T	dev	S*	S	T*	T	dev
burma14	540978.745	0.16	0.17	0.00%	540978.745	540978.75	0.06	5.97	0.00	540978.745	540978.75	0.08	5.32	0.00
ulysses16	1045426.76	0.52	0.55	0.00%	1045426.76	1045426.56	0.07	7.96	0.00	1045426.76	1045426.56	0.08	6.07	0.00
gr17	391773.68	0.44	0.47	0.00%	391773.68	391773.66	0.06	8.94	0.00	391773.68	391773.66	0.13	6.87	0.00
gr21	667131.56	32.67	38.64	0.00%	667131.56	667131.56	5.90	16.18	0.00	667131.56	667032.30	5.83	10.95	-0.01
ulysses22	1408527.99	75.59	92.73	0.00%	1408527.99	1408433.00	9.21	16.75	-0.01	1408527.99	1407771.00	0.67	12.50	-0.05
gr24	313651.42	5.89	48.70	0.00%	313651.42	313619.10	9.33	22.18	-0.01	313651.42	313535.60	16.53	15.48	-0.04
fri26	237759.19	206.00	223.45	0.00%	237759.19	237690.10	1.54	28.41	-0.03	237759.19	237624.30	12.32	18.41	-0.06
bays29	539347.37	36.31	378.92	0.00%	539347.37	538961.70	14.83	38.43	-0.07	539347.37	538981.00	13.70	28.95	-0.07
dantzig42	285605.55	1991.34	10800.00	0.52%	2471873.25	2466715.00	56.85	87.48	-0.21	2472919.00	2466184.00	68.47	85.12	-0.27
att48	4559007.06	9603.00	10800.00	2.51%	284900.28	284180.30	11.48	131.21	-0.25	284895.28	284191.00	42.76	124.45	-0.25
gr48	1935290.36	2480.66	10800.00	1.32%	4587733.50	4581998.00	64.47	206.49	-0.13	4597805.00	4584123.00	60.46	219.78	-0.30
hk48	4511601.81	7251.27	10800.00	1.22%	1929587.50	1921895.00	86.37	209.56	-0.40	1926223.75	1921710.00	10.55	232.50	-0.23
ei151	85482.75	9967.59	10800.00	44.99%	152334.92	151913.60	147.67	250.58	-0.28	152357.78	151925.60	209.69	261.20	-0.28
berlin52	1235108.12	4929.02	10800.00	55.86%	2722884.50	2718976.00	13.28	267.92	-0.14	2729865.75	2720083.00	178.67	323.06	-0.36
brazil58	11281984.18	9634.48	10800.00	3.87%	11461121.00	2718976.00	117.48	385.85	-76.28	11463258.00	11432682.00	35.14	462.65	-0.27
st70	-	-	10800.00	-	338038.62	337051.00	1091.21	854.01	-0.29	337896.59	337118.70	274.37	880.66	-0.23
ei176	-	-	10800.00	-	236625.89	236198.70	721.98	914.20	-0.18	236852.78	236179.50	276.21	923.28	-0.28
pr76	-	-	10800.00	-	52913480.00	52733805.00	1414.22	1163.21	-0.34	52959868.00	52768027.00	604.74	1388.64	-0.36
gr96	-	-	10800.00	-	33820656.00	33737586.00	487.36	1500.00	-0.25	33830080.00	33737084.00	293.76	1500.00	-0.27
rat99	-	-	10800.00	-	7822456.37	780396.80	50.38	1500.00	-0.26	781264.31	780111.00	1384.77	1500.00	-0.15
kroA100	-	-	10800.00	-	16027485.00	15995068.00	1054.29	1494.67	-0.20	16007513.00	15993466.00	801.31	1500.00	-0.09
kroB100	-	-	10800.00	-	15507184.00	15458129.00	696.23	1500.00	-0.32	15517892.00	15464435.00	636.95	1500.00	-0.34
ei1101	-	-	10800.00	-	317354.22	316515.60	10.79	1500.00	-0.26	316745.34	316341.80	1020.83	1500.00	-0.13
lin105	-	-	10800.00	-	11429048.00	11413877.00	583.18	1500.00	-0.13	11441579.00	11419229.00	1325.33	1500.00	-0.20
pr107	-	-	10800.00	-	54178324.00	54063147.00	448.93	1500.00	-0.21	54177276.00	54071862.00	1500.00	1500.00	-0.19
gr120	-	-	10800.00	-	4892669.50	4885372.00	395.97	1500.00	-0.15	4896112.50	4886659.00	1185.42	1500.00	-0.19
pr124	-	-	10800.00	-	65399888.00	65259224.00	1021.24	1500.00	-0.22	65371232.00	65263581.00	574.86	1500.00	-0.16
bier127	-	-	10800.00	-	57905416.00	57718236.00	1364.71	1500.00	-0.32	57894204.00	57706235.00	250.94	1500.00	-0.32
ch130	-	-	10800.00	-	4246087.50	4234849.00	1121.07	1500.00	-0.26	4254836.00	4237412.00	841.52	1500.00	-0.41



TABELA 5.5 – Resultados do CPLEX, BRKGA+CS e A-BRKGA+CS para instâncias do conjunto B.

Instância	CPLEX				A-BRKGA+CS				BRKGA+CS					
	sol <sub>C</sub>	T*	T	GAP	S*	S	T*	T	Dev	S*	S	T*	T	Dev
burma14	<b>-450.99</b>	7.84	23.23	0.00%	<b>-450.99</b>	-450.99	0.05	5.95	0.00	<b>-450.99</b>	-450.99	0.07	5.06	0.00
ulysses16	<b>398.77</b>	1471.41	1645.98	0.00%	<b>398.77</b>	398.77	0.10	8.21	0.00	<b>398.77</b>	398.77	0.20	5.88	0.00
gr17	<b>-255.93</b>	5730.20	10800.00	222.31%	<b>-255.93</b>	-255.93	0.55	8.52	0.00	<b>-255.93</b>	-255.93	0.10	6.22	0.00
gr21	352.03	10660.22	10800.00	634.40%	<b>1036.21</b>	1036.21	0.36	14.69	0.00	<b>1036.21</b>	1036.21	0.10	8.94	0.00
ulysses22	-1815.79	10763.17	10800.00	403.05%	<b>-114.06</b>	-114.06	0.21	16.90	0.00	<b>-114.06</b>	-114.06	0.50	10.23	0.00
gr24	422.16	10258.19	10800.00	240.77%	<b>592.70</b>	592.70	0.11	20.17	0.00	<b>592.70</b>	592.70	1.05	11.83	0.00
fri26	-584.55	504.84	10800.00	254.84%	<b>154.16</b>	154.16	3.62	25.24	0.00	<b>154.16</b>	152.16	7.90	13.94	-1.30
bays29	-792.27	994.81	10800.00	465.78%	<b>1033.42</b>	1033.42	10.57	36.20	0.00	<b>1033.42</b>	1033.42	2.30	18.51	0.00
dj38	-14921.94	4261.42	10800.00	200.18%	<b>79.70</b>	48.17	63.02	78.82	-39.55	<b>79.70</b>	59.99	27.56	39.37	-24.73
dantzig42	-377.75	2747.42	10800.00	665.69%	<b>1181.95</b>	1170.08	5.52	119.11	-1.00	<b>1181.95</b>	1161.66	58.72	67.77	-1.72
att48	-34997.32	5680.05	10800.00	202.74%	<b>11509.23</b>	11000.21	26.74	180.89	-4.42	<b>11384.19</b>	10618.18	4.15	93.24	-6.73
gr48	-15186.18	6600.11	10800.00	195.60%	<b>3222.88</b>	2968.09	142.21	170.24	-7.91	<b>3120.87</b>	2846.64	41.52	89.26	-8.79
hk48	-28065.98	9041.84	10800.00	217.55%	<b>10385.87</b>	9691.58	73.52	188.80	-6.68	<b>10132.88</b>	9507.07	83.61	96.82	-6.18
eil151	-2021.21	9783.72	10800.00	151.81%	<b>-4.75</b>	-22.89	55.46	208.29	381.89	<b>-6.71</b>	-31.93	107.82	117.95	375.86
berlin52	-48972.09	7874.70	10800.00	140.29%	<b>150.08</b>	-228.77	75.36	209.51	-252.43	<b>104.42</b>	-225.55	88.93	109.00	-316.00
brazil58	-189031.01	3904.91	10800.00	150.20%	<b>24532.86</b>	22405.51	172.21	333.56	-8.67	<b>23626.16</b>	21742.41	129.32	171.02	-7.97
st70	-9299.25	10800.00	10800.00	130.54%	<b>-76.13</b>	-124.23	128.42	669.21	63.18	<b>-49.33</b>	-125.55	159.07	365.88	154.51
eil76	-	-	10800.00	-	672.78	645.64	135.41	849.89	-4.03	<b>676.98</b>	633.26	342.44	570.33	-6.46
pr76	-	-	10800.00	-	<b>-128443.44</b>	-157740.40	329.10	730.39	22.81	<b>-138540.61</b>	-165986.80	278.70	444.19	19.81
gr96	-	-	10800.00	-	<b>-3591.47</b>	-17948.34	878.12	1344.83	399.75	<b>-10091.05</b>	-18797.72	241.18	1176.87	86.28
rat99	-	-	10800.00	-	<b>-39.85</b>	-215.76	675.15	1454.31	441.43	<b>-84.85</b>	-247.05	270.28	1428.41	191.16
kroA100	-	-	10800.00	-	<b>45349.34</b>	42688.74	773.23	1403.66	-5.87	<b>45022.53</b>	41844.28	81.78	1295.84	-7.06
kroB100	-	-	10800.00	-	<b>-12458.55</b>	-15460.95	582.39	1392.15	24.10	<b>-12964.93</b>	-15100.84	228.66	1259.72	16.47
eil101	-	-	10800.00	-	<b>-324.46</b>	-398.00	707.71	1481.23	22.67	<b>-297.43</b>	-393.02	1268.13	1456.59	32.14
lin105	-	-	10800.00	-	12579.35	10738.05	210.91	1500.00	-14.64	<b>14557.37</b>	11767.65	298.28	1450.78	-19.16
pr107	-	-	10800.00	-	258159.51	247378.60	1320.24	1428.66	-4.18	<b>263117.72</b>	249222.40	1445.38	1471.89	-5.28
gr120	-	-	10800.00	-	<b>7569.26</b>	6243.56	282.75	1500.00	-17.51	<b>7110.58</b>	6047.70	118.76	1500.00	-14.95
pr124	-	-	10800.00	-	251567.62	237369.20	1167.08	1500.00	-5.64	<b>259163.23</b>	235636.60	623.12	1396.29	-9.08
bier127	-	-	10800.00	-	<b>-190174.28</b>	-199969.60	980.34	1500.00	5.15	<b>-193390.01</b>	-204634.10	616.58	1500.00	5.81
ch130	-	-	10800.00	-	<b>-13712.49</b>	-14965.97	772.34	1500.00	9.14	<b>-14051.79</b>	-15269.40	601.10	1500.00	8.67

TABELA 5.6 – Resultados do CPLEX, BRKGA+CS e A-BRKGA+CS para instâncias do conjunto C.

