

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
Escola de Engenharia  
Programa de Pós-Graduação em Engenharia de Produção

Guido Pantuza Júnior

**Formulações e Abordagens Lagrangeanas  
para Problemas de Subrotas: Aplicações ao  
Caixeiro Viajante com Coleta de Prêmios e ao  
Dimensionamento e Sequenciamento de Lotes**

Belo Horizonte

2022

Guido Pantuza Júnior

**Formulações e Abordagens Lagrangeanas para Problemas  
de Subrotas: Aplicações ao Caixeiro Viajante com Coleta  
de Prêmios e ao Dimensionamento e Sequenciamento de  
Lotes**

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Escola de Engenharia da Universidade Federal de Minas Gerais, como parte integrante para a obtenção do título de Doutor em Engenharia de Produção.

Orientador: Maurício Cardoso de Souza

Belo Horizonte

2022

P198f

Pantuza Júnior, Guido.

Formulações e abordagens lagrangeanas para problemas de subrotas [recurso eletrônico]: aplicações ao caixeiro viajante com coleta de prêmios e ao dimensionamento e seqüenciamento de lotes / Guido Pantuza Júnior. - 2022.

1 recurso online (159 f. : il., color.) : pdf.

Orientador: Maurício Cardoso de Souza.

Tese (doutorado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Apêndices: f. 133-159.

Bibliografia: f. 124-132.

Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia de produção - Teses. 2. Otimização combinatória – Teses. 3. Programação inteira – Teses. I. Souza, Maurício Cardoso de. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 658.5(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS  
Escola de Engenharia  
Programa de Pós-Graduação em Engenharia de Produção

## ATA DE DEFESA DE TESE

### GUIDO PANTUZA JÚNIOR

Realizou-se, no dia 22 de fevereiro de 2022, às 08:30 horas, online em <https://meet.google.com/wyu-aovs-ebt>, da Universidade Federal de Minas Gerais, a 58ª defesa de tese, intitulada *Formulações e Abordagens Lagrangeanas para Problemas de Subrotas: Aplicações ao Caixeiro Viajante com Coleta de Prêmios e ao Dimensionamento e Sequenciamento de Lotes*, apresentada por GUIDO PANTUZA JÚNIOR, número de registro 2014769723, graduado no curso de ENGENHARIA DE PRODUÇÃO, como requisito parcial para a obtenção do grau de Doutor em ENGENHARIA DE PRODUÇÃO, à seguinte Comissão Examinadora: Prof(a). Mauricio Cardoso de Souza - Orientador - Orientador (UFMG), Prof(a). Franklina Maria Bragion de Toledo (USP), Prof(a). Reinaldo Morabito Neto (UFSCAR), Prof(a). Alexandre Salles da Cunha (UFMG), Prof(a). Martin Gomez Ravetti (UFMG).

A Comissão considerou a tese:

( X ) Aprovada

( ) Reprovada

Finalizados os trabalhos, lavrei a presente ata que, lida e aprovada, vai assinada por mim e pelos membros da Comissão.

Belo Horizonte, 22 de fevereiro de 2022.

**Belo Horizonte, 22 de fevereiro de 2022.**

Assinatura dos membros da banca examinadora:

Prof(a). Mauricio Cardoso de Souza - Orientador ( Doutor )

Prof(a). Franklina Maria Bragion de Toledo ( Doutora )

Prof(a). Reinaldo Morabito Neto ( Doutor )

Prof(a). Alexandre Salles da Cunha ( Doutor )

Prof(a). Martin Gomez Ravetti ( Doutor )



Documento assinado eletronicamente por **Mauricio Cardoso de Souza, Professor do Magistério Superior**, em 02/03/2022, às 19:57, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Martin Gomez Ravetti, Professor do Magistério Superior**, em 03/03/2022, às 14:03, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Alexandre Salles da Cunha, Professor do Magistério Superior**, em 03/03/2022, às 17:47, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

---



Documento assinado eletronicamente por **Reinaldoi Morabito Neto, Usuário Externo**, em 04/03/2022, às 06:55, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

---



Documento assinado eletronicamente por **Franklina Maria Bragion de Toledo, Usuária Externa**, em 09/03/2022, às 16:51, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

---



A autenticidade deste documento pode ser conferida no site [https://sei.ufmg.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **1281487** e o código CRC **8468A559**.

---

*Em memória de Guido Pantuza.*

# Agradecimentos

*Quando alguém reclamar que você cometeu um erro, diga-lhe que pode ser algo bom. Porque sem a imperfeição nem você e nem eu existiríamos....*

*Stephen Hawking.*

*Físico teórico e cosmólogo britânico (1942 – 2018)*

Ao professor Maurício, que não mediu esforços para me ajudar e sem o qual este trabalho não seria possível participando como orientador, pesquisador e amigo.

À minha família, em especial, Mayra e Rafael, pelo amor, carinho, incentivo e ajuda em todos os momentos. À minha mãe Edna e meus irmãos, os grandes responsáveis pela minha formação.

Ao Departamento de Engenharia de Produção da Escola de Engenharia da Universidade Federal de Minas Gerais e, em especial, aos seus professores por me proporcionarem os ensinamentos necessários.

A todos os meus amigos que me ajudaram e contribuíram para esta conquista, principalmente, àqueles da República Senzala.

Aos membros da banca pela contribuição indispensável para a qualidade deste trabalho.

A todos os professores que participaram, da minha formação profissional e crescimento pessoal durante este período.

Finalmente, a todos que, direta ou indiretamente, contribuíram para a conclusão deste trabalho.

*A distância do percurso até seu  
objetivo é proporcional à sua  
determinação e paciência, não desista,  
pois, enquanto há vida há esperança!*



# Resumo

Este trabalho trata do Problema do Caixeiro Viajante com Coleta de Prêmios – PCTSP, versão assimétrica. Ele consiste em encontrar uma rota de custo mínimo, que inicie no vértice raiz  $r$ , e visite cada vértice  $i$ , no máximo, uma única vez e colete um prêmio associado a cada vértice. O objetivo deste trabalho é propor novas formulações e um algoritmo híbrido para a resolução do PCTSP e problemas correlatos. Entre os problemas correlatos, escolheu-se o Problema de Dimensionamento de Lotes com Sequenciamento – LSSP. Este problema, assim como o PCTSP, pertence a um grupo de problemas de Otimização Combinatória conhecido como Problemas de Sub-Rotas – StP. Os problemas deste grupo são caracterizados por dois tipos de decisão: na seleção de quais itens estarão ativos na solução (programação); e seu respectivo sequenciamento (roteamento). Assim como os StP, o LSSP consiste em determinar o tamanho do lote de produção de cada produto (programação), bem como a sequência na qual os produtos serão produzidos (roteamento), sujeito às restrições de demanda e capacidade produtiva. Repare que para cada período  $t$  do horizonte de planejamento surge um PCTSP, ou seja, fixado  $t$ , o LSSP consiste em selecionar o subconjunto de itens que será produzido, bem como, a ordem de produção. Note que os itens que serão produzidos podem ser interpretados como os vértices que serão visitados. Logo, o PCTSP surge dentro do LSSP a partir da divisão do horizonte de planejamento em um conjunto finito de períodos. Dessa forma, o LSSP pode ser visto como um PCTSP com complicantes. Pois surge um PCTSP para cada período, e, devido à restrição de conservação do estado da máquina (*carryover*), a solução do período  $t$  depende do período anterior  $t - 1$ . Para resolver os problemas, inicialmente, são retiradas da literatura as principais formulações para cada problema. Em seguida, são apresentadas novas formulações para os problemas. Uma nova formulação proposta neste trabalho mostra-se a mais forte ao dominar as demais consideradas. Entretanto, seu uso para resolver os problemas até a otimalidade é restrito devido às suas dimensões. Neste caso, propõe-se um algoritmo híbrido baseado na relaxação Lagrangeana resultante desta formulação forte. Este algoritmo combina técnicas heurísticas, como um procedimento de busca local, com planos de corte implementados de acordo com um esquema *relax-and-cut*. Estes procedimentos são embutidos dentro do Algoritmo Volume. O qual é utilizado para resolver o problema dual Lagrangeano. Os resultados demonstram que a abordagem proposta é eficiente para o PCTSP e o LSSP. Ele foi capaz de limitar o valor da solução ótima dessas grandes instâncias em pequenos intervalos de otimalidade, com um tempo computacional moderado. Ainda, o algoritmo é eficiente para obter soluções viáveis em um tempo razoável. Ele destaca-se, especialmente, para as instâncias mais difíceis nas quais o método exato não foi sequer capaz de encontrar soluções viáveis.

**Palavras-chave:** Programação Inteira, Otimização Combinatória, Relaxação Lagrangeana, Problema do Caixeiro Viajante, Problema de Dimensionamento e Sequenciamento de Lotes.

# Abstract

This paper deals with the Traveling Salesman Problem with Prize Collection – PCTSP, asymmetric version. It consists of finding a minimum cost route that starts at the root node  $r$ , visits each node  $i$  at most once, and collects a prize associated with each node. The goal of this work is to propose new formulations and a hybrid algorithm for solving PCTSP and related problems. Among the related problems, the Lot Sizing with Sequencing Problem – LSSP – was chosen. This problem, like PCTSP, belongs to a group of Combinatorial Optimization problems known as Subroutine Problems – StP. Problems in this group are characterized by two types of decisions: in the selection of which items will be active in the solution (scheduling); and their respective sequencing (routing). Like StP, LSSP consists in determining the production lot size for each product (scheduling), as well as the sequence in which the products will be produced (routing), subject to demand and capacity constraints. Notice that for each period  $t$  of the planning horizon a PCTSP arises, that is, fixed  $t$ , the LSSP consists in selecting the subset of items that will be produced, as well as, the production order. Note that the items that will be produced can be interpreted as the vertices that will be visited. Therefore, the PCTSP arises within the LSSP from the division of the planning horizon into a finite set of periods. In this way, the LSSP can be seen as a PCTSP with complicants. Because a PCTSP arises for each period, and, due to the machine state conservation constraint (*carryover*), the solution of period  $t$  depends on the previous period  $t - 1$ . To solve the problems, initially, the main formulations for each problem are taken from the literature. Next, new formulations for the problems are presented. A new formulation proposed in this paper proves to be the strongest by dominating the others considered. However, its use to solve the problems up to optimality is restricted due to its dimensions. In this case, a hybrid algorithm based on the Lagrangean relaxation resulting from this tight formulation is proposed. This algorithm combines heuristic techniques, such as a local search procedure, with cutting plans implemented according to a relax-and-cut scheme. These procedures are embedded within the Volume Algorithm. Which is used to solve the dual Lagrangean problem. It was able to bound the optimal solution value of these large instances in small optimality intervals with moderate computational time. Also, the algorithm is efficient in obtaining feasible solutions in a reasonable amount of time. It especially stands out for the most difficult instances where the exact method was not even able to find feasible solutions.

**Keywords:** Integer Programming, Combinatorial Optimization, Lagrangian Relaxation, Traveling Salesman Problem, Lot Sizing and Scheduling Problem.

# Lista de figuras

Figura 1 – Ilustração da relação entre os politopos $P(F1)$ e $P(F2)$ . . . . .	22
Figura 2 – Ilustração de uma solução para a formulação <b>MCF</b> com o vértice raiz $r = 1$ e $ V  = 6$ . . . . .	36
Figura 3 – Ilustração de uma solução para a formulação <b>EPN</b> com o vértice raiz $r = 1$ e $ V  = 4$ . . . . .	42
Figura 4 – Ilustração de uma solução para a formulação <b>PSC</b> com o vértice raiz $r = 1$ e $ V  = 6$ . . . . .	43
Figura 5 – Ilustração do caminho mínimo para resolver LS-U (dado por (4.49)) com $ T  = 4$ . . . . .	100

# Lista de tabelas

Tabela 1	– Associação dos multiplicadores de Lagrange com as restrições da <b>EPN</b> . . . . .	44
Tabela 2	– Resultados da comparação da relaxação linear do grupo <i>realworld</i> para as formulações com tempo limite de 10800 segundos (PCTSP). . . . .	61
Tabela 3	– Resultados para as 39 instâncias, para além das 44 detalhadas na Tabela 2 para as quais ou <b>MCF</b> ou <b>EPN</b> resolveram a relaxação linear dentro do limite de tempo de 10.800 segundos. . . . .	62
Tabela 4	– Resultados da comparação da relaxação linear do grupo <i>crane</i> para as formulações com tempo limite de 10800 segundos (PCTSP). . . . .	63
Tabela 5	– Resultados da comparação da relaxação linear do grupo <i>stilt</i> para as formulações com tempo limite de 10800 segundos (PCTSP). . . . .	63
Tabela 6	– Resultado do Algoritmo 4 para as 78 instâncias do grupo <i>realworld</i> nas quais a relaxação linear de <b>EPN</b> é conhecida. . . . .	66
Tabela 7	– Resultado do Algoritmo 4 para as 45 instâncias do grupo <i>realworld</i> nas quais a relaxação linear de <b>EPN</b> não é conhecida. . . . .	69
Tabela 8	– Resultado do Algoritmo 4 para as 39 instâncias do grupo <i>crane</i> nas quais a relaxação linear de <b>EPN</b> é conhecida. . . . .	71
Tabela 9	– Resultado do Algoritmo 4 para as 30 instâncias do grupo <i>crane</i> nas quais a relaxação linear de <b>EPN</b> não é conhecida. . . . .	72
Tabela 10	– Resultado do Algoritmo 4 para as 30 instâncias do grupo <i>stilt</i> nas quais a relaxação linear de <b>EPN</b> é conhecida. . . . .	73
Tabela 11	– Resultado do Algoritmo 4 para as 30 instâncias do grupo <i>stilt</i> nas quais a relaxação linear de <b>EPN</b> não é conhecida. . . . .	75
Tabela 12	– Desigualdades válidas do Algoritmo 4 – PCTSP . . . . .	77
Tabela 13	– Associação dos multiplicadores de Lagrange com as restrições da <b>EPN'</b> . . . . .	94
Tabela 14	– Resultados da comparação da relaxação linear dos 24 grupos ( $P, T, \theta, cut$ ) de 10 instâncias cada (LSSP). . . . .	112
Tabela 15	– Resultado do Algoritmo 4 para as 138 instâncias nas quais a solução ótima não é conhecida. . . . .	115
Tabela 16	– Resultados da comparação da qualidade dos UB's. . . . .	119
Tabela 17	– Impacto das desigualdades válidas no Algoritmo 4 – LSSP . . . . .	119
Tabela 18	– Resultados da instâncias do grupo <i>realworld</i> resolvidas por todas as 6 formulações com tempo limite de 10800 segundos (PCTSP). . . . .	134
Tabela 19	– Resultados da instâncias do grupo <i>realworld</i> resolvidas por pelo menos uma formulação com tempo limite de 10800 segundos (PCTSP). . . . .	135

Tabela 20 – Resultados da relaxação linear da formulação <b>EPN</b> das instâncias do grupo <i>realword</i> . . . . .	137
Tabela 21 – Resultados da relaxação linear da formulação <b>EPN</b> das instâncias do grupo <i>stilt</i> . . . . .	140
Tabela 22 – Resultados da relaxação linear da formulação <b>EPN</b> das instâncias do grupo <i>crane</i> . . . . .	141
Tabela 23 – Resultados da instâncias resolvidas por todas as formulações com tempo limite de 3600 segundos (LSSP). . . . .	142
Tabela 24 – Resultados da instâncias resolvidas por pelo menos uma formulação com tempo limite de 3600 segundos (LSSP) . . . . .	144
Tabela 25 – Resultados detalhados da comparação da relaxação linear das 240 instâncias para as formulações – LSSP. . . . .	146

# Lista de abreviaturas e siglas

alg	Algoritmo
ATSP	Problema do Caixeiro Viajante Assimétrico ( <i>Asymmetric Traveling Salesman Problem</i> )
DDW	Técnica de decomposição de Dantzig-Wolfe
EnC	formulação <i>Enhanced circuit</i>
EPN	Formulação <i>Enhanced path-node</i>
FO	Função Objetivo
GRASP	Greedy Randomized Adaptive Search Procedure
GSEC	<i>Generalized Subtour Elimination Constraint</i>
ILS	Iterated Local Search
KP	Problema da Mochila ( <i>Knapsack Problem</i> )
LB	limite inferior ( <i>lower bound</i> )
LP	Relaxação Linear ( <i>linear relaxation</i> )
LSSP	Problema de Dimensionamento de Lotes com Sequenciamento ( <i>Lot Sizing Scheduling Problem</i> )
LS-U	<i>Single-Item Uncapacitated Lot-Sizing Problem</i>
MCF	Formulação <i>Multicommodity flow</i>
MSP	<i>Merchant Subtour Problem</i>
OC	Otimização Combinatória
OP	<i>Orienteering Problem</i>
PC-ST	<i>Prize and Collecting Steiner Tree Problem</i>
PCTSP	Problema do Caixeiro Viajante com Coleta de Prêmios ( <i>Prize Collecting Traveling Salesman Problem</i> )
PPF	Formulações de Caminho e Fluxo ( <i>Lifted Path and Flow Formulations</i> )

PLF	Problema de localização de facilidades
PSC	<i>Prize-single commodity formulation</i>
PTP	<i>Profitable Tour Problem</i>
QTST	<i>Quota Traveling Salesman Problem</i>
RCTSP	<i>Resource Constrained Travelling Salesman Problem</i>
resp	respectivamente
RHS	lado direito das restrições ( <i>Right hand side</i> )
RLT	<i>Reformulation-Linearization Technique</i>
SA	<i>Simulated Annealing</i>
SCF	Fluxo de uma Única Mercadoria ( <i>Single Commodity Flow Based Formulations</i> )
SEC	<i>(Subtour Elimination Constraint).</i>
SG	Subgradiente clássico
SSP	Problema de Caminho Mínimo ( <i>Shortest Path Problem</i> )
StP	Problemas de Sub-Rotas ( <i>Subtour Problems</i> )
SVM	<i>Framework Support Vector Machine</i>
TSP	Problema do Caixeiro Viajante ( <i>Traveling Salesman Problem</i> )
TW-PCTSP	Problema do Caixeiro Viajante com Coleta de Prêmios com janelas de tempo ( <i>Prize Collecting Traveling Salesman Problem with time windows</i> )
UB	limite superior ( <i>upper bounds</i> )
VNS	Variable neighborhood search.
VND	Variable Neighborhood Descent.
VOL	Algoritmo Volume ( <i>Volume Algorithm</i> )



# List of Algorithms

Algoritmo 1 – Subproblema $LR^y$ . . . . .	47
Algoritmo 2 – Subproblema $(LR^x)$ . . . . .	48
Algoritmo 3 – Subproblema $LR^z$ . . . . .	49
Algoritmo 4 – Abordagem por meio da relaxação Lagrangeana . . . . .	54
Algoritmo 5 – Subproblema $LR^{x_r}$ . . . . .	97
Algoritmo 6 – Busca Local para o LSSP . . . . .	104

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>21</b>
<b>1.1</b>	<b>Objetivos</b>	<b>25</b>
1.1.1	Objetivo geral	25
1.1.2	Objetivos específicos	25
<b>1.2</b>	<b>Organização do trabalho</b>	<b>26</b>
<b>2</b>	<b>FORMULAÇÕES E RELAXAÇÃO LAGRANGEANA PARA O PROBLEMA DO CAIXEIRO VIAJANTE COM COLETA DE PRÊMIOS</b>	<b>28</b>
<b>2.1</b>	<b>Definição do problema</b>	<b>28</b>
<b>2.2</b>	<b>Revisão da literatura</b>	<b>29</b>
<b>2.3</b>	<b>Formulações</b>	<b>32</b>
2.3.1	Formulação <b>GSEC</b>	34
2.3.2	Formulação <b>MCF</b>	35
2.3.3	Formulação <b>OPB</b>	36
2.3.4	Formulação <b>EnC</b>	38
2.3.5	Formulação <b>EPN</b>	40
2.3.6	Formulação <b>PSC</b>	41
<b>2.4</b>	<b>Relaxação Lagrangeana para o PCTSP</b>	<b>43</b>
2.4.1	Solução inicial	49
2.4.2	Busca local	50
2.4.3	Desigualdades válidas e cortes de otimalidade para o PCTSP	51
2.4.3.1	Cortes Cost-Cover	52
2.4.3.2	Desigualdades Cycle-Cover	52
<b>2.5</b>	<b>Algoritmo proposto para o PCTSP</b>	<b>53</b>
<b>3</b>	<b>EXPERIMENTOS COMPUTACIONAIS SOBRE O PROBLEMA DO CAIXEIRO VIAJANTE COM COLETA DE PRÊMIOS</b>	<b>57</b>
<b>3.1</b>	<b>Instâncias para o PCTSP</b>	<b>57</b>
<b>3.2</b>	<b>Resultados da relaxação linear</b>	<b>60</b>
<b>3.3</b>	<b>Resultados da abordagem Lagrangeana</b>	<b>64</b>
<b>3.4</b>	<b>Resultados das famílias de cortes de otimalidade e desigualdades válidas</b>	<b>76</b>

<b>4</b>	<b>FORMULAÇÕES E RELAXAÇÃO LAGRANGEANA PARA O PROBLEMA DE DIMENSIONAMENTO E SEQUENCIAMENTO DE LOTES . . . . .</b>	<b>79</b>
<b>4.1</b>	<b>Definição do problema . . . . .</b>	<b>80</b>
<b>4.2</b>	<b>Revisão da literatura . . . . .</b>	<b>81</b>
<b>4.3</b>	<b>Formulações . . . . .</b>	<b>86</b>
4.3.1	Formulação <b>AL</b> . . . . .	88
4.3.2	Formulação <b>SCF</b> . . . . .	90
4.3.3	Formulação <b>EPN'</b> . . . . .	91
4.3.4	Formulação <b>CSC</b> . . . . .	93
<b>4.4</b>	<b>Relaxação Lagrangeana para o LSSP . . . . .</b>	<b>94</b>
4.4.1	Solução Inicial . . . . .	100
4.4.2	Busca Local . . . . .	102
4.4.3	Desigualdades válidas para o LSSP . . . . .	105
4.4.3.1	Desigualdades $W_t$ . . . . .	105
4.4.3.2	Desigualdades $(L, S)$ . . . . .	106
<b>4.5</b>	<b>Algoritmo proposto para o LSSP . . . . .</b>	<b>107</b>
<b>5</b>	<b>EXPERIMENTOS COMPUTACIONAIS SOBRE O PROBLEMA DE DIMENSIONAMENTO E SEQUENCIAMENTO DE LOTES</b>	<b>109</b>
<b>5.1</b>	<b>Instâncias para o LSSP . . . . .</b>	<b>109</b>
<b>5.2</b>	<b>Resultados da relaxação linear . . . . .</b>	<b>110</b>
<b>5.3</b>	<b>Resultados da abordagem Lagrangeana . . . . .</b>	<b>113</b>
5.3.1	Comparação dos LB's e <i>gaps</i> de otimalidade . . . . .	114
5.3.2	Comparação dos UB's . . . . .	118
<b>5.4</b>	<b>Resultados das desigualdades válidas . . . . .</b>	<b>119</b>
<b>6</b>	<b>CONSIDERAÇÕES FINAIS . . . . .</b>	<b>121</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>124</b>
	<b>APÊNDICES . . . . .</b>	<b>133</b>
	<b>APÊNDICE A – RESULTADOS DOS MODELOS INTEIROS E DA RELAXAÇÃO LINEAR DE EPN . . . . .</b>	<b>134</b>
	<b>APÊNDICE B – RESULTADOS DOS MODELOS INTEIROS – LSSP . . . . .</b>	<b>142</b>
	<b>APÊNDICE C – RESULTADOS ADICIONAIS DA RELAXAÇÃO LINEAR – LSSP . . . . .</b>	<b>146</b>

**APÊNDICE D – PROCESSO DE CALIBRAÇÃO – PCTSP 156**

**APÊNDICE E – PROCESSO DE CALIBRAÇÃO – LSSP . 159**

# 1 Introdução

O Problema do Caixeiro Viajante com Coleta de Prêmios, conhecido na literatura como *Prize Collecting Traveling Salesman Problem* – PCTSP, é uma generalização do amplamente estudado Problema do Caixeiro Viajante (*Traveling Salesman Problem* – TSP). Ele diferencia-se do TSP, pois permite que nem todo vértice seja visitado. Assim, a cada vértice  $i$  está associado um prêmio  $p_i$  e uma penalidade  $w_i$ . Toda vez que um vértice é visitado, um prêmio é coletado, e, a cada vértice não visitado, uma penalidade é paga. Logo, os custos da rota são compostos pela distância entre os vértices visitados mais a penalidade dos vértices não visitados. O PCTSP tem como objetivo encontrar uma rota de custo mínimo, que inicie no vértice raiz  $r$ , que cada vértice  $i$  seja visitado, no máximo, uma única vez, que colete um prêmio mínimo  $g$ , e, no fim da rota, retorne ao vértice raiz.

O PCTSP, o qual é o foco deste trabalho, foi escolhido a partir da constatação de que ele possui uma grande variedade de aplicações. Como afirma Balas (1989), uma série de problemas de programação e roteamento, encontrados na literatura, podem ser formulados como variantes ou generalizações do PCTSP.

Esses problemas, de programação e roteamento, pertencem a um grupo que envolve dois tipos de decisão: a seleção de quais vértices serão visitados (programação); e qual a ordem das visitas (roteamento). De acordo com Kubo e Kasugai (1992), este grupo de problemas, pode ser chamado de Problemas de Sub-Rotas (*Subtour Problems* – StP). Alguns exemplos de problemas deste grupo são o *Orienteering Problem* – OP, Vohra, Levy e Golden (1987), o *Selective TSP*, Laporte e Martello (1990), o *Profitable Tour Problem* – PTP, Dell’Amico, Maffioli e Varbrand (1995), o *Merchant Subtour Problem* – MSP, Verweij e Aardal (2003), o *TSP with Profits*, Feillet, Dejax e Gendreau (2005) e o *Quota TSP* – QTSP, Ausiello et al. (2004).

Outro importante exemplo desse grupo é o Problema de Dimensionamento de Lotes com Sequenciamento, conhecido na literatura como *Lot Sizing and Scheduling Problem* – LSSP. Este consiste em determinar a programação da produção. Ou seja, determinar o tamanho do lote de produção de cada produto, bem como a sequência na qual os produtos serão produzidos. O objetivo é minimizar os custos envolvidos, sujeito às restrições de demanda e capacidade produtiva, em um horizonte de planejamento finito e conhecido.

Repare que para cada período  $t$  do horizonte de planejamento surge um PCTSP, ou seja, fixado  $t$ , o LSSP consiste em selecionar o subconjunto de itens que será produzido, bem como, a ordem de produção. Note que os itens que serão

produzidos podem ser interpretados como os vértices que serão visitados. Logo, o PCTSP surge dentro do LSSP a partir da divisão do horizonte de planejamento em um conjunto finito de períodos. Dessa forma, o LSSP pode ser visto como um PCTSP com complicantes, pois surge um PCTSP para cada período.

Entre os diversos métodos da literatura, este grupo de problemas pode ser resolvido por modelos de programação matemática, por intermédio de pacotes computacionais, que garantem a otimalidade da solução (solução ótima). Esses pacotes são conhecidos como *solvers* e seu desempenho está diretamente ligado à força e às dimensões da formulação matemática utilizada.

As dimensões da formulação são medidas por meio da quantidade de restrições e variáveis da formulação. A força de uma formulação pode ser medida por intermédio da distância relativa entre o valor de sua relaxação linear (*linear relaxation* – LP), e da solução ótima. Quanto menor esta distância mais forte (ou mais apertada) é a formulação. A relaxação linear consiste em ignorar o domínio das variáveis inteiras ou binárias. Geralmente, elas são consideradas como variáveis pertencentes ao conjunto dos números Reais, não negativos ( $\mathbb{R}_+$ ). Dessa forma, é possível afirmar que, dadas duas formulações  $F1$  e  $F2$ , aquela que obtiver o melhor valor para a relaxação linear, para toda e qualquer instância, é dita mais forte.

A força de uma formulação está relacionada ao poliedro descrito pela mesma. Considere, por exemplo, duas formulações diferentes,  $F1$  e  $F2$ , declaradas no mesmo espaço vetorial,  $x \in \mathbb{R}^2$ , tal que  $x \geq 0$ . Definido como  $P(F1)$  e  $P(F2)$  o poliedro associado a cada uma das formulações  $F1$  e  $F2$ . Se  $P(F1) \subset P(F2)$ , então  $F1$  é uma formulação mais forte, ou mais apertada, que  $F2$ , Fig. 1-(B). Ou seja, ao resolver as formulações linearmente relaxadas, o valor encontrado para  $F1$  será igual ou melhor que  $F2$ . Caso o resultado varie de acordo com a instância, ou seja, para alguns problemas  $F1$  é melhor, para outros  $F2$  é melhor, as formulações são ditas incomparáveis, Fig. 1-(A). Outro resultado possível é que o valor seja sempre igual, neste caso elas são ditas equivalentes,  $P(F1) \equiv P(F2)$ , Fig. 1-(C).

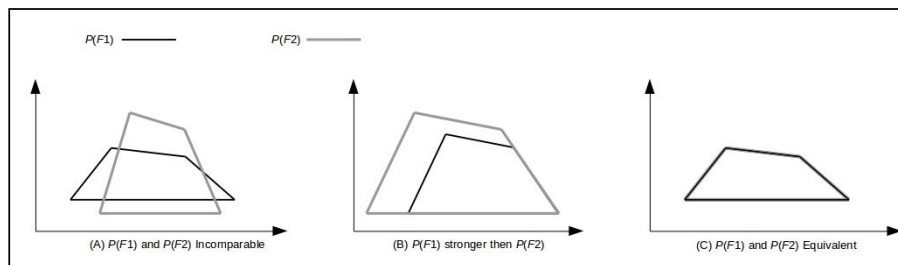


Figura 1 – Ilustração da relação entre os politopos  $P(F1)$  e  $P(F2)$

Porém, como foi observado por Godinho, Gouveia e Pesneau (2011) para o TSP assimétrico - (ATSP), para obter uma formulação mais forte, na maioria das vezes, é necessário aumentar o número de variáveis e/ou restrições. Dessa forma, a formulação pode tornar-se cara computacionalmente, em especial, devido à grande demanda de memória. Neste caso, seu uso para obter a solução ótima de um problema é inviável, mesmo para problemas de pequenas dimensões.

Por outro lado, formulações mais compactas demandam menos memória. Contudo, geralmente, elas são mais fracas e necessitam de um número de iterações maior para encontrar uma solução ótima, se comparado à uma formulação mais forte. Isto implica em um tempo de execução muito elevado, o que também inviabiliza seu uso para obter uma solução ótima de um problema.

Logo, para uma formulação apresentar um bom desempenho computacional, ela deve possuir um equilíbrio entre a sua dimensão e sua força. Isto pode ser observado nos dos trabalhos de Sherali, Sarin e Tsai (2006) e Guimarães, Klabjan e Almada-Lobo (2014).

No tocante à sua complexidade, os problemas deste grupo, em especial o PCTSP e o LSSP, são, respectivamente, classificados por Bienstock et al. (1993) e Bitran e Yanasse (1982) como NP-difícil (NP-*hard*). Isto significa que não se conhece um algoritmo polinomial capaz de encontrar a solução ótima. Ou seja, para instâncias com dimensões maiores, pode não ser possível encontrar a solução ótima em tempo hábil.

Para estes problemas a resolução por intermédio de uma abordagem Heurística pode ser mais indicada. Heurísticas são técnicas de resolução por meio de métodos de inteligência computacional. De acordo com Burke et al. (2003), elas consistem em um procedimento de decisão baseado em uma estrutura de controle repetitivo. Elas não possuem garantia da otimalidade da solução (ótimo global), porém espera-se que sejam capazes de encontrar boas soluções (ótimos locais), em tempo hábil.

Para problemas mais complexos ou para instâncias de dimensões maiores, como aqueles encontrados no mundo real, as heurísticas clássicas encontram dificuldade para gerar soluções razoáveis.

Entretanto, Gunawan, Lau e Vansteenwegen (2016) observaram que as heurísticas híbridas têm alcançado resultados satisfatórios. Além disso, elas são muito utilizadas na resolução de problemas combinatoriais, como os que pertencem ao StP. Na literatura é possível encontrar diversos trabalhos como Carvalho e Nascimento (2016), Yu et al. (2019) e Zhang et al. (2020) que obtiveram desempenho superior aos demais métodos. Esses trabalhos citados integram características das heurísticas clássicas, como os algoritmos de busca local, com a Relaxação Lagrangeana.

Os algoritmos de busca local são métodos de otimização que consistem no refinamento de uma solução. Ou seja, a partir de uma solução inicial, busca-se, a cada iteração, em sua vizinhança já definida, novas soluções que possam melhorar o valor da FO do problema.

A Relaxação Lagrangeana surgiu a partir da observação que problemas difíceis podem ser vistos como problemas fáceis, por meio da relaxação das restrições complicantes. Estas restrições relaxadas, ou dualizadas, são acrescentadas à FO e penalizadas por intermédio de multiplicadores, conhecidos como multiplicadores de Lagrange. O problema resultante é chamado de Problema Lagrangeano.

Assim, para resolver o PCTSP, são utilizadas seis formulações neste trabalho: duas delas usam ideias clássicas de eliminação de subrotas e formulações multimercadarias (*multicommodities*); duas são adaptações ao PCTSP de formulações fortes para o TSP assimétrico retiradas de Sherali, Sarin e Tsai (2006) e Godinho, Gouveia e Pesneau (2011); e duas são propostas nesse trabalho.

Definidas as formulações, elas são comparadas em termos do desempenho da relaxação linear. Nesta etapa da pesquisa é verificada a força das formulações, bem como o seu custo computacional com o uso de instâncias retiradas da literatura. Ainda nesta etapa, percebe-se que nenhuma formulação é capaz de resolver as instâncias mais difíceis até a otimalidade dentro de um limite de tempo razoável. Mesmo para a relaxação linear, a maioria das formulações não obteve sucesso com as instâncias mais difíceis (Ou seja, não foi capaz de resolver). Entretanto, uma formulação se destaca entre as demais. Ela apresenta excelentes valores para a relaxação linear, por isso ela pode ser considerada como uma formulação forte.

Dado que foi identificado uma formulação forte para o PCTSP e, que mesmo para a relaxação linear, não é possível resolver os problemas maiores. A relaxação Lagrangeana emerge neste contexto como uma boa alternativa para resolver os problemas. Segundo Beasley (1993), esta abordagem é adequada para gerar bons limites inferiores (*lower bounds* – LB) para a solução ótima. Ainda, como observado por Fisher (1985), os subproblemas Lagrangeanos, resultantes da relaxação, são uma fonte de informação valiosa para gerar boas soluções viáveis.

Logo, é proposto um algoritmo híbrido, que utiliza uma abordagem Lagrangeana combinada com heurísticas de busca local, para obter limites inferiores e superiores (*upper bounds* – UB) para resolver o PCTSP. Ainda, essa abordagem Lagrangeana faz uso de desigualdades válidas e/ou cortes de otimalidade da literatura, em um esquema de acordo com o procedimento *relax and cut* como em Guignard (2003) e Lucena (2005), para melhorar o LB do problema Lagrangeano. Os resultados mostram o potencial desta abordagem para gerar limites para a solução ótima e, em alguns casos, encontrar a solução ótima.



O sucesso dessa abordagem para o PCTSP despertou o interesse na sua aplicação para outros problemas correlatos. Deseja-se verificar se o seu desempenho é comparável. Neste contexto, optou-se pelo LSSP.

Ele foi escolhido, pois apesar de ser um problema correlato, esta relação não é tão evidente. O PCTSP surge dentro do LSSP a partir da divisão do horizonte de planejamento em um conjunto finito de períodos. Assim, para cada período, o conjunto de itens que serão produzidos pode ser interpretado como os vértices que serão visitados. Note que, dessa forma, o LSSP pode ser visto como um PCTSP com complicantes. Pois surge um PCTSP para cada período, e a solução do período  $t$  depende do período anterior  $t - 1$ . Esta condição de transferência do estado da máquina para o período seguinte é conhecida como restrição *carryover*.

Neste caso, são consideradas quatro formulações para este problema. A primeira foi proposta por Almada-Lobo et al. (2007) e a segunda por Guimarães, Klabjan e Almada-Lobo (2014). Como observado neste último trabalho, ambas estão entre as principais formulações da literatura para o LSSP. As outras duas são adaptações das mesmas formulações propostas para o PCTSP, neste trabalho. Nesta etapa da pesquisa é possível observar que o desempenho da formulação forte foi confirmado. Ou seja, novamente, ela apresentou os melhores valores para a relaxação linear.

A partir do bom desempenho computacional apresentado pelo algoritmo proposto na resolução do PCTSP. Assim como no caso das formulações, este algoritmo é adaptado e aplicado ao LSSP. Note que a formulação sobre a qual a relaxação Lagrangeana se baseia é a mesma. Novamente, os resultados mostram que é possível gerar limites para a solução ótima e, inclusive, encontrá-la em alguns casos.

## 1.1 Objetivos

### 1.1.1 Objetivo geral

O objetivo deste trabalho é propor novas formulações e um algoritmo híbrido para a resolução do Problema do Caixeiro Viajante com Coleta de Prêmios e problemas correlatos.

### 1.1.2 Objetivos específicos

Os objetivos específicos são:

1. Adaptar as melhores formulações do TSP para o PCTSP;
2. Propor novas formulações matemáticas para o PCTSP;
3. Desenvolver um algoritmo híbrido para resolver o PCTSP;

4. Adaptar as melhores formulações do PCTSP para o LSSP;
5. Propor novas formulações matemáticas para o LSSP;
6. Adaptar o algoritmo proposto para o LSSP;
7. Possibilitar maior eficiência para resolução de problemas combinatórios;
8. Contribuir com o desenvolvimento de metodologias que atendam problemas reais.

## 1.2 Organização do trabalho

Para a concretização deste trabalho foram determinados, neste primeiro capítulo, os norteadores da pesquisa, uma rápida contextualização e a justificativa contidos na introdução. A qual é seguida pelo objetivo principal, bem como os objetivos secundários. O restante deste trabalho está estruturado como segue.

No Capítulo 2 o problema foco deste trabalho é definido conjuntamente com uma revisão da literatura sobre diferentes abordagens para sua resolução. Então, as formulações consideradas são apresentadas. Na sequência, é apresentada a relaxação Lagrangeana da formulação de maior destaque, seguida de famílias de desigualdades válidas e cortes de otimalidade para o PCTSP. Finalmente, a heurística Lagrangeana híbrida proposta é explicada.

O Capítulo 3 mostra os resultados dos experimentos computacionais para o PCTSP. Primeiro, são descritas as instâncias retiradas da literatura do ATSP. Em seguida, os resultados da comparação das formulações, em termos da relaxação linear, são apresentados. Então, são mostrados os resultados da abordagem Lagrangeana proposta. Por último, é apresentado o impacto nos resultados provenientes da utilização das desigualdades válidas e dos cortes de otimalidade na abordagem proposta.

No Capítulo 4 um problema correlato ao PCTSP é definido (LSSP), e uma revisão da literatura sobre o tema é apresentada. Também são mostradas as principais formulações, retiradas da literatura. Então, é apresentada a relaxação Lagrangeana da formulação de maior destaque, bem como são apresentadas, brevemente, duas famílias de desigualdades válidas para o LSSP. No fim deste capítulo, é apresentada a adaptação da heurística Lagrangeana proposta, para o LSSP.

Analogamente, o Capítulo 5 mostra os resultados para o LSSP. Inicialmente, são discutidas as instâncias consideradas. Então, são apresentados os resultados da comparação entre as formulações consideradas para o LSSP. Na sequência, são mostrados os resultados da adaptação da abordagem Lagrangeana para o LSSP. Finalmente, também é apresentado o impacto nos resultados provenientes da utilização das desigualdades válidas.

---

Enfim, no Capítulo 6 são apresentadas as conclusões derivadas deste trabalho, assim como suas principais contribuições e possibilidades para trabalhos futuros.

## 2 Formulações e Relaxação Lagrangeana para o Problema do Caixeiro Viajante com Coleta de Prêmios

Neste capítulo, o PCTSP é formalmente apresentado e discutido. Inicialmente, o problema é apresentado e os trabalhos mais relevantes que tratam do PCTSP são citados. São apresentadas quatro formulações, nomeadas como: **GSEC**, **MCF**, **OPB** e **EnC**. A formulação clássica **GSEC**, baseada em eliminação de subrotas, foi, inicialmente, proposta para o TSP por Dantzig, Fulkerson e Johnson (1954) e adaptada ao PCTSP por Balas (1989). As demais foram adaptadas, neste trabalho, a partir de formulações propostas, originalmente, para o TSP. **MCF**, baseada no fluxo de multi mercadorias, foi proposta por Claus (1984). **OPB** e **EnC**, foram propostas, respectivamente, por Sherali, Sarin e Tsai (2006) e Godinho, Gouveia e Pesneau (2011). Estas duas são as que obtiveram o melhor desempenho em termos da relaxação linear para o ATSP, de acordo com os trabalhos de Oncan, Altinel e Laporte (2009) e Godinho, Gouveia e Pesneau (2011). Tal fato também foi confirmado pelos testes preliminares. Por isso, as demais formulações citadas nesses trabalhos foram omitidas.

Em seguida, são apresentadas duas novas formulações propostas neste trabalho, nomeadas como: **PSC** e **EPN**. A primeira é baseada no fluxo de mercadoria única, na qual, o fluxo representa o prêmio coletado. A segunda é baseada em restrições que descrevem subrotas mediante variáveis de fluxo e precedência.

Então, é apresentada a relaxação Lagrangeana da formulação de maior destaque, em relação ao valor da relaxação linear, **EPN**; bem como as famílias de desigualdades e cortes consideradas. O algoritmo proposto para resolver o PCTSP é apresentado na sequência.

### 2.1 Definição do problema

O PCTSP pode ser representado por um grafo  $G = (V, A)$ , com um conjunto de vértices,  $V = \{r, 1, 2, \dots, n-1\}$ , com  $n$  vértices, e  $A$  um conjunto de arcos (grafo direcionado). Ele consiste em determinar uma rota com origem no vértice raiz  $r \in V$ , a qual deve visitar um subconjuntos de vértices para coletar um determinado prêmio total. Para cada vértice  $i \in V \setminus \{r\}$  visitado, um prêmio  $p_i$ , não negativo, é coletado. Caso

um vértice não seja visitado, uma penalidade  $w_i$ , também não negativa, é incorrida.

Seja  $c_{ij}$  o custo do arco  $(i, j)$ , o objetivo é determinar uma rota, na qual o somatório dos custos dos arcos utilizados e das penalidades incorridas seja mínimo. Esta rota deve visitar cada vértice, no máximo, uma única vez e coletar um prêmio total maior ou igual ao prêmio mínimo  $g$ .

Este trabalho foca na versão Geral Assimétrica do PCTSP (*General Asymmetric Version*), ou seja, o custo  $c_{ij}$  do percurso entre os vértices  $i$  e  $j$  é diferente do percurso entre os vértices  $j$  e  $i$  ( $c_{ij} \neq c_{ji}$ ). A versão geral não presume a desigualdade triangular. Uma instância possui a propriedade da desigualdade triangular quando  $c_{ij} \leq c_{ik} + c_{kj}$  para todo  $i, j, k \in V$ . Isto porque, como observado em Cirasella et al. (2001), esta versão possui maior potencial de ser encontrado nos problemas do mundo real.

Além disso, destaca-se que no trabalho de Balas e Martin (1985), no qual o PCTSP foi introduzido, não havia especificação do nó raiz. Neste caso, qualquer vértice visitado pode ser considerado como origem da rota. Porém, na literatura, provavelmente a partir de Fischetti e Toth (1988), passou-se a estudar e abordar o problema com um nó raiz, o qual é fixo na rota e representa um depósito. Outros trabalhos como Fischetti e Toth (1988), Bienstock et al. (1993), Dell'Amico, Maffioli e Varbrand (1995) e Gomes, Diniz e Martinhon (2000) também consideram o vértice raiz.

## 2.2 Revisão da literatura

O PCTSP foi, inicialmente, estudado por Balas e Martin (1985) como um modelo para a programação da produção diária de uma fábrica de laminação de chapas de aço. Devido às características do processo de laminação, em especial, o desgaste dos rolos, a programação das ordens de produção é um ponto crítico do processo. Tal programação consiste na escolha de um subconjunto de chapas de aço associadas às suas ordens de execução. Este subconjunto deve satisfazer o limite inferior de peso e a produção precisa ser ordenada numa sequência que minimiza o tempo gasto no processo. Como a escolha do subconjunto de chapas limita as opções disponíveis para o seu sequenciamento, tais decisões necessitam ser resolvidas em conjunto. Este problema foi então chamado de *Prize Collecting Traveling Salesman Problem* e serviu como base para o desenvolvimento de um algoritmo que combina várias heurísticas para sua resolução.

Fischetti e Toth (1988) apresentam formulações matemáticas diferentes para o PCTSP. Eles exploram estas formulações por meio de diferentes relaxações, como a Relaxação Lagrangeana, para encontrar um LB mais forte com um procedimento aditivo. Este procedimento é testado em instâncias de pequenas dimensões. O mé-

todo mostrou-se eficaz para encontrar um LB melhor do que utilizar cada uma das formulações individualmente.

Balas (1989) estuda algumas propriedades estruturais do politopo do PCTSP. Neste trabalho ele identifica algumas famílias de facetas relacionadas ao politopo do TSP e do Problema da Mochila (*Knapsack Problem* – KP).

Dell’Amico, Maffioli e Varbrand (1995) resumem os principais resultados apresentados na literatura, até aquela data. Então novos LB’s são apresentados para a versão assimétrica do PCTSP e do PTP (*Profitable Tour Problem*). A partir de procedimentos retirados de outros trabalhos, os autores são capazes de resolver instâncias de grande porte do PTP assimétrico até a otimalidade. Entretanto, para o PCTSP só são apresentados os resultados para o LB.

A partir dos resultados do trabalho anterior, Dell’Amico, Maffioli e Sciomachen (1998) exploram uma Relaxação Lagrangeana e consideram a penalidade  $w_i = 0$  para todo  $i \in V$ , que resulta no PTP assimétrico como subproblema. Para testar o procedimento proposto, são adotadas instâncias aleatórias e do mundo real. De acordo com os autores, os resultados obtidos são satisfatórios.

Lopez, Carter e Gendreau (1998), assim como Balas e Martin (1985), modelam a programação da produção de uma laminação por meio de uma generalização do PCTSP. Neste trabalho é utilizada uma heurística, baseada no procedimento Busca Tabu, e em uma nova ideia chamada de Canibalização (*Cannibalization*). Os resultados computacionais sobre os dados da produção de uma empresa real indicam que o método proposto é capaz de melhorar significativamente a programação da empresa.

Os trabalhos de Gomes, Diniz e Martinhon (2000) e Melo e Martinhon (2004) comparam algumas versões de uma heurística baseada no método “GRASP”, proposto por Feo e Resende (1995), combinada com os métodos “VNS” e “VND”, ambos propostos por Mladenovic e Hansen (1997). Ambos utilizam instâncias da TSPLIB com os prêmios e penalidades gerados aleatoriamente. No primeiro, a versão com filtro apresentou o melhor resultado. No segundo, o conceito de distribuição de soluções progressivas (GRASP-Progressivo) como fase de construção é introduzido. De acordo com os autores, esta versão apresentou resultados médios promissores.

Bar-Yehuda, Even e Shahar (2005) abordam a versão do PCTSP com janelas de tempo (*time windows* – TW-PCTSP). Eles apresentam um  $O(\log n)$ -algoritmo aproximativo, no qual  $n$  denota o número de vértices, e um algoritmo de programação dinâmica. Ambos os algoritmos podem ser estendidos a outras versões do problema.

Chaves et al. (2007) apresentam uma nova formulação baseada em fluxo de mercadoria única. Além disso, eles também utilizam o GRASP com filtro, combinado com o VNS, proposto no trabalho de Gomes, Diniz e Martinhon (2000). Para testar

o algoritmo, os autores utilizaram instâncias geradas aleatoriamente. Os resultados mostram que foi possível encontrar o ótimo, para as instâncias nas quais ele era conhecido. Segundo os autores, o GRASP proposto também mostrou-se robusto. Ou seja, partindo de diferentes soluções iniciais, chegou-se a soluções finais que diferem da melhor solução encontrada com um pequeno desvio.

Ausiello, Bonifaci e Laura (2008) tratam da versão *on-line* do PCTSP. Nesta versão, os vértices são revelados ao longo do tempo. Os autores mostram como a abordagem proposta pode ser combinada com um algoritmo de aproximação, a fim de obter um algoritmo polinomial. Entretanto, os autores não apresentam resultados computacionais.

Tang e Wang (2008) adotam uma heurística baseada no método ILS combinada com as estratégias *guided kick* e *dynasearch*. Os resultados experimentais, em instâncias geradas aleatoriamente, mostram que a heurística proposta supera o algoritmo baseado no método Busca Tabu. Os resultados, das instâncias coletadas de um processo produtivo real, também mostram que o algoritmo proposto é mais eficaz, e eficiente, do que o método de programação atualmente adotado.

No trabalho de Bérubé, Gendreau e Potvin (2009) é apresentado um algoritmo *branch and cut* para resolver o PCTSP. Os autores exploraram alguns resultados clássicos, do politopo do TSP e do KP, para encontrar desigualdades válidas. Eles também derivaram algumas desigualdades válidas para o OP. São apresentados os resultados computacionais para instâncias simétricas com mais de 500 vértices.

Goemans (2009) apresentam um 1,91457-algoritmo aproximativo para o PCTSP. Este é desenvolvido a partir da combinação dos trabalhos de Bienstock et al. (1993) e de Goemans e Williamson (1995).

Pedro, Saldanha e Camargo (2013) tratam do PCTSP por meio de uma heurística híbrida. Eles combinam os procedimentos Busca Tabu e GENIUS (proposto por Gendreau, Hertz e Laporte (1992)), com as buscas locais 3-opt (proposta por Lin (1965)), e 2-opt (proposta por Croes (1958)). Para testar o algoritmo foram utilizadas as mesmas instâncias de Chaves et al. (2007). Para as instâncias mais fáceis, ele foi capaz de encontrar o ótimo e, para as instâncias mais complexas, ele encontrou soluções próximas do ótimo. Os resultados demonstram que o algoritmo proposto possui desempenho superior ao CPLEX e ao algoritmo proposto por Chaves et al. (2007) (GRASP com VNS). Ele é capaz de encontrar o ótimo para as instâncias com até 60 vértices. Ainda, ao considerar a complexidade do PCTSP, os autores consideram os tempos computacionais razoáveis.

Recentemente, Chan, Jiang e Jiang (2020) apresentaram um algoritmo aproximativo para o PCTSP, baseado nos trabalhos de Talwar (2004) e Bartal, Gottlieb e

Krauthgamer (2016). Os autores também desenvolveram novas formas de aplicar a estratégia dividir-para-conquistar e na decomposição de instâncias esparsas.

Gimadi e Tsidulko (2020) também apresentam algoritmos aproximativos, polinomiais, para o PCTSP e o QTSP. Ainda, eles estabelecem condições sob as quais os algoritmos são assintoticamente ótimos para instâncias aleatórias. Ou seja, condições nas quais os algoritmos tem uma complexidade que é igual ao número mínimo de operações para resolvê-lo.

Outros problemas correlatos ao PCTSP são o *Orienteering Problem* e o *Selective TSP*. O primeiro foi originalmente estudado por Vohra, Levy e Golden (1987) e Liu, Wang e Golden (1988) e o segundo estudado por Laporte e Martello (1990). Ambos os problemas consistem na maximização do prêmio coletado sujeito a um limite no custo da rota. O *Profitable Tour Problem* (PTP), estudado por Dell'Amico, Maffioli e Varbrand (1995), difere do PCTSP por não considerar a restrição de prêmio mínimo a ser coletado. Note que o PTP permite que uma rota vazia seja uma solução viável para o problema. Assim, a solução ótima pode ser uma rota vazia e o valor da função objetivo é igual ao somatório da penalidade  $w_i$  de todos os vértices  $i \in V$ ; ou uma rota incluindo o vértice raiz  $r$ . O QTSP também pode ser visto como uma variação do PCTSP, fazendo  $w_i = 0$ , para todo  $i \in V$ . O *Resource Constrained Travelling Salesman Problem* (RCTSP), proposto por Pekny e Miller (1990), busca uma rota de deslocamento que não consuma mais que uma certa quantidade pré-definida de recurso. O objetivo consiste em encontrar uma rota com custo mínimo.

## 2.3 Formulações

A primeira formulação para o PCTSP é atribuída a Balas e Martin (1985). Sejam as variáveis de decisão:

- $x_{ij} = 1$  se o arco  $(i, j)$  é utilizado na rota e 0 caso contrário;
- $y_i = 1$  se vértice  $i$  é visitado e 0 caso contrário.

O modelo, proposto por Balas e Martin (1985) é representado por (2.1)-(2.8):



$$\min \sum_{i \in V} \sum_{j \in V \setminus \{i\}} c_{ij} x_{ij} + \sum_{i \in V} w_i (1 - y_i) \quad (2.1)$$

sujeito a:

$$\sum_{j \in V \setminus \{i\}} x_{ij} - y_i = 0 \quad \forall i \in V \quad (2.2)$$

$$\sum_{i \in V \setminus \{j\}} x_{ij} - y_j = 0 \quad \forall j \in V \quad (2.3)$$

$$y_r = 1 \quad (2.4)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (2.5)$$

$$y_i \in \{0, 1\} \quad \forall i \in V \setminus \{r\} \quad (2.6)$$

$$\sum_{i \in V} p_i y_i \geq g \quad (2.7)$$

$$\text{Não existem subrotas desconexas da raiz } r \quad (2.8)$$

As restrições de designação, (2.2) e (2.3), determinam que se o vértice  $i$  é visitado, ele será visitado, no máximo, uma única vez. A restrição (2.4) garante que o vértice raiz  $r$  será utilizado. A restrição (2.7) assegura que o prêmio mínimo seja coletado. Seja  $b = \sum_{i \in V} p_i - g$ , a restrição (2.7) pode ser reescrita como (2.9):

$$\sum_{i \in V} p_i (1 - y_i) \leq b \quad (2.9)$$

Note que neste caso, esta restrição pode ser vista como a restrição de capacidade do KP. As restrições (2.5) e (2.6) definem o domínio das variáveis de decisão. A restrição (2.8) determina que não existam subrotas desconexas da raiz  $r$ .

Um dos principais desafios ao criar uma formulação para o PCTSP, e problemas correlatos, está na forma como representar as restrições que eliminam as subrotas (restrição (2.8)). Estas restrições são conhecidas como SEC's (*Subtour Elimination Constraints*).

Diversas formulações com este objetivo foram propostas para o TSP, o qual é um dos problemas de OC mais extensivamente estudado. Como já foi citado anteriormente no Capítulo 1, o PCTSP é uma generalização do TSP. Logo, a literatura relacionada a este problema, em especial, a versão assimétrica (ATSP), pode ser usada como uma importante fonte de ideias para desenvolver formulações e técnicas de resolução melhores para o PCTSP.

Provavelmente, a SEC mais conhecida foi proposta por Dantzig, Fulkerson e Johnson (1954) para o TSP (Restrições (2.10)):

$$\sum_{i, j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset V : r \in V \setminus S \quad (2.10)$$

De acordo com Dantzig, Fulkerson e Johnson (1954) e Balas (1995), estas restrições definem facetas para o TSP e, apesar de possuírem um número finito, este é exponencial.

Dado que estas restrições são da ordem  $O(2^n)$ , é necessário um algoritmo de separação (*Separation Algorithm*), que identifique as restrições violadas. Dessa forma, elas podem ser gradualmente acrescentadas ao modelo à medida que são identificadas.

Entretanto, é possível eliminar as subrotas utilizando um número polinomial de restrições. Neste caso, não é necessário um algoritmo de separação. A primeira formulação, com um número polinomial de restrições, para o ATSP foi proposta por Miller, Tucker e Zemlin (1960). Ela é conhecida como MTZ. A formulação MTZ é relativamente simples e flexível. Logo, ela pode ser facilmente aplicada a outros problemas que envolvam subrotas. Por isso, Desrochers e Laporte (1991) propõem uma extensão mais forte que MTZ (para o ATSP), denominada DL. Sherali e Driscoll (2002) também se aproveitam das características de MTZ. Eles apresentam novas SEC's mais fortes que DL (considerando o ATSP), conhecidas como SD. Uma formulação A é dita mais forte que uma outra formulação B se o poliedro associado à formulação A está contido no poliedro associado à formulação B, ver, por exemplo, Pochet e Wolsey (2010) para uma definição formal. Como consequência, para um problema de minimização, o valor da relaxação linear fornecido por uma formulação A mais forte que uma outra formulação B é sempre maior ou igual ao valor da relaxação linear fornecido pela formulação B.

Gavish e Graves (1978) adotam outra forma polinomial de representar as SEC's para o ATSP, chamada de GG. Neste caso, as restrições descrevem o fluxo de uma única mercadoria (*commodity*) que percorre os vértices do problema. Por isso, de acordo com Langevin, Soumis e Desrosiers (1990), esta classe de formulações recebem o nome de Fluxo de uma Única Mercadoria (*Single Commodity Flow Based Formulations – SCF*).

Fox, Gavish e Graves (1980) e Gouveia e Pires (2001) também apresentam novas formulações válidas para o ATSP, respectivamente conhecidas como FGG e GP. FGG aproveita-se da relação entre o ATSP e o Problema de Programação de Máquinas (*Machine Scheduling Problem*). GP são generalizações de MTZ por meio de restrições que descrevem a ordem de precedência entre os vértices.

### 2.3.1 Formulação GSEC

Segundo Fischetti e Toth (1988), devido às restrições (2.2) e (2.3), a restrição (2.10), pode ser reescrita como (2.11) (conhecida como *Connectivity Constraint*), a

qual é válida para o PCTSP.

$$\sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \geq y_k \quad \forall S \subset V : r \in S, k \in V \setminus S \quad (2.11)$$

Segundo Bérubé, Gendreau e Potvin (2009), a restrição (2.11) pode ser generalizada e reescrita segundo a restrição (2.12), conhecida como *Generalized Subtour Elimination Constraints* – GSECs.

$$\sum_{i \in S} \sum_{j \in S \setminus \{i\}} x_{ij} \leq \sum_{i \in S \setminus \{k\}} y_i \quad \forall S \subset V, 2 \leq |S| \leq |V| - 1, r \in V \setminus S, \forall k \in S \quad (2.12)$$

Considere a solução  $(\bar{x}, \bar{y}) \in \{0, 1\}^{|A| \times |V|}$  que satisfaça as restrições (2.2)-(2.7). Caso esta solução viole a restrição (2.12), ela pode ser encontrada por meio de um algoritmo de separação que consiste na resolução de  $|\bar{V}|$  problemas de fluxo máximo, no qual  $\bar{V} = \{v \in V \setminus \{r\} : \bar{y} > 0\}$ . Para cada  $v \in \bar{V}$ , o fluxo máximo é calculado entre  $r$  e  $v$ , no grafo formado pelos subconjuntos de vértices  $\bar{V}$  e de arcos  $\bar{A} = \{(i, j) \in A : \bar{x}_{ij} > 0\}$ , no qual  $\bar{x}_{ij}$  é a capacidade do arco  $(i, j) \in \bar{A}$ . O subconjunto  $S \subset V \setminus \{r\}$  no corte de capacidade mínima ( $\delta(S)$ ), que contém o vértice  $v$ , fornece uma restrição de subrota potencialmente violada, se a condição  $\delta(S) \geq 2$  não for satisfeita.

A formulação **GSEC** é dada por (2.1)-(2.7) e (2.12).

### 2.3.2 Formulação **MCF**

As formulações baseadas no fluxo de múltiplas mercadorias podem ser facilmente adaptadas do ATSP para o PCTSP. Em seu trabalho, Wong (1980) apresenta uma formulação (WONG), para o ATSP. Esta formulação utiliza a variável  $f_{ij}^{kl}$  que indica o fluxo da mercadoria com origem no vértice  $k$  e destino no vértice  $l$  que atravessa o arco  $(i, j)$ . Ou seja,  $f_{ij}^{rl}$  é igual a 1 apenas se a mercadoria sai do vértice  $r$  e chega no vértice  $l$  que passa pelo arco  $(i, j)$ . Analogamente,  $f_{ij}^{kr}$  é igual a 1 apenas se a mercadoria sai do vértice  $k$  e chega ao vértice  $r$  por meio do arco  $(i, j)$ . Repare que só existe fluxo de mercadorias entre dois vértices, desde que pelo menos um deles seja o vértice  $r$ .

Claus (1984) obtém uma formulação mais compacta para o ATSP. Ele adota a variável  $f_{ij}^k$  que descreve o fluxo de  $|V| - 1$  mercadorias através dos arcos  $(i, j)$ , com origem em cada vértice  $k$  (para cada mercadoria), com destino no vértice raiz  $r$ . Dessa forma, ele consegue eliminar metade das variáveis de mercadorias e suas respectivas restrições, se comparado com WONG.

A partir da formulação CLAUS é fácil derivar uma nova formulação para o PCTSP, nomeada neste trabalho como **MCF** (*Multicommodity flow formulation*).

Basta adotar a variável de fluxo  $f_{ij}^k$  e multiplicar o escalar do lado direito das restrições (*Right hand side – RHS*), por  $y_k$  para  $k \in V \setminus \{r\}$  e considerar o vértice origem fixo e igual à  $r$ .

$$\sum_{j \in V \setminus \{i\}} f_{ij}^k - \sum_{j \in V \setminus \{i\}} f_{ji}^k = \begin{cases} y_k & \text{se } k = i \\ 0 & \text{caso contrário} \end{cases} \quad \forall k, i \in V \setminus \{r\} \quad (2.13)$$

$$f_{ij}^k \leq x_{ij} \quad \forall (i, j) \in A, \forall k \in V \setminus \{r\} \quad (2.14)$$

$$f_{ij}^k \geq 0 \quad \forall (i, j) \in A, \forall k \in V \setminus \{r\} \quad (2.15)$$

Para **MCF**, a variável  $f_{ij}^k$  também pode ser interpretada como o caminho entre o vértice  $k$  e o vértice raiz  $r$ . Ou seja,  $f_{ij}^k = 1$  caso o arco  $(i, j) \in A$  esteja no percurso entre os vértices  $k$  e  $r$ , e zero, caso contrário. Um exemplo para uma solução é representado pela Figura 2.

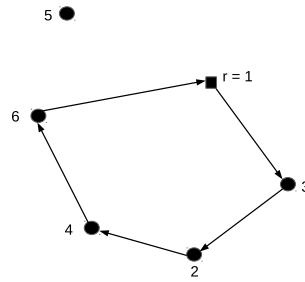


Figura 2 – Ilustração de uma solução para a formulação **MCF** com o vértice raiz  $r = 1$  e  $|V| = 6$ .

Para esta solução temos o vetor  $y = \{1, 1, 1, 1, 0, 1\}$ ,  $x_{1,3} = 1$ ,  $x_{3,2} = 1$ ,  $x_{2,4} = 1$ ,  $x_{4,6} = 1$ ,  $x_{6,1} = 1$ ,  $f_{3,2}^3 = 1$ ,  $f_{2,4}^3 = 1$ ,  $f_{4,6}^3 = 1$ ,  $f_{6,1}^3 = 1$ ,  $f_{2,4}^2 = 1$ ,  $f_{4,6}^2 = 1$ ,  $f_{6,1}^2 = 1$ ,  $f_{4,6}^4 = 1$ ,  $f_{6,1}^4 = 1$  e  $f_{6,1}^6 = 1$ , e as demais variáveis iguais a zero.

A formulação **MCF** é dada por (2.1)-(2.7), (2.13)-(2.15).

### 2.3.3 Formulação **OPB**

Sherali, Sarin e Tsai (2006) apresentam diversas formulações de tamanho polinomial para o ATSP. Eles as classificam como Formulações de Caminho e Fluxo *Lifted Path and Flow Formulations – PFF*. Os autores utilizam a técnica conhecida como *Reformulation-Linearization Technique (RLT)*, proposta por Sherali e Adams (1990). combinada com as restrições de precedência por meio das variáveis  $t_{ij}^k$  e  $v_{ij}$ . A variável  $t_{ij}^k$  é o artifício utilizado para manter a formulação linear (RLT) por meio do produto entre  $x_{ik}v_{kj}$ . Ela indica se o arco  $(i, k)$  faz parte do percurso e o vértice  $k$  precede o vértice  $j$ . Ou seja,  $t_{ij}^k = x_{ik}v_{kj}$  para todo  $i, j, k \in V \setminus \{r\}$  tal que  $i \neq j \neq v$ .

A variável de precedência  $v_{ij}$ ,  $i, j \in V \setminus \{r\}$ ,  $i \neq j$ , é contínua, porém possui uma conotação binária. Ou seja, ela assume o valor igual a 1 se o vértice  $i$  precede o vértice  $j$ . Esta relação de precedência não precisa ser direta. Isto significa que o vértice  $i$  pode estar em qualquer posição entre o vértice raiz  $r$  e o vértice  $j$ .

Neste trabalho, adaptamos para o PCTSP a formulação que forneceu o melhor resultado para a relaxação linear entre as propostas, nomeada como ATSP6 naquele artigo. Como no contexto do PCSTP, um vértice não faz necessariamente parte da rota, essa adaptação não é direta. Note que as variáveis  $v_{ij}$  e  $t_{ij}^k$  só podem assumir 1 se os vértices  $i$ ,  $j$  e  $k$  forem todos selecionados para fazer parte da rota. Este fato leva a modificações significativas na adaptação da formulação ATSP6 ao PCTSP, restrições (2.16)-(2.24).

$$v_{ij} + v_{ji} \leq y_i \quad \forall i, j \in V \setminus \{r\}, i \neq j \quad (2.16)$$

$$v_{ij} + 1 - y_j \geq x_{ri} \quad \forall i, j \in V \setminus \{r\}, i \neq j \quad (2.17)$$

$$v_{ji} + 1 - y_j \geq x_{ir} \quad \forall i, j \in V \setminus \{r\}, i \neq j \quad (2.18)$$

$$v_{ij} + x_{ji} + v_{jk} + v_{ki} \leq 2 \quad \forall i, j, k \in V \setminus \{r\}, i \neq j \neq k \quad (2.19)$$

$$t_{ij}^k \leq x_{ik} \quad \forall i, j, k \in V \setminus \{r\}, k \neq j \neq i \quad (2.20)$$

$$\sum_{k \in V \setminus \{r, i, j\}} t_{ij}^k + x_{ij} = v_{ij} \quad \forall i, j \in V \setminus \{r\}, i \neq j \quad (2.21)$$

$$\sum_{i \in V \setminus \{r, k, j\}} t_{ij}^k + x_{rk} + y_j - 1 \leq v_{kj} \quad \forall k, j \in V \setminus \{r\}, k \neq j \quad (2.22)$$

$$\sum_{i \in V \setminus \{r, k, j\}} t_{ij}^k + x_{rk} - y_j + 1 \geq v_{kj} \quad \forall k, j \in V \setminus \{r\}, k \neq j \quad (2.23)$$

$$t_{ij}^k \geq 0 \quad \forall i, j, k \in V \setminus \{r\}, k \neq j \neq i \quad (2.24)$$

As restrições (2.16) asseguram que, dado os vértices  $i$  e  $j$ ,  $i$  precede  $j$ , ou  $j$  precede  $i$ , apenas se ambos os vértices fizerem parte da subrota. Se um vértice  $i$  não for selecionado para fazer parte da mesma, então  $v_{ij} = v_{ji} = 0$ ,  $\forall j \in V \setminus \{r, i\}$ . (No contexto do ATSP, podemos impor que  $i$  precede  $j$ , ou vice-versa, para cada par de vértices, ou seja,  $v_{ij} + v_{ji} = 1$ ,  $\forall i, j \in V \setminus \{r\}, i \neq j$ .) Restrições (2.17) (resp. (2.18)) garantem que o primeiro (resp. último) vértice após (resp. antes) a raiz precede (resp. é precedido por) todos os outros vértices na subrota. O termo  $1 - y_j$  torna as restrições (2.17) e (2.18) satisfeitas se o vértice  $j$  não for visitado. Restrições (2.19) são uma versão melhorada da restrição de eliminação de subrotas de três vértices  $v_{ij} + v_{jk} + v_{ki} \leq 2$ , para mais detalhes consulte Sarin, Sherali e Bhootra (2005).

Sejam as restrições (2.20), (2.21) e (2.25) derivadas da aplicação da RLT de

primeira ordem por Sherali, Sarin e Tsai (2006):

$$\sum_{i \in V \setminus \{r, k, j\}} t_{ij}^k + x_{rk} = v_{kj} \quad \forall k, j \in V \setminus \{r\}, i \neq j \quad (2.25)$$

No contexto do PCTSP as restrições (2.25) devem ser modificadas, uma vez que não são satisfeitas para pares de vértices tais que  $k$  é o primeiro vértice após a raiz ( $x_{rk} = 1$ ) e  $j$  não é visitado ( $v_{kj} = 0$ ). No ATSP se  $x_{rk} = 1$  temos necessariamente  $v_{kj} = 1, \forall j \in V \setminus \{r, k\}$ . Portanto, as restrições (2.22) e (2.23) substituem (2.25) na formulação do PCTSP. Ambas as restrições são satisfeitas se  $v_{kj} = 0$ ; se  $y_j = 0$  e  $x_{rk} = 1$ , as restrições (2.20) implicam  $t_{ij}^k = 0, \forall i \in V \setminus \{r, k, j\}$ ; se  $y_j = 1$  então  $x_{rk} = t_{ij}^k = 0, \forall i \in V \setminus \{r, k, j\}$ , já que  $j$  precede  $k$  na subrota ou  $k$  não é visitado. Caso contrário,  $v_{kj} = 1$  implica  $y_j = 1$ , e (2.22) com (2.23) gera (2.25).