Instância	CPLEX				A-BRKGA+CS				BRKGA+CS					
	sol <sub>C</sub>	T*	T	GAP	S*	S	T*	T	Dev	S*	S	T*	T	Dev
burma14	-4382.58	35.95	37.80	0.00%	-4382.58	-4382.58	0.05	5.59	0.00	-4382.58	-4382.58	0.07	4.89	0.00
ulysses16	-7552.15	38.16	2027.34	0.00%	-7552.15	-7552.15	0.06	7.05	0.00	-7552.15	-7552.15	0.09	5.63	0.00
gr17	-2533.05	4154.95	10800.00	17.57%	-2518.18	-2518.18	0.07	8.74	0.00	-2518.18	-2518.18	0.49	6.06	0.00
gr21	-2952.52	9950.06	10800.00	43.08%	-2793.71	-2793.71	0.08	12.42	0.00	-2793.71	-2793.71	0.11	8.49	0.00
ulysses22	-11616.90	10765.53	10800.00	32.64%	-10726.12	-10726.12	0.09	17.15	0.00	-10726.12	-10726.12	0.25	9.63	0.00
gr24	-709.25	10784.16	10800.00	152.20%	-381.60	-381.60	1.15	18.30	0.00	-381.60	-381.60	0.18	11.57	0.05
fri26	-544.74	10800.00	10800.00	221.74%	-54.05	-54.05	1.30	25.14	0.00	-54.05	-54.05	0.18	13.81	0.00
bays29	-2454.39	10358.38	10800.00	94.92%	-1526.08	-1526.08	0.67	30.63	0.00	-1526.08	-1526.08	10.56	17.09	0.00
dj38	-33628.32	7093.38	10800.00	77.21%	-16323.99	-16323.99	0.44	67.68	0.00	-16323.99	-16323.99	0.23	33.24	0.00
dantzig42	133.32	3219.05	10800.00	2129.47%	1944.54	1923.37	126.69	121.21	-1.09	1944.54	1913.10	41.58	73.45	-1.62
at48	-51523.41	8275.50	10800.00	86.15%	-23027.95	-23106.67	3.98	155.08	0.34	-23027.95	-23138.10	12.12	77.35	0.48
gr48	-25912.34	8749.89	10800.00	95.59%	-9327.03	-9345.08	1.07	141.69	0.19	-9327.03	-9346.21	9.10	75.75	0.21
hk48	-48439.47	9783.30	10800.00	82.37%	-22170.77	-22198.52	49.06	139.38	0.13	-22170.77	-22243.41	57.04	71.20	0.33
eil151	403.81	10431.31	10800.00	919.73%	2699.44	2646.44	174.89	258.03	-1.96	2686.90	2617.99	86.03	129.27	-2.56
berlin52	-64634.27	9877.58	10800.00	93.95%	-18296.24	-18352.20	167.28	221.52	0.31	-18317.84	-18456.44	37.59	86.54	0.76
brazil58	-209329.54	6962.56	10800.00	91.30%	-61384.78	-61661.89	216.27	303.81	0.45	-61384.78	-61688.62	41.83	158.59	0.49
st70	-7354.87	10800.00	10800.00	178.10%	2370.46	2191.65	177.29	611.87	-7.54	2390.72	2162.09	181.43	373.85	-9.56
eil176	-	-	10800.00	-	4914.68	4821.67	720.53	734.54	-1.89	4898.41	4814.25	792.84	720.51	-1.72
pr76	-	-	10800.00	-	-524359.37	-530491.80	343.97	689.38	1.17	-526192.50	-531420.90	110.46	396.56	0.99
gr96	-	-	10800.00	-	-244170.83	-248055.80	1087.66	1308.98	1.59	-243754.44	-248103.70	26.86	1118.25	1.78
rat99	-	-	10800.00	-	735.25	522.96	505.07	1447.80	-28.87	632.99	518.22	48.93	1404.39	-18.13
kroA100	-	-	10800.00	-	-64474.61	-65007.84	198.95	1257.57	0.83	-64532.23	-65066.75	274.64	1170.08	0.83
kroB100	-	-	10800.00	-	-117031.14	-118457.90	541.86	1287.98	1.22	-117720.92	-118632.80	786.13	1232.18	0.77
eil101	-	-	10800.00	-	4656.28	4596.09	397.79	1483.00	-1.29	4745.54	4600.92	1211.42	1319.26	-3.05
lin105	-	-	10800.00	-	-62598.05	-63369.07	1088.46	1333.40	1.23	-61896.99	-63097.24	281.51	1427.69	1.94
pr107	-	-	10800.00	-	-163506.89	-165193.50	818.66	1376.97	1.03	-163872.11	-165041.10	31.42	1426.13	0.71
gr120	-	-	10800.00	-	-20219.13	-21148.39	580.17	1500.00	4.60	20537.48	-21140.62	1393.47	1500.00	-202.94
pr124	-	-	10800.00	-	-196417.47	-200211.80	1474.34	1500.00	1.93	-195745.12	-200318.50	73.23	1500.00	2.34
bier127	-	-	10800.00	-	-563419.06	-572220.80	1070.57	1500.00	1.56	-566039.37	-572754.50	1074.70	1500.00	1.19
chi30	-	-	10800.00	-	-35434.74	-36368.38	244.92	1500.00	2.63	-35063.98	-36300.84	1278.32	1500.00	3.53

TABELA 5.7 – Resultado dos testes WSR e Friedman para os conjuntos A, B e C.

Conjunto	Teste WSR		Teste Friedman		
	Z	p-valor	Chi-quadrado	df	p-valor
A	97	0.3464	0.7273	1	0.3938
B	164	0.0952	2.3333	1	0.1266
C	80	0.5521	1.0000	1	0.3173

conjunto B, a penalidade é usar um equilíbrio entre custos fixos e custos variáveis; e o Conjunto C não usou nenhuma penalidade.

O modelo proposto foi capaz de fornecer boas soluções para as instâncias do Conjunto A, mas não para as instâncias dos Conjuntos B e C. Isso ocorreu devido a sua relaxação linear. Enquanto no Conjunto A o modelo de relaxação linear era muito bom, aproximando-se da solução ótima, nos Conjuntos B e C, apresentou desempenho ruim. Por causa das penalidades, as instâncias do Conjunto A aproximaram o PCVMP do problema de atribuição, que possui propriedade de integralidade, em que a solução de relaxação linear é a solução ótima do modelo inteiro.

Ambos os métodos heurísticos propostos tiveram bom desempenho quanto a tempo computacional e qualidade da solução nas instâncias testadas. Em geral, o método A-BRKGA+CS encontrou melhores soluções (33) que o método BRKGA+CS (26), embora não tenha sido estatisticamente diferente de acordo com os testes de Wilcoxon e Friedman realizados. A principal diferença entre os dois métodos é que o método BRKGA+CS precisa ser calibrado, enquanto o método A-BRKGA+CS é adaptável.

TABELA 5.8 – Resultados da relaxação linear do modelo PCVMP nos conjuntos A, B e C.

Instância	Conjunto A				Conjunto B				Conjunto C			
	$GAP_{RL}^C$	$GAP_{RL}^A$	$GAP_{RL}^B$	Time(s)	$GAP_{RL}^C$	$GAP_{RL}^A$	$GAP_{RL}^B$	Tempo(s)	$GAP_{RL}^C$	$GAP_{RL}^A$	$GAP_{RL}^B$	Time(s)
burma14	0.22%	0.22%	0.22%	0.12	119.77%	119.77%	119.77%	1.64	72.76%	72.76%	72.76%	0.04
ulysses16	0.01%	0.01%	0.01%	0.35	92.01%	92.01%	92.01%	0.05	53.35%	53.35%	53.35%	0.06
gr17	0.19%	0.19%	0.19%	0.72	112.06%	112.06%	112.06%	0.06	200.05%	198.29%	198.29%	0.07
gr21	0.55%	0.55%	0.55%	1.17	90.90%	73.21%	73.21%	0.15	188.83%	173.30%	173.30%	0.13
ulysses22	1.07%	1.07%	1.07%	1.59	119.99%	101.26%	101.26%	0.14	135.90%	117.81%	117.81%	0.14
gr24	0.36%	0.36%	0.36%	2.08	78.27%	69.49%	69.49%	0.24	186.79%	146.70%	146.70%	0.75
fri26	0.59%	0.59%	0.59%	3.74	147.00%	87.60%	87.60%	2.11	154.80%	105.44%	105.44%	2.15
bays29	0.54%	0.54%	0.54%	5.90	122.37%	70.82%	70.82%	0.58	609.51%	416.80%	416.80%	0.55
dj38	2.31%	2.16%	2.12%	35.85	179.79%	99.57%	99.57%	1.39	564.35%	222.49%	222.49%	1.53
dantzig42	0.81%	1.05%	1.05%	45.53	116.00%	49.93%	49.93%	1.13	95.90%	40.13%	40.13%	1.11
att48	2.94%	2.33%	2.12%	85.63	189.79%	70.47%	70.79%	3.02	676.70%	247.14%	247.14%	1.88
gr48	1.50%	1.79%	1.96%	431.68	196.35%	79.55%	80.20%	14.10	3742.34%	1283.03%	1283.03%	5.39
hk48	1.59%	1.81%	1.81%	273.59	179.40%	70.62%	71.33%	4.38	506.37%	177.54%	177.54%	4.03
eil51	45.15%	2.26%	2.24%	400.22	277.89%	100.42%	100.59%	3.31	90.64%	37.42%	37.71%	2.55
berlin52	56.01%	3.03%	2.78%	1105.91	330.44%	99.29%	99.51%	6.42	1883.20%	461.39%	462.05%	5.89
brazil58	4.14%	2.61%	2.60%	572.05	289.09%	75.46%	76.37%	16.05	1157.06%	268.63%	268.63%	11.55
st70	-	3.30%	3.34%	1170.60	406.77%	102.51%	101.63%	15.34	223.23%	60.28%	59.94%	9.28
eil76	-	2.11%	2.02%	2102.69	-	64.98%	64.77%	19.21	-	26.94%	27.19%	26.20
pr76	-	4.02%	3.94%	3796.21	-	127.57%	129.74%	25.38	-	377.10%	378.77%	33.72
gr96	-	4.40%	4.38%	13350.90	-	101.18%	103.31%	57.53	-	333.25%	332.51%	115.79
rat99	-	3.95%	4.10%	23401.10	-	100.56%	101.20%	50.89	-	90.92%	92.19%	38.53
kroA100	-	3.42%	3.54%	13896.50	-	69.99%	70.20%	49.99	-	435.84%	436.32%	96.37
kroB100	-	4.86%	4.79%	9882.29	-	108.34%	108.68%	121.51	-	807.95%	813.30%	140.18
eil101	-	4.13%	4.31%	13409.00	-	111.70%	110.73%	78.29	-	48.52%	47.54%	87.04
lin105	-	4.62%	4.51%	16668.70	-	89.14%	87.43%	101.11	-	6326.84%	6254.86%	127.93
pr107	-	3.69%	3.69%	12816.00	-	53.18%	52.28%	59.36	-	734.29%	736.15%	93.32
gr120	-	3.89%	3.82%	100319.00	-	83.21%	84.22%	215.40	-	565.48%	-372.81%	198.73
pr124	-	3.62%	3.66%	118643.00	-	61.55%	60.39%	135.76	-	484.36%	482.36%	124.63
bier127	-	5.45%	5.47%	79501.30	-	136.23%	136.84%	448.98	-	425.33%	427.77%	566.28
ch130	-	5.93%	5.74%	111147.00	-	132.72%	133.53%	236.33	-	666.67%	660.74%	348.63

## 6 Conclusões

Neste trabalho, modelos matemáticos de otimização baseados em fluxo *multicommodity* foram propostos para dois problemas de otimização combinatória: o Problema de Minimização de Trocas de Ferramentas (PMTF) e o Problema do Caixeiro Viajante Multiproduto com Prioridades (PCVMP). [As principais contribuições deste trabalho além da contribuição descrita na frase anterior são:](#)

- O valor de relaxação linear fornecido pelos novos modelos desenvolvidos para o PMTF é pelo menos a diferença entre o número de ferramentas diferentes e a capacidade de ferramentas na máquina, e não depende da instância a ser resolvida;
- O limite inferior fornecido pelos novos modelos para o PMTF é, em geral, melhor do que os limites inferiores dados por outros modelos da literatura;
- Uma restrição que elimina metade das possíveis soluções simétricas para o PMTF foi proposta;
- O CPLEX, usando o melhor modelo de fluxo para o PMTF, determinado através de testes computacionais, conseguiu resolver 28,43% mais instâncias da literatura em comparação com outros modelos conhecidos anteriores;
- Ao relaxar as variáveis de fluxo do modelo desenvolvido para o PMTF, a solução e valor ótimo da relaxação linear associada são inteiros;
- As restrições de eliminação de subrota no modelo de fluxo desenvolvido para o PCVMP são implícitas.