Embora os experimentos computacionais relatados Sherali, Sarin e Tsai (2006) tenham mostrado que a ATSP6 é uma formulação forte para o ATSP, resultados computacionais preliminares mostraram que a formulação **OPB** possui uma relaxação linear fraca para o PCTSP devido à substituição de (2.25) por (2.22) e (2.23). Ou seja, a relaxação linear possui um grande desvio em relação à solução ótima. Assim, propõe-se uma adaptação alternativa do ATSP6 ao PCTSP que preserva as restrições (2.25). Para tanto, remove-se o termo  $1 - y_j$  das restrições (2.17) e faz-se  $v_{kj} = 1$  para cada vértice  $j$ , mesmo se  $j$  não for visitado, dado que  $k$  é o primeiro vértice após a raiz (restrições (2.27) abaixo). Dessa forma, dado um vértice  $k$  tal que  $x_{rk} = 1$ , as restrições (2.25) são satisfeitas mesmo se  $y_j = 0$  tal que  $v_{kj} = 1, \forall j \in V \setminus \{r, k\}$ . As restrições (2.26) substituem (2.16) para permitir  $v_{kj} = 1$  se  $y_j = 0$ , e as restrições (2.28) correspondem a (2.21) escritas como desigualdades a serem satisfeitas quando  $j$  não fizer parte da rota.

$$v_{ij} + v_{ji} \leq 1 \quad \forall i, j \in V \setminus \{r\}, i \neq j \quad (2.26)$$

$$v_{ij} \geq x_{ri} \quad \forall i, j \in V \setminus \{r\}, i \neq j \quad (2.27)$$

$$\sum_{k \in V \setminus \{r, i, j\}} t_{ij}^k + x_{ij} \leq v_{ij} \quad \forall i, j \in V \setminus \{r\}, i \neq j \quad (2.28)$$

Assim, a formulação **OPB** (*Ordered path-based*), é formada por (2.1)-(2.7), (2.18)-(2.20), (2.24), (2.25)-(2.28).

### 2.3.4 Formulação **EnC**

Uma outra forma sofisticada de representar as SEC's foi proposta por Godinho, Gouveia e Pesneau (2011), por meio de uma formulação de circuito melhorada, nomeada como EC-MCF++. A formulação explora a ideia de Godinho, Gouveia e Magnanti (2008) na qual cada vértice  $k$  pertence a um circuito. Esta ideia, desenvolvida em Godinho, Gouveia e Pesneau (2010), modela o ATSP com um grafo de duas

camadas com índice de saltos (*hop-indexed*). A primeira camada do grafo representa o caminho do vértice raiz  $r$  até o vértice  $k$ . A segunda camada representa o caminho de retorno, ou seja, o caminho entre o vértice  $k$  e o vértice raiz  $r$ .

A variável binária  $z_{ij}^h$  assume 1 se  $(i, j)$  é o arco na posição  $h$  do circuito que começa no vértice raiz  $r$ . Para cada vértice  $k$ , as variáveis binárias  $z_{ij}^{hk}$  e  $z_{ij}^{kh}$  indicam se o arco na posição  $h$  do circuito está ligado à primeira ou à segunda camada, ou seja,  $z_{ij}^{hk} = 1$  (resp.  $z_{ij}^{kh} = 1$ ) se  $(i, j)$  faz parte de o caminho da raiz para  $k$  (resp. de  $k$  para a raiz). A formulação do circuito é aprimorada com o uso de variáveis de precedência  $v_{ij}$ ,  $i, j \in V \setminus \{r\}, i \neq j$ .

Sejam as restrições (2.19) e (2.29)-(2.39):

$$\sum_{j \in V \setminus \{r\}} z_{rj}^{1k} = y_k \quad \forall k \in V \setminus \{r\} \quad (2.29)$$

$$\sum_{j \in V \setminus \{r, i\}} z_{ij}^{h+1, k} - \sum_{j \in V \setminus \{k, i\}} z_{ji}^{hk} = 0 \quad \forall k, i \in V \setminus \{r\}, k \neq i, h = 1, \dots, n-2 \quad (2.30)$$

$$\sum_{j \in V \setminus \{k\}} z_{kj}^{h+1, k} - \sum_{j \in V \setminus \{k\}} z_{jk}^{hk} = 0 \quad \forall k \in V \setminus \{r\}, h = 1, \dots, n-1 \quad (2.31)$$

$$\sum_{j \in V \setminus \{k, i\}} z_{ij}^{h+1, k} - \sum_{j \in V \setminus \{r, i\}} z_{ji}^{hk} = 0 \quad \forall k, i \in V \setminus \{r\}, k \neq i, h = 2, \dots, n-1 \quad (2.32)$$

$$\sum_{h=1}^{n-2} \sum_{i \in V \setminus \{k, j\}} z_{ij}^{hk} = v_{jk} \quad \forall j, k \in V \setminus \{r\}, j \neq k \quad (2.33)$$

$$z_{ij}^{hk} + z_{ij}^{kh} = z_{ij}^h \quad \forall k \in V \setminus \{r\}, \forall (i, j) \in A, h = 1, \dots, n \quad (2.34)$$

$$\sum_{h=1}^n z_{ij}^h = x_{ij} \quad \forall (i, j) \in A \quad (2.35)$$

$$v_{ij} + v_{ji} \leq y_j \quad i, j \in V \setminus \{r\}, j \neq i \quad (2.36)$$

$$z_{ij}^{hk}, z_{ij}^{kh} \in \{0, 1\} \quad \forall (i, j) \in A, \forall k \in V \setminus \{r\}, h = 1, \dots, n \quad (2.37)$$

$$z_{ij}^h \in \{0, 1\} \quad \forall (i, j) \in A, h = 1, \dots, n \quad (2.38)$$

$$v_{ij} \in \{0, 1\} \quad \forall i, j, k \in V \setminus \{r\}, k \neq j \neq i \quad (2.39)$$

A formulação EC-MCF++ pode ser facilmente adaptada ao PCTSP, basta garantir que haja um circuito que passa pelo vértice  $k$ , se e somente se,  $k$  fizer parte da rota, restrições (2.29). Assim, dado  $i, j \in V \setminus \{r\}, j \neq i$ , exatamente uma das variáveis

$v_{ij}$ , ou  $v_{ji}$ , assume 1 apenas se  $i$  e  $j$  fazem parte da rota. Caso contrário, ambas as variáveis assumem 0, restrições (2.36). As restrições restantes são escritas de acordo com a formulação originalmente proposta por Godinho, Gouveia e Pesneau (2011). As restrições (2.30)-(2.32) constroem os caminhos associados a cada camada do circuito que passa por um vértice visitado  $k$  e utiliza o índice de saltos. As restrições (2.33) vinculam as variáveis de índice de saltos com as variáveis de precedência. As restrições (2.34) garantem que um arco  $(i, j)$  esteja na mesma posição no circuito de todos os vértices  $k$  visitados, tanto na primeira como na segunda camada. As restrições (2.35) vinculam as variáveis de índice de saltos com as variáveis do arco original.

A formulação **EnC** (*Enhanced circuit*), é composta por (2.1)-(2.7), (2.19) e (2.29)-(2.39).

### 2.3.5 Formulação EPN

Esta formulação, proposta neste trabalho, adota as variáveis  $z_{ij}^k, \forall i, j, k \in V \setminus \{r\}, k \neq i$ . Dados três vértices diferentes entre si,  $z_{ij}^k$ , para todo  $i, j, k \in V, i \neq k \neq j$ , assume o valor 1 se, e somente se, todos os três vértices são visitados, e o arco  $(i, j)$  faz parte da rota no caminho do vértice raiz  $r$  até o vértice  $k$ . Ao contrário de  $z_{ij}^k$ , as variáveis  $z_{ik}^k$  e  $z_{ii}^k$ , tal que  $i \neq k$ , são definidas como:

- $z_{ik}^k$  : assume 1 se, e somente se, o vértice  $i$  é o primeiro vértice após a raiz  $r$  na rota, e o vértice  $k$  não é visitado;
- $z_{ii}^k$  : assume 1 se, e somente se, o vértice  $i$  é visitado, mas não é o primeiro vértice após a raiz  $r$  na rota e o vértice  $k$  precede  $i$  ou não é visitado.

A partir desse grupo de variáveis combinadas com as restrições (2.40)-(2.45):

$$\sum_{i \in V \setminus \{r, k\}} z_{ij}^k + x_{rj} = y_j \quad \forall k, j \in V \setminus \{r\}, j \neq k \quad (2.40)$$

$$\sum_{j \in V \setminus \{r\}} z_{ij}^k + x_{ik} = y_i \quad \forall i, k \in V \setminus \{r\}, i \neq k \quad (2.41)$$

$$1 + y_k - y_j \geq z_{ij}^k \quad \forall i, k, j \in V \setminus \{r\}, i \neq k \neq j \quad (2.42)$$

$$z_{ij}^k \leq x_{ij} \quad \forall i, j, k \in V \setminus \{r\}, k \neq j \neq i \quad (2.43)$$

$$x_{ir} + x_{ri} \leq y_i \quad \forall i \in V \setminus \{r\} \quad (2.44)$$

$$z_{ij}^k \in \{0, 1\} \quad \forall i, j, k \in V \setminus \{r\}, k \neq i \quad (2.45)$$

é possível propor uma nova formulação válida para o PCSTP.

Considere, inicialmente, as restrições (2.40), tal que  $j$  é visitado,  $y_j = 1$ , mas  $j$  não é o primeiro vértice após a raiz,  $x_{rj} = 0$ . Tem-se  $z_{jj}^k = 1$  se o vértice  $k$  precede



$j$  na rota ou não é visitado, caso contrário, há um arco  $(i, j)$ ,  $x_{ij} = 1$ , no caminho da raiz até o vértice  $k$  e  $z_{ij}^k = 1$ . Se o vértice  $j$  for visitado e for o primeiro vértice após a raiz,  $x_{rj} = 1$ , tem-se  $z_{jj}^k = 0$  e  $z_{ij}^k = 0, \forall i, k \in V \setminus \{r\}, j \neq k, i \neq k$ . Se o vértice  $j$  não for visitado, ambos os lados serão zero.

De maneira análoga, considere as restrições (2.41) e assuma que o vértice  $i$  é visitado,  $y_i = 1$ , mas não é o vértice subsequente à raiz,  $x_{ri} = 0$ . Assim,  $z_{ii}^k = 1$  se o vértice  $k$  precede  $i$  na subrota ou  $k$  não é visitado. Caso contrário, ou o arco  $(i, k)$  faz parte da solução ou existe um arco  $(i, j)$  no percurso entre o vértice raiz  $r$  e o vértice  $k$ . No último caso, considere que o vértice  $i$  é o primeiro vértice após a raiz,  $x_{ri} = 1$ , e o vértice  $k$  é visitado. Se o vértice  $i$  é o primeiro vértice após a raiz e o vértice  $k$  não é visitado, então temos  $z_{ik}^k = 1$ . Se o vértice  $i$  não for visitado, ambos os lados serão zero.

Restrições (2.42) e (2.43), para  $j \neq i \neq k$ , definem como zero o valor das variáveis  $z_{ij}^k$  quando o vértice  $k$  não é visitado e o vértice  $j$  está ativo. Analogamente, elas garantem que  $z_{ij}^k = 0$  quando o arco  $(i, j)$  não faz parte da solução. Resultados computacionais preliminares mostraram que as restrições (2.44) podem ser úteis para fortalecer o limite de relaxação linear fornecido pela formulação.

Figure 3 ilustra o uso das variáveis  $z_{ij}^k$  com uma solução viável para uma instância com 4 vértices na qual têm-se o vértice raiz  $r = 1$ . Nesta solução, as únicas variáveis  $z_{ij}^k$  que assumem o valor 1 são as seguintes:

- $z_{24}^4 = 1$ , uma vez que o vértice 2 é o primeiro vértice após a raiz e o vértice 4 não é visitado;
- $z_{33}^2 = 1$ , uma vez que o vértice 3 é visitado, mas não é o primeiro após a raiz e o vértice 2 precede 3,
- $z_{33}^4 = 1$ , uma vez que o vértice 3 é visitado, mas não é o primeiro vértice após a raiz e o vértice 4 não é visitado.

A nova formulação proposta, chamada de **EPN** (*Enhanced path-node*), é dada por (2.1)-(2.7), (2.40)-(2.45).

### 2.3.6 Formulação PSC

A formulação de mercadoria única (Formulação GG proposta por Gavish e Graves (1978)), segundo Oncan, Altinel e Laporte (2009), é conhecida por fornecer um limite fraco para a relaxação linear do ATSP. Entretanto, como observado por Guimarães, Klabjan e Almada-Lobo (2014), ela é boa para encontrar soluções viáveis.

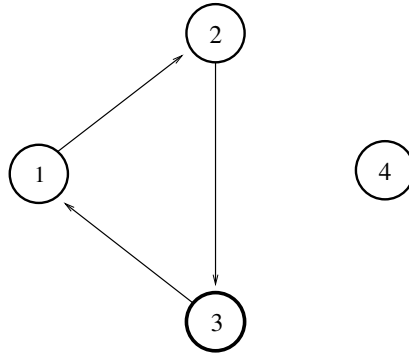


Figura 3 – Ilustração de uma solução para a formulação EPN com o vértice raiz  $r = 1$  e  $|V| = 4$ .

Assim, propomos uma formulação de mercadoria única, na qual o fluxo ao longo da subrota acumula o prêmio total coletado conforme os vértices são visitados. Ou seja, essa formulação é capaz de inibir as soluções  $S \subset V : 1 \in S, \sum_{i \in S} p_i < g$ , e coibir as subrotas desconexas da raiz (Repare que  $S \subseteq V$  é o subconjunto de vértices visitados). Para esta formulação, definimos a variável  $f_{ij}$  como a quantidade de prêmio  $p_i$ , coletada ao longo da rota formada pelos vértices pertencentes à  $S$ , tal que  $i$  e  $r \in S$ . Como a cada vértice visitado, um prêmio é coletado, ao retornar ao vértice raiz  $r$  o prêmio coletado deve ser maior que o prêmio mínimo  $g$ .

$$\sum_{j \in V \setminus \{i\}} (f_{ij} - p_i x_{ij}) - \sum_{j \in V \setminus \{r, i\}} f_{ji} = 0 \quad \forall i \in V \setminus \{r\} \quad (2.46)$$

$$\sum_{i \in V \setminus \{r\}} f_{ir} \geq g \quad (2.47)$$

$$f_{ij} \leq \bar{g} x_{ij} \quad \forall i, j \in V, i \neq r, i \neq j \quad (2.48)$$

$$f_{ij} \geq 0 \quad \forall i, j \in V, i \neq r, i \neq j \quad (2.49)$$

Nas restrições (2.46) o fluxo que sai de um vértice visitado  $i$  é aumentado de acordo com seu prêmio  $p_i$ . A restrição (2.47) garante que o fluxo que chega ao vértice raiz, por meio do arco  $(i, r)$ , é, pelo menos, igual ao prêmio mínimo  $g$ . Dessa forma, ao retornar ao vértice raiz  $r$  o prêmio coletado deve ser maior, ou igual, ao prêmio mínimo. Note que a restrição de prêmio mínimo (2.7) torna-se redundante e pode ser retirada da formulação. As restrições (2.48) vinculam as variáveis de fluxo com as variáveis de arco originais, nas quais  $\bar{g} = \sum_{i \in V \setminus \{r\}} p_i$ .

Um exemplo para uma solução é representado pela fig. 4.

Para esta solução temos o vetor  $y = \{1, 1, 1, 1, 0, 1\}$ ,  $x_{1,3} = 1$ ,  $x_{3,2} = 1$ ,  $x_{2,4} = 1$ ,  $x_{4,6} = 1$ ,  $x_{6,1} = 1$ ,  $f_{1,3} = 4$ ,  $f_{3,2} = 5$ ,  $f_{2,4} = 6$ ,  $f_{4,6} = 8$  e  $f_{6,1} = 8$ , e as demais

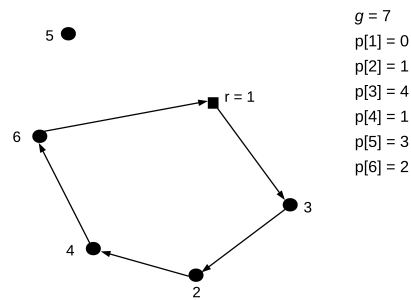


Figura 4 – Ilustração de uma solução para a formulação **PSC** com o vértice raiz  $r = 1$  e  $|V| = 6$

variáveis iguais a zero. Observe que o fluxo que chega ao vértice raiz ( $f_{6,1} = 8$ ) é maior que o prêmio mínimo  $g = 7$ .

Esta nova formulação, nomeada neste trabalho como **PSC** (*Prize-single commodity formulation*), é composta por (2.1)-(2.6) e (2.46)-(2.49).

## 2.4 Relaxação Lagrangeana para o PCTSP

Como será observado na Seção 3.2, a formulação **EPN** (descrita na Seção 2.3.5), apresenta os melhores resultados, em termos da relaxação linear. Na verdade, ela possui não só o melhor *gap* médio, mas também um *gap* melhor ou igual para todas as instâncias. No entanto, a sua relaxação linear é muitas vezes cara em termos computacionais. De fato, para 105 das 252 instâncias testadas, não foi possível resolver a relaxação linear dentro de 3 horas.

Dentro desse cenário, a formulação **EPN** surge como a melhor candidata, entre as formulações estudadas, para a técnica da relaxação Lagrangeana. Logo, ela foi escolhida para ser explorada por meio dessa abordagem. Como pode ser observado a seguir na Seção 3.3, Cap. 3, a relaxação Lagrangeana, proposta a partir da formulação **EPN**, pode fornecer LB's mais fortes do que a relaxação linear de **EPN**, visto que não possui a propriedade de integralidade.

Dessa forma, nesta seção é mostrado com detalhes a relaxação Lagrangeana da formulação **EPN** para o PCTSP, bem como o problema dual Lagrangeano decorrente da dualização das restrições complicantes. Também são apresentados os subproblemas Lagrangeanos resultantes e os respectivos procedimentos de resolução. Em seguida, são apresentados os procedimentos solução inicial, busca local e as famílias de cortes de otimalidade e desigualdades válidas; utilizados no Algoritmo 4 apresentado na Seção 2.5, o qual é utilizado para resolver o PCTSP.

A Tabela 1 mostra os multiplicadores de Lagrange e suas respectivas restrições complicantes da formulação **EPN**. As restrições (2.4), (2.5), (2.6), (2.7) e (2.45) são

deixadas para os subproblemas Lagrangeanos. Como **EPN** é uma formulação válida sem (2.42), esta restrição é desconsiderada na relaxação Lagrangeana.

**Tabela 1 – Associação dos multiplicadores de Lagrange com as restrições da EPN.**

Multiplicadores	Restrições
$\pi_i \in \mathbb{R}, \forall i \in V$	(2.2)
$\varpi_j \in \mathbb{R}, \forall j \in V$	(2.3)
$\lambda_{jk} \in \mathbb{R}, \forall j, k \in V \setminus \{r\}, j \neq k$	(2.40)
$\mu_{ik} \in \mathbb{R}, \forall i, k \in V \setminus \{r\}, i \neq k$	(2.41)
$\nu_{ij}^k \in \mathbb{R}_+, \forall i, j, k \in V \setminus \{r\}, i \neq j \neq k$	(2.43)
$\rho_i \in \mathbb{R}_+, \forall i \in V \setminus \{r\}$	(2.44)

Repare que  $W = \sum_{i \in V} w_i$  é uma constante e o vértice raiz  $r$  é sempre igual a um (Restrição (2.4)). Logo, a constante  $W$  e a parcela  $(\pi_r + \varpi_r - w_r)$  podem ser omitidas e consideradas apenas no final. A partir da formulação **EPN**, repetida aqui para maior comodidade do leitor:

$$\min \sum_{i \in V} \sum_{j \in V \setminus \{i\}} c_{ij} x_{ij} + \sum_{i \in V} w_i (1 - y_i) \quad (2.1)$$

sujeito a:

$$\sum_{j \in V \setminus \{i\}} x_{ij} - y_i = 0 \quad \forall i \in V \quad (2.2)$$

$$\sum_{i \in V \setminus \{j\}} x_{ij} - y_j = 0 \quad \forall j \in V \quad (2.3)$$

$$y_r = 1 \quad (2.4)$$

$$\sum_{i \in V} p_i y_i \geq g \quad (2.7)$$

$$\sum_{i \in V \setminus \{r, k\}} z_{ij}^k + x_{rj} = y_j \quad \forall k, j \in V \setminus \{r\}, j \neq k \quad (2.40)$$

$$\sum_{j \in V \setminus \{r\}} z_{ij}^k + x_{ik} = y_i \quad \forall i, k \in V \setminus \{r\}, i \neq k \quad (2.41)$$

$$z_{ij}^k \leq x_{ij} \quad \forall i, j, k \in V \setminus \{r\}, k \neq j \neq i \quad (2.43)$$

$$x_{ir} + x_{ri} \leq y_i \quad \forall i \in V \setminus \{r\} \quad (2.44)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (2.5)$$

$$y_i \in \{0, 1\} \quad \forall i \in V \setminus \{r\} \quad (2.6)$$

$$z_{ij}^k \in \{0, 1\} \quad \forall i, j, k \in V \setminus \{r\}, k \neq i \quad (2.45)$$

O PCTSP Lagrangeano pode ser formulado como:

$$\begin{aligned}
 \min(D) = & \sum_{i \in V \setminus \{r\}} \left( \pi_i + \varpi_i + \rho_i - w_i + \sum_{k \in V \setminus \{r, i\}} (\lambda_{ik} + \mu_{ik}) \right) y_i \quad (2.50) \\
 & + \sum_{i, j \in V \setminus \{r\} | i \neq j} (c_{ij} - \pi_i - \varpi_j - \mu_{ij} + \sum_{k \in V \setminus \{r, i, j\}} \nu_{ij}^k) x_{ij} \\
 & + \sum_{j \in V \setminus \{r\}} (c_{rj} + \rho_j - \pi_r - \varpi_j - \sum_{k \in V \setminus \{r, j\}} \lambda_{jk}) x_{rj} + \\
 & \sum_{i \in V \setminus \{r\}} (c_{ir} + \rho_i - \pi_i - \varpi_r) x_{ir} \\
 & + \sum_{k, i, j \in V \setminus \{r\} | k \neq j \neq i} (-\lambda_{jk} - \mu_{ik} + \nu_{ij}^k) z_{ij}^k \\
 & + \sum_{k, j \in V \setminus \{r\} | k \neq j} (-\lambda_{jk} - \mu_{jk}) z_{jj}^k + \sum_{k, i \in V \setminus \{r\} | k \neq i} (-\mu_{ik}) z_{ik}^k
 \end{aligned}$$

sujeito a:

$$\begin{aligned}
 y_r &= 1 \\
 \sum_{i \in V \setminus \{r\}} p_i y_i &\geq g \\
 y_i &\in \{0, 1\} \quad \forall i \in V \\
 x_{ij} &\in \{0, 1\} \quad \forall i, j \in V \\
 z_{ij}^k &\in \{0, 1\} \quad \forall i, j, k \in V \setminus \{r\}, k \neq i
 \end{aligned}$$

Os multiplicadores de Lagrange,  $\pi$ ,  $\varpi$ ,  $\lambda$ ,  $\mu$ , das restrições complicantes (2.2), (2.3), (2.40) e (2.41) podem assumir qualquer valor real, visto que as restrições são igualdades. Repare que estas restrições foram escolhidas como complicantes, pois, a partir de suas respectivas relaxações, são obtidos subproblemas Lagrangeanos mais fáceis de se resolver. Como tem-se um problema de minimização, devido à relação de desigualdade ( $\leq$ ) das restrições (2.43) e (2.44), o domínio de seus respectivos multiplicadores é positivo, ou seja,  $\rho \geq 0$  e  $\nu \geq 0$ .

Observe que o problema Lagrangeano resultante pode ser decomposto em três subproblemas independentes:  $LR^y$ ,  $LR^x$ , e  $LR^z$ ; associados, respectivamente, com as variáveis  $y$ ,  $x$  e  $z$ .

O subproblema  $LR^y$  é dado pela FO (2.51), sujeito às restrições (2.7) e (2.6):

$$\begin{aligned}
 \min \quad & \sum_{i \in V \setminus \{r\}} \left( \pi_i + \varpi_i - \rho_i - w_i + \sum_{k \in V \setminus \{r, i\}} (\lambda_{ik} + \mu_{ik}) \right) y_i & (2.51) \\
 & \sum_{i \in V \setminus \{r\}} p_i y_i \geq g \\
 & y_i \in \{0, 1\} \quad \forall i \in V \setminus \{r\}
 \end{aligned}$$

Segundo Guignard (2003), o KP não possui a Propriedade de Integralidade, e que a restrição (2.7) é equivalente à restrição de capacidade da mochila (Seção 2.3). Logo, para explorar esta característica, faça:

$$\bar{y}_i = 1 - y_i, i \in V \setminus \{r\}; \quad (2.52)$$

$$U = \sum_{i \in V \setminus \{r\}} p_i - g; \quad (2.53)$$

$$\bar{u}_i = \pi_i + \varpi_i - \rho_i - w_i + \sum_{k \in V \setminus \{r, i\}} (\lambda_{ik} + \mu_{ik}) \quad (2.54)$$

Dessa forma,  $LR^y$  é equivalente ao KP dado por (2.55), (2.56) e (2.57) :

$$\max \sum_{i \in V \setminus \{r\}} \bar{u}_i \bar{y}_i \quad (2.55)$$

$$\sum_{i \in V \setminus \{r\}} p_i \bar{y}_i \leq U \quad (2.56)$$

$$\bar{y}_i \in \{0, 1\} \quad \forall i \in V \setminus \{r\} \quad (2.57)$$

Para Toth (1980), a partir desta transformação é possível resolver ( $LR^y$ ) utilizando um método de Programação Dinâmica. Este método resolve várias mochilas menores, recursivamente, até atingir a capacidade da mochila. Ou seja, dada a mochila de tamanho  $U$  e os itens, vértices  $\bar{y}_i$ , com seus respectivos prêmios  $p_i$ , o método, primeiramente, resolve o problema para uma mochila com capacidade 0. Em seguida resolve o problema para uma mochila com capacidade 1, depois para uma mochila de tamanho 2 e assim, sucessivamente, até obter uma solução para a mochila com capacidade  $U$ . Este método pseudo-polinomial possui complexidade  $O(nU)$ , no qual  $n = |V|$  é o número de itens (vértices) e  $U$  a capacidade da mochila (prêmio excedente), e é exemplificado pelo Alg. 1

---

**Algorithm 1** Subproblema  $LR^y$ 


---

```

1:  $U \leftarrow \sum_{i \in \bar{V}} (p_i) - g$ 
2: for  $u = 0$  to  $U$  do
3:    $M[0; u] \leftarrow 0$ 
4: end for
5: for  $i = 0$  to  $|\bar{V}|$  do
6:    $M[i; 0] \leftarrow 0$ 
7: end for
8: for  $i = 1$  to  $|\bar{V}|$  do
9:   for  $u = 0$  to  $U$  do
10:     $rc \leftarrow \alpha_{i-1}$ 
11:     $a \leftarrow (rc + M[i-1; u - p_{i-1}])$ 
12:     $b \leftarrow M[i-1][u];$ 
13:    if  $((p_{i-1} \leq u) \text{ and } (a < b))$  then
14:       $M[i; u] \leftarrow M[i-1; u - p_{i-1}] + rc$ 
15:    else
16:       $M[i; u] \leftarrow M[i-1; u]$ 
17:    end if
18:   end for
19: end for
20:  $i \leftarrow |\bar{V}|$ 
21:  $u \leftarrow U$ 
22: while  $(M[i; u] \neq 0)$  do
23:   if  $(M[i; u] = M[i-1][u])$  then
24:      $i \leftarrow i - 1$ 
25:      $\bar{y}_i \leftarrow 0$ 
26:   else
27:      $i \leftarrow i - 1$ 
28:      $\bar{y}_i \leftarrow 1$ 
29:      $u \leftarrow u - p_i$ 
30:   end if
31: end while

```

---

Na primeira fase, linhas 1-7, os parâmetros são inicializados. As linhas 8-19 são responsáveis pela resolução recursiva dos subproblemas, retornando apenas a solução ótima do subproblema da mochila com capacidade  $u = \{0, \dots, U\}$ . A cada iteração deste laço de repetição, é verificado se o item  $i$ , deve ser alocado à mochila de capacidade  $u$ . Os itens que pertencem à mochila, ou seja, os vértices que não serão visitados são definidos nas linhas 20-29.

O problema  $LR^x$  é definido pela FO (2.58), sujeito à restrição (2.5):

$$\begin{aligned}
 \min \quad & \sum_{i,k \in V \setminus \{r\} | i \neq k} (c_{ik} - \pi_i - \varpi_k - \mu_{ik} + \sum_{j \in V \setminus \{r\}} \nu_{ij}^k) x_{ik} \\
 & + \sum_{k \in V \setminus \{r\}} (c_{rk} + \rho_k - \pi_r - \varpi_k - \sum_{j \in V \setminus \{r,k\}} \lambda_{kj}) x_{rk} \\
 & + \sum_{i \in V \setminus \{r\}} (c_{ir} + \rho_i - \pi_i - \varpi_r) x_{ir}
 \end{aligned} \tag{2.58}$$

s. a:

$$x_{ik} \in \{0, 1\} \quad \forall i, k \in V$$

Este subproblema pode ser resolvido com uma simples inspeção, como exemplificado pelo Algoritmo 2.

---

**Algorithm 2** Subproblema ( $LR^x$ )
 

---

```

1: for ( $\forall i, k \in V, i \neq k$ ) do
2:   if ( $rc[x_{ik}] < 0$ ) then
3:      $x_{ik} = 1$ 
4:   else
5:      $x_{ik} = 0$ 
6:   end if
7: end for
    
```

---

A variável  $x_{ij}$  recebe o valor 1 se seu custo reduzido for negativo e 0 caso contrário. O cálculo do custo reduzido do arco ( $rc[x_{ij}]$ ), depende se este é o arco de saída do vértice raiz (Eq. (2.59)), um arco qualquer da subrota entre dois vértices  $i$  e  $j \in V \setminus \{r\}$  (Eq. (2.60)), ou se é um arco de chegada no vértice raiz (Eq. (2.61)) :

- Para um arco  $(r, j)$

$$rc[x_{rj}] = c_{rj} + \rho_j - \pi_r - \varpi_j - \sum_{k \in V \setminus \{r, j\}} \lambda_{jk} \quad (2.59)$$

- Para um arco  $(i, j)$

$$rc[x_{ij}] = c_{ij} - \pi_i - \varpi_j - \mu_{ij} - \sum_{k \in V \setminus \{r, i, j\}} \nu_{ij}^k \quad (2.60)$$

- Para um arco  $(i, r)$

$$rc[x_{ir}] = c_{ir} + \rho_i - \pi_i - \varpi_r \quad (2.61)$$

O problema  $LR^z$  é composto pela FO (2.62) e pela restrição de domínio (2.45):

$$\min \sum_{k, i, j \in V \setminus \{r\} | k \neq j; i \neq j} (-\lambda_{jk} - \mu_{ik} + \nu_{ij}^k) z_{ij}^k + \sum_{k, i \in V \setminus \{r\} | k \neq i} (-\mu_{ik}) z_{ik}^k \quad (2.62)$$

s. a:

$$z_{ij}^k \in \{0, 1\} \quad \forall k, i, j \in V \setminus \{r\}, k \neq i$$

Analogamente, o  $LR^z$  também pode ser facilmente resolvido por inspeção. Sejam  $i, j, k \in V \setminus \{r\}$  três vértices sujeitos à restrição de domínio (2.45), seus respectivos custos reduzidos  $rc$  são dados por:

- Associado à variável  $z_{ij}^k, i, j, k \in V \setminus \{r\}, i \neq j \neq k$

$$\nu_{ij}^k - \lambda_{jk} - \mu_{ik} \quad (2.63)$$

- Associado à variável  $z_{jj}^k, j, k \in V \setminus \{r\}, j \neq k$

$$-\lambda_{jk} - \mu_{ik} \quad (2.64)$$



- Associado à variável  $z_{ik}^k, i, k \in V \setminus \{r\}, i \neq k$ 

$$-\mu_{ik} \tag{2.65}$$

O subproblema  $LR^z$  é resolvido pelo Algoritmo 3, o qual atribui valor 1 às variáveis com o custo reduzido negativo e 0 caso contrário:

---

**Algorithm 3** Subproblema  $LR^z$ 


---

```

1: for ( $\forall k, i, j \in V \setminus \{r\} | i \neq j$ ) do
2:   if ( $rc[z_{ij}^k] < 0$ ) then
3:      $z_{ij}^k \leftarrow 1$ 
4:   else
5:      $z_{ij}^k \leftarrow 0$ 
6:   end if
7: end for

```

---

Na Seção 2.5 é apresentado o algoritmo proposto para resolver o PCSTP. Para tanto, é adotada uma abordagem híbrida que combina heurísticas com a relaxação Lagrangeana, baseada na formulação **EPN** apresentada nesta seção. Logo, são apresentados a seguir os procedimentos heurísticos de solução inicial, busca local e planos de corte.

### 2.4.1 Solução inicial

Nesta seção, é descrito o procedimento heurístico guloso utilizado para obter uma solução inicial viável  $UB^0$  para o PCTSP. Dado um subconjunto de vértices  $H \subseteq V, r \in H$ , tal que  $\sum_{i \in H} p_i \geq g$ , o procedimento consiste em construir uma rota para  $H$  por meio do Concorde<sup>1</sup>. Em seguida, uma busca local de duas fases é utilizada para refinar a solução.

O Concorde é um algoritmo especializado, escrito em C, para resolver o TSP simétrico e está disponível em <<http://www.math.uwaterloo.ca/tsp/concorde.html>>. Ele já foi utilizado com sucesso para encontrar a solução ótima de todas as instâncias da TSPLIB e para problemas de até 85 mil vértices. A biblioteca de procedimentos do Concorde inclui mais de 700 funções, por meio das quais é possível criar códigos especializados para problemas correlatos ao TSP, em especial, aqueles que tratam de sequenciamento, tal como o PCTSP. Para resolver um problema de forma exata o Concorde necessita de um *solver* externo de Programação Linear. Neste trabalho optou-se pelo CPLEX 12.7.