Considerando o PMTF com tamanhos uniformes de ferramentas, foram propostos cinco modelos de otimização linear inteira mista, todos eles tendo relaxação linear igual ao limitante inferior trivial, ou seja, a diferença entre a quantidade de ferramentas e a capacidade da máquina. Testes computacionais realizados indicaram uma performance superior do Modelo denominado FM-PMTF-1D em relação aos demais modelos de fluxo propostos. O melhor desempenho pode ser explicado pelo fato deste modelo ter menos variáveis de decisão e menos restrições. Este modelo foi comparado com os modelos

anteriormente propostos na literatura, e observou-se um desempenho superior tanto em quantidade de instâncias resolvidas na otimalidade, quanto em tempo computacional.

Duas matheurísticas foram propostas para resolver o PMTF, ambas baseadas em busca por proximidade. Os resultados mostraram que considerando os conjuntos de instâncias de pequeno e médio portes, a matheurística VNDLB teve um desempenho satisfatório, obtendo as mesmas soluções que o estado da arte, ou fornecendo soluções próximas. Entretanto, nas instâncias de grande porte, a VNDLB apresentou um desempenho não competitivo. Um método exato baseado em ordenamento de tarefas, visando a eliminação de simetrias, também foi proposto. O método obteve resultados limitados resolvendo algumas instâncias na otimalidade e fornecendo limitantes inferiores melhores que os obtidos pelo CPLEX, considerando um tempo máximo de execução de 3.600 segundos.

O outro problema abordado, PCVMP, também foi modelado por fluxo *multicommodity*. O modelo obteve bons resultados quando aplicado à pequenas instâncias. Dois BRKGAs também foram propostos para resolução do PCVMP, sendo um deles, um BRKGA adaptativo. Os resultados foram muito satisfatórios e não foram comparados com nenhum outro da literatura, uma vez que este problema, de nosso conhecimento, ainda não havia sido abordado na literatura.

Os resultados obtidos indicam que modelar problemas via fluxo *multicommodity* pode ser interessante e promissor. O uso destes modelos dentro de técnicas metaheurísticas podem ser também explorados.

Os resultados apresentados nesta tese refletem os desenvolvimentos feitos até a data da defesa. Apesar dos dois problemas terem seus estudos iniciados em paralelo, um esforço maior foi dado ao PMTF por conta dos resultados intermediários que foram sendo observados e que se mostraram mais promissores ou encorajadores para que novos desenvolvimentos e resultados fossem buscados. O esforço dedicado ao estudo do PCVMP foi, desta forma, menor. No entanto alguns bons resultados foram alcançados. Não foi possível dedicar mais tempo a desenvolvimentos adicionais ao PCVMP devido ao prazo planejado para o término da tese. Eventuais estudos que poderiam ser prosseguidos estão indicados na Seção 6.2.

## 6.1 Estudos futuros para o PMTF

Estudos futuros podem ser considerados no sentido de propor novas inequações válidas para os modelos de fluxo *multicommodity* visando eliminar simetrias. Estudar a possibilidade de um modelo do tipo *single commodity* ao invés do modelo *multicommodity*. Quanto ao VNDLB, com o intuito de acelerar a sua convergência, uma combinação com outros métodos de busca local seria interessante.

Modificações na ramificação do método exato proposto também é um possível caminho para estudo futuro. Por exemplo, ao invés de ramificar em relação as tarefas, poderia-se ramificar em relação às ferramentas, olhando para aquelas que primeiro deixariam a caixa de ferramentas da máquina, de acordo, por exemplo, com os graus dos nós do grafo MOSP.

Extensões também podem ser estudadas, considerando tamanhos não uniformes de ferramentas, máquinas paralelas, tempo de processamento, compartilhamento das ferramentas pelas máquinas, e tarefas que utilizam mais ferramentas que a máquina suporta. Outras possíveis variações também podem ser feitas considerando a vida útil das ferramentas e o não conhecimento prévio das tarefas a serem produzidas, ou seja, considerar uma versão estocástica do problema.

Métodos clássicos de resolução de problemas de fluxo em rede podem ser estudados, como geração de colunas e *branch-and-price*.

## 6.2 Estudos futuros para o PCVMP

No caso do PCVMP, estudos futuros podem ser feitos no sentido de desenvolver outros decodificadores nos BRKGAs propostos. Por exemplo, sequências parciais de clientes podem ser consideradas para criar todo o sequenciamento. Outros métodos de busca local também podem ser aplicados e métodos para melhorar os limites superiores do modelo podem ser desenvolvidos. Um processo de recodificação pode ser desenvolvido para retornar a solução obtida por busca local para a população atual.

Extensões do PCVMP podem ser estudadas, como por exemplo o problema de roteamento. Podem ser considerados janela de tempo, peso de carga, emissão de gás carbônico e determinação de rotas *online*. Outras metaheurísticas e métodos de busca local podem ser estudados, como o uso do VNDLB.

Métodos clássicos de resolução de problemas de fluxo em rede podem ser estudados para ambos problemas, como geração de colunas e *branch-and-price*.

# Referências

- ADJIASHVILI, D.; BOSIO, S.; ZEMMER, K. Minimizing the number of switch instances on a flexible machine in polynomial time. **Operations Research Letters**, v. 43, n. 3, p. 317–322, 2015.
- AGAPIOU, J. S. Sequence of operations optimization in single-stage multifunctional systems. **Journal of Manufacturing Systems**, v. 10, n. 3, p. 194–208, 1991.
- AHMADI, E.; GOLDENGORIN, B.; SÜER, G. A.; MOSADEGH, H. A hybrid method of 2-TSP and novel learning-based GA for job sequencing and tool switching problem. **Applied Soft Computing Journal**, v. 65, p. 214–229, 2018.
- AHUJA, R.; MAGNANTI, T.; ORLIN, J. **Network Flows: Theory, Algorithms and Applications**. [S.l.]: Prentice Hall, 1993.
- AKYUZ, M. H.; ONCAN, T.; Kuban Altinel, I. Beam search heuristics for the single and multi-commodity capacitated Multi-facility Weber Problems. **Computers and Operations Research**, v. 40, n. 12, p. 3056–3068, 2013.
- AL-FAWZAN, M. A.; AL-SULTAN, K. S. A tabu search based algorithm for minimizing the number of tool switches on a flexible machine. **Computers and Industrial Engineering**, v. 44, n. 1, p. 35–47, 2003.
- ALBAREDA-SAMBOLA, M.; ALONSO-AYUSO, A.; ESCUDERO, L. F.; FERNÁNDEZ, E.; PIZARRO, C. Fix-and-Relax-Coordination for a multi-period location-allocation problem under uncertainty. **Computers and Operations Research**, v. 40, n. 12, p. 2878–2892, 2013.
- AMAYA, J. E.; COTTA, C.; FERNÁNDEZ, A. J. A memetic algorithm for the tool switching problem. In: BLESÁ, M.; BLUM, C.; COTTA, C.; FERNÁNDEZ, A.; GALLARDO, J.; ROLI, A.; SAMPELS, M. (Ed.). **Hybrid Metaheuristics. HM 2008. Lecture Notes in Computer Science**. [S.l.]: Springer, Berlin, Heidelberg, 2008. v. 5296, p. 190–202.
- AMAYA, J. E.; COTTA, C.; FERNÁNDEZ-LEIVA, A. J. Memetic cooperative models for the tool switching problem. **Memetic Computing**, v. 3, n. 3, p. 199–216, 2011.
- AMAYA, J. E.; COTTA, C.; FERNÁNDEZ-LEIVA, A. J. Solving the tool switching problem with memetic algorithms. **Artificial Intelligence for Engineering Design, Analysis and Manufacturing**, v. 26, n. 2, p. 221–235, 2012.



- AMAYA, J. E.; COTTA, C.; FERNÁNDEZ-LEIVA, A. J. Cross entropy-based memetic algorithms: An application study over the tool switching problem. **International Journal of Computational Intelligence Systems**, v. 6, n. 3, p. 559–584, 2013.
- AMAYA, J. E.; COTTA, C.; FERNÁNDEZ-LEIVA, A. J.; GARCÍA-SÁNCHEZ, P. Deep memetic models for combinatorial optimization problems: application to the tool switching problem. **Memetic Computing**, p. 1–20, 2019.
- AMAYA, J. E.; COTTA, C.; LEIVA, A. J. F. Hybrid cooperation models for the tool switching problem. In: GONZÁLEZ, J.; PELTA, D.; CRUZ, C.; TERRAZAS, G.; KRASNOGOR, N. (Ed.). **Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)**. **Studies in Computational Intelligence**. [S.l.]: Springer, Berlin, Heidelberg, 2010. v. 284, p. 39–52.
- ANGEL, R. D.; CAUDLE, W. L.; NOONAN, R.; WHINSTON, A. Computer-Assisted School Bus Scheduling. **Management Science**, v. 18, n. 6, p. 279–288, 1972.
- APPLEGATE, D.; BIXBY, R.; CHVATAL, V.; COOK, W. **The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics)**. [S.l.]: Princeton University Press, 2007. ISBN 0691129932.
- ARAUJO, S. A. de; ARENALES, M. N.; CLARK, A. R. Lot sizing and furnace scheduling in small foundries. **Computers and Operations Research**, v. 35, n. 3, p. 916–932, 2008.
- ARCHETTI, C.; SPERANZA, M. G.; VIGO, D. Vehicle Routing Problems with Profits. In: TOTH, P.; VIGO, D. (Ed.). **Vehicle Routing: Problems, Methods, and Applications, Second Edition**. [S.l.]: Society for Industrial and Applied Mathematics, 2014. cap. 10, p. 273–297.
- AZEVEDO, T. d.; CARVALHO, M. Uma Avaliação Precisa da Modelagem do Problema de Minimização de Troca de Ferramentas como o Problema do Caixeiro Viajante. In: **XLIX Simpósio Brasileiro de Pesquisa Operacional**. Blumenau: [s.n.], 2017.
- BAENA, D.; CASTRO, J.; GONZÁLEZ, J. A. Fix-and-relax approaches for controlled tabular adjustment. **Computers and Operations Research**, v. 58, p. 41–52, 2015.
- BALAS, E. The prize collecting traveling salesman problem. **Networks**, v. 19, n. 6, p. 621–636, 1989.
- BARD, J. F. A heuristic for minimizing the number of tool switches on a flexible machine. **IIE Transactions (Institute of Industrial Engineers)**, v. 20, n. 4, p. 382–391, 1988.
- BAYKASOGLU, A.; OZSOYDAN, F. B. Minimizing tool switching and indexing times with tool duplications in automatic machines. **International Journal of Advanced Manufacturing Technology**, v. 89, n. 5-8, p. 1775–1789, 2017.
- BAZARAA, M.; JARVIS, J.; SHERALI, H. **Linear Programming and Network Flows**. [S.l.]: John Wiley & Sons, 2010.