O método de construção define um subconjunto  $H^0 \subseteq V$  por meio de um procedimento guloso. Ele ordena, de forma decrescente, o conjunto de vértices  $i \in V$  de acordo com a relação  $(w_i/p_i)$ . Então, os vértices  $i$  são escolhidos de acordo com o

<sup>1</sup> Escrito por David Applegate, Robert E. Bixby, Vašek Chvátal, e William J. Cook.

maior valor, até que o prêmio mínimo  $g$ , seja coletado (Eq. (2.7)). O conjunto  $H^0$  é formado pelos vértices visitados ( $i \in H^0$  se  $y_i = 1$ ). Em seguida, a rota ótima local  $S^0$  é encontrada por meio do Concorde para todo  $i$  ativo. A rota é dita ótima local, pois é a melhor rota possível para o subconjunto  $H^0 \subseteq V$ , entretanto, não há garantias de que seja a solução ótima global. Após a construção da solução inicial  $S^0$ , é aplicada uma busca local, Seção 4.4.2, para refinar a solução.

Note que este trabalho trata da versão Assimétrica do PCTSP. Como o Concorde resolve apenas a versão simétrica do TSP, é necessário utilizar o procedimento proposto por Kumar e Li (2000) para transformar uma instância do ATSP com  $|H^0|$  vértices em uma instância de TSP simétrico com  $2|H^0|$  vértices.

### 2.4.2 Busca local

A busca local é utilizada para refinar uma solução viável  $S$  obtida a partir de um subconjunto  $H \subseteq V$ ,  $r \in H$ , tal que,  $\sum_{i \in H} p_i \geq g$ . O objetivo é embutir o esquema heurístico na abordagem da relaxação Lagrangeana. Assim, espera-se que subconjuntos de vértices promissores  $H$  possam ser produzidos durante a resolução dos subproblemas Lagrangeanos.

O procedimento consiste em construir uma rota para  $H$ , caso esta não exista, e refinar esta solução por meio de uma busca local. Este refinamento é feito por meio de buscas nas soluções vizinhas de  $S$  com sucessivos acréscimos e retiradas de vértices da solução corrente.

Uma vez definida uma solução para o PCTSP, esta é refinada por meio de uma busca local dividida em duas fases. A primeira fase tenta melhorar a solução atual com a inserção de vértices na rota. Um vértice  $i \in V \setminus H$  é inserido entre os vértices  $j$  e  $k$  pertencentes à rota, ou seja,  $H = H \cup \{i\}$ , se  $c_{ji} + c_{ik} < w_i + c_{jk}$ . Nesta etapa, o vértice  $i$  é inserido na primeira posição de melhora.

Uma vez que não há mais movimentos de melhora com vértices em  $V \setminus H$ , a segunda fase tenta melhorar a solução atual removendo vértices da rota. Seja  $j, i, k \in H$ ,  $i \neq r$ , três vértices sucessivos na rota, e  $e_i = w_i + c_{jk} - c_{ji} - c_{ik}$ . O conjunto de vértices candidatos a deixarem a rota é dado por  $H' = \{i \in H \setminus \{r\} : e_i < 0 \text{ e } \sum_{u \in H \setminus \{i\}} p_u \geq g\}$ . Os vértices removidos da rota são escolhidos a partir da solução do seguinte problema de otimização na qual a variável  $s_i$ ,  $i \in H'$ , assume 1 se o vértice  $i$  for removido, e 0 caso contrário.

$$\min \sum_{i \in H'} e_i s_i \quad (2.66)$$

$$\sum_{i \in H'} p_i s_i \leq \sum_{i \in H'} p_i - g \quad (2.67)$$

$$s_i + s_j \leq 1 \quad \forall i, j \in H' : \text{arco } (i, j) \text{ faz parte da rota} \quad (2.68)$$

$$s_i \in \{0, 1\} \quad \forall i \in H' \quad (2.69)$$

A função objetivo (2.66) mede o quanto é possível melhorar na solução atual com a retirada dos vértices. Restrições (2.67) garantem que o prêmio ainda seja coletado pelos vértices restantes na rota. As restrições (2.68) asseguram que dois vértices sucessivos na rota não serão removidos. Repare que, dado  $j, i, k \in H$  três vértices sucessivos na rota, o valor  $e_i$  representa a economia com a retirada do vértice  $i$  da mesma. Logo, caso dois vértices consecutivos sejam retirados simultaneamente da rota, por exemplo:  $j$  e  $i$ ; o valor  $e_i$  e  $e_j$  deixam de ser válidos. Seja  $Y^*$  o conjunto de vértices para os quais  $s_i = 1$  na solução ótima para (2.66)-(2.69), o Concorde é utilizado para encontrar uma rota ótima com os vértices em  $H \setminus Y^*$ . Então, retorna-se à primeira fase e a heurística é repetida até que nenhum vértice possa ser retirado ou acrescentado à solução.

A heurística parte de um subconjunto  $H \subseteq V$ , tal que  $y_i = 1$  para todo  $i \in H$ , obtido durante a resolução do subproblema  $LR^y$ . Para evitar retrabalho, antes de chamar o Concorde, sempre é verificado se o subconjunto  $H$  corrente já foi resolvido pelo mesmo. Em caso afirmativo, este procedimento não é realizado.

### 2.4.3 Desigualdades válidas e cortes de otimalidade para o PCTSP

O LB resultante da Relaxação Lagrangeana pode ser fortalecido com a adição de desigualdades válidas (ou cortes), que são atualmente violadas pela solução Lagrangeana em um esquema *relax-and-cut*. Como em Guignard (2003) e Lucena (2005), normalmente, os cortes de uma família de desigualdades, com um número exponencial de candidatos, são dualizados à medida que são identificados na solução corrente.

Nesta seção, famílias de cortes de otimalidade e de desigualdades válidas são, respectivamente, apresentadas: cortes *Cost-Cover* – CC; e desigualdades *Cycle-Cover* - CY. Elas são adotadas a partir de Fischetti e Toth (1988) e Bérubé, Gendreau e Potvin (2009) que as utilizam para fortalecerem os LB's do PCTSP e, dessa forma, aprimorarem o desempenho de seus respectivos algoritmos. Ambas precisam ser identificadas por meio de algoritmos de separação, que também são descritos a seguir. Eles são utilizadas a partir da solução do subproblema Lagrangeano.

### 2.4.3.1 Cortes Cost-Cover

A família de cortes de otimalidade *Cost-Cover* foram propostas por Leifer e Rosenwein (1994) para o *Orienteering Problem* (OP) e foram utilizadas para o PCTSP Simétrico por Bérubé, Gendreau e Potvin (2009). Naquele trabalho, os autores consideraram apenas dois casos especiais:  $|S| = 2$  e  $|S| = 3$ . Dessa forma é mais fácil identificar cortes violados. Eles utilizam um UB da solução ótima para definir uma restrição da mochila em termos dos custos que pode ser usada para derivar cortes de otimalidade. Estes cortes são válidos quando o custo dos arcos satisfazem a desigualdade triangular. Entretanto, neste trabalho, nem todas as instâncias apresentam esta propriedade.

Seja  $\bar{U}$  um UB para o valor da solução ótima de uma instância do PCTSP; e  $\bar{L}_S$  um LB para o valor ótimo da rota resultante do subconjunto  $S \subseteq V$ ,  $r \in S$ , ou seja, um LB para uma instância do TSP em  $S$ . Então, segundo Bérubé, Gendreau e Potvin (2009), se  $\bar{L}_S > \bar{U}$ , e os custos satisfazem a desigualdade triangular, o corte de otimalidade (2.70) :

$$\sum_{i \in S} y_i \leq |S| - 1 \quad (2.70)$$

é válido para o PCTSP.

A cada iteração, verificamos se o subconjunto  $S$ , composto pelos vértices para os quais  $y_i$  é igual a 1 na solução ótima de  $LR^y$ , é inédito. Em caso afirmativo, um novo UB para o PCTSP, de valor  $f(S)$ , é encontrado ao somar o custo da rota do TSP para  $S$  mais as penalidades dos vértices  $i \in V \setminus S$ . Dado  $S^*$  a melhor solução até o momento e  $f(S^*)$  seu respectivo valor, os cortes *Cost-Cover*, Eq. (2.70), que são possivelmente violados podem ser identificados. Se  $f(S) \geq f(S^*)$ , a solução  $S$  é acrescida ao corte. Caso  $f(S) < f(S^*)$ , a solução  $S^*$  é acrescida ao corte e são atualizados os parâmetros  $f(S^*) \leftarrow f(S)$  e  $S^* \leftarrow S$ .

### 2.4.3.2 Desigualdades Cycle-Cover

As *Cycle-Cover Inequalities* foram propostas por Fischetti e Toth (1988) para o PCTSP. Elas exploram o fato de que toda solução factível é um ciclo e deve coletar o prêmio mínimo  $g$ .

Seja  $Y \subset V$  um subconjunto de vértices, tal que  $r \in Y$  e  $\sum_{i \in Y} p_i < g$ , então, segundo Fischetti e Toth (1988), (2.71) :

$$\sum_{i,j \in Y} x_{ij} \leq \sum_{i \in Y} y_i - 1 \quad (2.71)$$

é válida para o PCTSP.

Conforme observado por Bérubé, Gendreau e Potvin (2009), o algoritmo de separação utilizado para **GSEC** (Seção 2.3.1), baseado no problema de fluxo máximo pode ser usado na tentativa de identificar possíveis desigualdades *Cycle-Cover* violadas. Considere  $\bar{G} = (\bar{V}, \bar{A})$  como um grafo auxiliar obtido a partir da solução  $(\bar{x}, \bar{y})$ , tal que  $\bar{V} = \{i \in V : \bar{y}_i > 0\}$  e  $\bar{A} = \{(i, j) \in A : \bar{x}_{ij} > 0\}$ . Defina  $\bar{x}_{ij}$  como a capacidade do arco  $(i, j) \in \bar{A}$ . Para cada  $i \in \bar{V} \setminus \{r\}$ , basta verificar a soma dos prêmios dos vértices pertencentes ao conjunto  $Y$  no corte de capacidade mínima contendo o vértice  $r$ . Se esta soma for menor do que  $g$ , a inequação (2.71) é violada. O algoritmo de separação é baseado no algoritmo de fluxo máximo, que pode ser usado a partir da implementação proposta por Dinitz (2006), disponível em <<https://gist.github.com/gabber12/5257878>>. Esta implementação possui uma complexidade  $O(n^2m)$  para cada rodada do problema de fluxo máximo entre cada vértice  $i \in \bar{V}$  e  $r$ , sendo  $n$  o número de vértices e  $m$  o número de arestas.

## 2.5 Algoritmo proposto para o PCTSP

Esta seção descreve o esquema geral do Algoritmo proposto para resolver o PCTSP, cujos objetivos são os seguintes:

- (i) obter um LB próximo da solução ótima;
- (ii) encontrar soluções primais ótimas ou quase ótimas com um esquema heurístico híbrido que explora a solução do problema Lagrangeano;
- (iii) e delimitar a solução ótima dentro de um pequeno intervalo de valores com esforço computacional razoável.

O Algoritmo 4 mostra o pseudocódigo da abordagem Lagrangeana proposta para resolver o PCTSP. Utiliza-se o algoritmo Volume (VOL), proposto por Barahona e Anbil (2000), para resolver o problema dual Lagrangeano. Para mais detalhes sobre o VOL veja Barahona e Anbil (2002) e Bahiense, Barahona e Porto (2003).

**Algorithm 4** Abordagem por meio da relaxação Lagrangeana

---

```

1: Inicializa parâmetros de acordo com o VOL
2:  $UB \leftarrow$  solução primal viável
3: while critério de parada não for satisfeito do
4:    $it \leftarrow it + 1$ 
5:   if  $t \bmod h_{it} \leftarrow 0$  then
6:      $LB_{it} \leftarrow 0$ 
7:   end if
8:   Atualize  $\theta$  e os multiplicadores de Lagrange  $\eta$ 
9:   Resolva os subproblemas Lagrangeanos e obtenha  $H'$  e um  $LB'$ 
10:  Atualize  $v^{it}$ , e faça  $v_k^{it} = 0$  quando possível
11:  Atualize  $\alpha^{it}$ ,  $d^{it}$ ,  $\bar{\chi}^{it}$ 
12:  if  $LB' > LB$  then
13:     $LB \leftarrow LB'$ 
14:     $LB_{it} \leftarrow LB'$ 
15:     $H \leftarrow H'$ 
16:  else
17:    if  $LB' > LB_{it}$  then
18:       $LB_{it} \leftarrow LB'$ 
19:       $H \leftarrow H'$ 
20:    end if
21:  end if
22:  if  $H'$  não foi gerado até o momento then
23:    Encontre uma rota ótima para  $H'$ 
24:    if  $\sum_{i \in H'} y_i \geq g$  then
25:      calcule  $UB'$ 
26:      if  $UB' < UB$  then
27:         $UB \leftarrow UB'$ 
28:      end if
29:    end if
30:    Verifique se alguma inequação válida é violada pelo conjunto  $H'$ 
31:  end if
32:  if  $t \bmod h_{it} = 0$  e conjunto  $H$  não foi gerado até o momento then
33:    Aplique a heurística em  $H$  para obter uma solução inicial viável e calcule  $UB'$ 
34:    if  $UB' < UB$  then
35:       $UB \leftarrow UB'$ 
36:    end if
37:  end if
38: end while

```

---

A inicialização na Linha 1 define como 0 o contador de iteração  $it$  e o contador do bloco de  $h_{it}$  iterações. os demais parâmetros do VOL:  $\eta$  é o vetor de multiplicadores de Lagrange;  $LB$  a solução do problema dual Lagrangeano;  $\chi$  a solução primal resultante dos subproblemas; e  $v^0$  o vetor de subgradientes.

Na Linha 2 uma solução inicial é gerada por meio de um procedimento guloso. Ao final deste procedimento, é definido o primeiro  $UB$  para o problema, ou seja, a primeira solução viável. O procedimento é descrito na Seção 2.4.1.

O laço de repetição principal é executado enquanto os critérios de parada não são atendidos. São considerados os mesmos critérios de parada do VOL, apresentados em Barahona e Anbil (2000).

Na Linha 4 o contador de iterações é aumentado. Em seguida, na linha 6, o parâmetro  $LB_{it}$  que armazena o melhor  $LB$  é definido como 0. Isto é feito no início de cada bloco de  $h_{it}$  iterações.

Uma vez que o passo  $\theta^t$  e a direção  $d^t$  já foram definidos, é possível atualizar o valor dos multiplicadores de Lagrange  $\eta$ , linha 8,

Em seguida na Linha 9, o valor  $LB'$  é calculado a partir da resolução dos subproblemas Lagrangeanos de acordo com a Eq. (2.72) :

$$LB' \leftarrow LR^y + LR^x + LR^z + W + (\pi_r + \varpi_r - w_r) \quad (2.72)$$

A cada iteração do algoritmo, os subproblemas Lagrangeanos são respectivamente resolvidos pelos procedimentos descritos na Seção 2.4.

O valor dos subgradientes  $v^{it}$  é atualizado na linha 10. Para melhorar a convergência do algoritmo é adotado o mesmo ajuste no vetor  $v$  proposto por Beasley (1993). Tal ajuste consiste em fazer  $v_k = 0$  se  $v_k > 0$  e  $\eta_k = 0$ . Repare que este ajuste aplica-se apenas aos multiplicadores de Lagrange relativos às restrições do tipo menor ou igual ( $\leq$ ).

Os valores de  $\alpha^{it}$ ,  $d^{it}$ ,  $\bar{\chi}^{it}$  são atualizados na linha 11. Esta etapa segue as mesmas regras definidas para o VOL por Barahona e Anbil (2000).

Caso o  $LB'$  seja o melhor até o momento, os valores  $LB$ ,  $LB_{it}$ , e  $H$  são atualizados nas linhas 13, 14, e 15, respectivamente. Caso contrário, é verificado na linha 17 se  $LB'$  é o melhor  $LB$  dentro do bloco de  $h_{it}$  iterações. Em caso afirmativo  $LB_{it}$  e  $H$  são atualizados, respectivamente, nas linhas 18 e 19.

O conjunto  $H$  é formado pelo vértice raiz  $r$  e pelo subconjunto de vértices  $y_i = 1$  para  $i \in V$ . Este subconjunto fornece o melhor  $LB$  até o momento ou o melhor valor dentro do bloco de  $h_{it}$  iterações,  $LB_{it}$ . Esses vértices são utilizados pelos procedimentos heurísticos, Seção 2.4.2, para encontrar soluções viáveis  $UB$ .

Na linha 22 é verificado se o conjunto  $H'$  não foi gerado em uma iteração anterior. Caso  $H'$  seja inédito, e se  $\sum_{i \in H'} y_i \geq g$ , o procedimento para gerar uma solução viável é chamado. Este procedimento é semelhante ao apresentado na Seção 2.4.2, exceto pela busca local que não é utilizada nesta etapa. Ou seja, é resolvido um TSP para o subconjunto de vértices  $H'$ , por intermédio do Concorde. Encontrado uma solução viável  $UB$ , na linha 24, é verificado se esta é a melhor solução até o momento. Em caso afirmativo, o valor de  $UB'$  é atualizado na linha 27.

Observe que na linha 30 ocorre a busca por desigualdades válidas/cortes de otimalidade violados apenas se o conjunto  $H'$  não foi gerado em uma iteração anterior (caso contrário, os cortes, se houver, já estão dualizados nas funções objetivo dos subproblemas Lagrangeanos). Identificado um corte/desigualdade violada, este é dualizado e mantido até o final do algoritmo.

Os corte/desigualdades são identificados por meio de algoritmos de separação. Para detalhes destes algoritmos reveja a Seção 2.4.3. Depois de identificadas, elas são dualizadas e acrescentadas ao problema dual Lagrangeano. Ou seja, para cada corte identificado, um multiplicador de Lagrange  $\bar{\eta}$  é associado. Este multiplicador

será utilizado para o cálculo do custo reduzido e na resolução dos subproblemas Lagrangeanos. A variável dual  $\bar{\eta}$  associada à nova inequação é inicializada como zero. Uma vez que todas os cortes/desigualdades são do tipo  $\leq$  o valor de  $\bar{\eta}$  é atualizado segundo a Eq. (2.73)

$$\bar{\eta} = \max\{0, \bar{\eta} + \theta V\} \quad (2.73)$$

Repare que o mesmo ajuste no vetor  $v$  proposto por Beasley (1993), utilizado para melhorar a convergência do algoritmo, também pode ser aplicado. Logo, temos que  $v_k = 0$  se  $v_k > 0$  e  $\bar{\eta}_k = 0$ . Uma vez que  $\bar{\eta}$  é acrescentada ao problema dual Lagrangeano, ela não será retirada. Assim, ela será considerada na resolução dos subproblemas Lagrangeanos nas iterações seguintes, até o fim da execução do algoritmo.

Na linha 32 é verificado no final de um bloco de  $h_{it}$  iterações se o conjunto  $H$  de vértices é inédito. Este conjunto gera o melhor  $LB$  dentro deste bloco de iterações. No caso afirmativo, a heurística de busca local é aplicada, linha 33, e o  $UB$ , possivelmente, atualizado, linha 35.



## 3 Experimentos Computacionais sobre o Problema do Caixeiro Viajante com Coleta de Prêmios

O presente capítulo apresenta os resultados dos experimentos computacionais realizados para o PCTSP. Primeiro, são apresentadas as instâncias utilizadas, seguidas pelos resultados da comparação das formulações, descritas na Seção 2.3, em termos da relaxação linear. Na seção seguinte, os resultados para o Algoritmo 4, proposto na Seção 2.5, são apresentados. Este algoritmo explora as informações da solução proveniente dos subproblemas. Estas informações são utilizadas para encontrar soluções viáveis para a instância do PCTSP. Isto é feito por meio de um procedimento heurístico que aplica uma busca local depois de encontrar uma solução. Estas informações também são utilizadas para identificar famílias de cortes/desigualdades violadas, as quais são dualizadas e acrescentadas aos subproblemas. Dessa forma, o Algoritmo 4 é capaz de encontrar um LB melhor. No fim do capítulo, o impacto do uso destas famílias, no Algoritmo 4, também é mostrado.

Todos os experimentos foram executados em um PC Intel® Core™ i7-2600, 3,40 GHz, 16 GB de RAM, sob uma distribuição Linux 64 bits. As formulações foram implementadas com a linguagem C, por meio da interface do *solver* Gurobi versão 7.5.2, 64 bits, o qual foi utilizado para tentar resolver o problema inteiro até a otimalidade, assim como as relaxações lineares. A implementação da abordagem Lagrangeana (Algoritmo 4, Seção 2.5), por meio da linguagem C++, foi baseada no código fornecido pelo COIN-OR *Project*, disponível em <<https://projects.coin-or.org/Vol>>. Também é utilizado a interface do Concorde combinado com o CPLEX versão 12.7, 64 bits. O CPLEX foi utilizado pois o Gurobi, já utilizado anteriormente, não possui interface com o Concorde. Ainda, para o PCTSP, o Gurobi versão 7.5.2, 64 bits, também é utilizado para resolver o problema resultante da segunda etapa da busca local apresentada na Seção 2.4.2.

### 3.1 Instâncias para o PCTSP

Nesta seção, são apresentadas as instâncias utilizadas neste trabalho, inicialmente propostas para o ATSP. Logo, elas precisam ser adaptadas ao PCTSP. Os detalhes desta adaptação, tais como, a forma como são determinados os pesos  $p_i$  e as penalidades  $w_i$ , são expostos a seguir. Ao todo foram usadas 252 instâncias do PCTSP

no experimento descrito neste trabalho.

As instâncias foram geradas a partir dos grafos de instâncias para o ATSP a partir da TSPLIB, proposta por Reinelt (1991), e do trabalho de Cirasella et al. (2001) e estão disponíveis, respectivamente, em <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/> e <https://sites.google.com/site/atspinstances/cirasella-instances>. Neste último, elas são utilizadas para comparar diversos algoritmos propostos para o TSP. Neste trabalho, consideramos três grupos de instâncias utilizados por Cirasella et al. (2001):

- *Specific Instances: TSPLIB and Other Sources* – `realworld`.
- *Tilted Drilling Machine Instances with Sup Norm* – `stilt`.
- *Random Euclidean Stacker Crane Instances* – `crane`.

Estas são escolhidas a partir da observação dos resultados de Cirasella et al. (2001); por meio do qual é possível constatar que elas estão entre as mais difíceis de serem resolvidas. Ainda segundo o autor, elas também são as que estão mais próximas de possíveis aplicações reais.

O grupo `realworld` inclui 28 instâncias retiradas da TSPLIB, proposta por Reinelt (1991), mais 13 novas instâncias de outras fontes (aplicações). Para as instâncias retiradas da TSPLIB têm-se: quatro `rbg` retiradas de um problema de guindaste (semelhante à `crane`); duas `ft` de um problema de sequenciamento de tarefas, retiradas de Fischetti e Toth (1992); `ry48p` e `kro124p` são, originalmente, instâncias euclidianas simétricas, as quais são transformadas em assimétricas a partir de perturbações aleatórias em suas matrizes de distância, conforme descrito por Fischetti e Toth (1992); `p43` vem de um problema de sequenciamento da indústria química; `br17` é de uma fonte desconhecida; e 18 `ftv` de um problema de entrega de produtos farmacêuticos, retiradas de Fischetti, Toth e Vigo (1994). Entre as 18 `ftv`, 9 foram geradas a partir da instância `ftv170.atsp`, que tem 171 nós. Para gerar a `ftv80` considere apenas os primeiros 81 nós, para os primeiros 91 nós a `ftv90` e assim por diante até 161 nós conforme descrito em Fischetti e Toth (1997). Para as 13 novas instâncias têm-se: duas `td` de um gerador proposto por Bruce Hillyer of Lucent para o problema de sequenciamento de uma leitora magnética; cinco `dc` foram fornecidas por Adam Buchsbaum da AT&T Labs; duas `code` de um problema de otimização descrito em Young et al. (1997); quatro `atex` fornecidas por Victor Manuel da Universidade Carlos III de Madrid. Para mais detalhes sobre as instâncias veja Cirasella et al. (2001).

O grupo `stilt` é composto por 20 grafos diferentes que variam entre 100 e 316 vértices. Segundo Cirasella et al. (2001) ele foi inspirado no processo de perfuração no qual deseja-se fazer uma série de furos em uma superfície inclinada, e a broca é

movida usando dois motores. Cada motor é responsável por mover a broca em uma direção diferente. Os parâmetros são escolhidos para que o deslocamento vertical, eixo  $y$ , seja mais rápido do que o deslocamento horizontal, eixo  $x$ . Ainda no eixo  $y$ , o movimento para baixo é mais rápido do que para cima. Para estas máquinas, é adotada a distância máxima entre os furos nas direções  $x$  e  $y$ .

O grupo `crane` é composto por 23 grafos diferentes que variam entre 66 e 316 vértices. Ele foi inspirado no problema conhecido como *Stacker Crane Problem*. Este consiste no trabalho de um guindaste que deve pegar um objeto  $i$  no local de origem  $s_i$  e entregá-lo em seu destino  $d_i$ . O objetivo é ordenar essas tarefas de forma a minimizar o tempo gasto pelo guindaste. Este pode ser visto como um ATSP no qual o vértice  $c^i$  corresponde ao par  $(s_i, d_i)$  do objeto  $i$ . Neste caso, a distância entre os vértices  $c^i$  e  $c^j$  é a distância entre o destino do objeto  $i$  ( $d_i$ ) e a origem do objeto  $j$  ( $s_j$ ).

Também é importante destacar que dos 84 grafos, 36 não satisfazem a desigualdade triangular, ou seja,  $c_{ij} > c_{ik} + c_{kj}$ , para algum trio de vértices  $i, k, j \in V$ , a saber: `br17`, `kro124`, `ry48p`, `p43`, `code198`, `code253`, `dc126`, `td100_1`, `td316_10`, `rbg323`, `rbg358`, `rbg403`, e `rbg443` do grupo `realworld`; e as 23 do grupo `crane`. Geralmente, as instâncias que não satisfazem a desigualdade triangular são mais difíceis de resolver. Isto porque muitos algoritmos são capazes de explorar essa característica, tornando-os melhores.

Assim, no total, 84 grafos assimétricos, que variam entre 16 e 443 vértices, serviram de base para as instâncias PCTSP. Como dito anteriormente, elas foram originalmente propostas para o ATSP. Logo, os demais parâmetros tais como a penalidade e o prêmio associados à cada vértice  $i$ , e o prêmio mínimo  $g$  foram gerados conforme é proposto por Fischetti e Toth (1988) e Bérubé, Gendreau e Potvin (2009).

Para cada um dos 84 grafos, 3 instâncias do PCTSP assimétrico foram geradas a partir do prêmio mínimo que deve ser coletado como  $g = (\tau/100) \sum_{i \in V \setminus \{r\}} p_i$ ,  $\tau \in \{20, 50, 80\}$ , o que resulta um total de 252 instâncias usadas neste experimento. Ou seja, para cada grafo, temos três instâncias que diferem entre si pela porcentagem mínima do prêmio total que deve ser coletado, a saber: 20%, 50% e 80%.

Para todas as 252 instâncias, o prêmio  $p_i$ , para todo  $i \in V \setminus \{r\}$ , são números inteiros e escolhidos aleatoriamente na distribuição uniforme no intervalo  $[1, 100]$ .

Observe o caso no qual a penalidade  $w_i$  associada ao vértice  $i$  é muito grande, se comparada ao custo do arco. Neste caso, a solução do PCTSP é equivalente ao do TSP, pois a solução ótima de ambos será composta por todos os vértices.

Por isso, o valor para a penalidade  $w_i$  são números inteiros escolhidos aleatoriamente de acordo com o grupo de instâncias ao qual pertence:

- Para o grupo `realworld`:  $w_i =$  distribuição uniforme no intervalo  $[1, 10^2]$ .
- Para o grupo `stilt`:  $w_i =$  distribuição uniforme no intervalo  $[1, 10^5]$ ;
- Para o grupo `crane`:  $w_i =$  distribuição uniforme no intervalo  $[1, 10^5]$ ;

### 3.2 Resultados da relaxação linear

Aqui são apresentados os resultados para a comparação das formulações descritas na Seção 2.3. Esta comparação consiste no desempenho em termos da relaxação linear, ou seja, na capacidade da formulação gerar uma solução para a relaxação linear próxima da solução ótima e no respectivo esforço computacional. Os resultados para o modelo inteiro são mostrados no Apêndice A.

Dado que a formulação **GSEC** possui um número de restrições exponenciais, utilizou-se um algoritmo de separação para identificar as restrições violadas, tal como descrito na Seção 2.3.1. Este algoritmo pode ser resolvido em tempo polinomial. Para tanto, foi utilizado um algoritmo de fluxo-máximo para verificar se existe algum corte mínimo ( $\delta(S)$ ) entre os vértices  $r$  e  $i$ , tal que  $y_i > 0$ , na solução da relaxação linear que viole a restrição  $\delta(S) \geq 2$ . O algoritmo de separação implementado adota o algoritmo de fluxo-máximo baseado na implementação proposta por Dinitz (2006), disponível em <https://gist.github.com/gabber12/5257878>, a qual possui uma complexidade  $O(n^2m)$ , sendo  $n$  o número de vértices e  $m$  o número de arcos.

As Tabelas 2, 4 e 5 resumem, respectivamente, os resultados para os grupos `realworld`, `crane` e `stilt`. Elas apresentam a comparação da relaxação linear das 6 formulações em termos da força da relaxação linear ( $gap$ ), e do esforço computacional. Apenas **PSC** foi capaz de obter os valores inteiros ótimos para todas as 192 instâncias do grupo `crane` e `realword`, por meio do Gurobi, sem limite de tempo. Para o grupo `stilt`, **PSC** foi capaz de resolver 41 dos 60 problemas até a otimalidade, com limite de tempo. As outras formulações, incluindo **GSEC**, ficaram sem memória para instâncias maiores. O  $gap$  representa o desvio relativo da relaxação linear  $LB$ , em relação à solução ótima  $opt$  que é calculado segundo a Equação (3.1):

$$gap = (Opt - LB)/Opt \quad (3.1)$$

Para as instâncias nas quais a solução ótima não é conhecida, é feita a substituição da solução ótima  $opt$ , pela melhor solução conhecida dentro do limite de 3 horas. Os problemas foram executados com um limite de tempo de 10.800 segundos. Na parte superior das Tabelas 2, 4 e 5 para cada formulação, é apresentado na primeira linha a quantidade de problemas-teste do grupo para os quais o problema de relaxação linear foi resolvido dentro do limite de tempo. Em seguida, na segunda linha ( $AVG\_gap$ ), são apresentados o  $gap$  médio para as esses problemas-teste que foram resolvidos.

Tabela 2 – Resultados da comparação da relaxação linear do grupo `realworld` para as formulações com tempo limite de 10800 segundos (PCTSP).

		Formulações					
		<b>GSEC</b> Seção 2.3.1	<b>MCF</b> Seção 2.3.2	<b>OPB</b> Seção 2.3.3	<b>EnC</b> Seção 2.3.4	<b>EPN</b> Seção 2.3.5	<b>PSC</b> Seção 2.3.6
	#SOLVED <sub>LP</sub>	123	77	57	44	78	123
	AVG_gap	10,93	1,77	1,77	1,76	<b>1,48</b>	11,31
	AVG_tempo	2	22	131	2716	45	2
Instância							
$\tau = 20$							
ft70	gap	3,93	1,62	1,62	1,62	<b>1,42</b>	1,74
	tempo	47	24	172	8086	55	22
ft53	gap	4,89	0,68	0,66	<b>0,65</b>	<b>0,65</b>	13,63
	tempo	2	17	75	947	18	1
ry48p	gap	19,60	10,61	10,61	9,93	<b>8,67</b>	18,26
	tempo	1	25	116	703	35	1
atex1	gap	8,96	8,96	8,96	8,96	<b>0,00</b>	8,96
	tempo	0	0	0	38	0	0
atex4	gap	21,72	0,42	0,42	0,42	<b>0,00</b>	21,70
	tempo	0	21	191	8247	17	0
$\tau = 50$							
atex1	gap	25,65	22,38	22,38	22,38	<b>21,00</b>	32,05
	tempo	0	0	0	45	0	0
atex4	gap	20,29	0,02	0,02	0,02	<b>0,00</b>	19,90
	tempo	0	17	162	8923	16	1
$\tau = 80$							
atex1	gap	34,93	2,73	2,73	2,73	<b>2,58</b>	22,91
	tempo	0	0	0	62	0	0

Da mesma forma, na terceira linha são relatados os tempos computacionais médios (AVG\_tempo), em segundos, para obter o *LB* sobre esses mesmos problemas-teste. De fato, **MCF**, **OPB**, **EnC** e **EPN** forneceram o mesmo *gap* para quase todas as instâncias (**GSEC** e **PSC** obtiveram *gap* pior para todas as instâncias). Logo, são detalhados apenas os resultados para as instâncias nas quais o *gap* de pelo menos uma dessas 4 formulações é diferente. Os valores destacados em negrito indicam o melhor resultado.

Para o grupo `realworld`, a relaxação linear de apenas 44, das 123 instâncias, foram resolvidas por todas as 6 formulações dentro do limite tempo. Entretanto, a Tabela 3 mostra os resultados para as formulações **GSEC**, **MCF**, **EPN** e **PSC** para as 39 instâncias, para além das 44 detalhadas na Tabela 2, para as quais ou **MCF** ou **EPN** resolveram a relaxação linear, dentro do limite de tempo. A parte superior da Tabela 3 mostra os resultados médios para as 28 das 39 instâncias analisadas nesta tabela para a qual tanto **MCF** como **EPN** resolveram o problema do relaxamento linear. Como pode ser visto, as formulações **MCF** e **EPN** forneceram os melhores limites da relaxação linear com os custos computacionais mais elevados quando comparadas com **PSC** e **GSEC**. Ainda, a formulação **EPN** obteve o melhor *gap* médio e um tempo computacional médio menor do que a formulação **MCF**. Mais especificamente, **MCF** e **EPN** forneceram os mesmos limites para todas estas 28 instâncias, exceto para a instância `krol124`  $\tau = 20$ . Os resultados detalhados para esta instância, bem como as demais instâncias resolvidas por apenas uma formulação, são mostrados na parte inferior da Tabela 3. O símbolo “–” indica que a formulação não resolveu a relaxação

Tabela 3 – Resultados para as 39 instâncias, para além das 44 detalhadas na Tabela 2 para as quais ou **MCF** ou **EPN** resolveram a relaxação linear dentro do limite de tempo de 10.800 segundos.

		Formulations			
		<b>GSEC</b>	<b>MCF</b>	<b>EPN</b>	<b>PSC</b>
		Section 2.3.1	Section 2.3.2	Section 2.3.5	Section 2.3.6
	AVG_gap	12,45	0,83	<b>0,82</b>	10,40
	AVG_time	5	1234	1025	11
Instance					
$\alpha = 0, 2$					
krol124	gap	24,40	3,09	<b>2,84</b>	15,69
	time	25	4993	1968	16
ftv120	gap	24,29	20,78	–	21,67
	time	31	3164	–	36
ftv130	gap	11,72	7,57	–	9,55
	time	10	9378	–	45
dc126	gap	0,18	–	0,05	0,11
	time	1	–	481	5
dc134	gap	0,22	–	0,03	0,12
	time	63	–	4523	41
$\alpha = 0, 5$					
ftv120	gap	24,29	20,78	–	21,67
	time	29	5212	–	36
ftv130	gap	11,72	7,57	–	9,55
	time	38	10709	–	45
dc126	gap	2,18	–	0,50	1,05
	time	2	–	3247	12
dc134	gap	0,22	–	0,03	0,12
	time	72	–	4678	44
$\alpha = 0, 8$					
ftv120	gap	24,29	20,78	–	21,67
	time	32	3912	–	36
dc126	gap	5,08	–	0,17	3,20
	time	12	–	4797	28
dc134	gap	0,53	–	0,10	0,30
	time	94	–	5303	76

linear dentro do limite de tempo. A formulação **OPB** não aparece na Tabela 3 porque foi capaz de resolver a relaxação linear dentro do limite de tempo apenas para 13 dos 39 instâncias representadas na Tabela 3. Ela obteve os mesmos limites obtidos por **MCF** e **EPN** mas exigindo tempo computacional muito superior.

Para o grupo *crane*, mostrado na Tabela 4, as formulações **MCF** e **EPN** conseguiram resolver a relaxação linear, dentro do limite de tempo, para as mesmas 39 instâncias. A formulação **EnC** não foi capaz de resolver nenhuma das relaxações lineares e a formulação **OPB** resolveu apenas 6 das 69 instâncias. Dentre essas 6, ela forneceu o mesmo *gap* que **MCF**. A Tabela 4 tem a mesma estrutura da Tabela 2. A formulação **EPN** obteve melhores *gaps* do que a formulação **MCF** para 6 das 39 instâncias, conforme detalhado na parte de baixo da Tabela 4.

Para o grupo *stilt*, mostrado na Tabela 5, as formulações **EnC** e **OPB** não resolvem as relaxações lineares em nenhuma das 60 instâncias. Assim, são apresentados apenas os resultados para as formulações **GSEC**, **MCF**, **EPN** e **PSC**. A Tabela 5 tem

Tabela 4 – Resultados da comparação da relaxação linear do grupo `crane` para as formulações com tempo limite de 10800 segundos (PCTSP).

		Formulações				
		<b>GSEC</b>	<b>MCF</b>	<b>EPN</b>	<b>PSC</b>	
		Seção 2.3.1	Seção 2.3.2	Seção 2.3.5	Seção 2.3.6	
		<i>#SOLVED<sub>L<sub>P</sub></sub></i>	69	39	39	69
		AVG_gap	8,96	0,65	<b>0,61</b>	7,31
		AVG_tempo	33	623	721	24
Instância						
$\tau = 20$						
crane66_2	gap	8,26	1,73	<b>1,42</b>	8,34	
	tempo	1	71	45	2	
crane100_2	gap	9,62	1,81	<b>1,59</b>	11,92	
	tempo	25	712	836	20	
crane100_3	gap	8,39	1,11	<b>0,42</b>	8,13	
	tempo	23	569	824	30	
crane100_4	gap	10,05	0,15	<b>0,11</b>	8,09	
	tempo	22	711	858	22	
crane100_8	gap	8,78	0,52	<b>0,38</b>	8,2	
	tempo	25	621	753	21	
$\tau = 50$						
crane100_3	gap	8,04	0,27	<b>0,26</b>	7,75	
	tempo	24	609	731	42	

Tabela 5 – Resultados da comparação da relaxação linear do grupo `stilt` para as formulações com tempo limite de 10800 segundos (PCTSP).

		Formulations				
		<b>GSEC</b>	<b>MCF</b>	<b>EPN</b>	<b>PSC</b>	
		Seção 2.3.1	Seção 2.3.2	Seção 2.3.5	Seção 2.3.6	
		<i>#SOLVED<sub>L<sub>P</sub></sub></i>	60	30	30	60
		AVG_gap	15,61	2,91	<b>2,68</b>	15,33
		AVG_tempo	44	645	824	23
Instância						
$\tau = 20$						
stilt100_3	gap	12,2	9,88	<b>8,66</b>	19,8	
	tempo	34	636	796	25	
stilt100_6	gap	13,4	8,20	<b>7,84</b>	19,6	
	tempo	34	761	878	24	
stilt100_7	gap	14,1	6,50	<b>5,25</b>	18,2	
	tempo	36	744	851	22	
stilt100_8	gap	13,9	6,97	<b>5,21</b>	13,4	
	tempo	33	723	881	21	
stilt100_9	gap	15,4	3,77	<b>2,28</b>	13,9	
	tempo	40,0	595,9	811,0	28,1	
$\tau = 50$						
stilt100_3	gap	19,9	1,98	<b>1,64</b>	17,6	
	tempo	36	519	739,0	17	
stilt100_7	gap	15,5	1,15	<b>1,12</b>	18,7	
	tempo	47	694	773	24	
stilt100_8	gap	21,4	1,21	<b>0,98</b>	12,0	
	tempo	29,0	655,00	792,0	29,0	

a mesma estrutura da Tabela 2. A formulação **EPN** obteve melhores *gaps* do que a formulação **MCF**, para 8 das 30 instâncias que as duas formulações conseguiram resolver, dentro do limite de tempo, conforme detalhado na parte de baixo da Tabela 5.

Observa-se pelos experimentos que a formulação proposta **EPN**, quando capaz de resolver o problema de relaxação linear dentro do limite de tempo, obteve os melhores desvios da relaxação linear entre as consideradas neste estudo. Este fato indica que esta formulação, para o PCTSP, pode ser mais forte que as demais. Além disso, tais desvios foram obtidos com esforço computacional consideravelmente menor do que o necessário para resolver os problemas de relaxação linear das outras duas formulações, fornecendo os limites inferiores mais fortes, a saber, **OPB** e **EnC**. Por outro lado, **EnC**, se comparada com **MCF** e **OPB**, apresentou um *gap* melhor ou igual para todos os problemas-teste. Logo, ela pode ser dominante em relação à **MCF** e **OPB**. A formulação **OPB** obteve um *gap* melhor ou igual para todas as instâncias se comparada com **MCF**. Isto pode indicar que, para o PCTSP, **OPB** pode ser dominante em relação a **MCF**. As formulações **GSEC** e **PSC** fornecem limites de relaxação linear muito piores que as demais e são incomparáveis entre si. Ou seja, para algumas instâncias o *gap* de **PSC** é melhor do que **GSEC**, para outras **GSEC** é melhor. Neste caso, não há relação de dominância entre elas.

### 3.3 Resultados da abordagem Lagrangeana

Agora são relatados os resultados detalhados obtidos com a abordagem Lagrangeana proposta na Seção 2.5 por meio do Algoritmo 4 para o PCTSP.

Para realizar os experimentos computacionais foram adotadas as mesmas 252 instâncias apresentadas na Seção 3.1. Repare que os grafos de `br17`, `kro124`, `ry48p`, `p43`, `rbg323`, `rbg358`, `rbg403` e `rbg443`, pertencentes ao grupo `realworld` e de todos os grafos do grupo `crane` não satisfazem a desigualdade triangular, portanto, a família de desigualdades *cost-cover* não foram aplicados nas instâncias com base nesses grafos. O Algoritmo 4 é executado até que o critério de parada do VOL na implementação fornecida seja atendida: ou  $\|v\| \leq 0,0001$ ; ou se  $\frac{|LB-UB|}{|LB|} < 0,0001$  ou se  $UB - LB < 1$ . O último critério é utilizado pois todos os custos e penalidades associados às instâncias utilizadas neste experimento são inteiros.

O Algoritmo Volume possui parâmetros que controlam seu comportamento. Esses parâmetros influenciam diretamente os resultados e a convergência do algoritmo. Portanto, uma fase de calibração é necessária. Os trabalhos de Barahona e Anbil (2000), Bahiense, Maculan e Sagastizábal (2002) e Frangioni, Gendron e Gorgone (2017) foram usados para definir o conjunto inicial de valores possíveis para os parâmetros além de novos valores. Para mais detalhes sobre esta etapa, veja o Apêndice D, no qual a calibração é devidamente detalhada. Várias combinações foram testadas, como nas



obras mencionadas acima. Por uma questão de clareza e concisão, os valores usados em comum por todas as instâncias para obter os resultados apresentados são:

- $grp = 0,0001$ ;  $pap = 0,0001$ ;
- $greentestinvl = 1$ ;  $lambdainit = 0,1$ ;  $alphainit = 0,1$ ;

Os demais parâmetros variam de acordo com a instância:

- se  $|V| < 70$ , exceto para as instâncias *crane*, *atex*, *ft53*, *p43*, e *ry48p*:  $redtestinvl = 4$ ;  $yellowtestinvl = 2$ ;  $alphamin = 0,0001$ ;  $alphafactor = 0,9$ ;  $alphaint = 100$ .
- se *atex*:  $redtestinvl = 4$ ;  $yellowtestinvl = 2$ ;  $alphamin = 0,0001$ ;  $alphafactor = 0,5$ ;  $alphaint = 100$ .
- se  $|V| \geq 70$ , exceto para *crane* e *stilt*, e incluindo as instâncias *ft53*, *p43*, e *ry48p*:  $redtestinvl = 10$ ;  $yellowtestinvl = 2$ ;  $alphamin = 0,00001$ ;  $alphafactor = 0,03$ ;  $alphaint = 150$ .
- se *crane*:  $redtestinvl = 10$ ;  $yellowtestinvl = 10$ ;  $alphamin = 0,00001$ ;  $alphafactor = 0,5$ ;  $alphaint = 100$ .
- se *stilt*:  $redtestinvl = 10$ ;  $yellowtestinvl = 10$ ;  $alphamin = 0,00001$ ;  $alphafactor = 0,5$ ;  $alphaint = 150$ .

Ainda, o Algoritmo 4 utiliza o parâmetro  $h_{it} = 100$ . Ou seja, a cada 100 iterações o procedimento heurístico para solução viável é chamado se um novo conjunto  $H$  (conjunto de variáveis primais de interesse) é obtido dentro deste bloco de iterações.

As Tabelas 6, 8 e 10 mostram, respectivamente, os resultados para as instâncias dos grupos *realworld*, *crane* e *stilt*. Essas tabelas apresentam os resultados para as instâncias nas quais **EPN** foi capaz de resolver a relaxação linear dentro do limite de tempo de 10.800 segundos.

A primeira coluna identifica a instância. O número associado com cada instância dá a cardinalidade de  $V$ , por exemplo,  $|V| = 17$  para *br17*, exceto para *atex*. As instâncias *atex* possuem, respectivamente, 16, 32, 48 e 72 vértices.

A segunda coluna (*gap UB*) apresenta o desvio relativo  $100 \times (UB - Opt)/Opt$  entre a melhor solução  $UB$  encontrada durante a execução do algoritmo e o valor da solução ótima  $Opt$ .

A terceira coluna (*gap LB*) apresenta o desvio relativo  $100 \times (Opt - LB)/Opt$  entre o  $LB$  retornado pela relaxação Lagrangeana e o valor ótimo  $Opt$ .

A quarta coluna (*gap LP*) apresenta o desvio relativo  $100 \times (LP - LB)/LP$  entre o *LB* retornado pela relaxação Lagrangeana e o valor da relaxação linear *LP* da formulação **EPN**. Portanto, um valor negativo nesta coluna significa que o valor da solução Lagrangeana é melhor do que o valor da solução da relaxação linear.

A quinta coluna (*gap Δ*) apresenta o desvio relativo  $100 \times (UB - LB)/UB$  entre o melhor *UB* encontrado durante o algoritmo e o *LB* retornado pela relaxação Lagrangeana. Ou seja, o intervalo no qual a solução ótima reside, calculado exclusivamente com os limites fornecidos pelo Algoritmo 4.

A sexta coluna apresenta o número de iterações do Algoritmo 4 até atingir o critério de parada.

As sétima e oitava colunas apresentam, respectivamente, o tempo total de execução em segundos, do Algoritmo 4 e do problema inteiro misto (MIP), resolvido por meio de **PSC** pelo Gurobi, sem limite de tempo para o grupo *realworld*, e com um limite de 10.800 seg. (3 horas) para *crane* e *stilt*.

Tabela 6 – Resultado do Algoritmo 4 para as 78 instâncias do grupo *realworld* nas quais a relaxação linear de **EPN** é conhecida.

Instância	<i>gap UB</i>	<i>gap LB</i>	<i>gap LP</i>	<i>gap Δ</i>	<i>it</i>	tempo Alg 4	MILP
$\tau = 20$							
br17	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	4181	2	0
ft53	1,20	2,21	1,57	3,37	4633	105	46
ft70	1,34	1,27	0,71	2,57	2708	286	63
ftv33	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	5679	19	1
ftv35	<b>0,00</b>	<b>0,00</b>	-0,49	<b>0,00</b>	7157	57	1
ftv38	<b>0,00</b>	0,59	-0,21	0,59	2583	47	5
ftv44	<b>0,00</b>	0,08	0,08	0,08	7906	76	2
ftv47	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	5130	203	3
ftv55	<b>0,00</b>	1,74	0,65	1,74	3491	599	8
ftv64	<b>0,00</b>	0,26	0,26	0,26	10456	316	14
ftv70	<b>0,00</b>	1,98	0,54	1,98	11608	607	15
ftv80	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	4638	387	723
ftv90	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	5993	400	825
ftv100	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	4167	424	974
ftv110	<b>0,00</b>	<b>0,00</b>	-0,05	<b>0,00</b>	6057	665	2132
kro124	2,20	4,06	1,26	6,13	16435	2498	136
p43	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	37210	517	13
ry48p	0,09	8,04	-0,69	8,13	4985	81	14
atex1	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	1655	41	0
atex3	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	1312	34	1
atex4	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	4058	212	95
atex5	0,33	1,19	0,02	1,52	7674	113	17
dc112	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	4429	306	412
dc126	1,25	1,15	1,10	2,36	1169	551	707
dc134	<b>0,00</b>	<b>0,00</b>	-0,03	<b>0,00</b>	3632	678	1099

*Continua na próxima página*

Tabela 6 – Continuação da página anterior

Instância	gap UB	gap LB	gap LP	gap $\Delta$	it	tempo Alg 4	MILP
td100_1	1,24	1,65	1,01	2,85	2356	452	556
Média	0,30	0,93	0,22	1,21	6589	372	302
$\tau = 50$							
br17	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	4181	1	0
ft53	1,47	2,97	1,28	4,38	7657	193	28
ft70	1,64	1,29	1,08	2,89	2708	148	38
ftv33	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	5270	40	1
ftv35	0,34	0,43	-0,06	0,77	7157	59	1
ftv38	<b>0,00</b>	0,59	-0,21	0,59	6395	52	3
ftv44	<b>0,00</b>	0,16	0,16	0,16	6652	83	2
ftv47	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	5852	223	3
ftv55	0,13	1,74	0,65	1,86	3580	394	9
ftv64	<b>0,00</b>	0,26	0,26	0,26	10277	472	12
ftv70	0,24	2,28	0,84	2,51	8924	483	13
ftv80	0,29	0,07	0,07	0,36	6831	379	734
ftv90	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	5062	436	827
ftv100	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	4167	464	995
ftv110	<b>0,00</b>	<b>0,00</b>	-0,05	<b>0,00</b>	6057	813	2105
kro124	2,15	2,50	0,97	4,55	9228	1484	206
p43	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	38524	581	17
ry48p	0,34	2,95	0,00	3,28	6109	121	8
atex1	0,4	14,58	-8,13	14,90	1983	35	2
atex3	1,0	3,59	-0,21	4,58	2412	33	3
atex4	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	4867	392	120
atex5	1,0	0,44	0,02	1,47	7759	227	35
dc112	<b>0,00</b>	<b>0,00</b>	-0,46	<b>0,00</b>	4720	389	803
dc126	1,3	1,85	1,35	3,06	2138	693	1075
dc134	<b>0,00</b>	<b>0,00</b>	-0,03	<b>0,00</b>	4704	673	1284
td100_1	1,1	1,17	0,80	2,22	2761	492	630
Média	0,44	1,42	-0,06	1,84	6768	360	344
$\tau = 80$							
br17	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	4181	1	1
ft53	1,25	1,39	0,96	2,60	1083	193	64
ft70	1,16	1,15	0,95	2,29	5466	57	56
ftv33	<b>0,00</b>	0,09	-1,30	0,09	5027	50	1
ftv35	<b>0,00</b>	0,67	-0,31	0,67	6570	73	2
ftv38	0,49	1,80	-0,02	2,29	6559	127	3
ftv44	<b>0,00</b>	1,60	0,02	1,60	6761	190	6
ftv47	0,07	1,45	0,00	1,52	9482	19	6
ftv55	0,06	1,74	0,65	1,80	5593	475	10
ftv64	0,13	<b>0,00</b>	-0,07	0,13	920	67	60
ftv70	<b>0,00</b>	0,87	-3,13	0,87	8548	397	101
ftv80	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	864	676	726
ftv90	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	3795	633	828
ftv100	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	5709	375	1026
ftv110	<b>0,00</b>	<b>0,00</b>	-0,05	<b>0,00</b>	6041	718	2295
kro124	2,28	2,13	1,15	4,31	772	217	537

Continua na próxima página

Tabela 6 – Continuação da página anterior

Instância	gap $UB$	gap $LB$	gap $LP$	gap $\Delta$	it	tempo Alg 4	MILP
p43	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	8992	118	6
ry48p	0,11	1,80	-0,16	1,90	675	19	13
atex1	0,5	1,87	-0,74	2,35	3014	63	1
atex3	1,1	1,08	-0,43	2,17	2566	58	50
atex4	1,3	4,39	-0,11	5,65	4317	317	412
atex5	1,2	3,13	0,02	4,23	8073	312	366
dc112	<b>0,00</b>	<b>0,00</b>	-0,27	<b>0,00</b>	5278	521	4484
dc126	1,3	1,73	1,56	2,97	2086	603	1127
dc134	0,4	0,17	0,07	0,56	5263	996	5578
td100_1	1,3	1,81	1,63	3,05	3291	576	988
Média	0,49	1,11	0,02	1,58	4651	302	721

Repare na Tabela 6, que mostra os resultados para o grupo `realworld`, o Algoritmo 4 consegue obter os valores ótimos para 43 de 78 instâncias, resultados destacados em negrito. A heurística mostra-se satisfatória, pois obtém soluções próximas do ótimo quando o ótimo não foi encontrado. Na verdade, apenas para 18 das 78 instâncias a coluna  $gap\ UB$  é maior que 1%, e seu valor máximo é de 2,28%.

O valor da relaxação Lagrangeana obtém  $LB$ 's apertados na maioria dos casos. De fato, o valor médio é de 1,15%, em 32 das 78 instâncias o valor de  $LB$  é igual à solução ótima e apenas para 6 instâncias o  $gap\ LB$  é maior do que 3%.

Em relação ao  $gap\ LP$ , o Algoritmo 4 retorna uma média de 0,06%. Os limites obtidos pela relaxação Lagrangeana mostram-se mais apertados que a relaxação linear para 24 de 78 instâncias, embora apenas para 3 instâncias a melhora é maior que 1%.

De maneira geral, o Algoritmo 4 reporta pequenos intervalos para a solução ótima calculados a partir de seus resultados,  $gap\ \Delta$ . O valor médio foi de 1,51%, tal que para 44 de 78 instâncias, é possível limitar os valores em um  $gap\ \Delta$  menor que 1%, e apenas para 4 casos, esse  $gap\ \Delta$  é maior do que 5%.

Resolver o PCTSP de forma exata por meio de **PSC** é mais rápido para a maioria das instâncias na Tabela 6. No entanto, pode-se ver que o Algoritmo 4 encontra soluções ótimas consideravelmente mais rápido do que executar **PSC** com o Gurobi, para instâncias maiores, tais como as `dc`, `td` e `ftv90`, `ftv100` e `ftv110` com os 3 valores de  $\tau$ . Observe ainda que para  $\tau = 80$ , o tempo médio do Algoritmo 4 foi menor do que o método exato. Também é possível observar que o tempo para executar o Algoritmo 4 não é diretamente proporcional ao número de iterações executadas. Note que mais iterações podem ser executadas em uma instância do que em outra em um menor tempo computacional. Isso ocorre porque, em alguns casos, o subproblema

da mochila  $LR^y$  retorna frequentemente o mesmo conjunto de vértices. Neste caso, o Concorde e a rotina heurística para solução viável não são chamados com tanta frequência, economizando tempo computacional.

Tabela 7 – Resultado do Algoritmo 4 para as 45 instâncias do grupo `realworld` nas quais a relaxação linear de **EPN** não é conhecida.

Instância	<i>gap UB</i>	<i>gap LB</i>	<i>gap Δ</i>	<i>it</i>	tempo Alg 4	MILP
$\tau = 20$						
ftv120	<b>0,00</b>	0,10	0,10	7356	1422	2519
ftv130	0,05	0,05	0,09	7356	1162	3179
ftv140	0,09	0,09	0,18	7792	1738	8262
ftv150	0,04	<b>0,00</b>	0,04	6453	1672	9342
ftv160	<b>0,00</b>	0,32	0,32	7321	2269	8199
ftv170	<b>0,00</b>	0,41	0,41	7905	3263	10164
rbg323	0,51	1,69	2,16	25512	17502	129954
rbg358	0,11	0,30	0,40	30215	16752	132963
rbg403	0,05	2,34	2,33	38112	22207	180595
rbg443	0,42	1,31	1,71	37034	29642	183949
dc176	1,2	0,36	1,54	5179	986	3408
dc188	0,8	0,86	1,67	7635	1692	3344
code198	2,3	1,78	4,01	8455	1961	3425
code253	2,5	1,68	4,04	9592	2593	4652
td316_10	0,6	0,61	1,17	6874	4657	5427
Média	0,58	0,79	1,34	14186	7301	45959
$\tau = 50$						
ftv120	<b>0,00</b>	0,10	0,10	6100	1030	3278
ftv130	0,05	0,05	0,09	6100	926	3154
ftv140	0,40	0,53	0,93	5989	2166	8231
ftv150	0,31	0,08	0,39	6886	1877	9124
ftv160	0,08	0,32	0,40	11180	3537	8192
ftv170	<b>0,00</b>	0,41	0,41	7905	2966	10800
rbg323	0,33	1,69	1,92	21247	19988	133136
rbg358	0,33	1,01	1,30	31325	16440	130826
rbg403	0,13	2,34	2,38	40234	22991	205795
rbg443	0,14	1,31	1,42	40519	31854	198349
dc176	1,2	0,36	1,54	9138	1502	3520
dc188	0,8	0,86	1,67	7905	1859	3114
code198	2,3	1,78	4,01	8993	2060	3418
code253	2,5	1,68	4,04	10589	2805	6437
td316_10	1,4	0,57	1,96	7853	4876	5104
Média	0,67	0,87	1,50	14798	7792	48832
$\tau = 80$						
ftv120	<b>0,00</b>	0,15	0,15	8664	1265	2735
ftv130	0,23	0,37	0,60	8664	1545	2973
ftv140	0,40	0,67	1,06	9283	1178	8386
ftv150	0,04	0,12	0,16	8397	1560	8192
ftv160	0,08	0,51	0,59	6827	2381	8208
ftv170	<b>0,00</b>	0,41	0,41	6968	2623	9912

*Continua na próxima página*

Tabela 7 – Continuação da página anterior

Instância	gap $UB$	gap $LB$	gap $\Delta$	$it$	tempo Alg 4	MILP
rbg323	0,26	1,44	1,67	24854	18887	147954
rbg358	0,17	0,30	0,40	35027	18806	154563
rbg403	0,11	1,85	1,91	41682	24102	229627
rbg443	0,14	1,31	1,42	41301	32059	234349
dc176	1,6	1,07	2,64	17521	1791	12538
dc188	1,0	0,89	1,86	9648	1976	3533
code198	2,3	1,78	4,01	9452	2158	3406
code253	2,8	1,60	4,26	11552	3763	40863
td316_10	2,0	0,75	2,71	9749	4973	8892
Média	0,74	0,88	1,59	16639	7938	58409

A Tabela 7 mostra os resultados para as 45 instâncias restantes do grupo `realworld`. Para estas instâncias **EPN** não foi capaz de resolver a relaxação linear dentro do limite de tempo. Portanto, ele tem as mesmas colunas da Tabela 6, exceto por *gap LP*. No geral, o Algoritmo 4 obteve um *gap UB* médio de 0,66%; um *gap  $\Delta$*  médio de 1,47%; e um tempo médio muito inferior se comparado com o método exato (resolver a formulação **PSC** por intermédio do Gurobi).

As instâncias identificadas por `rbg` são as maiores e, de longe, as mais difíceis do grupo `realworld`. O esforço computacional necessário para obter soluções ótimas para este grupo de instâncias por meio do método exato (**PSC**, com o Gurobi) é enorme. São necessários quase dois dias de execução (47,7 horas) em média. No pior caso de nossos experimentos computacionais, a instância `rbg443`  $\tau = 80$ , levou 65 horas de computação para obter o valor ótimo. Por outro lado, o Algoritmo 4 mostra ser uma boa alternativa para limitar a solução ótima em um tempo computacional moderado. O Algoritmo 4 obteve em média um *gap  $\Delta$*  de 2,53% sobre as instâncias `rbg`, levando 6,3 horas em média. Os *LB*'s retornados pela relaxação Lagrangeana são apertados mesmo para essas instâncias maiores e mais difíceis. Para 8 das 12 `rbg`, o *gap LB* é menor que 1%, e apenas para `rbg403`  $\tau = 80$  é maior que 2%.

O Algoritmo 4 foi muito bom para lidar com as instâncias `ftv`. Ele conseguiu obter soluções com um *gap UB* menor que 0,5% para todas as 18 instâncias neste grupo. Ele também foi capaz de encontrar a solução ótima para 7 delas. Ainda para as instâncias `ftv`, a relaxação Lagrangeana forneceu *LB*'s apertados com um *gap* menor que 1% em relação ao ótimo. Além disso, para todas as 18 instâncias neste grupo (por exemplo `ftv150`  $\tau = 20$ ), o *LB* é igual à solução ótima, mas a heurística não conseguiu gerar esta solução primal. Assim, somente com os resultados fornecidos pelo Algoritmo 4 é possível limitar a solução ótima em um intervalo menor que 1% para todas as instâncias `ftv`, exceto 1 (`ftv140`  $\tau = 80$ , *gap  $\Delta$*  = 1,06 %). Observe que

esses resultados foram obtidos com consideravelmente menos esforço computacional do que o necessário para obter as soluções ótimas com o método exato.

Tabela 8 – Resultado do Algoritmo 4 para as 39 instâncias do grupo `crane` nas quais a relaxação linear de `EPN` é conhecida.

Instância	gap <i>UB</i>	gap <i>LB</i>	gap <i>LP</i>	gap $\Delta$	<i>it</i>	tempo Alg 4	MILP
$\tau = 20$							
crane66_0	0,30	2,87	-0,84	3,15	3744	104	6
crane66_1	<b>0,00</b>	<b>0,00</b>	-2,17	<b>0,00</b>	3924	109	4
crane66_2	<b>0,00</b>	<b>0,00</b>	-1,44	<b>0,00</b>	3920	112	5
crane100_0	1,34	1,53	0,39	2,83	12321	333	21
crane100_1	0,39	0,92	0,59	1,30	13186	347	36
crane100_2	0,95	1,97	0,39	2,90	11618	314	367
crane100_3	0,40	0,80	0,38	1,19	11988	324	47
crane100_4	0,99	0,71	0,60	1,68	12502	329	57
crane100_5	1,05	0,85	0,63	1,88	12654	333	19
crane100_6	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	12210	330	17
crane100_7	<b>0,00</b>	<b>0,00</b>	0,00	<b>0,00</b>	11894	313	9
crane100_8	1,29	0,93	0,55	2,19	12358	334	28
crane100_9	0,97	1,21	0,78	2,16	11803	319	45
Média	0,55	0,91	-0,07	1,45	10122	272	51
$\tau = 50$							
crane66_0	<b>0,00</b>	2,45	-0,05	2,45	4824	134	27
crane66_1	1,52	0,52	-0,12	2,01	5460	156	14
crane66_2	0,09	0,74	0,00	0,83	5145	147	65
crane100_0	1,44	2,46	1,81	3,84	15542	409	401
crane100_1	0,59	2,01	1,51	2,58	15429	417	108
crane100_2	1,30	1,41	0,92	2,68	17252	454	325
crane100_3	1,46	1,36	1,10	2,78	16796	442	51
crane100_4	1,29	1,19	0,37	2,44	16465	445	3706
crane100_5	<b>0,00</b>	<b>0,00</b>	-0,08	<b>0,00</b>	15884	418	17
crane100_6	0,46	0,81	0,42	1,26	15651	423	21
crane100_7	<b>0,00</b>	<b>0,00</b>	-0,05	<b>0,00</b>	16758	441	7
crane100_8	1,43	0,42	0,18	1,82	16391	443	54
crane100_9	1,66	2,28	0,38	3,87	16948	446	457
Média	0,94	1,30	0,54	2,21	13555	363	404
$\tau = 80$							
crane66_0	0,49	1,17	-0,06	1,65	6545	187	8
crane66_1	0,77	3,18	1,20	3,91	6876	191	29
crane66_2	1,41	1,39	0,41	2,76	6545	187	18
crane100_0	1,59	0,89	0,48	2,44	19722	519	74
crane100_1	0,82	2,56	1,66	3,36	18981	513	53
crane100_2	1,61	2,72	-0,38	4,27	17784	468	176
crane100_3	1,69	0,83	0,51	2,48	18848	496	23
crane100_4	1,64	1,50	0,33	3,09	20254	533	668
crane100_5	0,49	2,39	1,63	2,87	19228	506	44
crane100_6	0,75	1,26	0,57	1,99	17612	476	18
crane100_7	0,89	0,78	0,53	1,66	18130	490	10

*Continua na próxima página*

Tabela 8 – Continuação da página anterior

Instância	<i>gap UB</i>	<i>gap LB</i>	<i>gap LP</i>	<i>gap Δ</i>	<i>it</i>	tempo Alg 4	MILP
crane100_8	1,49	0,01	-0,10	1,48	19000	500	15
crane100_9	1,87	1,28	0,68	3,09	18352	496	61
Média	1,29	1,47	0,52	2,72	16270	435	92

A Tabela 8, mostra os resultados para o grupo `crane`. Neste caso, o Algoritmo 4 obteve os valores ótimos para 6 das 39 instâncias, destacados em negrito. Ele também obteve soluções próximas do ótimo quando o mesmo não foi encontrado. Note que todas as 39 instâncias possuem o *gap UB* menor que 2%. Novamente, o valor da relaxação Lagrangeana obteve LB's apertados na maioria dos casos. O valor do *gap LB* médio é de 1,22%, porém, em apenas 6 das 39 instâncias o valor de LB é igual à solução ótima e apenas para 1 instância o *gap LB* é maior do que 3%. Em relação ao *gap LP*, têm-se uma média de 0,33%. Os limites obtidos pela relaxação Lagrangeana mostraram-se mais apertados que a relaxação linear para 11 das 39 instâncias. Quanto ao *gap Δ*, o valor médio foi de 2,12%, tal que para 7 instâncias, este valor é menor que 1%, porém, nenhuma instância apresentou *gap Δ* maior do que 5%.

Tabela 9 – Resultado do Algoritmo 4 para as 30 instâncias do grupo `crane` nas quais a relaxação linear de **EPN** não é conhecida.

Instância	<i>gap UB</i>	<i>gap LB</i>	<i>gap Δ</i>	<i>it</i>	tempo Algo 4	MILP	<i>gap GRB</i>
$\tau = 20$							
crane316_10	0,78	0,74	1,51	205694	5413	1595	0,00
crane316_11	0,67	0,59	1,25	228912	6024	10802	1,52
crane316_12	0,73	0,56	1,28	220476	5802	10800	1,90
crane316_13	1,12	0,05	1,16	213256	5612	7096	0,00
crane316_14	0,86	0,60	1,45	229900	6050	10800	5,80
crane316_15	0,94	0,77	1,69	220818	5811	10800	1,90
crane316_16	1,01	0,42	1,42	220096	5792	10800	3,79
crane316_17	0,93	0,29	1,21	221160	5820	10802	1,15
crane316_18	0,52	0,78	1,29	215042	5659	10800	2,02
crane316_19	0,56	0,68	1,23	223326	5877	7984	0,00
Média	0,81	0,55	1,35	219868	5786	9228	1,81
$\tau = 50$							
crane316_10	1,04	0,96	1,99	211698	5571	1511	0,00
crane316_11	0,94	0,23	1,15	230090	6055	10802	1,42
crane316_12	1,72	0,55	2,23	218576	5752	10800	2,55
crane316_13	1,58	0,84	2,39	231154	6083	10342	0,00
crane316_14	1,35	0,18	1,51	203756	5362	10802	1,30
crane316_15	1,44	0,08	1,50	208316	5482	10800	1,85
crane316_16	1,75	0,81	2,52	214548	5646	10800	5,57
crane316_17	1,13	1,17	2,28	228646	6017	10802	2,09

Continua na próxima página



Tabela 9 – Continuação da página anterior

Instância	gap $UB$	gap $LB$	gap $\Delta$	$it$	tempo Algo 4	MILP	gap GRB
crane316_18	0,81	0,31	1,10	202654	5333	10800	1,87
crane316_19	1,10	0,70	1,78	212344	5588	5068	0,00
Média	1,29	0,58	1,85	216178	5689	9253	1,67
$\tau = 80$							
crane316_10	1,15	0,89	2,01	211242	5559	10802	0,30
crane316_11	0,99	1,85	2,82	204136	5372	10800	1,72
crane316_12	1,80	0,17	1,94	207252	5454	10800	1,90
crane316_13	1,42	0,89	2,28	230014	6053	3023	0,00
crane316_14	1,49	1,69	3,13	214624	5648	10802	1,85
crane316_15	1,62	0,41	2,00	206872	5444	10800	2,15
crane316_16	1,89	0,86	2,70	220096	5792	10800	2,46
crane316_17	1,28	0,41	1,66	230052	6054	10801	5,45
crane316_18	0,92	0,31	1,22	230280	6060	10800	2,00
crane316_19	1,45	0,49	1,91	205238	5401	9237	0,00
Média	1,40	0,80	2,17	215981	5684	9866	1,78

A Tabela 9 mostra os resultados para as 30 instâncias, restantes do grupo *crane*, para as quais **EPN** não foi capaz de resolver a relaxação linear. Para as instâncias nas quais a solução ótima não é conhecida, é feita a substituição da solução ótima *opt*, pela melhor solução conhecida dentro do limite de 3 horas. Ela tem as mesmas colunas da Tabela 7 mais a coluna *gap GRB*. Esta última mostra o desvio relativo entre o *UB* e o *LB* obtido pelo método exato (**PSC** por intermédio do Gurobi) no final do limite de tempo de 10800 segundos (3 horas). Considerando os valores médios, ambos os métodos exato e heurístico alcançaram valores próximos, tal que o *gap GRB* médio é de 1,75% e o *gap  $\Delta$*  é de 1,79%. Entretanto, para metade das instâncias o *gap  $\Delta$*  é menor do que o *gap GRB*. Em relação ao tempo computacional, o Algoritmo 4 teve um desempenho superior ao método exato.