- BECCENERI, J. C.; YANASSE, H. H.; SOMA, N. Y. A method for solving the minimization of the maximum number of open stacks problem within a cutting process. **Computers and Operations Research**, v. 31, n. 14, p. 2315–2332, 2004.
- BEEZÃO, A. **O problema de minimização de troca de ferramentas**. Tese (Doutorado) — Universidade de São Paulo, 2016.
- BEEZÃO, A. C.; CORDEAU, J.-F.; LAPORTE, G.; YANASSE, H. H. Scheduling identical parallel machines with tooling constraints. **European Journal of Operational Research**, v. 257, n. 3, p. 834–844, 2017.
- BEKTAS, T. Formulations and Benders decomposition algorithms for multidepot salesmen problems with load balancing. **European Journal of Operational Research**, v. 216, n. 1, p. 83–93, 2012.
- BENTLEY, J. J. Fast algorithms for geometric traveling salesman problems. **ORSA journal on computing**, v. 4, n. 4, p. 387–411, 1992.
- BLUM, C.; PUCHINGER, J.; RAIDL, G. R.; ROLI, A. Hybrid metaheuristics in combinatorial optimization: A survey. **Applied Soft Computing**, v. 11, n. 6, p. 4135–4151, 2011.
- BOCK, F. Mathematical programming solution of traveling salesman examples. In: GRAVES, R. L.; WOLFE, P. (Ed.). **Recent advances in mathematical programming**. [S.l.]: McGraw-Hill, New York, 1963.
- BOSCHETTI, M. A.; MANIEZZO, V.; ROFFILLI, M.; Bolufé Röhler, A. Matheuristics: Optimization, simulation and control. In: BLESA, M.; BLUM, C.; GASPERO, L. D.; ROLI, A.; SAMPELS, M.; SCHAERF, A. (Ed.). **Hybrid Metaheuristics. HM 2009. Lecture Notes in Computer Science**. [S.l.]: Springer, Berlin, Heidelberg, 2009. v. 5818, p. 171–177.
- BRAGIN, M. A.; LUH, P. B.; YAN, J. H.; YU, N.; STERN, G. A. Convergence of the Surrogate Lagrangian Relaxation Method. **Journal of Optimization Theory and Applications**, v. 164, n. 1, p. 173–201, 2014.
- BURCHETT, D. L.; RICHARD, J. P. Multi-commodity variable upper bound flow models. **Discrete Optimization**, v. 17, p. 89–122, 2015.
- BURGER, A. P.; JACOBS, C. G.; VUUREN, J. H. van; VISAGIE, S. E. Scheduling multi-colour print jobs with sequence-dependent setup times. **Journal of Scheduling**, v. 18, n. 2, p. 131–145, 2015.
- CALMELS, D. The job sequencing and tool switching problem: state-of-the-art literature review, classification, and trends. **International Journal of Production Research**, v. 57, n. 15-16, p. 5005–5025, 2019.
- CATANZARO, D.; GOUVEIA, L.; LABBÉ, M. Improved integer linear programming formulations for the job Sequencing and tool Switching Problem. **European Journal of Operational Research**, v. 244, n. 3, p. 766–777, 2015.

- CHAVES, A. A.; GONÇALVES, J. F.; LORENA, L. A. N. Adaptive biased random-key genetic algorithm with local search for the capacitated centered clustering problem. **Computers & Industrial Engineering**, v. 124, p. 331–346, 2018.
- CHAVES, A. A.; LORENA, L. A. N.; SENNE, E. L. F.; RESENDE, M. G. C. Hybrid method with CS and BRKGA applied to the minimization of tool switches problem. **Computers & Operations Research**, v. 67, p. 174–183, 2016.
- CHAVES, A. A.; SENNE, E. L. F.; YANASSE, H. H. Uma nova heurística para o problema de minimização de trocas de ferramentas. **Gestao e Producao**, v. 19, n. 1, p. 17–30, 2012.
- COOK, W. **Concorde TSP Solver**. 2016.  
<http://www.math.uwaterloo.ca/tsp/concorde.html>. Accessed 26 november 2019.
- CRAMA, Y.; KOLEN, A. W. J.; OERLEMANS, A. G.; SPIEKSMAN, F. C. R. Minimizing the number of tool switches on a flexible machine. **International Journal of Flexible Manufacturing Systems**, v. 6, n. 1, p. 33–54, 1994.
- CRAMA, Y.; MOONEN, L. S.; SPIEKSMAN, F. C. R.; TALLOEN, E. The tool switching problem revisited. **European Journal of Operational Research**, v. 182, n. 2, p. 952–957, 2007.
- CROWDER, H.; JOHNSON, E. L.; PADBERG, M. Solving Large-Scale Zero-One Linear Programming Problems. **Operations Research**, v. 31, n. 5, p. 803–834, 1983.
- CUNHA, J. O.; KRAMER, H. H.; MELO, R. A. Effective matheuristics for the multi-item capacitated lot-sizing problem with remanufacturing. **Computers and Operations Research**, v. 104, p. 149–158, 2019.
- DADASHI, H.; MOSLEMI, S.; MIRZAZADEH, A. Optimization of a New Tool Switching Problem in Flexible Manufacturing Systems with a Tool Life by a Genetic Algorithm. **International Journal of Industrial and Manufacturing Systems Engineering**, v. 1, n. 3, p. 52–58, 2016.
- DANTZIG, G.; FULKERSON, R.; JOHNSON, S. Solution of a Large-Scale Traveling-Salesman Problem. **Journal of the Operations Research Society of America**, v. 2, n. 4, p. 393–410, 1954.
- DANTZIG, G. B.; FULKERSON, D. R. **12. On the Max-Flow Min-Cut Theorem of Networks**. [S.l.]: Princeton University Press, 1957. 215–222 p.
- DEHAYBE, H. **An Ant Colony System for solving the Job Sequencing and Tool Switching Problem**. 2018.
- DERELI, T.; FILIZ, I. H. Allocating optimal index positions on tool magazines using genetic algorithms. **Robotics and Autonomous Systems**, v. 33, n. 2, p. 155–167, 2000.
- DESROCHERS, M.; LAPORTE, G. Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. **Operations Research Letters**, v. 10, n. 1, p. 27–36, 1991.

- DIESSEN, T. van. **Modelling and optimization of tool replacements on parallel machines**. 2019.
- DJELLAB, H.; DJELLAB, K.; GOURGAND, M. A new heuristic based on a hypergraph representation for the tool switching problem. **International Journal of Production Economics**, v. 64, n. 1, p. 165–176, 2000.
- DOLAN, E.; MOREÉ, J. Benchmarking optimization software with performance profiles. **Mathematical Programming**, v. 91, p. 201–213, 2002.
- DORIGO, M. **Optimization, learning and natural algorithms**. Tese (Doutorado) — Dipartimento di Elettronica, Politecnico di Milano, Itália, 1992.
- DORNELES, Á. P.; ARAÚJO, O. C. de; BURIOL, L. S. A column generation approach to high school timetabling modeled as a multicommodity flow problem. **European Journal of Operational Research**, v. 256, n. 3, p. 685–695, 2017.
- FARUGHI, H.; DOLATABADIAA, M.; MORADI, V.; KARBASI, V.; MOSTAFAYI, S. Minimizing the number of tool switches in flexible manufacturing cells subject to tools reliability using genetic algorithm. **Journal of industrial and systems engineering**, v. 10, n. 1, p. 1–18, 2017.
- FATHI, Y.; BARNETTE, K. W. Heuristic procedures for the parallel machine problem with tool switches. **International Journal of Production Research**, v. 40, n. 1, p. 151–164, 2002.
- FERREIRA, D.; MORABITO, R.; RANGEL, S. Relax and fix heuristics to solve one-stage one-machine lot-scheduling models for small-scale soft drink plants. **Computers and Operations Research**, v. 37, n. 4, p. 684–691, 2010.
- FISCHETTI, M.; LODI, A. Local branching. **Mathematical Programming Series B**, v. 98, n. 1-3, p. 23–47, 2003.
- FISCHETTI, M.; MONACI, M. Proximity search for 0-1 mixed-integer convex programming. **Journal of Heuristics**, v. 20, n. 6, p. 709–731, 2014.
- FLOOD, M. M. The Traveling-Salesman Problem. **Operations Research**, v. 4, n. 1, p. 61–75, 1956.
- FORD, L. R.; FULKERSON. A suggested computation for maximal multi-commodity network flows. **Management Science**, v. 5, n. 1, p. 97–101, 1957.
- FORD, L. R.; FULKERSON, D. R. Notes on Linear Programming: Part XX Maximal Flow Through a Network. RAND Corporation, 1954.
- FORD, L. R.; FULKERSON, D. R. A simple algorithm for finding maximal network flows and an application to the Hitchcock problem. RAND Corporation, 1955.
- FORD, L. R.; FULKERSON, D. R. **Flows in networks**. [S.l.]: The RAND Corporation, 1962. 332 p.
- FOURER, R.; GAY, D. M.; KERNIGHAN, B. W. **AMPL: A Modeling Language for Mathematical Programming**. [S.l.]: Cengage Learning, 2002. 540 p. ISBN 0-534-38809-4.

- FRIEDMAN, M. A Comparison of Alternative Tests of Significance for the Problem of m Rankings. **The Annals of Mathematical Statistics**, v. 11, n. 1, p. 86–92, 1940.
- FUKUDA, E. H. **Algoritmo do volume e otimização não diferenciável**. Tese (Doutorado) — Instituto de Matemática e Estatística – Universidade de São Paulo, 2007.
- FULKERSON, D. **Journal of the Society for Industrial and Applied Mathematics**, v. 9, n. 1, p. 18–27, 1961.
- FULKERSON, D.; GROSS, D. Incidence Matrices and Interval Graphs. **Pacific Journal of Mathematics**, v. 15, n. 3, p. 835–855, 1965.
- FURRER, M.; MÜTZE, T. An algorithmic framework for tool switching problems with multiple objectives. **European Journal of Operational Research**, v. 259, n. 3, p. 1003–1016, 2017.
- GAMILA, M. A.; MOTAVALLI, S. A modeling technique for loading and scheduling problems in FMS. **Robotics and Computer-Integrated Manufacturing**, v. 19, p. 45–54, 2003.
- GAO, C.; WEISE, T.; LI, J. In: **Advances in Swarm Intelligence. ICSI 2014. Lecture Notes in Computer Science**, pages = 27–35, title = **Improve the 3-flip neighborhood local search by random flat move for the set covering problem**, editor = Tan, Y. and Shi, Y. and Coello, C.A.C., publisher = Springer, Cham, volume = 8794, year = 2014. [S.l.: s.n.].
- GENDREAU, M.; HERTZ, A.; LAPORTE, G. New Insertion and Postoptimization Procedures for the Traveling Salesman Problem. **Operations Research**, v. 40, n. 6, p. 1086–1094, 1992.
- GEOFFRION, A. M. Lagrangean relaxation for integer programming. In: . [S.l.: s.n.], 1974. v. 2, p. 82–114.
- GHIANI, G.; GRIECO, A.; GUERRIERO, E. An exact solution to the TLP problem in an NC machine. **Robotics and Computer-Integrated Manufacturing**, v. 23, n. 6, p. 645–649, 2007.
- GHIANI, G.; GRIECO, A.; GUERRIERO, E. Solving the Job Sequencing and Tool Switching Problem as a nonlinear least cost Hamiltonian cycle problem. **Networks**, v. 55, n. 4, p. 379–385, 2010.
- GHRAYEB, O. A.; PHOJANAMONGKOLKIJ, N.; FINCH, P. R. A mathematical model and heuristic procedure to schedule printed circuit packs on sequencers. **International Journal of Production Research**, v. 41, n. 16, p. 3849–3860, 2003.
- GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computers and Operations Research**, v. 13, n. 5, p. 533–549, 1986.
- GÖKGÜR, B.; HNICH, B.; ÖZPEYNIRCI, S. Parallel machine scheduling with tool loading: a constraint programming approach. **International Journal of Production Research**, v. 56, n. 16, p. 5541–5557, 2018.

- GOMEZ, A. **Modelo para sequenciamento de partes e ferramentas em um sistema de manufatura flexível com restrições as datas de vencimento e a capacidade do magazine**. Tese (Doutorado) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 1996.
- GÓMEZ, A. T.; LORENA, L. A. N. Modelagem de sistemas de manufatura flexíveis considerando restrições temporais e a capacidade do magazine. **Gestão & Produção**, v. 5, n. 1, p. 69–80, 1998.
- GONÇALVES, J. F.; RESENDE, M. G. Biased random-key genetic algorithms for combinatorial optimization. **Journal of Heuristics**, v. 17, n. 5, p. 487–525, 2011.
- GONÇALVES, J. F.; RESENDE, M. G. A biased random-key genetic algorithm for the unequal area facility layout problem. **European Journal of Operational Research**, v. 246, n. 1, p. 86–107, 2015.
- GONÇALVES, J. F.; RESENDE, M. G. Random-key genetic algorithms. In: MARTÍ, R.; PARDALOS, P.; RESENDE, M. (Ed.). **Handbook of Heuristics**. [S.l.]: Springer, Cham, 2018. p. 703–715.
- GONDZIO, J.; GONZÁLEZ-BREVIS, P.; MUNARI, P. Large-scale optimization with the primal-dual column generation method. **Mathematical Programming Computation**, v. 8, n. 1, p. 47–82, 2016.
- GRÖTSCHEL, M.; JÜNGER, M.; REINELT, G. Optimal control of plotting and drilling machines: A case study. **ZOR Zeitschrift für Operations Research Methods and Models of Operations Research**, v. 35, n. 1, p. 61–84, 1991.
- GUIGNARD, M. Lagrangean relaxation. **Top**, v. 11, n. 2, p. 151–200, 2003.
- HARRIS, T.; ROSS, F. Fundamentals of a Method for Evaluating Rail Net Capacities. Research Memorandum RM-1573, The RAND Corporation Santa Monica CA., 1955.
- HELD, M.; KARP, R. M. The Traveling-Salesman Problem and Minimum Spanning Trees. **Operations Research**, v. 18, n. 6, p. 1138–1162, 1970.
- HELD, M.; KARP, R. M. The traveling-salesman problem and minimum spanning trees: Part II. **Mathematical Programming**, v. 1, n. 1, p. 6–25, 1971.
- HERTZ, A.; LAPORTE, G.; MITTAZ, M.; STECKE, K. E. Heuristics for minimizing tool switches when scheduling part types on a flexible machine. **IIE Transactions (Institute of Industrial Engineers)**, v. 30, n. 8, p. 689–694, 1998.
- HIRVIKORPI, M.; SALONEN, K.; KNUUTILA, T.; NEVALAINEN, O. S. The general two-level storage management problem: A reconsideration of the KTNS-rule. **European Journal of Operational Research**, v. 171, n. 1, p. 189–207, 2006.
- HITCHCOCK, F. L. The Distribution of a Product from Several Sources to Numerous Localities. **Journal of Mathematics and Physics**, v. 20, n. 1–4, p. 224–230, 1941.
- HOFFMAN, A. J.; KOLEN, A. W. J.; SAKAROVITCH, M. Totally-Balanced and Greedy Matrices. **SIAM Journal on Algebraic Discrete Methods**, v. 6, n. 4, p. 721–730, 1985.

- HOP, N. V.; NAGARUR, N. The scheduling problem of pcbs for multiple non-identical parallel machines. **European Journal of Operational Research**, v. 158, p. 577–594, 2004.
- ILOG Using the CPLEX Callable Library. [S.l.], 2006.
- INGBER, L. Simulated annealing: Practice versus theory. **Mathematical and Computer Modelling**, v. 18, n. 11, p. 29–57, 1993.
- JÄGER, G.; MOLITOR, P. Algorithms and experimental study for the traveling salesman problem of second order. In: YANG, B.; DU, D.; WANG, C. (Ed.). **Combinatorial Optimization and Applications. COCOA 2008. Lecture Notes in Computer Science**. [S.l.: s.n.], 2008. v. 5165, p. 211–224.
- JANG, S. Y.; PARKT, J.; PARK, N. An integrated decision support system for FMS production planning and scheduling problems. **International Journal of Advanced Manufacturing Technology**, v. 11, n. 2, p. 101–110, 1996.
- KARAKAYALI, I.; AZIZOGLU, M. Minimizing total flow time on a single flexible machine. **International Journal of Flexible Manufacturing Systems**, v. 18, n. 1, p. 55–73, 2006.
- KARZAN, F. K.; AZIZOGLU, M. The tool transporter movements problem in flexible manufacturing systems. **International Journal of Production Research**, v. 46, n. 11, p. 3059–3084, 2008.
- KELLY, J. D.; MANN, J. L. Flowsheet decomposition heuristic for scheduling: A relax-and-fix method. **Computers and Chemical Engineering**, v. 28, n. 11, p. 2193–2200, 2004.
- KHAN, B. K.; GUPTA, B. D.; Sen Gupta, D. K.; KUMAR, K. D. A generalized procedure for minimizing tool changeovers of two parallel and identical CNC machining centres. **Production Planning and Control**, v. 11, n. 1, p. 62–72, 2000.
- KIM, Y. D. Heuristics for flowshop scheduling problems minimizing mean tardiness. **Journal of the Operational Research Society**, v. 44, n. 1, p. 19–28, 1993.
- KONAK, A.; KULTUREL-KONAK, S. An ant colony optimization approach to the minimum tool switching instant problem in flexible manufacturing system. In: **Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Scheduling, CI-Sched 2007**. Honolulu, HI, USA: [s.n.], 2007. p. 43–48.
- KONAK, A.; KULTUREL-KONAK, S.; AZIZOGLU, M. Minimizing the number of tool switching instants in Flexible Manufacturing Systems. **International Journal of Production Economics**, v. 116, n. 2, p. 298–307, 2008.
- KOOPMANS, T. C.; BECKMANN, M. Assignment Problems and the Location of Economic Activities. **Econometrica**, v. 25, n. 1, p. 53, 1957.
- KOU, L. T. Polynomial Complete Consecutive Information Retrieval Problems. **SIAM Journal on Computing**, v. 6, n. 1, p. 67–75, 1977.

- LAPORTE, G.; SALAZAR-GONZÁLEZ, J. J.; SEMET, F. Exact algorithms for the job sequencing and tool switching problem. **IIE Transactions (Institute of Industrial Engineers)**, v. 36, n. 1, p. 37–45, 2004.
- LAWLER, E. L.; LENSTRA, J. K.; Rinnooy Kan, A. H. G.; SHMOYS, D. B. **The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization**. [S.l.]: Wiley, 1991. 476 p. ISBN 978-0-471-90413-7.
- LETCHFORD, A. N.; SALAZAR-GONZÁLEZ, J. J. Stronger multi-commodity flow formulations of the Capacitated Vehicle Routing Problem. **European Journal of Operational Research**, v. 244, n. 3, p. 730–738, 2015.
- LINHARES, A.; YANASSE, H. H. Connections between cutting-pattern sequencing, VLSI design, and flexible machines. **Computers and Operations Research**, v. 29, n. 12, p. 1759–1772, 2002.
- LOZANO, L.; SMITH, J. C.; KURZ, M. E. Solving the traveling salesman problem with interdiction and fortification. **Operations Research Letters**, v. 45, n. 3, p. 210–216, 2017.
- MAHARJAN, B.; MATIS, T. I. Multi-commodity flow network model of the flight gate assignment problem. **Computers and Industrial Engineering**, v. 63, n. 4, p. 1135–1144, 2012.
- MALAGUTI, E.; MARTELLO, S.; SANTINI, A. The traveling salesman problem with pickups, deliveries, and draft limits. **Omega**, v. 74, p. 50–58, 2018.
- MALDONADO, M.; RANGEL, S.; FERREIRA, D. A study of different subsequence elimination strategies for the soft drink production planning. **Journal of Applied Research and Technology**, v. 12, n. 4, p. 631–641, 2014.
- MATZLIACH, B. The online tool switching problem with non-uniform tool size. **International Journal of Production Research**, v. 36, n. 12, p. 3407–3420, 1998.
- MATZLIACH, B.; TZUR, M. Storage management of items in two levels of availability. **European Journal of Operational Research**, v. 121, n. 2, p. 363–379, 2000.
- MCGEOCH, L. A.; SLEATOR, D. D. A strongly competitive randomized paging algorithm. **Algorithmica**, v. 6, n. 1-6, p. 816–825, 1991.
- MECLER, J.; SUBRAMANIAN, A.; VIDAL, T. A Simple and Effective Hybrid Genetic Search for the Job Sequencing and Tool Switching Problem. **arxiv.org/pdf/1910.10021**, 2019.
- MELNYK, S. A.; GHOSH, S.; RAGATZ, G. L. Tooling constraints and shop floor scheduling: A simulation study. **Journal of Operations Management**, v. 8, n. 2, p. 69–89, 1989.
- MENGER, K. Das botenproblem. **Ergebnisse eines Mathematischen Kolloquiums** 2, p. 11–12, 1932.
- MESQUITA, M.; MOZ, M.; PAIAS, A.; PATO, M. A decompose-and-fix heuristic based on multi-commodity flow models for driver rostering with days-off pattern. **European Journal of Operational Research**, v. 245, n. 2, p. 423–437, 2015.



- MILLER, C. E.; ZEMLIN, R. A.; TUCKER, A. W. Integer Programming Formulation of Traveling Salesman Problems. **Journal of the ACM (JACM)**, v. 7, n. 4, p. 326–329, 1960.
- MLADENVIĆ, N.; HANSEN, P. Variable neighborhood search. **Computers & Operations Research**, v. 24, n. 11, p. 1097–1100, 1997.
- MLADENVIĆ, N.; HANSEN, P. Variable neighborhood search: Methods and recent applications. In: **Third Metaheuristics International Conference**. Angra dos Reis, Brasil: [s.n.], 1999. p. 275–280.
- MÖHRING, R. H. Graph Problems Related to Gate Matrix Layout and PLA Folding. In: **Computational Graph Theory**. [S.l.]: Springer, Vienna, 1990. p. 17–51.
- MUNGUÍA, L. M.; AHMED, S.; BADER, D. A.; NEMHAUSER, G. L.; GOEL, V.; SHAO, Y. A parallel local search framework for the Fixed-Charge Multicommodity Network Flow problem. **Computers and Operations Research**, v. 77, p. 44–57, 2017.
- OERLEMANS, A. **Production planning for flexible manufacturing systems**. Tese (Doutorado) — University of Limburg, Holanda, 1992.
- ÖNCAN, T.; ALTINEL, I. K.; LAPORTE, G. A comparative analysis of several asymmetric traveling salesman problem formulations. **Computers and Operations Research**, v. 36, n. 3, p. 637–654, 2009.
- ÖZPEYNIRCI, S.; GÖKGÜR, B.; HNIC, B. Parallel machine scheduling with tool loading. **Applied Mathematical Modelling**, v. 40, n. 9-10, p. 5660–5671, 2016.
- PAIVA, G. S.; CARVALHO, M. A. M. Improved heuristic algorithms for the Job Sequencing and Tool Switching Problem. **Computers and Operations Research**, v. 88, p. 208–219, 2017.
- PAPADIMITRIOU, C. H. The euclidean traveling salesman problem is *np*-complete. **Theoretical Computer Science**, v. 4, n. 3, p. 237–244, 1977.
- PISINGER, D.; ROPKE, S. Large Neighborhood Search. In: GENDREAU, M.; POTVIN, J.-Y. (Ed.). **Handbook of Metaheuristics. International Series in Operations Research & Management Science**. [S.l.]: Springer, Boston, MA, 2010. v. 146, p. 399–419.
- PRASETYO, H.; FAUZA, G.; AMER, Y.; LEE, S. H. Survey on applications of biased-random key genetic algorithms for solving optimization problems. In: **2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)**. [S.l.]: IEEE, 2015. p. 863–870.
- PRIVAULT, C.; FINKE, G. Modelling a tool switching problem on a single NC-machine. **Journal of Intelligent Manufacturing**, v. 6, n. 2, p. 87–94, 1995.
- PRIVAULT, C.; FINKE, G. K-server problems with bulk requests: An application to tool switching in manufacturing. **Annals of Operations Research**, v. 96, n. 1-4, p. 255–269, 2000.