Tabela 10 – Resultado do Algoritmo 4 para as 30 instâncias do grupo *stilt* nas quais a relaxação linear de **EPN** é conhecida.

Instância	gap $UB$	gap $LB$	gap $LP$	gap $\Delta$	$it$	tempo Algo 4	MILP
$\tau = 20$							
stilt100_0	0,44	3,86	0,16	4,29	18537	501	165
stilt100_1	1,03	4,95	-2,46	5,92	20976	552	21562
stilt100_2	1,05	4,14	-0,19	5,13	21926	577	1121
stilt100_3	1,02	3,45	-5,70	4,43	21014	553	23291
stilt100_4	0,35	3,03	0,72	3,37	20558	541	197
stilt100_5	0,65	2,92	1,61	3,55	19418	511	21
stilt100_6	1,11	8,56	0,78	9,56	21470	565	4530
stilt100_7	0,31	5,83	0,61	6,12	19314	522	676

Continua na próxima página

Tabela 10 – Continuação da página anterior

Instância	gap $UB$	gap $LB$	gap $LP$	gap $\Delta$	it	tempo Algo 4	MIP
stilt100_8	0,48	6,58	1,44	7,02	21964	578	1959
stilt100_9	0,95	2,72	0,45	3,64	20976	552	299
Média	0,74	4,60	-0,26	5,30	20615	545	5382
$\tau = 50$							
stilt100_0	1,06	1,27	0,24	2,31	21497	581	420
stilt100_1	0,98	0,66	0,35	1,62	21867	591	974
stilt100_2	1,50	1,19	0,46	2,65	22792	616	268
stilt100_3	1,31	2,72	1,09	3,98	21571	583	280
stilt100_4	0,60	3,40	1,13	3,98	20202	546	746
stilt100_5	1,81	4,29	0,94	5,99	21386	578	1681
stilt100_6	1,18	3,09	1,05	4,22	24700	650	1380
stilt100_7	0,52	1,93	0,81	2,44	23104	608	810
stilt100_8	0,89	2,22	1,25	3,08	23199	627	644
stilt100_9	1,23	2,40	0,11	3,58	22755	615	797
Média	1,11	2,32	0,74	3,38	22307	600	800
$\tau = 80$							
stilt100_0	1,75	3,33	0,92	4,99	24013	649	961
stilt100_1	1,83	2,53	0,89	4,28	25916	682	369
stilt100_2	1,63	1,83	0,86	3,41	26603	719	623
stilt100_3	1,28	4,36	2,60	5,57	28934	782	272
stilt100_4	0,94	2,35	1,78	3,26	29488	776	166
stilt100_5	1,88	3,86	1,75	5,63	26524	698	168
stilt100_6	1,27	3,57	2,62	4,78	30818	811	634
stilt100_7	0,62	0,90	0,09	1,51	26159	707	557
stilt100_8	1,15	3,72	0,96	4,81	28157	761	604
stilt100_9	1,58	3,18	0,89	4,69	26942	709	160
Média	1,39	2,96	1,34	4,29	27355	729	451

A Tabela 10, mostra os resultados do grupo *stilt*, para as 30 instâncias nas quais a relaxação linear de **EPN** é conhecida. Neste caso, ao contrário dos outros grupos, o Algoritmo 4 não obteve os valores ótimos em nenhum dos casos. Entretanto, ele obteve soluções próximas do ótimo. De fato, todas as instâncias possuem o *gap UB* menor que 2%. Porém, o valor do *gap LB* médio é 3,29%, o qual é superior aos demais grupos. Em relação ao *gap LP*, têm-se uma média de 0,61%. Os limites obtidos pela relaxação Lagrangeana mostraram-se mais apertados que a relaxação linear para 3 das 30 instâncias. Quanto ao *gap Δ*, o valor médio foi de 4,12%, resultado pior que dos demais grupos. A coluna tempo **PSC** (MIP) é omitida, pois, todas as instâncias foram executadas até o limite de 3 horas.

Tabela 11 – Resultado do Algoritmo 4 para as 30 instâncias do grupo *stilt* nas quais a relaxação linear de **EPN** não é conhecida.

Instância	<i>gap UB</i>	<i>gap LB</i>	<i>gap Δ</i>	<i>it</i>	tempo Alg 4	<i>gap GRB</i>
$\tau = 20$						
stilt316_10	1,74	0,83	2,53	276678	7281	5,25
stilt316_11	1,16	1,44	2,57	262010	6895	5,52
stilt316_12	0,67	2,13	2,78	259806	6837	7,03
stilt316_13	1,61	2,11	3,66	269534	7093	4,26
stilt316_14	1,13	1,03	2,14	266418	7011	4,08
stilt316_15	1,01	0,25	1,25	272270	7165	6,05
stilt316_16	1,39	1,61	2,96	263264	6928	6,88
stilt316_17	1,59	0,95	2,50	266722	7019	5,82
stilt316_18	1,33	0,27	1,58	267634	7043	6,58
stilt316_19	0,88	3,99	4,83	269458	7091	2,61
Média	1,25	1,46	2,68	267379	7036	5,41
$\tau = 50$						
stilt316_10	–	–	2,47	265620	6990	–
stilt316_11	1,90	0,69	2,54	273714	7203	9,39
stilt316_12	0,00	2,21	2,21	261060	6870	10,75
stilt316_13	2,05	0,44	2,44	255702	6729	6,95
stilt316_14	1,63	1,74	3,31	270256	7112	7,66
stilt316_15	1,45	0,73	2,14	270864	7128	6,63
stilt316_16	1,87	0,93	2,74	264328	6956	9,76
stilt316_17	-0,01	2,47	2,46	266418	7011	12,76
stilt316_18	2,13	0,35	2,43	252092	6634	7,88
stilt316_19	-0,20	2,51	2,31	273638	7201	20,62
Média	1,20	1,34	2,51	265341	6983	10,27
$\tau = 80$						
stilt316_10	2,66	0,61	3,18	264214	6953	9,25
stilt316_11	0,00	3,58	3,58	258742	6809	14,92
stilt316_12	0,05	2,85	2,90	276184	7268	10,86
stilt316_13	-0,03	6,06	6,03	267026	7027	15,91
stilt316_14	1,86	0,80	2,61	276754	7283	7,25
stilt316_15	2,26	0,15	2,36	264632	6964	0,72
stilt316_16	1,98	3,73	5,60	266152	7004	12,75
stilt316_17	1,25	2,33	3,54	267596	7042	11,98
stilt316_18	-0,02	3,69	3,67	260870	6865	16,70
stilt316_19	-25,91	28,28	3,21	274816	7232	93,33
Média	-1,59	5,21	3,67	267699	7045	19,37

A Tabela 11 apresenta os resultados para as 30 instâncias restantes do grupo *stilt* para as quais **EPN** não foi capaz de resolver a relaxação linear. Ela tem as mesmas colunas da Tabela 8, incluindo a coluna *gap GRB*. Assim como para *crane*, as instâncias nas quais a solução ótima não é conhecida, a solução *opt* é substituída pela melhor solução conhecida. Repare que para a instância *stilt316\_10*,  $\tau = 50$

nenhuma solução viável  $opt$  foi encontrada, logo os resultados das colunas  $gap UB$ ,  $gap LB$  e  $gap GRB$  são indicados por “-”. O Algoritmo 4 obteve um  $gap \Delta$  médio de 2,95%, enquanto o método exato apresentou o  $gap GRB$  médio de 11,68%. Ou seja, a abordagem Lagrangeana teve um resultado muito superior ao método exato para este grupo de instâncias. Na verdade, o  $gap \Delta$  é menor do que o  $gap GRB$  em quase todas as instâncias, exceto em duas:  $stilt316\_19$ ,  $\tau = 20$ ; e  $stilt316\_15$ ,  $\tau = 80$ . Ainda, o Algoritmo 4 encontrou soluções melhores do que o método exato em 4 instâncias. Em relação ao tempo computacional, o Algoritmo 4 teve um desempenho superior ao método exato, uma vez que ele obteve um intervalo para a solução ótima mais apertado e em menor tempo.

Para o grupo `crane`, em especial, as instâncias maiores e mais difíceis do grupo ( $|V| = 316$ ), o método exato só encontrou a solução ótima dentro do limite de 3 horas, em 8 casos. Apesar do Algoritmo 4 não encontrar nenhuma solução ótima, ele também se apresenta como uma boa alternativa para limitar a solução ótima em um tempo computacional moderado. A saber, ele apresentou um  $gap \Delta$  médio menor que 2% com um tempo médio de execução de 1,5 hora.

Para o grupo `stilt`, o método exato (**PSC MIP**) encontrou o seu pior desempenho médio. Ele não foi capaz de resolver, até a otimalidade, nenhuma das instâncias com 316 vértices. Ainda, ele não encontrou nenhuma solução viável para uma das instâncias e reportou um  $gap GRB$  de 93% para  $stilt316\_19$   $\tau = 80$ . Entretanto, o Algoritmo 4 obteve um desempenho semelhante aos demais grupos. Tal fato confirma a sua eficiência para limitar a solução ótima em um tempo computacional razoável.

### 3.4 Resultados das famílias de cortes de otimalidade e desigualdades válidas

A seguir são apresentados os resultados para o impacto dos cortes de otimalidade e das desigualdades válidas, apresentados na Seção 2.4.3, no resultado final do Algoritmo 4.

A Tabela 12 mostra a influência desses para o PCTSP. Nela, são mostrados os desvios relativos à solução ótima,  $gap\Delta$ , calculado segundo a Equação (3.2):

$$gap\Delta = 100 \times (UB - opt)/opt \quad (3.2)$$

na qual,  $UB$  é a solução obtida pelo Algoritmo 4 e  $opt$  a solução ótima (ou a melhor solução conhecida). Além disso, são mostrados o número de iterações. Na primeira coluna, são apresentadas as instâncias do grupo `realworld`. Note que foram escolhidas as que mostraram-se mais difíceis para o alg. proposto. Nas quatro colunas seguintes, são apresentados os resultados para os casos nos quais: ambas as famílias foram utilizadas; apenas uma família; e sem as famílias; tal que:

- $\Delta_{cc}$  indica o caso no qual ambas as famílias foram utilizadas (versão final);
- $\Delta_{co}$  indica o caso no qual apenas a família *Cost Cover*;
- $\Delta_{cy}$  indica o caso no qual apenas a família *Cycle Cover*;
- $\Delta_{no}$  indica o caso no qual nenhuma família é utilizada.

As demais colunas, apresentam o número de iterações para cada caso citado, da mesma forma que é apresentado o  $gap\Delta$ . Os valores destacados com negrito indicam o melhor resultado.

Tabela 12 – Desigualdades válidas do Algoritmo 4 – PCTSP

Instância	$gap\Delta$				Iterações			
	$\Delta_{cc}$	$\Delta_{co}$	$\Delta_{cy}$	$\Delta_{no}$	$\Delta_{cc}$	$\Delta_{co}$	$\Delta_{cy}$	$\Delta_{no}$
$\tau = 20$								
ft53	<b>3,37</b>	9,42	13,34	18,38	<b>4633</b>	6708	<b>4633</b>	9095
ft70	<b>2,57</b>	2,57	2,57	12,31	<b>2708</b>	<b>2708</b>	2944	8033
ftv170	<b>0,41</b>	7,86	7,37	18,31	<b>7905</b>	7918	8092	9705
rbg323	-	-	<b>2,16</b>	3,33	-	-	<b>25512</b>	27638
rbg358	-	-	<b>0,40</b>	0,51	-	-	<b>30215</b>	33738
rbg403	-	-	<b>2,33</b>	2,47	-	-	<b>38112</b>	42539
rbg443	-	-	<b>1,71</b>	1,79	-	-	<b>37034</b>	41993
Valor Médio	2,12	6,62	4,27	16,34	5082	5778	20935	24677
$\tau = 50$								
ft53	<b>4,38</b>	12,09	12,01	16,56	<b>7657</b>	8705	<b>7657</b>	9174
ft70	<b>2,89</b>	10,65	10,65	17,45	<b>2708</b>	2911	2765	4696
ftv170	<b>0,41</b>	0,41	1,47	17,96	<b>7905</b>	7918	8092	8832
rbg323	-	-	<b>1,92</b>	3,16	-	-	<b>21247</b>	26755
rbg358	-	-	<b>1,30</b>	1,71	-	-	<b>31325</b>	32891
rbg403	-	-	<b>2,38</b>	2,47	-	-	<b>40234</b>	44874
rbg443	-	-	<b>1,42</b>	1,42	-	-	<b>40519</b>	43522
Valor Médio	2,56	7,72	4,45	17,32	6090	6511	21691	24392
$\tau = 80$								
ft53	<b>2,60</b>	7,62	7,32	18,67	<b>1083</b>	1919	<b>1083</b>	2119
ft70	<b>2,29</b>	12,61	11,08	17,47	<b>5466</b>	7797	<b>5466</b>	9341
ftv170	<b>0,41</b>	0,41	0,41	16,35	<b>6968</b>	7019	7007	8839
rbg323	-	-	<b>1,67</b>	3,01	-	-	<b>24854</b>	27131
rbg358	-	-	<b>0,40</b>	1,11	-	-	<b>35027</b>	38689
rbg403	-	-	<b>1,91</b>	5,45	-	-	<b>41682</b>	45791
rbg443	-	-	<b>1,42</b>	1,50	-	-	<b>41301</b>	43824
Valor Médio	1,77	6,88	3,46	9,08	4506	5578	22346	25105

Pode-se observar que o uso de cortes de otimalidade e/ou desigualdades válidas melhorou o desempenho do Algoritmo 4. Cada família foi individualmente útil

para melhorar os resultados. De fato, a redução média do  $gap\Delta$  para  $rgb$  foi, aproximadamente, de 17% para 7%; para as demais a redução foi ainda maior, aproximadamente, de 17% para 2%. Lembre-se que a família *Cost Cover* só é válida para o PCTSP quando a instância satisfaz a propriedade da desigualdade triangular (Seção 2.4.3.1). Logo, para as 12 instâncias  $rgb$ , que não satisfazem esta propriedade, as colunas  $\Delta_{cc}$  e  $\Delta_{co}$  são indicadas por “-”. Além do valor da solução  $UB$ , a convergência do algoritmo também é aperfeiçoada, pois o número de iterações necessário foi sempre menor para a versão do algoritmo com o uso das mesmas.

## 4 Formulações e Relaxação Lagrangeana para o Problema de Dimensionamento e Sequenciamento de Lotes

O bom desempenho da abordagem proposta para o PCTSP habilita a sua aplicação aos demais problemas que pertencem à categoria dos Problemas de Sub-Rotas, ou seja, deseja-se verificar se o desempenho da abordagem proposta também será satisfatório para outros problemas correlatos.

Como citado anteriormente, o PCTSP possui diversos problemas correlatos, dos quais optou-se por escolher o Problema de Dimensionamento e Sequenciamento de Lotes em uma máquina – LSSP. Este problema foi escolhido porque é amplamente estudado e possui diversas aplicações na indústria, como por exemplo: na indústria têxtil, Silva e Magalhaes (2006); embalagens, Marinelli, Nenni e Sforza (2007); recipientes de vidro, Almada-Lobo, Oliveira e Carravilla (2008); ração animal, Toso, Morabito e Clark (2009); indústria madeireira, Gaudreault et al. (2011); bebidas, Ferreira, Morabito e Rangel (2010), Baldo et al. (2014) e Toscano, Ferreira e Morabito (2020); embalagens moldadas de celulose, Martínez, Toso e Morabito (2016); setor automotivo Diaz-Madroñero et al. (2018); e injetora de plástico, Klement et al. (2021).

Neste trabalho, optou-se por considerar a versão do LSSP com os tempos de ajuste das máquinas dependentes da sequência de produção. Neste caso, o LSSP também é conhecido na literatura como *Lot Sizing and Sequence Problem*, como pode ser observado em Toso, Morabito e Clark (2009) e Perraudat, Dauzère-Pérès e Mason (2021).

Logo, o LSSP é apresentado, assim como os detalhes da versão considerada. Em seguida, os principais trabalhos da literatura que tratam do problema são apresentados. Na sequência, ele é formalmente definido e as principais formulações para este problema são mostradas. A primeira é proposta por Almada-Lobo et al. (2007) e a segunda por Guimarães, Klabjan e Almada-Lobo (2014). Como observado neste último trabalho, ambas estão entre as principais formulações da literatura para a versão do LSSP considerada. Assim como para o PCTSP, também é apresentada a relaxação Lagrangeana da formulação de maior destaque, bem como as famílias de desigualdades válidas utilizadas. A adaptação para o LSSP da abordagem proposta é, então,

apresentada.

## 4.1 Definição do problema

O LSSP pode ser representado por um conjunto  $I$  de produtos e um conjunto  $T$  de períodos. O conjunto  $I = \{1, 2, \dots, n\}$  é formado por  $n$  produtos e  $T = \{1, 2, \dots, |T|\}$  é formado pelos períodos  $t$  no qual o horizonte de planejamento é dividido. Este problema é muito importante no processo de planejamento e controle da produção e consiste em determinar a programação da produção. Ou seja, determinar a quantidade  $q_i^t$  produzida de cada produto  $i$  (tamanho do lote) para cada período  $t$  e a seqüência na qual os produtos serão produzidos, dentro de cada período (ordens de produção).

Dado:  $c_{ij}$  o custo incorrido para alterar a configuração da máquina do produto  $i$  para o produto  $j$ ; e  $w_i$  o custo unitário de manutenção do estoque do produto  $i$  de um período para o outro. O objetivo é determinar a programação da produção, na qual o somatório destes custos envolvidos seja mínimo. Este programa deve atender a demanda  $d_i^t$  de cada produto  $i$  e respeitar a capacidade produtiva  $C_t$  dos recursos compartilhados pelos produtos para todo período  $t$ .

Em relação às características dos recursos produtivos, considera-se uma máquina única com capacidade limitada de produção, para cada período de tempo. Os custos de ajuste desta máquina dependem da seqüência de produção dos itens e são considerados assimétricos. Ou seja, o custo  $c_{ij}$  do ajuste da máquina do produto  $i$  para o  $j$  é diferente do custo do ajuste do produto  $j$  para o  $i$  ( $c_{ij} \neq c_{ji}$ ). Esta diferença é muito comum em diferentes tipos de processos produtivos como recipientes de vidro, Almada-Lobo et al. (2007), e nutrição animal, Toso, Morabito e Clark (2009).

Além disso, é considerado o caso monoestágio. Ou seja, é considerado que os produtos são produzidos em um único meio de produção compartilhado, em apenas uma única operação. Entretanto, a produção de um produto é independente de outro. Isto significa que não há relação de precedência entre os itens produzidos. O problema multiestágio é caracterizado pela existência de produtos predecessores de um produto final. Ou seja, é preciso fabricar um componente do produto final, antes de produzi-lo.

Outra característica considerada é que o ajuste da máquina (*setup*), no final do período, é mantido para o período seguinte (*carryover*). Isto significa que, dado um período  $t$ , a configuração de máquina para o último produto  $i$ , produzido em  $t$ , é mantida caso se inicie o período  $t + 1$  com a produção de  $i$ . Porkka, Vepsäläinen e Kuula (2003) afirmam que diferentes trabalhos demonstram que esta restrição é capaz de reduzir o número de vezes que o *setup* ocorre e incrementar a capacidade de produção significamente.



Para Menezes, Clark e Almada-Lobo (2011), esta estratégia é muito útil para processos de capacidade restrita se comparada à demanda. Dessa forma é possível aumentar a taxa de utilização dos recursos e, conseqüentemente, reduzir os custos de produção.

Além disso, considera-se que não é permitido que o *setup* seja realizado mais de uma vez, para o mesmo produto, dentro de um mesmo período. Ainda, também não é permitido que o *setup* comece em um período e termine no próximo. Ou seja, as operações de ajuste da máquina devem terminar dentro do mesmo período no qual começaram.

Na versão considerada, neste trabalho, não é permitida a entrega com atrasos (*backlog*), ou que a demanda seja parcialmente atendida, o que representa vendas perdidas. Ou seja, a demanda  $d_i^t$  do produto  $i$  é integralmente atendida no período  $t$ . Entretanto, é permitido antecipar a produção, isto implica nos custos de estocagem do produto cuja produção é adiantada de um período para outro. Note que os pressupostos apresentados nesta seção, que caracterizam a versão do LSSP considerada, são, usualmente, consideradas na literatura, ver por exemplo Guimarães, Klabjan e Almada-Lobo (2014).

## 4.2 Revisão da literatura

Rogers (1958), provavelmente, é o primeiro trabalho que aborda ambos os problemas, determinar o tamanho do lote econômico e sequenciar as ordens de produção, de forma simultânea. Tradicionalmente, primeiro era determinado os lotes de produção e, em seguida, o problema de sequenciamento era resolvido. O trabalho descreve uma abordagem computacional para encontrar um programa de produção viável e de custo mínimo em um único centro produtivo.

Fleischmann (1990) abordam o LSSP por meio do algoritmo *branch-and-bound* baseado na relaxação Lagrangeana para determinar os limites inferiores e as soluções viáveis. Os problemas Lagrangeanos são resolvidos por programação dinâmica. Os resultados computacionais são relatados para vários exemplos da literatura com até 12 produtos e 122 períodos ou 3 produtos e 250 períodos. O algoritmo produz soluções ótimas ou pelo menos soluções viáveis com limites inferiores apertados em alguns minutos. Os autores comparam os resultados com o problema clássico de dimensionamento de lotes capacitados.

Drex1 e Kimms (1997) procuram explicar as diferenças dos principais modelos e fornecer algumas recomendações de leitura para um entendimento inicial do problema. Eles focam no caso capacitado, dinâmico e determinístico do LSSP. Modelos de programação matemática que consideram o horizonte de planejamento subdividido

em vários períodos discretos são apresentados para esta versão. O artigo conclui com algumas sugestões para futuras atividades de pesquisa.

Meyr (2000) utiliza um algoritmo dual associado às meta-heurísticas *Threshold Accepting* (TA) e *Simulated Annealing* (SA), respectivamente propostas por Dueck e Scheuer (1990) e Kirkpatrick, Gelatt e Vecchi (1983). O algoritmo é composto de duas etapas. Na primeira, as meta-heurísticas TA ou SA são utilizadas para gerar sequências de produção. Em seguida, um procedimento de otimização determina o tamanho do lote e os custos de armazenagem associados a essas soluções. Esse algoritmo é baseado no problema de fluxo máximo com custo mínimo. Ele utiliza um método de busca que elimina soluções com base na resolução do problema dual associado.

Haase e Kimms (2000) propõem uma formulação inteira mista para o problema. Esta é baseada nas “sequências eficientes” que são pré determinadas e escolhidas para cada período. O problema é então resolvido com o uso de um algoritmo do tipo *branch-and-bound* para problemas que variam de 3 itens e 15 períodos a 10 itens e 3 períodos.

Gupta e Magnusson (2005) tratam de uma fábrica de lixas e abrasivos. Eles utilizam uma heurística que parte de uma solução inicial gulosa. Para cada período, uma sequência é construída de forma gulosa, de acordo com os custos de ajustes das máquinas. A cada iteração, uma busca local é realizada a partir do último período. Ela busca por capacidade produtiva ociosa nos períodos anteriores e verifica se antecipar a produção gera soluções melhores. De acordo com os autores, a heurística gera bons resultados em um tempo hábil.

Almada-Lobo et al. (2007) propuseram duas novas formulações para o LSSP, denominadas AL1 e AL2. Ambas as formulações incorporam todos os recursos necessários da configuração de transição (*carryover*) e não permitem entrega com atraso (*backlog*). A formulação AL1 possui um número polinomial de restrições e AL2 um número exponencial de restrições, as quais podem ser separadas em tempo polinomial. Os autores comparam a força da relaxação Linear e também propõem novas desigualdades válidas. Estas desigualdades válidas são utilizadas para tornar ambas as formulações mais fortes. Os experimentos demonstraram que a formulação AL2 é mais forte que AL1. Os autores também apresentam uma heurística de cinco etapas que é eficaz tanto para encontrar uma solução viável (mesmo para instâncias fortemente capacitadas) quanto para produzir boas soluções para esses problemas.

Araujo, Arenales e Clark (2008) abordam o LSSP incluindo a possibilidade de atrasos na entrega do produto (*backorder*) em uma pequena siderúrgica. Uma formulação de programação inteira mista (MIP) do problema é proposta. Porém, para instâncias maiores, ela é ineficiente para resolver dentro de um tempo computacional razoável. Como resultado, uma abordagem *relax-and-fix* mais eficiente é desenvolvida.

Ainda, três variantes de um método de busca local também são desenvolvidos e testados por meio de instâncias baseadas na literatura. Os testes baseados na siderúrgica resultam em uma redução substancial dos atrasos na entrega, no estoque de produto acabado, no melhor uso da capacidade e na definição de um cronograma mais eficaz em comparação com a prática atual.

Ferreira, Morabito e Rangel (2010) tratam do problema de fabricação de refrigerantes em plantas de pequeno porte, no qual uma linha de produção é composta de dois estágios principais: a preparação do líquido e o engarrafamento. Esse problema é modelado como um problema com uma máquina e monoestágio, pois o engarrafamento é o gargalo produtivo. Propostas de algoritmos do tipo *relax-and-fix* são elaboradas por meio de diferentes variáveis de decisão. Experimentos realizados em instâncias reais mostram que a heurística de melhor desempenho associava a técnica *relax-and-fix* com as variáveis de estado de preparação de máquina.

James e Almada-Lobo (2011) resolvem o LSSP, por meio de duas metaheurísticas: Busca Tabu e VNS. A qualidade das soluções é medida por meio dos resultados obtidos em Almada-Lobo et al. (2007). Os autores concluem que a melhor metaheurística, para a instância testada, depende do tamanho do problema. Entretanto, se o limite de tempo for desconsiderado, o VNS apresenta melhor desempenho do que a Busca Tabu.

Menezes, Clark e Almada-Lobo (2011) também adotam uma nova formulação para o caso com *setup crossover*, ou seja, quando um ajuste de máquina pode começar em um período e terminar em outro. Eles comparam o desempenho de sua formulação com outras que não permitem tal ajuste. Dessa forma, é demonstrado que é possível reduzir os custos de produção, especialmente quando a capacidade é pequena se comparada com a demanda.

Amorim et al. (2013) abordam o LSSP para a indústria de bens de consumo com alto giro de estoques, caracterizada por sistemas de produção contínua que envolvem capacidades limitadas e complexos ajustes de máquinas. Este é o caso da produção de cerveja e laticínios, por exemplo. O trabalho tenta uma análise sistemática sobre as formulações da literatura. Uma reformulação foi proposta, o que, na maioria dos casos, supera o desempenho das formulações existentes para o conjunto de instâncias aleatórias.

Guimarães, Klabjan e Almada-Lobo (2013) desenvolvem um novo modelo matemático para o LSSP sem desigualdade triangular. A formulação explora a ideia de programar produtos com base na seleção de sequências de produção conhecidas. O modelo é a base de uma *matheuristic*, que incorpora princípios de *pricing* em heurísticas baseadas em MIP de construção e refinamento, denominada  $P\&MIP_{eval}$ . O desempenho dessa abordagem é comparada com outras heurísticas da literatura

por meio da instância de Almada-Lobo et al. (2007). Os resultados demonstram que  $P\&MIP_{eval}$  possui os melhores resultados, até agora, para a instância adotada, sendo o  $GAP$  médio máximo de 1,93%.

Guimarães, Klabjan e Almada-Lobo (2014) desenvolveram uma estrutura de classificação bidimensional para o LSSP. Dessa forma é possível um melhor estudo das principais abordagens de modelagem para o LSSP. Também foi apresentado uma nova formulação para o problema, que utiliza restrições de eliminação de subrotas baseadas no fluxo de uma única mercadoria – SCF. Os experimentos, que utilizam as instâncias Almada-Lobo et al. (2007), demonstram que o desempenho de SCF foi superior em relação às demais em termos de tempos de execução e soluções viáveis, tal que que 81 de 160 problemas-teste foram resolvidos até a otimalidade.

Oliveira e Santos (2017) propuseram uma reformulação simples para AL2 por meio da variável binária  $y'_{it}$ , a qual indica se o produto  $i$  será produzido no período  $t$ . Por meio desta reformulação é possível definir uma nova regra de ramificação *branching*. Os testes computacionais utilizaram a instância de Almada-Lobo et al. (2007). Através da priorização da variável  $y'$  foi possível reduzir significativamente o tempo de execução para resolver o LSSP por meio de um *solver* comercial. A saber, foi possível encontrar a solução ótima, em menos de uma hora, para 176 do 240 testados, *versus* AL2 que resolveu 157.

Copil et al. (2017) apresentam um revisão da literatura detalhada para o problema de dimensionamento de lotes e problemas correlatos, incluindo o LSSP. Como pode ser observado neste trabalho, métodos como decomposição Lagrangeana ou a geração de colunas foram aplicados com sucesso apenas para modelos baseados em micro períodos e dimensionamento discreto de lotes e sequenciamento – DLSP – e para modelos baseados na configuração contínua – CSLP. No grande maioria dos casos, as heurísticas são o principal método adotado para resolução do LSSP. Os métodos mais utilizados, baseados em MIP, adotam abordagens como *Fix and Optimize* e *Relax and Fix*. Estes métodos decompõem o problema original em subproblemas menos complexos que são sucessivamente resolvidos. No entanto, esses subproblemas, geralmente, não utilizam a formulação original. Eles apenas utilizam um número reduzido de variáveis inteiras envolvidas pela fixação ou relaxamento das mesmas. Assim, essas abordagens dependem fortemente do poder de solucionadores modernos de MIP.

Mahdieh, Clark e Bijari (2018) desenvolvem e testam modelos de programação inteira mista para custos sem a desigualdade triangular. Assim como em Guimarães, Klabjan e Almada-Lobo (2014), usam uma formulação baseada no ATSP que permite vários lotes de um produto em um período e impõe um tamanho mínimo de lote em vários períodos. Este novo modelo relaxa todas as limitações de separação física entre

os períodos. O modelo é desenvolvido primeiro para uma única máquina e depois estendido para outras configurações de máquina, incluindo máquinas paralelas e linhas de fluxo flexíveis. Testes computacionais demonstram a flexibilidade dos modelos propostos.

Vaez, Sabouhi e Jabalameli (2019) focam no LSSP bi-objetivo que maximiza o lucro total e minimiza as emissões de CO<sub>2</sub>. Além disso, um método de resolução **EPN**' *silon*-restrito é utilizado para transformar o modelo em mono-objetivo. O modelo proposto também considera as penalidades de antecipação/atraso com janela de tempo de entrega. Além disso, o modelo é testado em um estudo de caso de uma empresa siderúrgica e são apresentadas conclusões práticas e percepções gerenciais. De acordo com os autores, os resultados mostram que o modelo proposto é útil para os gestores no processo de planejamento de suas operações.

Toscano, Ferreira e Morabito (2020) trazem um novo modelo de programação inteira mista (MIP) para a produção de bebidas à base de frutas, laticínios, entre outros. O modelo envolve produção sincronizada em dois estágios, limpezas temporais obrigatórias e mudanças dependentes da programação. Este trabalho representa de forma adequada os processos de produção industrial, de sucos pasteurizados e de sucos com sabor de frutas, de diferentes empresas. Os autores combinam o MIP com heurísticas para resolver as instâncias inspiradas em uma grande empresa multinacional. Os testes computacionais mostram que eles são competitivos, quando comparados ao modelo e às soluções atuais da empresa.

Melega, Araujo e Morabito (2020) abordam a versão do LSSP integrada com o problema de corte de estoque, com dois estágios. No estágio um, uma máquina de corte é usada para cortar objetos grandes em pedaços menores. As peças cortadas, são utilizadas para montar os produtos finais na fase dois, na qual os produtos finais são produzidos para atender a demanda do cliente. Para resolver o problema integrado de dois estágios, os autores apresentam métodos de solução baseados em uma abordagem de *branch-and-price*. Um estudo computacional é conduzido com dados aleatórios e uma análise mostra o impacto dos métodos na solução. Além disso, o desempenho e os benefícios desta abordagem integrada são comparados por meio de uma simulação empírica da prática tradicional.

Ríos-Solís et al. (2020) aborda o LSSP em um cenário que envolve um ambiente de produção por injeção de plástico, no qual as peças são produzidas por meio de moldes. Cada peça pode ser processada em um conjunto de moldes com diferentes taxas de produção em diferentes máquinas. A taxa de produção varia de acordo com as atribuições da peça, do molde e da máquina. Uma heurística iterativa de dois estágios baseada em programação matemática é proposta. Primeiro, o tamanho do lote dos produtos é determinado junto das atribuições da máquina de moldagem.

A segunda etapa determina se existe uma sequência viável sem sobreposição. Para os autores, essa abordagem nos permite lidar com um ambiente mais complexo. Este incorpora tempos ociosos e considerações operacionais da linha de montagem. Os resultados experimentais mostram que essa heurística converge para uma solução viável com poucas iterações, obtendo soluções competitivas para a empresa tanto em termos de qualidade quanto em tempo de execução.

### 4.3 Formulações

Segundo Krarup e Bilde (1977), o problema de dimensionamento de lotes clássico pode ser reformulado como um problema de localização de facilidades – PLF. O PLF consiste em decidir o local onde instalar uma facilidade, de modo que as demandas sejam satisfeitas com custo mínimo. Normalmente, esse custo está associado à distância da demanda ao ponto onde a facilidade está localizada e à quantidade de facilidades instaladas. Em relação ao problema de dimensionamento de lotes, a facilidade representa a quantidade produzida de um item em um determinado período, que é destinada ao atendimento da demanda deste item no mesmo período ou em um período posterior, mediante o custo de estocagem. Esta reformulação pode ser vista em diversos trabalhos tais como Sürle e Stadtler (2003), Bernardes, Araujo e Rangel (2010).

Como é possível observar em Guimarães, Klabjan e Almada-Lobo (2014), esta técnica também pode ser aplicada ao LSSP. De acordo com os autores, por meio desta reformulação, é possível melhorar o desempenho das formulações. Tal fato foi confirmado por testes preliminares. Logo, este trabalho também adota esta reformulação. Assim, substituindo  $q_i^t$  (variável da formulação original) pela nova variável de decisão:

- $q_{lt}^i$ : a quantidade produzida do item  $i$ , no período  $l$ , para atender a demanda no período  $t$ .

E as demais variáveis de decisão:

- $s_i^t$ : estoque do item  $i$  ao final do período  $t$ .
- $x_{ij}^t$ : 1 se o item  $i$  é produzido antes do item  $j$  no período  $t$ , e 0 caso contrário.

E os seguintes parâmetros de entrada:

- $w_i$ : custo do estoque do item  $i$ .
- $d_i^t$ : demanda do item  $i$  no período  $t$ .

- $C_t$ : capacidade produtiva no período  $t$ .
- $a_i$ : tempo necessário para produzir uma unidade do item  $i$ .
- $c_{ij}$ : custo de *setup* para produzir o item  $j$  depois do item  $i$ .
- $f_{ij}$ : tempo de *setup* para produzir o item  $j$  depois do item  $i$ .
- $M_i^t$ : limite máximo de produção do item  $i$  no período  $t$ , tal que  $M_i^t = \min\{\frac{C_t}{a_i}, \sum_{u=t}^T d_i^u\}$ .

O LSSP, com a reformulação de localização de facilidades, é representado por (4.1)-(4.10) :

$$\min \sum_{t \in T} \sum_{i \in I} w_i s_i^t + \sum_{t \in T} \sum_{i \in I} \sum_{j \in I} c_{ij} x_{ij}^t \quad (4.1)$$

sujeito a:

$$\sum_{t=l}^{|T|} \sum_{i \in I} a_i q_{lt}^i + \sum_{i \in I} \sum_{j \in I \setminus \{i\}} f_{ij} x_{ij}^l \leq C_l \quad \forall l \in T \quad (4.2)$$

$$\sum_{l=1}^t q_{lt}^i = d_i^t \quad \forall i \in I, \forall t \in T \quad (4.3)$$

$$\sum_{l=1}^t \sum_{t'=t+1}^{|T|} q_{lt'}^i = s_i^t \quad \forall i \in I, \forall t = 1, \dots, (|T| - 1) \quad (4.4)$$

$$x_{ij}^t \in \{0, 1\} \quad \forall i, j \in I, \forall t \in T \quad (4.5)$$

$$s_i^t \geq 0 \quad \forall i \in I, \forall t \in T \quad (4.6)$$

$$q_{lt}^i \geq 0 \quad \forall t \in T, \forall l \in T, i \in I \quad (4.7)$$

$$q_{lt}^i \leq d_i^t \sum_{j \in I} x_{ji}^l \quad \forall i \in I, \forall l, t \in T : t \geq l \in T \quad (4.8)$$

$$\text{Preserve o estado da máquina (carryover)} \quad (4.9)$$

$$\text{Determine uma sequência viável de produção} \quad (4.10)$$

As restrições (4.3) asseguram que toda a demanda será contemplada sem atrasos. As restrições (4.4) determinam os níveis dos estoques como o excesso de produção do período. As restrições (4.2) garantem que a capacidade dos recursos produtivos é respeitada. As restrições (4.5)-(4.7) dizem respeito ao domínio das variáveis de decisão. As restrições (4.8) ativam o ajuste da máquina para todo item produzido. A restrição (4.9) determina que o estado da máquina no fim do período  $t$  e no início do período

posterior  $t + 1$  seja igual. A restrição (4.10) determina que, os itens produzidos devem ser ordenados.

Do mesmo modo que a diferença entre as formulações para o PCTSP está na forma como representar as restrições que eliminam as subrotas; as formulações do LSSP diferenciam-se, principalmente, na forma como é representado o sequenciamento das ordens de produção, e como o estado da máquina é preservado de um período para outro. Outra característica em comum, também apontada por Guimarães, Klabjan e Almada-Lobo (2014), está na possibilidade de aplicar as SEC's do ATSP para o LSSP. Entretanto, como as principais formulações do ATSP já foram estudadas no Capítulo 2, e observado o desempenho superior da formulação **EPN** sobre elas, as mesmas não são consideradas nesta seção.

### 4.3.1 Formulação **AL**

Almada-Lobo et al. (2007) propuseram duas novas formulações para o LSSP, denominadas como  $F_1$  e  $F_2$ . Ambas as formulações incorporam todas as especificações da versão do LSSP consideradas neste trabalho, tal como a preservação do estado da máquina. A formulação  $F_1$  possui um número polinomial de restrições e  $F_2$  um número exponencial de restrições, as quais podem ser separadas em tempo polinomial. Os autores demonstraram que a formulação  $F_2$  é mais forte que a formulação  $F_1$  e, nos experimentos computacionais apresentados em Almada-Lobo et al. (2007), foram, efetivamente, obtidos melhores limites de relaxação linear com a formulação  $F_2$ . Além disso, os resultados apresentados por Guimarães, Klabjan e Almada-Lobo (2014) mostram que a formulação  $F_2$  tem um desempenho melhor com a reformulação PLF. Logo, é apresentada apenas a formulação  $F_2$  de Almada-Lobo et al. (2007), com a reformulação PLF.

Seja a variável de decisão:

- $\alpha_i^t$ : 1 se a máquina está ajustada para o produto  $i$  no início do período  $t$ , e 0 caso contrário.

A restrição (4.8) pode ser substituída por:

$$q_{it}^i \leq d_i^t \left( \sum_{j \in I} x_{ji}^l + \alpha_i^l \right) \quad \forall i \in I, l, t \in T : t \geq l \quad (4.11)$$

Dado as restrições (4.12)-(4.17), originalmente apresentadas por Almada-Lobo et al.



(2007):

$$\sum_{i \in S} \sum_{j \notin S} x_{ij}^t + \sum_{i \in S} \alpha_i^{(t+1)} \geq \sum_{j \in I} x_{jk}^t \quad \forall t \in T, \forall k \in S, S \subset I \quad (4.12)$$

$$\sum_{i \in I} \alpha_i^t = 1 \quad \forall t \in T \quad (4.13)$$

$$\alpha_i^t + \sum_{j \in I} x_{ji}^t = \alpha_i^{(t+1)} + \sum_{j \in I} x_{ij}^t \quad \forall i \in I, \forall t \in T \quad (4.14)$$

$$\sum_{j \in I} x_{ji}^t \leq 1 \quad \forall i \in I, \forall t \in T \quad (4.15)$$

$$\sum_{j \in I} x_{ij}^t \leq 1 \quad \forall i \in I, \forall t \in T \quad (4.16)$$

$$\alpha_i^t \geq 0 \quad \forall i \in I, \forall t \in T \quad (4.17)$$

Das quais: as restrições (4.12)-(4.13) determinam o sequenciamento da produção e o respectivo estado da máquina; as restrições (4.14) asseguram que a o estado da máquina no fim do período  $t$  e no início do período  $t + 1$  seja o mesmo (*carryover*), ou seja, asseguram um o balanceamento do fluxo e carregam o estado de configuração da máquina para o próximo período, caso não seja realizada nenhum ajuste no período ( $x_{ij}^t = 0$ ), o estado de configuração da máquina é mantido para o próximo período, se houver um ajuste de entrada e nenhuma de saída para o produto  $i$  no período  $t$ , significa que esta configuração foi a última a ser realizada na máquina no período  $t$  e, conseqüentemente, a máquina é ajustada para o produto  $i$  no início do próximo período ( $\alpha_i^{t+1} = 1$ ), por outro lado, se houver um ajuste saída e nenhum de entrada, isto significa que a máquina é configurada para o produto  $i$  no início do período  $t$  ( $\alpha_i^t = 1$ ); as restrições (4.15) e (4.16) asseguram que se um produto for produzido no período  $t$ , apenas um ajuste pode ser feito; e as restrições (4.17) definem o domínio das variáveis de decisão.

A formulação **AL**, considerada neste trabalho, é composta por (4.1)-(4.7), (4.11), (4.12)-(4.17).

Como dito anteriormente, esta formulação possui um número exponencial de restrições. Logo, para viabilizar seu uso, ela precisa de um algoritmo de separação para tentar identificar as restrições violadas, o qual é descrito a seguir.

Considere uma solução  $(\bar{x}, \bar{\alpha}, \bar{q}, \bar{s})$  que satisfaça as restrições (4.2)-(4.4), (4.6), (4.7) e (4.11)-(4.17) e  $\bar{x}_{ij}^t \geq 0 \quad \forall t, i, j$ . O algoritmo de separação consiste na resolução do problema de fluxo máximo para cada produto  $i$  e para cada período  $t$ . Dado um período  $t$ , o grafo  $G_t = (I, A_t)$  no qual  $A_t = \{(i, j) : \bar{x}_{ij}^t > 0\}$ , com a capacidade no arco  $(i, j)$  definida como  $\bar{x}_{ij}^t$ . Adicione o vértice destino  $r$  ao grafo  $G_t$ , obtenha  $G_t^r$  e defina a capacidade dos arcos  $(i, r)$  como  $\bar{\alpha}_{i(t+1)}$ . Seja  $\bar{Z}_i$  o valor do fluxo máximo calculado entre  $i$  e  $r$ , no grafo  $G_t^r$  para  $i$ . Caso exista algum produto  $i$  tal

que  $\sum_{j \in I} x_{ji}^t > \bar{Z}_i$ , então a restrição (4.12) é violada e  $i \in S$ . Assim, o sub conjunto de itens  $S$  é composto pelos itens  $i$  que formam uma subrota. Já que o problema de fluxo máximo em um grafo direcionado pode ser resolvido por meio de um algoritmo polinomial, esse problema de separação também é polinomial. Para mais detalhes veja Almada-Lobo et al. (2007).

### 4.3.2 Formulação SCF

Guimarães, Klabjan e Almada-Lobo (2014) se aproveitam do fato de que o ATSP pode ser uma importante fonte de conteúdo para desenvolver bons modelos e métodos para o LSSP. Eles apresentam uma nova formulação para o problema inspirada na formulação proposta por Gavish e Graves (1978) para o ATSP. Esta formulação é baseada na já conhecida formulação de fluxo de mercadoria única. Os autores comparam o desempenho desta e de outras formulações, incluindo a proposta por Gupta e Magnusson (2005) e uma inspirada na MCF (Seção 2.3.2). A formulação denominada por SCF apresenta os melhores resultados em termos da resolução até a otimalidade. Ou seja, ela resolveu o maior número de problemas até a otimalidade dentro de um limite de tempo pré-estabelecido. .

Sejam as variáveis de decisão:

- $g_{ij}^t$ , a qual representa o fluxo de uma mercadoria (*commodity*) que passa pelo arco  $(i, j)$  no período  $t$ ;
- $y_i^t$  a qual indica se o item  $i$  é produzido no período  $t$ ;

e o vértice fictício  $r$ , o qual indica o estado inicial da máquina e desempenha a função de fonte (origem) do fluxo. Ele possui custos de *setup* e demanda iguais a zero. No início de cada período  $t$  o fluxo começa, e termina, com o vértice  $r$ . Uma vez que ele representa um produto artificial, não há custo ou perda de tempo envolvidos.

As subrotas desconexas do vértice  $r$  são eliminadas por meio das restrições (4.18)-(4.22). Dessa forma, ela garante a conectividade do grafo induzido pelas variáveis  $x$  diferentes de zero.

$$\sum_{j \in I} g_{rj}^t = \sum_{j \in I} y_j^t \quad \forall t \in T \quad (4.18)$$

$$\sum_{j \in I \cup \{r\}} g_{ji}^t = y_i^t + \sum_{j \in I} g_{ij}^t \quad \forall i \in I, \forall t \in T \quad (4.19)$$

$$y_i^t \in \{0, 1\} \quad \forall i \in I, \forall t \in T \quad (4.20)$$

$$g_{ri}^t \leq |I| \alpha_i^t \quad \forall i \in I, \forall t \in T \quad (4.21)$$

$$g_{ij}^t \leq |I| x_{ij}^t \quad \forall i \in I, \forall j \in I \setminus \{i\}, \forall t \in T \quad (4.22)$$

As restrições (4.18) forçam o fluxo da mercadoria para fora do vértice  $r$ . Repare que a quantidade total de fluxo que sai de  $r$  deve ser igual ao número de produtos produzidos no período. As restrições (4.19) asseguram o balanceamento do fluxo do vértice  $i$ . Dessa forma, é assegurado que um fluxo unitário é enviado para cada vértice ativo, o que corresponde a um caminho da origem  $r$  para cada produto produzido no período. As restrições (4.20) definem o domínio da variável de decisão. Ambas as restrições (4.21) e (4.22) impõem um limite superior na quantidade de fluxo que atravessa os arcos. Note que (4.22) também é responsável por ativar a variável  $x$ . Ou seja, ativar o ajuste na máquina. Além disso, as restrições (4.21) impõem que o fluxo só pode deixar o vértice raiz para o primeiro produto na rota, enquanto (4.22) garantem que o fluxo só atravessa arcos ativos na solução corrente.

A formulação **SCF** é formada por (4.1)-(4.7), (4.11), (4.13), (4.14) e (4.18)-(4.22).

### 4.3.3 Formulação **EPN'**

Esta formulação é inicialmente proposta para o PCTSP, na Seção 2.3.5. Logo, nesta seção, é apresentada a adaptação de **EPN** para o LSSP. Como dito anteriormente, para cada período  $t$ , o PCTSP surge como uma estrutura do LSSP somada à restrição do ajuste inicial da máquina *carryover*. Esta adaptação consiste na inclusão da dimensão relativa aos períodos na variável  $z$  e na retirada e adição de restrições.

Considere o produto fictício  $r$  tal que  $c_{ir} = c_{ri} = 0$ , assim como em **SCF**. Ou seja, ele representa o estado da máquina no início de cada período. Por meio da variável  $z_{ij}^{tk}, \forall t \in T, \forall i, j, k \in I, k \neq i \neq j$ , a qual indica se todos os três itens  $i, j$  e  $k$  são produzidos, e, conseqüentemente, o ajuste da máquina entre os produtos  $i$  e  $j$ , indicado pelo arco  $(i, j)$ , é realizado no período  $t$ . Esta variável também pode ser interpretada como uma indicação de que o item  $j$  será produzido imediatamente após  $i$  e ambos são produzidos antes do item  $k$  no período  $t$ . Entretanto, as variáveis  $z_{ik}^{tk}$  e  $z_{ii}^{tk}$ , tal que  $i \neq k$ , são interpretadas como:

- $z_{ik}^{tk}$  : assume 1 se, e somente se, para o período  $t$ , o item  $i$  é o primeiro item produzido no período  $t$  e o produto  $k$  não será produzido.
- $z_{ii}^{tk}$  : assume 1 se, e somente se, o item  $i$  é produzido, porém ele não é o primeiro e o item  $k$  precede  $i$  ou não é produzido no período  $t$ .

A partir das restrições (4.23)-(4.32) combinadas com as variáveis descritas acima têm-se:

$$q_{lt}^i \leq d_i^t y_i^l \quad \forall i \in I, \forall l \in T, t = l, \dots, |T| \quad (4.23)$$

$$x_{ir}^{t-1} = x_{ri}^t \quad \forall i \in I, t \in T \quad (4.24)$$

$$\sum_{j \in (I \cup \{r\}) \setminus \{i\}} x_{ij}^t = y_i^t \quad \forall i \in I \cup \{r\}, t \in T \quad (4.25)$$

$$\sum_{i \in (I \cup \{r\}) \setminus \{j\}} x_{ij}^t = y_j^t \quad \forall j \in I \cup \{r\}, t \in T \quad (4.26)$$

$$y_r^t = 1 \quad \forall t \in T \quad (4.27)$$

$$\sum_{i \in I \setminus \{r, k\}} z_{ij}^{tk} = y_j^t - x_{rj}^t \quad \forall t \in T, \forall j, k \in I \setminus \{r\}, k \neq j \quad (4.28)$$

$$\sum_{j \in I \setminus \{r\}} z_{ij}^{tk} = y_i^t - x_{ik}^t \quad \forall t \in T, \forall i, k \in I \setminus \{r\}, k \neq i \quad (4.29)$$

$$z_{ij}^{tk} \leq x_{ij}^t \quad \forall t \in T, \forall k, i, j \in I, k \neq j, k \neq i, i \neq j \quad (4.30)$$

$$y_i^t \in \{0, 1\} \quad \forall i \in I, \forall t \in T \quad (4.31)$$

$$z_{ij}^{tk} \geq 0 \quad \forall t \in T, \forall k, i, j \in I, k \neq i \quad (4.32)$$

As restrições (4.25) e (4.26) determinam que se o item  $i$  é produzido, ele será produzido apenas uma única vez por período. As restrições (4.27) garantem o estado da máquina no início de cada período. As restrições (4.30) e (4.32) definem o domínio das variáveis de decisão. As restrições (4.28) e (4.29) determinam o sequenciamento da produção para cada período por meio da descrição das relações de precedência e fluxo. Todas estas restrições já foram descritas anteriormente e fazem parte da formulação original, exceto pela adição da dimensão relativa aos períodos.

Porém, para garantir a viabilidade da solução, novas restrições precisam ser adicionadas à formulação. Para garantir que o estado da máquina no início do período seja igual ao último estado do período anterior, as restrições (4.24) são utilizadas. As restrições (4.23) asseguram e ativação da produção do item  $i$  no período  $t$  corrente.

Repare que as restrições (2.44), presentes na formulação original, proíbem a produção de um único produto  $i$  no período corrente. Logo, para uma formulação válida para o LSSP elas precisam ser omitidas. Além disso, as restrições (2.42) são redundantes, por isso elas também são desconsideradas nesta adaptação.

A formulação **EPN'** (a qual é uma adaptação de **EPN** para o LSSP) é representada por (4.1)-(4.7) e (4.23)-(4.32).

#### 4.3.4 Formulação **CSC**

Nesta seção é descrita a formulação inspirada na ideia de **PSC**, descrita na Seção 2.3.6. A formulação **PSC** descreve o fluxo ao longo da rota que representa o prêmio coletado ao visitar um vértice. Observe que ao adotar esta interpretação é possível eliminar a restrição de prêmio mínimo (restrição (2.7)). Ela, assim como a restrição de capacidade do LSSP (restrição (4.2)), pode ser vista como a restrição da mochila.

Porém, ao contrário do PCTSP, o LSSP não possui prêmio associado aos itens  $i$ , logo a formulação **PSC** não é válida. Neste caso, note que ao invés de considerar o fluxo como o prêmio coletado, correlacionado à restrição da mochila, ele pode ser considerado como a capacidade do recurso produtivo, presente no LSSP. Dessa forma, a restrição da mochila existente no LSSP pode ser omitida.

Para que isto seja possível considere a seguinte variável de decisão:

- $\tau_{ij}^t$ : representa o fluxo do arco  $(i, j)$  no período  $t$ .

e as restrições (4.33) e (4.34)

$$\sum_{i \in I \cup \{r\}} (\tau_{ij}^t + f_{ij} x_{ij}^t) + \sum_{u=t}^{|T|} a_j q_{tu}^j = \sum_{i \in I \cup \{r\}} \tau_{ji}^t \quad \forall j \in I, \forall t \in T \quad (4.33)$$

$$0 \leq \tau_{ij}^t \leq C_t x_{ij}^t \quad \forall i, j \in I \cup \{r\}, \quad \forall t \in T \quad (4.34)$$

Note que o fluxo, representado pela variável  $\tau_{ij}^t$ , pode ser interpretado como a parcela utilizada da capacidade disponível. Ou seja, a partir da restrição (4.33),  $\tau_{ij}^t$  pode ser entendida como o tempo gasto para realizar o ajuste da máquina do produto  $i$  para o produto  $j$  somado ao tempo necessário para produzir  $q$  unidades do produto  $j$ , para o período  $t$ .

Além disso, por meio de (4.34), o fluxo que é capaz de passar pelo arco  $(i, j)$  é, no máximo, igual à capacidade  $C$ . Logo, a restrição de capacidade (4.2) torna-se redundante e pode ser retirada da formulação. Como citado anteriormente, as formulações baseadas em fluxo, principalmente de mercadoria única, são boas para encontrar soluções viáveis.

A formulação nomeada como *Capacity-single commodity*, **CSC** (a qual é inspirada na ideia da formulação **PSC** proposta para o PCTSP) é composta por (4.1), (4.3)–(4.7), (4.23)–(4.27), (4.33) e (4.34).

#### 4.4 Relaxação Lagrangeana para o LSSP

A formulação **EPN'** é uma adaptação de **EPN** (proposta para o PCTSP), logo elas possuem as mesmas características desejáveis para a relaxação Lagrangeana. De fato, como será observado na Seção 5.2, a formulação **EPN'** (bem como **EPN**), obteve o melhor desempenho em termos da relaxação linear.

Por isso, nesta seção, é mostrado com detalhes a relaxação Lagrangeana da formulação **EPN'** para o LSSP, assim como o problema dual Lagrangeano decorrente da dualização das restrições complicantes. Também são apresentados os subproblemas Lagrangeanos resultantes e os respectivos procedimentos de resolução. Na sequência, são apresentados os procedimentos heurísticos solução inicial, busca local e desigualdades válidas; utilizados na adaptação do Algoritmo 4 apresentado na Seção 2.5, para resolver o LSSP.

A Tabela 13 mostra os multiplicadores de Lagrange e suas respectivas restrições complicantes da formulação **EPN'**. As restrições (4.3)–(4.7), (4.24), (4.27), (4.31) e (4.32) são deixadas para os subproblemas Lagrangeanos.

**Tabela 13 – Associação dos multiplicadores de Lagrange com as restrições da EPN'.**

Multiplicadores	Restrições
$\rho_t \in \mathbb{R}_+, \forall t \in T$	(4.2)
$\omega_{lt}^i \in \mathbb{R}_+, \forall i \in I, \forall l \in T, t = l, \dots,  T $	(4.23)
$\pi_i^t \in \mathbb{R}, \forall i \in I \cup \{r\}, t \in T$	(4.25)
$\varpi_j^t \in \mathbb{R}, \forall j \in I \cup \{r\}, t \in T$	(4.26)
$\lambda_{jk}^t \in \mathbb{R}, \forall t \in T, \forall j, k \in I, k \neq j$	(4.28)
$\mu_{ik}^t \in \mathbb{R}, \forall t \in T, \forall i, k \in I, k \neq i$	(4.29)
$\nu_{ij}^{tk} \in \mathbb{R}_+, \forall t \in T, \forall k, i, j \in I, k \neq j, k \neq i, i \neq j$	(4.30)

Repare que os multiplicadores de Lagrange  $\pi$ ,  $\varpi$ ,  $\lambda$  e  $\mu$ , correspondentes às restrições complicantes (4.25), (4.26), (4.28) e (4.29) possuem o domínio livre. Ou seja, dado que estas restrições são igualdades, eles podem assumir qualquer valor incluindo valores negativos, positivos ou zero. Os demais multiplicadores  $\rho$ ,  $\omega$  e  $\nu$  possuem o domínio positivo, pois têm-se um problema de minimização e as restrições do tipo menor ou igual ( $\leq$ ).

Dessa forma, o LSSP Lagrangeano pode ser definido por (4.35):

$$\begin{aligned}
 \min \sum_{t \in T} \sum_{i \in I} h_i s_{it} &+ \sum_{i \in I} \sum_{l \in T} \sum_{t=l}^{|T|} (\rho_l a_i + \omega_{lt}^i) q_{lt}^i & (4.35) \\
 &+ \sum_{t \in T} \sum_{j \in I} (\pi_r^t + \varpi_j^t + \sum_{k \in I \setminus \{j\}} \lambda_{jk}^t) x_{rj}^t + \sum_{t \in T} \sum_{i \in I} (\pi_i^t + \varpi_r^t) x_{ir}^t \\
 &+ \sum_{t \in T} \sum_{i \in I} \sum_{j \in I \setminus \{i\}} (c_{ij} + \rho_t f_{ij} + \pi_i^t + \varpi_j^t + \mu_{ij}^t - \sum_{k \in I \setminus \{i,j\}} \nu_{ij}^{tk}) x_{ij}^t \\
 &- \sum_{t \in T} \sum_{i \in I} \left( \pi_i^t + \varpi_i^t + \sum_{j \in I \setminus \{i\}} (\lambda_{ij}^t + \mu_{ij}^t) + \sum_{l=t}^{|T|} d_i^l \omega_{il}^i \right) y_i^t - \sum_{t \in T} (\pi_r^t + \varpi_r^t) y_r^t \\
 &+ \sum_{t \in T} \sum_{k \in I} \sum_{i \in I \setminus \{k\}} \sum_{j \in I \setminus \{k,i\}} (\lambda_{jk}^t + \mu_{ik}^t + \nu_{ij}^{tk}) z_{ij}^{tk} \\
 &+ \sum_{t \in T} \sum_{k \in I} \sum_{j \in I \setminus \{k\}} (\lambda_{jk}^t + \mu_{jk}^t) z_{jj}^{tk} + \sum_{t \in T} \sum_{k \in I} \sum_{i \in I \setminus \{k\}} \mu_{ik}^t z_{ik}^{tk} - \sum_{l \in T} \rho_l C_l
 \end{aligned}$$

sujeito a:

$$\begin{aligned}
 \sum_{l=1}^t q_{lt}^i &= d_{it} & \forall i \in I, \forall t \in T \\
 \sum_{l=1}^t \sum_{t'=t+1}^{|T|} q_{lt'}^i &= s_{it} & \forall i \in I, \forall t = 1, \dots, (|T| - 1) \\
 x_{ir}^{t-1} &= x_{ri}^t & \forall i \in I, t \in T \\
 y_{rt} &= 1 & \forall t \in T \\
 z_{ij}^{tk} &\geq 0 & \forall t \in T, \forall k, i, j \in I, k \neq i \\
 s_{it} &\geq 0 & \forall t \in T, \forall i \in \bar{I} \\
 q_{lt}^i &\geq 0 & \forall t \in T, \forall l \in T, \forall i \in \bar{I} \\
 y_{it} &\in \{0, 1\} & \forall i \in \bar{I}, \forall t \in T \\
 x_{ij}^t &\in \{0, 1\} & \forall i, j \in \bar{I}, \forall t \in T
 \end{aligned} \tag{4.36}$$

Uma vez que  $C(T) = \sum_{t \in T} \rho_t C_{it}$  é constante, ele será omitido. Observe que o vértice raiz é sempre igual a um (Restrição (4.27)). Assim, a parcela  $(\pi_r + \varpi_r)$  pode ser vista como uma constante. Bem como  $C(T)$ , ela também pode ser omitida e ambas subtraídas do valor obtido ao final da resolução do problema Lagrangeano.

Assim, têm-se os novos subproblemas Lagrangeanos  $LR^y$ ,  $LR^x$ ,  $LR^{x_r}$ ,  $LR^z$  e  $LR^{qs}$ .

O problema  $LR^y$  é definido pela FO, Eq. (4.37), sujeito às restrições de domínio binário para  $y$ :

$$\max \sum_{t \in T} \sum_{i \in I} \left( \pi_i^t + \varpi_i^t + \sum_{j \in I \setminus \{i\}} (\lambda_{ij}^t + \mu_{ij}^t) + \sum_{l=t}^{|T|} d_l^i \omega_{il}^i \right) y_i^t \quad (4.37)$$

s. a:

$$y_i \in \{0, 1\} \quad \forall i \in I$$

Seja  $rc[y_i^t]$ , Eq. (4.38):

$$rc[y_i^t] = \pi_i^t + \varpi_i^t + \sum_{j \in I} (\lambda_{ij}^t + \mu_{ij}^t) + \sum_{l=t}^{|T|} d_l^i \omega_{il}^i \quad \forall t \in T, i \in I \quad (4.38)$$

o custo reduzido de  $y_i^t$ . Para resolver  $LR^y$ , para cada período  $t$ , basta atribuir o valor 1 aos produtos com custo reduzido positivo. Ou seja, para todo  $t$  e todo  $i$  se  $rc[y_i^t] > 0$ , então  $y_i^t = 1$ , senão  $y_i^t = 0$ . Isto é equivalente a ativar a produção de  $i$  no período  $t$ .

O problema  $LR^x$  é definido por (4.39):

$$\min \sum_{t \in T} \sum_{i \in I} \sum_{j \in I \setminus \{i\}} \left( c_{ij} + \rho_t f_{ij} + \pi_i^t + \varpi_j^t + \mu_{ij}^t - \sum_{k \in I \setminus \{i, j\}} \nu_{ij}^{tk} \right) x_{ij}^t \quad (4.39)$$

s. a:

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in I, i \neq j$$

Seja  $rc[x_{ij}^t]$ , Eq. (4.40):

$$rc[x_{ij}^t] = c_{ij} + \rho_t f_{ij} + \pi_i^t + \varpi_j^t + \mu_{ij}^t - \sum_{k \in I} \nu_{ij}^{tk} \quad \forall t \in T \text{ e } i, j \in I | i \neq j \quad (4.40)$$

o custo reduzido de  $x_{ij}^t$ . Assim como  $LR^y$ , o subproblema  $LR^x$  também é resolvido por meio de uma simples inspeção. Para todo  $t \in T$  e todo  $i$  e  $j \in I : i \neq j$  se  $rc[x_{ij}^t] < 0$  então  $x_{ij}^t = 1$  senão  $x_{ij}^t = 0$ .

Note que  $c_{rj}$ ,  $c_{ir}$ ,  $f_{rj}$  e  $f_{ir}$  são sempre iguais a zero, portanto eles também podem ser omitidos. Dessa forma, o problema  $LR^{xr}$  é definido por (4.41):

$$\min \sum_{t \in T} \sum_{j \in I} \left( \pi_r^t + \varpi_j^t + \sum_{k \in I \setminus \{j\}} \lambda_{jk}^t \right) x_{rj}^t + \sum_{t \in T} \sum_{i \in I} (\pi_i^t + \varpi_r^t) x_{ir}^t$$

s. a:

$$\begin{aligned} x_{ir}^{t-1} &= x_{ri}^t & \forall i \in I, t \in T \setminus \{1\} \\ x_{rj}^t &\in \{0, 1\} & \forall j \in I, t \in T \\ x_{ir}^t &\in \{0, 1\} & \forall i \in I, t \in T \end{aligned} \quad (4.41)$$



Sejam, respectivamente,  $rc[x_{rj}^t]$  e  $rc[x_{ir}^t]$ , Equações (4.42) e (4.43):

- o custo reduzido de  $x_{rj}^t$  para  $\forall j \in I, t \in T$

$$rc[x_{rj}^t] = \pi_r^t + \varpi_j^t + \sum_{k \in I \setminus \{j\}} \lambda_{jk}^t \quad (4.42)$$

- o custo reduzido de  $x_{ir}^t$  para  $\forall i \in I, t \in T$

$$rc[x_{ir}^t] = \pi_i^t + \varpi_r^t \quad (4.43)$$

O subproblema  $LR^{x_r}$  é resolvido pelo Algoritmo 5.

---

**Algorithm 5** Subproblema  $LR^{x_r}$

---

```

1: for  $t \in T$  e  $i \in I$  do
2:   if  $t == 1$  then
3:     if  $(rc[x_{ri}^t] < 0)$  then
4:        $x_{ri}^t \leftarrow 1$ 
5:     else
6:        $x_{ri}^t \leftarrow 0$ 
7:     end if
8:   else if  $(t == |T|)$  then
9:     if  $(rc[x_{ir}^t] < 0)$  then
10:       $x_{ir}^t \leftarrow 1$ 
11:    else
12:       $x_{ir}^t \leftarrow 0$ 
13:    end if
14:   else
15:     if  $(rc[x_{ir}^{t-1}] + rc[x_{ri}^t] < 0)$  then
16:        $rc[x_{ir}^{t-1}] \leftarrow 1$ 
17:        $rc[x_{ri}^t] \leftarrow 1$ 
18:     else
19:        $rc[x_{ir}^{t-1}] \leftarrow 0$ 
20:        $rc[x_{ri}^t] \leftarrow 0$ 
21:     end if
22:   end if
23: end for

```

---

Observe a restrição que aborda o estado da máquina (*carryover*):

$$x_{ir}^{t-1} = x_{ri}^t, \forall i \in I, t \in T$$

note que ela assegura que o estado da máquina no início de cada período  $t$  seja igual ao final do período anterior  $t - 1$ . A partir dela, têm-se três cenários diferentes.

No primeiro, considere  $t = 1$ , ou seja, o primeiro período (Linha 2 do Alg. 5). Como não existe  $t = 0$ , a máquina pode ser ajustada para qualquer um dos produtos  $i$ . Logo, basta verificar se o custo reduzido de  $x_{ri}^1$  é negativo para algum  $i$ . Em caso afirmativo ele recebe o valor 1.

No segundo, considere  $t = |T|$ , ou seja, o último período (Linha 8 do Alg. 5). Como não existe  $t = |T| + 1$ , no fim do período  $|T|$  a máquina pode ser ajustada para qualquer produto ativo. Analogamente, basta verificar se o custo reduzido de  $x_{ir}^{|T|}$  é negativo para algum  $i$ .

Entretanto, para os demais cenários, ou seja,  $1 < t < |T|$  (Linha 14 do Alg. 5). O ajuste da máquina deve considerar, simultaneamente, os custos reduzidos do períodos  $t - 1$  e  $t$ . Dessa forma, se  $rc[x_{ir}^{t-1}] + rc[x_{ri}^t] < 0$  para algum  $i$ , então a máquina deve ser ajustada para o produto  $i$  no fim do período  $t - 1$  ( $x_{ir}^{t-1} = 1$ ) e no início do período  $t$  ( $x_{ri}^t = 1$ ).