- PUCHINGER, J.; RAIDL, G. R. Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In: MIRA, J.; ÁLVAREZ, J. (Ed.). **Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach. IWINAC 2005. Lecture Notes in Computer Science**. [S.l.]: Springer, Berlin, Heidelberg, 2005. v. 3562, p. 41–53.
- PUREZA, V.; MORABITO, R.; LUNA, H. P. Modeling and solving the traveling salesman problem with priority prizes. **Pesquisa Operacional**, v. 38, n. 3, p. 499–522, 2018.
- RAACK, C. **Capacitated Network Design – Multi-Commodity Flow Formulations, Cutting Planes, and Demand Uncertainty**. Tese (Doutorado) — Technische Universität Berlin, Alemanha, 2012.
- RADULY-BAKA, C.; KNUUTILA, T.; NEVALAINEN, O. **Minimising the Number of Tool Switches with Tools of Different Sizes**. [S.l.], 2005.
- RADULY-BAKA, C.; NEVALAINEN, O. S. The modular tool switching problem. **European Journal of Operational Research**, v. 242, n. 1, p. 100–106, 2015.
- RAGHAVAN, U. N.; ALBERT, R.; KUMARA, S. Near linear time algorithm to detect community structures in large-scale networks. **Physical Review E - Statistical, Nonlinear, and Soft Matter Physics**, v. 76, n. 3, 2007.
- RATLIFF, H. D.; ROSENTHAL, A. S. Order-Picking in a Rectangular Warehouse: A Solvable Case of the Traveling Salesman Problem. **Operations Research**, v. 31, n. 3, p. 507–521, 1983.
- REINELT, G. **TSP Test Data Homepage**. 2009.  
<http://www.math.uwaterloo.ca/tsp/data/index.html>. Accessed 26 november 2019.
- REINELT, G. **TSPLib Homepage**. 2013.  
<http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/index.html>. Accessed 26 november 2019.
- RODGERS, J. L.; NICEWANDER, W. Thirteen ways to look at the correlation coefficient. **American Statistician**, v. 42, n. 1, p. 59–66, 1988.
- ROTHENBÄCHER, A. K.; DREXL, M.; IRNICH, S. Branch-and-price-and-cut for a service network design and hub location problem. **European Journal of Operational Research**, v. 255, n. 3, p. 935–947, 2016.
- RUBINSTEIN, R. The Cross-Entropy Method for Combinatorial and Continuous Optimization. **Methodology And Computing In Applied Probability**, v. 1, n. 2, p. 127–190, 1999.
- RUPE, J.; KUO, W. Solutions to a modified tool loading problem for a single FMM. **International Journal of Production Research**, v. 35, n. 8, p. 2253–2268, 1997.
- SALONEN, K.; RADULY-BAKA, C.; NEVALAINEN, O. S. A note on the tool switching problem of a flexible machine. **Computers and Industrial Engineering**, v. 50, n. 4, p. 458–465, 2006.

- SALONEN, K.; SMED, J.; JOHNSON, M.; NEVALAINEN, O. Grouping and sequencing PCB assembly jobs with minimum feeder setups. **Robotics and Computer-Integrated Manufacturing**, v. 22, n. 4, p. 297–305, 2006.
- SARMADI, H.; GHOLAMI, S. Modeling of Tool Switching Problem in a Flexible Manufacturing Cell: with two or More Machines. In: **International Conference on Mechanical and Electrical Technology, 3rd, (ICMET-London 2011)**. [S.l.]: ASME Press, 2011. v. 1-3, p. 2345–2349.
- SARUBBI, J. **Um modelo linear para o problema do caixeiro viajante com demandas heterogêneas**. 2003.
- SARUBBI, J. **Problemas de Roteamento com Custos de Carga**. Tese (Doutorado) — Universidade Federal de Minas Gerais, 2008.
- SARUBBI, J.; LUNA, H. The multicommodity traveling salesman problem. In: **INOC - International Network Optimization Conference**. Bélgica: [s.n.], 2007.
- SARUBBI, J.; MATEUS, G. R.; LUNA, H. P.; MIRANDA, G. J. de. Model and algorithms for the Multicommodity Traveling Salesman Problem. In: **7th International Conference on Hybrid Intelligent Systems (HIS 2007)**. Kaiserslautern, Germany: IEEE, 2007. p. 113–119.
- SARUBBI, J.; MIRANDA, G.; LUNA, H. P.; MATEUS, G. A cut-and-branch algorithm for the multicommodity traveling salesman problem. In: **2008 IEEE International Conference on Service Operations and Logistics, and Informatics**. Beijing, China: [s.n.], 2008. v. 2, p. 1806–1811.
- SCHRIJVER, A. On the history of the transportation and maximum flow problems. **Mathematical Programming**, v. 91, n. 3, p. 437–445, 2002.
- SCHWERDFEGER, S.; BOYSEN, N. Order picking along a crane-supplied pick face: The SKU switching problem. **European Journal of Operational Research**, v. 260, n. 2, p. 534–545, 2017.
- SENNE, E.; YANASSE, H. Beam search algorithms for minimizing tool switches on a flexible manufacturing system. In: **XI WSEAS International Conference on Mathematical and Computational Methods in Science and Engineering**. Baltimore, USA: [s.n.], 2009. v. 1, p. 68–72.
- SHERALI, H. D.; DRISCOLL, P. J. On tightening the relaxations of miller-tucker-zemlin formulations for asymmetric traveling salesman problems. **Operations Research**, v. 50, n. 4, p. 656–669, 2002.
- SHIRAZI, R.; FRIZELLE, G. D. Minimizing the number of tool switches on a flexible machine: An empirical study. **International Journal of Production Research**, v. 39, n. 15, p. 3547–3560, 2001.
- SILVA, A. **Abordagens de Otimização para apoiar a Elaboração e Análise de Roteiros Turísticos**. Tese (Doutorado) — Universidade Federal de São Carlos, 2017.

- SILVA, A. A. da; MORABITO, R.; PUREZA, V. Optimization approaches to support the planning and analysis of travel itineraries. **Expert Systems with Applications**, v. 112, p. 321–330, 2018.
- SILVA, T. T.; CHAVES, A. A.; YANASSE, H. H. A new multicommodity flow model for the Job Sequencing and Tool Switching Problem. **International Journal of Production Research**. In press.
- SILVA, T. T.; RANGEL, S. Limites duais para o problema integrado de dimensionamento e sequenciamento de lotes modelado com restrições multicommodity. In: **Congresso Nacional de Matemática Aplicada e Computacional**. [S.l.: s.n.], 2016.
- SMED, J.; JOHNSON, M.; PURANEN, M.; LEIPÄLÄ, T.; NEVALAINEN, O. Job grouping in surface mounted component printing. **Robotics and Computer-Integrated Manufacturing**, v. 15, n. 1, p. 39–49, 1999.
- SONG, C.-Y.; HWANG, H. Optimal tooling policy for a tool switching problem of a flexible machine with automatic tool transporter. **International Journal of Production Research**, v. 40, n. 4, p. 873–883, 2002.
- SPEARS, V. M.; SPEARS, V. M.; De Jong, K. A. On the virtues of parameterized uniform crossover. In: **In Proceedings of the Fourth International Conference on Genetic Algorithms**. [S.l.: s.n.], 1991. p. 230–236.
- STECKE, K. E. Formulation and Solution of Nonlinear Integer Production Planning Problems for Flexible Manufacturing Systems. **Management Science**, v. 3, n. 29, p. 273–288, 1983.
- STÜTZLE, T. **Iterated local search for the quadratic assignment problem**. Tese (Doutorado) — TU Darmstadt, 1999.
- TADAYON, B.; SMITH, J. C. Algorithms for an Integer Multicommodity Network Flow Problem with Node Reliability Considerations. **Journal of Optimization Theory and Applications**, v. 161, n. 2, p. 506–532, 2014.
- TANG, C. S.; DENARDO, E. V. Models Arising from a Flexible Manufacturing Machine, Part I: Minimization of the Number of Tool Switches. **Operations Research**, v. 36, n. 5, p. 767–777, 1988.
- TANG, C. S.; DENARDO, E. V. Models arising from a flexible manufacturing machine, part II: minimization of the number of switching instants. **Operations Research**, v. 36, n. 5, p. 778–784, 1988.
- TOSO, E. A.; MORABITO, R.; CLARK, A. R. Lot sizing and sequencing optimisation at an animal-feed plant. **Computers and Industrial Engineering**, v. 57, n. 3, p. 813–821, 2009.
- TZUR, M.; ALTMAN, A. Minimization of tool switches for a flexible manufacturing machine with slot assignment of different tool sizes. **IIE Transactions (Institute of Industrial Engineers)**, v. 36, n. 2, p. 95–110, 2004.

- UMETANI, S.; ARAKAWA, M.; YAGIURA, M. Relaxation heuristics for the set multicover problem with generalized upper bound constraints. **Computers and Operations Research**, v. 93, p. 90–100, 2018.
- Van Hop, N. The tool-switching problem with magazine capacity and tool size constraints. **IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans.**, v. 35, n. 5, p. 617–628, 2005.
- WIDMER, M. Job Shop Scheduling with Tooling Constraints: A Tabu Search Approach. **The Journal of the Operational Research Society**, v. 42, n. 1, p. 75, 1991.
- WILCOXON, F. Individual Comparisons by Ranking Methods. **Biometrics Bulletin**, v. 1, n. 6, p. 80, 1945.
- WOLSEY, L. A. **Integer Programming: Volume 52 of Wiley Series in Discrete Mathematics and Optimization**. [S.l.: s.n.], 1998. 288 p.
- WONG, R. Integer programming formulations of the traveling salesman problem. In: **IEEE international conference of circuits and computers**. [S.l.: s.n.], 1980. p. 149–52.
- YAGIURA, M.; IBARAKI, T. Efficient 2 and 3-flip neighborhood search algorithms for the MAX SAT: Experimental evaluation. **Journal of Heuristics**, v. 7, n. 5, p. 423–442, 2001.
- YANASSE, H.; LAMOSA, M. An application of the generalized travelling salesman problem: the minimization of tool switches problem. In: **International Annual Scientific Conference of the German Operations Research Society**. [S.l.: s.n.], 2005. p. 90.
- YANASSE, H.; LAMOSA, M. On solving the minimization of tool switches problem using graphs. In: **XII ICIEOM**. Fortaleza, CE, Brasil: [s.n.], 2006.
- YANASSE, H. H. On a pattern sequencing problem to minimize the maximum number of open stacks. **European Journal of Operational Research**, v. 100, n. 3, p. 454–463, 1997.
- YANASSE, H. H. A transformation for solving a pattern sequencing problem in the wood cut industry. **Pesquisa Operacional**, v. 17, n. 1, p. 57–70, 1997.
- YANASSE, H. H. Um novo limitante inferior para o problema de minimização de trocas de ferramentas. In: **Anais do XLI Simpósio Brasileiro de Pesquisa Operacional**. Porto Seguro, Brasil: [s.n.], 2009. p. 2841–2848.
- YANASSE, H. H. Sobre o problema de minimização de troca de ferramentas: um caso especial. In: **Anais do XLIII Simpósio Brasileiro de Pesquisa Operacional**. Ubatuba, Brasil: [s.n.], 2011. p. 2117–2124.
- YANASSE, H. H. Um algoritmo polinomial para o caso de árvores do problema de minimização de trocas de ferramentas. In: **Anais do XLIV Simpósio Brasileiro de Pesquisa Operacional**. Rio de Janeiro, Brasil: [s.n.], 2012. p. 3385–3391.

YANASSE, H. H.; De Cassia Meneses Rodrigues, R.; SENNE, E. L. F. Um algoritmo enumerativo baseado em ordenamento parcial para resolução do problema de minimização de trocas de ferramentas. **Gestao e Producao**, v. 16, n. 3, p. 370–381, 2009.

YANASSE, H. H.; PINTO, M. J. The minimization of tool switches problem as a network flow problem with side constraints. In: **XXXIV Simpósio Brasileiro de Pesquisa Operacional**. Rio de Janeiro, Brasil: [s.n.], 2002.

ZEBALLOS, L. J. A constraint programming approach to tool allocation and production scheduling in flexible manufacturing systems. **Robotics and Computer-Integrated Manufacturing**, v. 26, n. 6, p. 725–743, 2010.

ZHANG, J. H.; HINDUJA, S. Determination of the Optimum Tool Set for a Given Batch of Turned Components. **CIRP Annals - Manufacturing Technology**, v. 44, n. 1, p. 445–450, 1995.

ZHOU, B. H.; XI, L. F.; CAO, Y. S. A beam-search-based algorithm for the tool switching problem on a flexible machine. **International Journal of Advanced Manufacturing Technology**, v. 25, n. 9-10, p. 876–882, 2005.

# Anexo A - Resultados do PMTF

Resultados obtidos pelo CPLEX nos modelos MF-PMTF-1, MF-PMTF-1A, MF-PMTF-1B, MF-PMTF-1C e MF-PMTF-1D, respectivamente, considerando as instâncias fornecidas por Catanzaro *et al.* (2015).

TABELA A.1 – Resultados dos modelos de fluxo para as instâncias introduzidas por Catanzaro *et al.* (2015) - Grupo A.

	MF-PMTF-1		MF-PMTF-1A		MF-PMTF-1B		MF-PMTF-1C		MF-PMTF-1D	
	S	T	S	T	S	T	S	T	S	T
A1-1	10.00	6.17	10.00	3.08	10.00	4.31	10.00	8.17	10.00	7.00
A1-2	7.00	6.02	7.00	0.52	7.00	0.52	7.00	0.47	7.00	0.28
A1-3	8.00	2.39	8.00	0.97	8.00	2.66	8.00	1.86	8.00	2.03
A1-4	9.00	5.30	9.00	1.92	9.00	3.52	9.00	7.41	9.00	5.52
A1-5	9.00	5.27	9.00	1.70	9.00	5.39	9.00	3.17	9.00	5.81
A1-6	10.00	7.86	10.00	4.27	10.00	5.70	10.00	7.42	10.00	7.20
A1-7	8.00	5.64	8.00	0.72	8.00	0.63	8.00	1.84	8.00	1.52
A1-8	7.00	8.19	7.00	0.30	7.00	0.47	7.00	0.49	7.00	0.31
A1-9	9.00	2.47	9.00	1.56	9.00	2.17	9.00	1.92	9.00	1.75
A1-10	8.00	4.39	8.00	1.11	8.00	1.05	8.00	1.66	8.00	1.91
Média	8.50	5.37	8.50	1.61	8.50	2.64	8.50	3.44	8.50	3.33
A2-1	6.00	4.09	6.00	0.48	6.00	0.63	6.00	0.74	6.00	0.38
A2-2	5.00	0.22	5.00	0.31	5.00	0.20	5.00	0.09	5.00	0.22
A2-3	5.00	1.27	5.00	0.08	5.00	0.23	5.00	0.11	5.00	0.11
A2-4	6.00	5.86	6.00	0.78	6.00	1.34	6.00	0.72	6.00	2.06
A2-5	6.00	4.53	6.00	0.95	6.00	0.88	6.00	0.73	6.00	1.61
A2-6	7.00	20.56	7.00	2.86	7.00	7.53	7.00	6.56	7.00	7.77
A2-7	6.00	9.36	6.00	0.39	6.00	0.42	6.00	0.66	6.00	0.41
A2-8	5.00	0.39	5.00	0.16	5.00	0.14	5.00	0.13	5.00	0.20
A2-9	6.00	6.64	6.00	0.33	6.00	0.45	6.00	0.66	6.00	0.42
A2-10	6.00	5.89	6.00	0.42	6.00	0.66	6.00	0.74	6.00	0.63
Média	5.80	5.88	5.80	0.68	5.80	1.25	5.80	1.11	5.80	1.38
A3-1	4.00	0.30	4.00	0.19	4.00	0.13	4.00	0.13	4.00	0.17
A3-2	4.00	0.30	4.00	0.08	4.00	0.09	4.00	0.08	4.00	0.05
A3-3	4.00	0.11	4.00	0.13	4.00	0.08	4.00	0.08	4.00	0.05
A3-4	4.00	0.20	4.00	0.11	4.00	0.36	4.00	0.30	4.00	0.22
A3-5	4.00	0.47	4.00	0.22	4.00	0.34	4.00	0.16	4.00	0.22
A3-6	5.00	23.31	5.00	0.66	5.00	0.61	5.00	1.22	5.00	0.72
A3-7	4.00	0.19	4.00	0.20	4.00	0.25	4.00	0.09	4.00	0.08
A3-8	4.00	0.13	4.00	0.22	4.00	0.14	4.00	0.09	4.00	0.09
A3-9	4.00	0.11	4.00	0.20	4.00	0.16	4.00	0.08	4.00	0.13
A3-10	4.00	0.53	4.00	0.25	4.00	0.30	4.00	0.27	4.00	0.14
Média	4.10	2.56	4.10	0.23	4.10	0.25	4.10	0.25	4.10	0.19
A4-1	3.00	0.11	3.00	0.09	3.00	0.08	3.00	0.06	3.00	0.06
A4-2	3.00	0.13	3.00	0.09	3.00	0.08	3.00	0.08	3.00	0.08
A4-3	3.00	0.09	3.00	0.06	3.00	0.08	3.00	0.08	3.00	0.08
A4-4	3.00	0.09	3.00	0.08	3.00	0.06	3.00	0.08	3.00	0.09
A4-5	3.00	0.09	3.00	0.13	3.00	0.08	3.00	0.08	3.00	0.08
A4-6	3.00	0.09	3.00	0.08	3.00	0.06	3.00	0.08	3.00	0.06
A4-7	3.00	0.25	3.00	0.08	3.00	0.08	3.00	0.08	3.00	0.06
A4-8	3.00	0.06	3.00	0.09	3.00	0.05	3.00	0.08	3.00	0.06
A4-9	3.00	0.42	3.00	0.08	3.00	0.08	3.00	0.09	3.00	0.08
A4-10	3.00	0.09	3.00	0.08	3.00	0.08	3.00	0.09	3.00	0.06
Média	3.00	0.14	3.00	0.09	3.00	0.07	3.00	0.08	3.00	0.07



TABELA A.2 – Resultados dos modelos de fluxo para as instâncias introduzidas por Catanzaro *et al.* (2015) - Grupo B.

	MF-PMTF-1		MF-PMTF-1A		MF-PMTF-1B		MF-PMTF-1C		MF-PMTF-1D	
	S	T	S	T	S	T	S	T	S	T
B1-1	19.00	2509.69	19.00	859.34	19.00	2211.06	19.00	2643.66	19.00	1181.38
B1-2	27.00	3600.00	26.00	3600.00	26.00	3600.00	26.00	3600.00	26.00	2147.53
B1-3	23.00	2073.52	23.00	2665.61	23.00	3600.00	23.00	3600.00	23.00	2615.28
B1-4	18.00	598.80	18.00	495.22	18.00	379.28	18.00	951.91	18.00	860.97
B1-5	20.00	3600.00	20.00	2420.23	21.00	3600.00	20.00	3600.00	20.00	1861.33
B1-6	21.00	753.50	21.00	932.17	21.00	742.92	21.00	1085.25	21.00	1215.22
B1-7	15.00	720.45	15.00	503.09	15.00	328.38	15.00	2072.42	15.00	178.50
B1-8	22.00	3600.00	22.00	2606.97	22.00	3600.00	22.00	3600.00	22.00	3258.36
B1-9	19.00	729.45	19.00	384.66	19.00	864.22	19.00	1985.80	19.00	553.45
B1-10	22.00	1308.22	22.00	830.31	22.00	1553.47	22.00	1716.89	22.00	806.19
Média	20.60	1241.95	20.50	1299.73	20.60	1013.22	20.50	1742.65	20.50	1467.82
B2-1	14.00	3600.00	14.00	1415.59	14.00	3116.47	14.00	3600.00	14.00	3600.00
B2-2	17.00	3600.00	17.00	2715.61	17.00	3600.00	17.00	3379.89	17.00	1568.22
B2-3	14.00	1655.33	14.00	437.14	14.00	758.05	14.00	1802.98	14.00	606.44
B2-4	12.00	1924.25	12.00	513.72	12.00	722.30	12.00	1837.66	12.00	2095.69
B2-5	15.00	3600.00	15.00	3600.00	15.00	1466.70	15.00	3600.00	15.00	3600.00
B2-6	14.00	918.19	14.00	381.67	14.00	801.45	14.00	2215.69	14.00	1421.56
B2-7	10.00	14.14	10.00	3.47	10.00	2.23	10.00	2.69	10.00	8.77
B2-8	13.00	1434.58	13.00	640.69	13.00	541.03	13.00	974.94	13.00	1431.45
B2-9	13.00	3600.00	13.00	484.38	13.00	2258.72	13.00	1370.44	13.00	1025.58
B2-10	15.00	3199.86	15.00	597.19	15.00	1069.19	15.00	550.09	15.00	578.27
Média	13.70	1524.39	13.70	798.83	13.70	1192.90	13.70	1516.80	13.70	1092.00
B3-1	10.00	30.06	10.00	4.69	10.00	3.81	10.00	13.56	10.00	20.30
B3-2	12.00	3600.00	12.00	1584.02	12.00	968.39	12.00	2382.97	12.00	3600.00
B3-3	10.00	3600.00	10.00	312.19	10.00	564.77	10.00	637.23	10.00	1917.39
B3-4	8.00	30.20	8.00	3.03	8.00	1.48	8.00	1.59	8.00	42.16
B3-5	11.00	3600.00	11.00	3600.00	11.00	1618.77	11.00	569.88	11.00	1376.14
B3-6	10.00	3600.00	10.00	72.58	10.00	502.97	10.00	147.06	10.00	401.28
B3-7	8.00	2.52	8.00	1.75	8.00	1.53	8.00	0.67	8.00	0.75
B3-8	9.00	17.80	9.00	1.77	9.00	10.55	9.00	7.03	9.00	164.83
B3-9	9.00	41.14	9.00	1.48	9.00	8.08	9.00	8.31	9.00	3.34
B3-10	10.00	126.86	10.00	30.38	10.00	30.45	10.00	44.27	10.00	143.42
Média	9.70	41.43	9.70	223.54	9.70	371.08	9.70	381.26	9.70	452.18
B4-1	8.00	0.38	8.00	0.22	8.00	0.49	8.00	0.33	8.00	0.23
B4-2	8.00	21.72	8.00	2.23	8.00	59.48	8.00	63.36	8.00	57.97
B4-3	7.00	1.05	7.00	3.17	7.00	4.02	7.00	1.53	7.00	0.22
B4-4	6.00	1.45	6.00	0.34	6.00	0.44	6.00	1.72	6.00	1.59
B4-5	8.00	1.06	8.00	0.81	8.00	0.53	8.00	0.41	8.00	0.33
B4-6	7.00	3.47	7.00	0.73	7.00	1.47	7.00	0.59	7.00	0.86
B4-7	6.00	0.94	6.00	0.22	6.00	0.30	6.00	0.31	6.00	0.47
B4-8	7.00	5.02	7.00	3.77	7.00	0.95	7.00	6.05	7.00	2.16
B4-9	7.00	1.06	7.00	0.80	7.00	0.94	7.00	0.98	7.00	0.20
B4-10	8.00	2.09	8.00	0.36	8.00	0.50	8.00	1.41	8.00	4.84
Média	7.20	3.82	7.20	1.27	7.20	6.91	7.20	7.67	7.20	6.89

TABELA A.3 – Resultados dos modelos de fluxo para as instâncias introduzidas por Catanzaro *et al.* (2015) - Grupo C.

	MF-PMTF-1	MF-PMTF-1A	MF-PMTF-1B	MF-PMTF-1C	MF-PMTF-1D
	S	S	S	S	S
C1-1	97.00	100.00	101.00	106.00	99.00
C1-2	108.00	110.00	108.00	115.00	107.00
C1-3	94.00	90.00	96.00	96.00	93.00
C1-4	98.00	95.00	94.00	97.00	97.00
C1-5	95.00	92.00	94.00	98.00	93.00
C1-6	107.00	110.00	109.00	111.00	105.00
C1-7	98.00	99.00	100.00	106.00	100.00
C1-8	95.00	96.00	98.00	96.00	93.00
C1-9	101.00	101.00	103.00	106.00	99.00
C1-10	78.00	82.00	82.00	85.00	80.00
Média	97.10	97.50	98.50	101.60	96.60
C2-1	78.00	80.00	78.00	82.00	78.00
C2-2	86.00	86.00	84.00	86.00	86.00
C2-3	73.00	75.00	75.00	79.00	75.00
C2-4	79.00	78.00	77.00	77.00	76.00
C2-5	76.00	77.00	78.00	76.00	74.00
C2-6	85.00	86.00	88.00	88.00	86.00
C2-7	77.00	78.00	78.00	83.00	77.00
C2-8	76.00	74.00	76.00	79.00	75.00
C2-9	80.00	79.00	79.00	84.00	77.00
C2-10	64.00	61.00	67.00	71.00	63.00
Média	77.40	77.40	78.00	80.50	76.70
C3-1	57.00	62.00	58.00	60.00	56.00
C3-2	63.00	63.00	61.00	67.00	61.00
C3-3	55.00	57.00	57.00	59.00	55.00
C3-4	56.00	52.00	56.00	61.00	55.00
C3-5	60.00	56.00	55.00	58.00	54.00
C3-6	61.00	62.00	62.00	69.00	61.00
C3-7	56.00	57.00	59.00	60.00	54.00
C3-8	54.00	57.00	54.00	58.00	51.00
C3-9	59.00	60.00	63.00	66.00	57.00
C3-10	48.00	52.00	50.00	51.00	46.00
Média	56.90	57.80	57.50	60.90	55.00
C4-1	32.00	34.00	36.00	37.00	31.00
C4-2	40.00	35.00	36.00	41.00	37.00
C4-3	31.00	30.00	36.00	35.00	30.00
C4-4	33.00	33.00	31.00	33.00	34.00
C4-5	34.00	33.00	33.00	34.00	34.00
C4-6	34.00	36.00	36.00	39.00	36.00
C4-7	32.00	33.00	35.00	35.00	34.00
C4-8	30.00	31.00	34.00	33.00	31.00
C4-9	35.00	34.00	36.00	38.00	34.00
C4-10	26.00	29.00	31.00	27.00	29.00
Média	32.70	32.80	34.40	35.20	33.00

TABELA A.4 – Resultados dos modelos de fluxo para as instâncias introduzidas por Catanzaro *et al.* (2015) - Grupo D.

	MF-PMTF-1	MF-PMTF-1A	MF-PMTF-1B	MF-PMTF-1C	MF-PMTF-1D
	S	S	S	S	S
D1-1	219.00	229.00	231.00	282.00	217.00
D1-2	183.00	179.00	196.00	228.00	176.00
D1-3	228.00	228.00	244.00	308.00	221.00
D1-4	221.00	230.00	229.00	0.00	216.00
D1-5	199.00	205.00	230.00	0.00	201.00
D1-6	240.00	256.00	255.00	547.00	238.00
D1-7	218.00	220.00	250.00	309.00	219.00
D1-8	193.00	194.00	211.00	581.00	191.00
D1-9	223.00	231.00	228.00	0.00	217.00
D1-10	230.00	247.00	241.00	0.00	228.00
Média	215.40	221.90	231.50	375.83	212.40
D2-1	190.00	188.00	203.00	239.00	187.00
D2-2	157.00	163.00	169.00	210.00	155.00
D2-3	195.00	195.00	197.00	249.00	193.00
D2-4	194.00	199.00	197.00	249.00	191.00
D2-5	183.00	176.00	188.00	625.00	186.00
D2-6	210.00	211.00	211.00	260.00	202.00
D2-7	182.00	192.00	192.00	562.00	179.00
D2-8	173.00	178.00	175.00	208.00	168.00
D2-9	192.00	191.00	194.00	255.00	189.00
D2-10	194.00	206.00	204.00	243.00	196.00
Média	187.00	189.90	193.00	310.00	184.60
D3-1	159.00	157.00	159.00	208.00	153.00
D3-2	126.00	146.00	133.00	0.00	127.00
D3-3	160.00	156.00	164.00	224.00	154.00
D3-4	149.00	162.00	183.00	590.00	158.00
D3-5	148.00	147.00	148.00	183.00	149.00
D3-6	167.00	171.00	182.00	197.00	168.00
D3-7	153.00	156.00	165.00	207.00	179.00
D3-8	136.00	143.00	148.00	172.00	146.00
D3-9	162.00	164.00	167.00	196.00	169.00
D3-10	158.00	165.00	167.00	577.00	160.00
Média	151.80	156.70	161.60	255.40	156.30
D4-1	115.00	115.00	122.00	473.00	112.00
D4-2	103.00	106.00	121.00	498.00	94.00
D4-3	114.00	122.00	114.00	128.00	114.00
D4-4	121.00	113.00	136.00	434.00	115.00
D4-5	110.00	116.00	120.00	144.00	108.00
D4-6	124.00	136.00	142.00	165.00	120.00
D4-7	115.00	117.00	136.00	498.00	111.00
D4-8	97.00	102.00	136.00	486.00	99.00
D4-9	119.00	124.00	127.00	186.00	118.00
D4-10	128.00	119.00	142.00	544.00	119.00
Média	114.60	117.00	129.60	355.60	111.00

# Anexo B - Publicações

Neste anexo, apresentam-se as publicações e apresentações de trabalho relacionadas ao PMTF e PCVMP, realizadas com resultados obtidos desta tese.

## Lista de artigos publicados:

- SILVA, T. T.; CHAVES, A. A. ; YANASSE, H. H. ; LUNA, H. P. L. . The multi-commodity traveling salesman problem with priority prizes: a mathematical model and metaheuristics. COMPUTATIONAL & APPLIED MATHEMATICS, v. 38, p. 188, 2019.

## Lista de artigos aceitos para publicação em revista:

- SILVA, T. T.; CHAVES, A. A. ; YANASSE, H. H. A New Multicommodity Flow Model for the Job Sequencing and Tool Switching Problem. INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH, 2020.

## Resumos expandidos publicados em anais de congressos:

- SILVA, T. T.; CHAVES, A. A. ; YANASSE, H. H. . A Relax-and-Fix with Fix-and-Optimize Heuristic Applied to Job Sequencing and Tool Switching Problem. In: INFORMS-ALIO International Conference, 2019, Cancún. Proceedings of INFORMS-ALIO 2019, 2019.

## Resumos publicados em anais de congressos:

- SILVA, T. T.; YANASSE, H. H. ; CHAVES, A. A. . A new mixed integer linear programming model for solving the minimization of tool switches problem. In: IFORS 2017, 2017, Quebec. Conference Program and Abstract Book, 2017. v. 1.
- SILVA, T. T.; YANASSE, H. H. ; CHAVES, A. A. ; LUNA, H. P. L. . Uma metaheurística Simulated Annealing para o Problema do Caixeiro Viajante *Multicommodity* com custos fixos, variáveis e prioridades. In: XXXVII CNMAC - Congresso Nacional de Matemática Aplicada e Computacional, 2017, São José dos Campos, SP. XXXVII CNMAC 2017, 2017.

**Apresentações de Trabalho:**

- SILVA, T. T.; CHAVES, A. A. ; YANASSE, H. H. . Um modelo de fluxo *multicommodity* para o problema de minimização de trocas de ferramentas com tamanhos de ferramentas não uniformes. ONPCE 2019.
- SILVA, T. T.; CHAVES, A. A. ; YANASSE, H. H. . Um modelo de fluxo para o problema de minimização de troca de ferramentas. ONPCE 2017.

## FOLHA DE REGISTRO DO DOCUMENTO

1. CLASSIFICAÇÃO/TIPO TD	2. DATA 10 de março de 2020	3. DOCUMENTO Nº DCTA/ITA/TD-003/2020	4. Nº DE PÁGINAS 164
5. TÍTULO E SUBTÍTULO: Novas formulações de fluxo para problemas de otimização combinatória			
6. AUTOR(ES): <b>Tiago Tiburcio da Silva</b>			
7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES): Instituto Tecnológico de Aeronáutica – ITA / Universidade Federal de São Paulo – UNIFESP			
8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR: Fluxo multicommodity; Problema de minimização de trocas de ferramentas; Problema do caixeiro viajante.			
9. PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO: Problema do caixeiro viajante; Modelos matemáticos; Otimização; Matemática computacional; Computação.			
10. APRESENTAÇÃO: <span style="float: right;"><input checked="" type="checkbox"/> Nacional    <input type="checkbox"/> Internacional</span> ITA/UNIFESP, São José dos Campos. Curso de Doutorado. Programa de Pós-Graduação em Pesquisa Operacional. Área de Métodos de Otimização. Orientador: Prof. Dr. Antônio Augusto Chaves. Coorientador: Prof. Dr. Horacio Hideki Yanasse. Defesa em 18/02/2020. Publicada em 2020.			
11. RESUMO: Neste trabalho aborda-se o Problema de Minimização de Trocas de Ferramentas (PMTF) e o Problema do Caixeiro Viajante Multiproduto com Prioridades (PCVMP). O PMTF consiste em determinar um sequenciamento de tarefas, de tal modo que a quantidade de trocas de ferramentas entre as tarefas seja a menor possível. Cada tarefa requer um conjunto de ferramentas distinto, e supõe-se que cada um destes conjuntos não contenha mais ferramentas do que suporta a máquina. Já o PCVMP consiste em determinar uma rota de entrega de mercadorias considerando ao mesmo tempo, o cliente e o vendedor, ou seja, minimizando os custos totais do vendedor e maximizando as preferências dos clientes. Neste estudo tem-se como objetivo modelar, baseado em fluxo <i>multicommodity</i> , os problemas citados. Modelos matemáticos de otimização foram propostos assim como alguns resultados teóricos foram desenvolvidos. No caso do PMTF, o melhor modelo proposto foi comparado com os modelos existentes na literatura, mostrando um melhor desempenho tanto em quantidade de instâncias resolvidas na otimalidade, quanto no valor da relaxação linear e no tempo de execução. Mostrou-se que o valor da relaxação linear nos modelos propostos corresponde a diferença entre a quantidade de ferramentas e a capacidade da máquina. Algumas metaheurísticas baseadas em busca por proximidade e um método exato enumerativo considerando eliminação de simetria foram propostos e comparados com os resultados da literatura. Já no caso do PCVMP, o modelo proposto se mostrou eficiente em resolver instâncias de pequeno e médio porte. Duas metaheurísticas, BRKGA e BRKGA adaptativo, ambas com busca local, também foram propostas para o PCVMP, apresentando bons resultados.			
12. GRAU DE SIGILO: <input checked="" type="checkbox"/> OSTENSIVO <input type="checkbox"/> RESERVADO <input type="checkbox"/> SECRETO			