O problema  $LR^z$  é definido por (4.44):

$$\begin{aligned} \min \quad & \sum_{t \in T} \sum_{k \in I} \sum_{i \in I \setminus \{k\}} \sum_{j \in I \setminus \{k, i\}} (\lambda_{jk}^t + \mu_{ik}^t + \nu_{ij}^{tk}) z_{ij}^{tk} \\ & + \sum_{t \in T} \sum_{k \in I} \sum_{j \in I \setminus \{k\}} (\lambda_{jk}^t + \mu_{jk}^t) z_{jj}^{tk} + \sum_{t \in T} \sum_{k \in I} \sum_{i \in I \setminus \{k\}} \mu_{ik}^t z_{ik}^{tk} \end{aligned} \quad (4.44)$$

s. a:

$$z_{ij}^{tk} \geq 0 \quad \forall t \in T \text{ e } k, i, j \in I : k \neq i$$

Bem como os demais anteriormente apresentados nesta seção, o subproblema  $LR^z$  é resolvido por meio de uma simples inspeção. Dado os três produtos  $i, j$  e  $k$ , pertencentes à  $I$ , os custos reduzidos das variáveis  $z$  são calculados de acordo com as Equações (4.45), (4.46) e (4.47) da seguinte forma:

- o custo reduzido de  $z_{ij}^{tk}$  para  $\forall t \in T$  e  $k, i, j \in I, k \neq j, i \neq j, k \neq i$

$$rc[z_{ij}^{tk}] = (\lambda_{jk}^t + \mu_{ik}^t + \nu_{ij}^{tk}) \quad (4.45)$$

- o custo reduzido de  $z_{jj}^{tk}$  para  $\forall t \in T$  e  $k, j \in I, k \neq j$

$$rc[z_{jj}^{tk}] = (\lambda_{jk}^t + \mu_{jk}^t) \quad (4.46)$$

- o custo reduzido de  $z_{ik}^{tk}$  para  $\forall t \in T$  e  $k, i \in I, k \neq i$

$$rc[z_{ik}^{tk}] = \mu_{ik}^t \quad (4.47)$$

Para resolver o subproblema  $LR^z$ , faça para todo  $t \in T$  e  $\forall k, i, j \in I : i \neq k$ : se  $rc[z_{ij}^{tk}] < 0$  então  $z_{ij}^{tk} = 1$ , senão  $z_{ij}^{tk} = 0$ . Ou seja, atribua o valor 1 para as variáveis que apresentam o  $rc$  negativo, e 0, caso contrário.

O subproblema  $LR^{qs}$  é definido por (4.48):

$$\min \sum_{t \in T} \sum_{i \in I} h_i s_{it} + \sum_{i \in I} \sum_{l \in T} \sum_{t=l}^{|T|} (\rho_l a_i + \omega_{lt}^i) q_{lt}^i$$

sujeito a:

$$\sum_{l=1}^t q_{lt}^i = d_{it} \quad \forall i \in I, \forall t \in T \quad (4.48)$$

$$\sum_{l=1}^t \sum_{t'=t+1}^{|T|} q_{lt'}^i = s_{it} \quad \forall i \in I, \forall t = 1, \dots, (|T| - 1)$$

$$s_{it} \geq 0 \quad \forall t \in T, \forall i \in \bar{I}$$

$$q_{lt}^i \geq 0 \quad \forall t \in T, \forall l \in T, i \in \bar{I}$$

O  $LR^{qs}$  resultante é um problema em função das variáveis de decisão  $s$  e  $q$ , que definem, respectivamente, o estoque e a quantidade produzidas. Neste problema resultante não há nenhuma restrição de capacidade. Logo, têm-se um problema conhecido como *Multi-Item Uncapacitated Lot-Sizing*. Uma vez que não há concorrência entre os itens por recursos (*Uncapacitated*), é possível resolver  $LR^{qs}$  para cada item  $i$  individualmente. Assim, têm-se um *Single-Item Uncapacitated Lot-Sizing Problem* – (LS-U) para cada item  $i \in I$ , representado por (4.49):

$$\min \sum_{t \in T} h s_t + \sum_{l \in T} \sum_{t=l}^{|T|} (\rho_l a + \omega_{lt}) q_{lt}$$

sujeito a:

$$\sum_{l=1}^t q_{lt} = d_t \quad \forall t \in T \quad (4.49)$$

$$\sum_{l=1}^t \sum_{t'=t+1}^{|T|} q_{lt'} = s_t \quad \forall t = 1, \dots, (|T| - 1)$$

$$s_t \geq 0 \quad \forall t \in T$$

$$q_{lt} \geq 0 \quad \forall t \in T, \forall l \in T$$

Para resolver (4.49), Eppen e Martin (1987) propuseram uma reformulação baseada no Problema de Caminho Mínimo (*Shortest path problem* – SPP). Segundo Ahuja et al. (1990), este é um problema fácil de ser resolvido, ou seja, ele pode ser resolvido em tempo hábil, mesmo para instâncias maiores. Esta reformulação explora a propriedade do Intervalo de Regeneração ( $T^R$ ).

Um Intervalo de Regeneração é definido como um intervalo de períodos subsequentes ( $T^R = \{l, l + 1, \dots, l'\}$ ), denotado por  $[l, l']$ , no qual, a produção  $q$

do item ocorre somente no primeiro período  $l$  do intervalo e é capaz de satisfazer a demanda de todos os períodos do intervalo, ou seja:

$$q^l = \sum_{t=l}^{l'} d_t$$

Ainda, segundo Manne (1961), para caracterizar este intervalo, repare que o estoque  $s^{l-1}$  no início do intervalo é igual ao estoque final  $s^{l'}$  e ambos são iguais a zero, ou seja,  $s^{l-1} = s^{l'} = 0$ , entretanto,  $s^t > 0$  para  $t = l, l + 1, \dots, l' - 1$ .

Uma solução ótima para (4.49) pode ser decomposta em uma sequência de Intervalos de Regeneração, veja Pochet e Wolsey (2010). Assim, seja  $G(L)$  um grafo orientado no qual cada arco  $(l, t)$ , tal que  $l < t$ , corresponde ao Intervalo de Regeneração  $[l, t - 1]$ , tal que o conjunto de nós  $L = \{1, 2, \dots, |T|, \bar{l}\}$ , é formado pelos períodos  $t \in T$  e o pelo nó fictício de destino  $\bar{l}$ . O custo  $c'_{l,t}$  associado ao arco  $(l, t)$  é dado por (4.50):

$$c'_{l,t} = \begin{cases} \sum_{k=l}^{t-1} (\rho_l a + \omega_{lk}) d_k + \sum_{k=l}^{t-2} \sum_{t'=k+1}^{t-1} h_k d_{t'} & \text{se } t - l \geq 2; \\ (\rho_l a + \omega_{ll}) d_l & \text{caso contrário.} \end{cases} \quad \forall l \in T, t \in L, l < t \quad (4.50)$$

Logo, é possível resolver (4.49) por um algoritmo para o SPP no grafo  $G(L)$ , exemplificado pela Fig. 5.

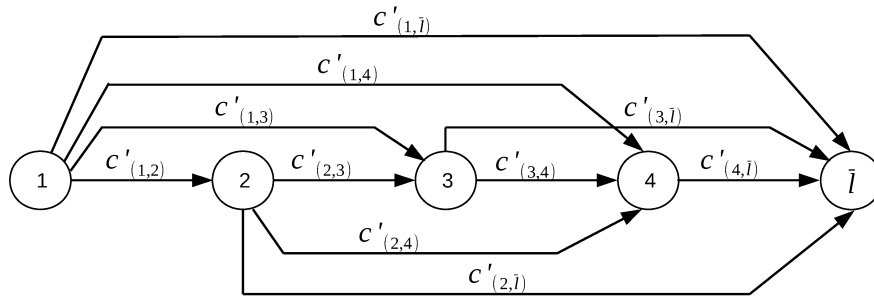


Figura 5 – Ilustração do caminho mínimo para resolver LS-U (dado por (4.49)) com  $|T| = 4$

Assim como para o PCTSP, o algoritmo proposto para resolver o LSSP é descrito na Seção 2.5. A seguir, são apresentados os procedimentos heurísticos de solução inicial, busca local e planos de corte.

#### 4.4.1 Solução Inicial

Na tentativa de encontrar uma solução inicial ( $UB^0$ ) utilizou-se uma heurística construtiva gulosa. Ela constrói a solução iterativamente período a período,

começando pelo período 1. O procedimento é inicializado com a atribuição do valor  $10^9$  para  $UB^0$ . Então, a produção dos itens é ativada de acordo com o parâmetro  $\varphi_i$ , em seguida, é feito o sequenciamento dos itens ativados. Para todo  $i \in I$ , o parâmetro  $\varphi_i$  é calculado segundo a Equação (4.51).

$$\varphi_i = \frac{2 \sum_{j \in I \setminus \{i\}} \frac{c_{ij}}{|I|}}{\sum_{t \in T} \frac{d_{it} h_i}{|T|}} \quad \forall i \in I \quad (4.51)$$

Este parâmetro representa a relação entre o custo médio de ajuste da máquina e o custo com o estoque para um período. Note que se o valor de  $\varphi$  for maior que 1 ( $\varphi > 1$ ), isto indica que os custos com ajustes podem ser relativamente maiores do que com o estoque. Dessa forma, para estes itens é preferível que eles tenham sua produção antecipada e que sejam estocados. Para os itens com valores de  $\varphi$  menores que 1 ( $\varphi < 1$ ), é preferível que eles não sejam estocados.

Inicialmente, a produção de todos os itens é ativada no período  $t = 1$  de acordo com a demanda do período corrente. Ou seja, para todo  $i$  que possua a demanda maior que zero ( $d_{it} > 0$ ), tem-se que  $q_{11}^i = d_{i1}$ . Caso a capacidade produtiva não tenha sido totalmente utilizada, a produção dos itens é adiantada segunda a Regra de Antecipação:

- Se  $\varphi \geq 2,5$  a produção do item é antecipada em dois períodos, desde que tenha capacidade disponível.
- Se  $1 \leq \varphi < 2,5$  a produção do item é antecipada em um período, desde que tenha capacidade disponível.
- Se  $\varphi < 1$  a produção do item não é antecipada.

Lembre-se que os tempos de ajuste devem ser considerados para o cálculo da utilização da capacidade e estes tempos dependem do sequenciamento. Assim, é considerado o tempo médio aproximado como  $2 \sum_j \frac{c_{ij}}{|I|}$ . Este processo é repetido até que a capacidade do período 1 tenha sido utilizada. Uma vez terminada o estágio de ativação da produção, é feito o sequenciamento por meio do Concorde (citado anteriormente na Seção 2.4.1). Novamente, será preciso transformar a instância de acordo com o procedimento de Kumar e Li (2000). Além disso, para toda vez que o Concorde é chamado, antes, é verificado se o subconjunto de itens já foi sequenciado em alguma iteração anterior. Em caso afirmativo, esta etapa é desconsiderada. Definido a sequência de produção, é verificado se a restrição de capacidade foi respeitada. Caso haja inviabilidade, o maior lote de produção antecipado, ou seja, aquele que consome mais tempo, será reduzido proporcionalmente à demanda do período mais distante de  $t$ . Este processo é repetido até que a restrição de capacidade seja respeitada.

A partir do período 2, é verificado quais itens devem ser ativados obrigatoriamente para atender a demanda. Ou seja, os itens que não tiveram a demanda do período corrente  $t$  antecipada. Em seguida é verificado se existe algum item com  $\varphi > 1$  ativo. Em caso afirmativo, sua produção será antecipada de acordo com a Regra de Antecipação, citada anteriormente, de acordo com a prioridade na qual foi apresentada. Assim como no fim de todo período, ao terminar o estágio de ativação da produção, é feito o sequenciamento por meio do Concorde seguido do processo de verificação de viabilidade da restrição de capacidade.

Para garantir que a restrição *carryover* seja respeitada, o primeiro item que será produzido no período corrente será sempre igual ao último do período anterior. Neste caso, a produção deste item não pode ser antecipada. Assim, um item ativo com valor de  $\varphi < 1$ , caso exista, é escolhido aleatoriamente para ser o último item produzido, logo ele não é considerado durante a etapa de sequenciamento. Caso não exista pelo menos um item ativo com  $\varphi < 1$ , um item ativo qualquer é escolhido aleatoriamente para ser o último item produzido, de acordo com a Regra de Antecipação ( (i)  $2, 5 \geq \varphi$  e (ii)  $1 < \varphi \leq 2, 5$ ). Assim, este procedimento é repetido até que todos os períodos sejam contemplados.

Esta heurística não garante a obtenção de uma solução viável. Entretanto, os experimentos realizados mostraram que ela é eficaz para as instâncias adotadas. Pois, ao ser utilizada, ela obteve sucesso para gerar uma solução inicial em todos os casos.

#### 4.4.2 Busca Local

Assim como para o PCTSP (Seção 2.4.2), o objetivo é introduzir um esquema heurístico dentro da relaxação Lagrangeana com o intuito de encontrar boas soluções para o LSSP.

Seja  $Y$  a solução do subproblema Lagrangeano  $LR^y$  (descrito na Seção 4.4). O procedimento consiste, a cada iteração, em realizar  $NP$  perturbações sobre  $Y$ . Uma perturbação constitui uma alteração no estado da variável  $y_i^t$ , escolhida aleatoriamente. Ou seja, se  $y_i^t = 1$  então seu valor é alterado para zero ( $y_i^t \leftarrow 0$ ); caso contrário (se  $y_i^t = 0$ ) seu valor é ajustado para um ( $y_i^t \leftarrow 1$ ). No fim de sua execução, o procedimento retorna a melhor solução encontrada.

Para impedir que ocorram mais de uma perturbação, dentro de um mesmo intervalo de  $NP$  perturbações, sobre a mesma variável  $y_i^t$ , adotou-se uma *lista restrita*. Ela contém os índices  $i$  e  $t$  das variáveis  $y_i^t$  já selecionadas.

Repare que, a partir de  $Y$ , é possível determinar em qual período  $t$  um item  $i$  deve ser produzido (ativo). Entretanto, para gerar uma solução, também é preciso determinar as quantidades e o sequenciamento da produção.

Para determinar as quantidades, utiliza-se o conceito do Intervalo de Regeneração (descrito na Seção 4.4). Para todo período  $t$ , a produção de um item  $i \in I$  deve ser ativada se  $y_i^t = 1$ . A quantidade produzida, no período  $t$ , é igual à demanda do período  $t$  somada às demandas dos períodos posteriores  $t'$ , para os quais a produção não está ativa (ou seja,  $y_i^{t'} = 0$ ). São considerados os períodos consecutivos até que  $y_i^{t''} = 1$ , ou seja,  $t < t' < t''$ .

O sequenciamento é determinado de forma semelhante ao PCTSP. Para determinar o sequenciamento é fixado um período  $t$  e os itens  $i$ , para os quais  $y_i^t = 1 \forall i \in I$ , são transformados em uma instância do TSP assimétrico. Esta é transformada em uma instância do TSP simétrico, a qual é resolvida pelo Concorde.

Entretanto, para garantir que a restrição de *carryover* (conservação do estado da máquina) seja respeitada, novamente, o parâmetro  $\varphi_i$  (Seção 4.4.1) é considerado. Ainda, durante a etapa de sequenciamento, é verificado se existe algum item  $i$  ativo, tal que  $\varphi_i < 1$ , em dois períodos consecutivos ( $t$  e  $t + 1$ ), a partir do período 1. Em caso afirmativo, um destes itens é selecionado de forma aleatória e sua posição é fixada como o último do período  $t$  e primeiro do período  $t + 1$ . Não é permitido que o mesmo item  $i$  seja o primeiro e último a ser produzido, dentro de um mesmo período. Caso não exista um item  $i$  ativo, tal que  $\varphi_i < 1$ , então é escolhido um item  $i$  ativo, aleatoriamente, seguindo a regra: (1º)  $1 \leq \varphi_i < 2,5$  e (2º)  $\varphi_i \geq 2,5$ .

Após o sequenciamento, ao contrário do procedimento guloso (Seção 4.4.1), não é verificada a viabilidade da solução. Ou seja, não é verificado se a restrição de capacidade é violada. Ela é considerada apenas durante o cálculo do valor de  $f$ . Caso ela seja violada, uma penalidade é incorrida. O valor da solução  $f$  é calculado de acordo com a Eq. (4.52).

$$f = \sum_{t \in T} \sum_{i \in I} h_i s_{it} + \sum_{t \in T} \sum_{i \in I} \sum_{j \in I} c_{it} x_{ij}^t + \Delta_{cap} \sum_t C_t \quad (4.52)$$

Na qual  $\Delta_{cap}$  representa a soma, para todo  $t \in T$ , dos desvios positivos em relação à capacidade  $C_t$ , calculada segundo a Eq. (4.53):

$$\Delta_{cap} = \sum_{t \in T} \Delta_t \quad (4.53)$$

Tal que  $\Delta_t$  indica o quanto a capacidade é extrapolada no período  $t$ , e é calculada pela Eq. (4.54):

$$\Delta_t = \max \left\{ 0; \sum_{i \in I} a_i q_{it} + \sum_{i \in I} \sum_{j \in I \setminus \{i\}} f_{ij} x_{ij}^t - C_t \right\} \quad (4.54)$$

Logo,  $f$  é penalizada proporcionalmente ao tempo no qual excede a capacidade.

Assim, o Algoritmo 6 resume o procedimento de busca local.

---

**Algorithm 6** Busca Local para o LSSP

---

```

1: Dado  $Y$ , gere a solução  $S$  e calcule  $f$  (valor da solução)
2:  $NP \leftarrow 1$ 
3:  $\bar{f} \leftarrow f$ 
4: while  $NP \leq |I|$  do
5:    $pert \leftarrow 0$ 
6:    $lista\ restrita \leftarrow \emptyset$ 
7:   while  $pert < NP$  do
8:     Escolha  $t$  aleatoriamente, tal que  $1 < t \leq |T|$ 
9:     Escolha  $i$  aleatoriamente, tal que  $y_i^t \notin lista\ restrita$ 
10:    Faça  $y_i^t \in lista\ restrita$ 
11:    Defina  $Y'$  fazendo: Se  $y_i^t = 1$  então  $y_i^t \leftarrow 0$ ; senão  $y_i^t \leftarrow 1$ 
12:     $pert \leftarrow pert + 1$ 
13:  end while
14:  A partir de  $Y'$  construa uma solução  $S'$  e calcule  $f'$ 
15:  if  $f' < \bar{f}$  then
16:     $\bar{f} \leftarrow f'$ 
17:     $S \leftarrow S'$ 
18:  end if
19:   $NP \leftarrow NP + 1$ 
20: end while
21: Retorne  $\bar{f}$ 

```

---

A partir de  $Y$ , o procedimento determina as quantidades produzidas para todo item  $i \in I$ , para todo período  $t \in T$ . Então, os itens são ordenados pelo Concorde, dessa forma uma solução  $S$  é gerada (Linha 1). No caso de uma solução  $S$  já estiver disponível, como por exemplo, quando é feito o refinamento da solução inicial, as etapas de determinação das quantidades e sequenciamento são desconsideradas.

Nas linhas 2 e 3, são inicializados, respectivamente, o número de perturbações  $NP$  e o valor da melhor solução  $\bar{f}$ , até o momento.

A cada iteração principal, o contador de perturbações  $pert$  é inicializado, bem como, a *lista restrita* (Linhas 5 e 6).

Para cada perturbação  $pert$ , primeiro, um período  $t$  é escolhido aleatoriamente (Linha 8). Na Linha 9, um item  $i$  também é escolhido aleatoriamente, tal que ainda não tenham sido escolhido dentro do intervalo corrente de  $NP$  perturbações. Esta etapa é repetida até se encontre um candidato inédito.

Definidos  $i$  e  $t$ ,  $y_i^t$  é adicionado à *lista restrita* na Linha 10 e, na Linha 11, o estado do mesmo é alterado. Ou seja, se ele está ativo (resp. inativo) ele será trocado para inativo (resp. ativo).

Finalizada a etapa de perturbações sobre  $Y$ ,  $Y'$  é determinado. Então, na Linha 14, a nova solução  $S'$  é gerada e  $f'$  é calculado (Equação. (4.52)).

Assim, a melhor solução é atualizada nas Linhas 16 e 17. Em seguida, na Linha 19, o contador  $NP$  é incrementado. A busca local é repetida, a partir de  $\bar{S}$ , até que não encontre solução de melhora.

No fim da execução, a busca local retorna a melhor solução encontrada durante o procedimento, desde que ela seja viável. Caso a solução obtida seja inviável,



o UB não será atualizado.

#### 4.4.3 Desigualdades válidas para o LSSP

Assim como no PCTSP, são apresentadas duas famílias de desigualdades válidas: desigualdades  $W_t$ ; e desigualdades  $(L, S)$ . Elas foram retiradas de Almada-Lobo et al. (2007), que utilizam estas famílias de desigualdades para fortalecerem os resultados da relaxação linear das formulações propostas naquele trabalho.

Semelhante ao PCTSP, elas são igualmente utilizadas a partir da solução do subproblema Lagrangeano.

##### 4.4.3.1 Desigualdades $W_t$

A primeira família de desigualdades válidas apresentada é proposta para o LSSP por Almada-Lobo et al. (2007). Ela não é uma família no espaço original, ou seja, é preciso considerar novas dimensões para o problema, o que significa novas variáveis. Considere  $W_t$  como uma variável binária, que assume o valor 0 se pelo menos um ajuste de máquina for realizado no período  $t$ , e 1 caso contrário. Então, segundo Almada-Lobo et al. (2007), as seguintes famílias de desigualdades válidas (4.55)-(4.58), nomeadas como desigualdades  $W_t$ , são válidas para o LSSP.

$$\alpha_i^t \leq \sum_{j \in I} x_{ij}^t + W_t \quad \forall i \in I, \forall t \in T \quad (4.55)$$

$$\alpha_i^t \leq \sum_{j \in I} x_{ji}^{t-1} + W_{t-1} \quad \forall i \in I, \forall t \in T \quad (4.56)$$

$$1 - W_t \leq \sum_{i \in I} \sum_{j \in I} x_{ij}^t \quad \forall t \in T \quad (4.57)$$

$$W_t \leq 1 - \frac{\sum_{i \in I} \sum_{j \in I} x_{ij}^t}{|I|} \quad \forall t \in T \quad (4.58)$$

Note que, existem apenas  $|T|$  variáveis  $W_t$  e que seu domínio binário está implícito. Ou seja, não precisa ser declarado explicitamente por uma restrição. Além disso, a identificação de uma desigualdade, ou corte, possivelmente violado é verificado por meio de uma simples inspeção das desigualdades (4.55)-(4.58). Logo, a introdução desta nova variável, e das respectivas desigualdades violadas, não aumenta o tempo computacional significativamente.

A família de desigualdades  $W_t$  pode ser facilmente identificada por meio de uma simples conferência. A cada iteração é verificado se as desigualdades (4.55)-(4.58) são violadas a partir da solução obtida pela resolução do subproblema Lagrangeano  $LR^x$ . Lembre-se que a formulação **EPN'** não prevê as variáveis  $\alpha_i^t$ . Neste caso é feita

a sua substituição por  $x_{ri}^t$  ( $\alpha_i^t = x_{ri}^t$ ) para todo  $i \in I$  e para todo  $t \in T$ . Para mais detalhes veja Almada-Lobo et al. (2007).

#### 4.4.3.2 Desigualdades $(L, S)$

Além das desigualdades  $W_t$ , também é adotada a família de desigualdades  $(L, S)$ . Ela foi originalmente proposta por Barany, Roy e Wolsey (1984) para o problema de dimensionamento de lotes. Naquele trabalho, considera-se apenas a versão com um único produto. Almada-Lobo et al. (2007) adaptaram esta família para o LSSP, o qual considera múltiplos itens e o sequenciamento da produção. Ela foi adaptada e a família de desigualdades válidas para o LSSP é representada por (4.59).

$$d_{1l}^i \leq \sum_{t \in L \setminus S} d_{tl}^i \left( \sum_{j \in I} x_{ij}^t + \alpha_i^t \right) + \sum_{t \in S} \sum_{u=t}^{|T|} q_{tu}^i \quad \forall i \in I, \forall l \in T, \forall S \subseteq L \quad (4.59)$$

Na qual  $L = \{1, \dots, l\}$  tal que  $l = 1, \dots, |T|$  e  $d_{tl}^i = \sum_{l'=t}^l d_{il'}^i$ .

Para identificar as desigualdades  $(L, S)$ , mais violadas, adotou-se o mesmo algoritmo de separação proposto por Barany, Roy e Wolsey (1984), o qual é explicado a seguir. Antes, repare que o algoritmo foi adaptado para a versão com múltiplos itens e que possui uma complexidade da ordem de  $O(|I||T|^2)$ .

Sejam:  $\bar{y}$  a solução do subproblema  $LR^y$ , tal que  $y \in \{1, 0\}$ ; e  $\bar{q}$  a solução do subproblema  $LR^{\bar{q}}$  que satisfaz as restrições (4.3) e (4.7), para  $l = 1, \dots, T$ . Para encontrar algum corte possivelmente violado procure, para todo  $i \in I$ , por um conjunto  $S_l \subseteq L = \{1, \dots, l\}$  no qual:

$$t \in S_l \text{ se } \bar{q}_i^t \leq d_{tl}^i \bar{y}_i^t \quad (4.60)$$

$$t \in L \setminus S_l \text{ se } \bar{q}_i^t > d_{tl}^i \bar{y}_i^t \quad (4.61)$$

Fixado  $i \in I$ , o algoritmo de separação consiste em definir o valor de  $d_{tl}^i$  como a demanda acumulada no intervalo  $[t, l]$ , tal que  $1 \leq l \leq T$  e  $t \in T$ . Em seguida, os conjuntos  $S_l$  e  $L \setminus S_l$  são construídos a partir de (4.60) e (4.61), para cada período  $t \in T$ . Por fim é verificado se existe um conjunto  $L \setminus S_l$  para o qual  $\sum_{t \in L \setminus S_l} d_{tl}^i \bar{y}_i^t < d_{tl}^i$ . Em caso afirmativo, então, existe uma desigualdade  $(L, S)$  violada, representada pelo conjunto  $L \setminus S_l$ . Para mais detalhes sobre este alg. veja Barany, Roy e Wolsey (1984).

Bem como para a família de desigualdades válidas  $W_t$ , também é necessário fazer a substituição da variável  $\alpha_i^t$  por  $x_{ri}^t$  em (4.59). Dessa forma, as desigualdades  $(L, S)$  também podem ser combinadas com **EPN'** e **CSC**.

## 4.5 Algoritmo proposto para o LSSP

Os algoritmos adotados para resolver o PCTSP e LSSP são, ambos, baseados na estrutura do Algoritmo 4, apresentada na Seção 2.5. A diferença entre os algoritmos reside em:

- Linha 2: procedimento de construção da solução inicial.
- Linha 9: subproblemas Lagrangeanos.
- Linha 23: procedimento para gerar uma solução viável.
- Linha 30: desigualdades válidas.
- Linha 33: procedimento de busca local.

Assim, a seguir, a adaptação da estrutura do Algoritmo 4 é explicada.

Na Linha 1, são inicializados o contador de iteração  $it$  e os demais parâmetros do VOL. Em seguida, na Linha 2 uma solução inicial é gerada por meio de um procedimento guloso. Este é descrito na Seção 4.4.1. O primeiro  $UB$  para o problema (primeira solução viável) é, assim, definido. Novamente, o laço de repetição principal é executado enquanto os critérios de parada não são atendidos.

Na Linha 4, o contador de iterações é incrementado e, na linha 6, o parâmetro  $LB_{it}$  (melhor  $LB$ ) é definido como 0. O passo  $\theta^t$ , bem como os multiplicadores de Lagrange  $\eta$  são atualizados na linha 8. Então, os subproblemas Lagrangeanos, apresentados na Seção 4.4 são resolvidos e o valor  $LB'$  é calculado de acordo com a Eq. (4.62) :

$$LB' \leftarrow LR^y + LR^x + LR^{x_r} + LR^z + LR^{q_s} - C(T) - (\pi_r + \varpi_r) \quad (4.62)$$

Repare que, a cada iteração, os subproblemas Lagrangeanos são respectivamente resolvidos pelos procedimentos descritos na Seção 4.4. No caso do LSSP, o conjunto  $H'$  é formado pelo subconjunto de vértices  $y_i^t = 1$  para  $i \in I$  e  $t \in T$ .

Na linha 10, o valor dos subgradientes  $v^{it}$  é atualizado. Destaca-se que é adotado o mesmo ajuste no vetor  $v$ , para as restrições do tipo menor ou igual ( $\leq$ ), proposto por Beasley (1993), a saber:  $v_k = 0$  se  $v_k > 0$  e  $\eta_k = 0$ . Na sequência, os valores de  $\alpha^{it}$ ,  $d^{it}$ ,  $\bar{\chi}^{it}$  são atualizados na linha 11.

Na linha 12 é verificado se o  $LB'$  é o melhor até o momento. Em caso afirmativo, os valores  $LB$ ,  $LB_{it}$ , e  $H$  são respectivamente atualizados nas linhas 13, 14, e 15. Caso contrário, é, então, verificado na linha 17 se  $LB'$  é o melhor  $LB$  dentro do bloco de  $h_{it}$  iterações. Em caso afirmativo  $LB_{it}$  e  $H$  são respectivamente atualizados nas linhas 18 e 19.

Na linha 22 é verificado se o conjunto  $H'$  é inédito. Repare que a partir deste subconjunto é possível gerar o melhor  $LB$  até o momento ou o melhor valor dentro do bloco de  $h_{it}$  iterações,  $LB_{it}$ . Caso seja verdade, é chamado um procedimento para gerar uma solução viável  $UB$ . Este é descrito na Seção 4.4.2 como parte do procedimento de busca local. Encontrado uma solução viável  $UB$ , o valor de  $UB'$  é atualizado na linha 27, caso  $UB$  seja a melhor solução até o momento.

Na linha 30 é realizada a busca por desigualdades válidas possivelmente violadas. Assim como para o PCTSP, caso seja identificada uma desigualdade violada, esta é dualizada e mantida até o final do algoritmo. As desigualdades violadas são identificadas por meio de algoritmos de separação explicados na Seção. 4.4.3.

Finalmente, na linha 32, é verificado no final de um bloco de  $h_{it}$  iterações se o conjunto  $H$  é inédito. No caso afirmativo, na linha 33, a heurística de busca local, apresentada na Seção 4.4.2 é aplicada, e o  $UB$ , possivelmente, atualizado, linha 35.

## 5 Experimentos Computacionais sobre o Problema de Dimensionamento e Sequenciamento de Lotes

Na presente seção, são apresentados os resultados provenientes do LSSP. Primeiro, são apresentadas as instâncias originalmente propostas em Almada-Lobo et al. (2007) e James e Almada-Lobo (2011). Então, os resultados da comparação das formulações, apresentadas na Seção 4.3, são mostradas. Estas são comparadas em termos da relaxação linear. Em seguida, são apresentados os resultados para a estrutura do Algoritmo 4 adaptado ao LSSP, mostrado na Seção 4.5. Assim como para o PCTSP, este algoritmo explora as informações da solução proveniente dos subproblemas para encontrar soluções viáveis para os problemas e identificar desigualdades válidas violadas. Estes resultados são comparados com os resultados obtidos com a resolução da formulação **SCF** por meio do *solver* Gurobi. Finalmente, são mostrados os resultados para o impacto do uso de desigualdades válidas no Algoritmo 4.

Assim como para o PCTSP, todos os experimentos foram executados no mesmo PC Intel® Core™ i7-2600, 3,40 GHz, 16 GB de RAM, sob uma distribuição Linux 64 bits. As formulações também foram implementadas com a linguagem C, por meio da interface do *solver* Gurobi versão 7.5.2, 64 bits, o qual foi utilizado para tentar resolver o problema inteiro até a otimalidade, assim como as relaxações lineares. Novamente, a implementação do algoritmo, da linguagem C++, foi baseada no código fornecido pelo COIN-OR *Project*, disponível em <<https://projects.coin-or.org/Vol>>. Também é utilizado a interface do Concorde combinado com o CPLEX versão 12.7, 64 bits.

### 5.1 Instâncias para o LSSP

Nesta seção são apresentadas as instâncias consideradas. Elas são as mesmas utilizadas no trabalho de Almada-Lobo et al. (2007), James e Almada-Lobo (2011) Oliveira e Santos (2017) e Guimarães, Klajban e Almada-Lobo (2013). Elas foram gentilmente cedidas pelos autores do trabalho James e Almada-Lobo (2011) para a realização desta pesquisa.

Cada instância deste grupo é construída pelos elementos  $P, T, \theta, cut$ . No qual:

$P$  representa o número de produtos;  $T$  representa o número de períodos;  $\theta$  representa o custo do *setup* da máquina em função do tempo; e *cut* representa a taxa de utilização da capacidade. Eles são definidos pelos seguintes critérios:

- $P \in \{15, 25\}$ .
- $T \in \{5, 10, 15\}$ .
- $\theta \in \{50, 100\}$ .
- $cut \in \{0, 6; 0, 8\}$ .

Os seguintes parâmetros de entrada do problema foram gerados a partir de uma distribuição uniforme e depois arredondados para o número inteiro mais próximo. Os intervalos considerados são:

- $w_i$  = distribuição uniforme no intervalo  $[2, 9]$  para todo  $i \in I \setminus \{r\}$ .
- $d_i^t$  = distribuição uniforme no intervalo  $[40, 59]$  para todo  $i \in I \setminus \{r\}$  e para todo  $t \in T$ .
- $f_{ij}$  = distribuição uniforme no intervalo  $[5, 10]$  para todo  $i, j \in I \setminus \{r\}$ .

Os demais parâmetros de entrada foram calculados a partir dos elementos que foram gerados, conforme citado anteriormente. As especificações consideradas para estes parâmetros são:

- $C_t = \frac{\sum_{i \in I \setminus \{r\}} d_i^t}{cut}$ , tal que  $cut \in \{0, 6; 0, 8\}$ .
- $a_i = 1$  para todo  $i \in I \setminus \{r\}$ .
- $c_{ij} = \theta f_{ij}$ , tal que  $\theta \in \{50, 100\}$ .

Para cada combinação dos elementos dos conjuntos  $P$ ,  $T$ ,  $\theta$  e *cut*, dez problemas foram gerados. Dessa forma, a partir dos elementos definidos nesta seção, um total de 240 problemas-teste foram gerados aleatoriamente, deste total 102 foram introduzidos em Almada-Lobo et al. (2007) e as 138 restantes em James e Almada-Lobo (2011).

## 5.2 Resultados da relaxação linear

Na presente seção são apresentados os resultados para a comparação entre as formulações descritas na Seção 4.3. Esta comparação consiste no desempenho em

termos da relaxação linear, ou seja, na qualidade do LB fornecido e no esforço computacional. Também são expostas as particularidades da implementação das formulações.

Dado que a formulação **AL** possui um número de restrições exponenciais, ela necessita de um algoritmo de separação para identificar as possíveis restrições violadas. Este algoritmo pode ser resolvido em tempo polinomial, logo ele não possui um impacto significativo no desempenho computacional. Portanto, foi utilizado um algoritmo de fluxo máximo, tal como descrito Seção 4.3.1, baseado na implementação proposta por Dinitz (2006) (o mesmo utilizado para **GSEC**).

Os resultados detalhados da resolução do problema com o domínio inteiro misto, por meio do Gurobi, são apresentados no Apêndice B.

A Tabela 14 resume, os resultados para as 240 instâncias. Ela apresenta a comparação da relaxação linear das 4 formulações em termos da força da relaxação linear (*gap*), e do esforço computacional (tempo). Além disso, também são apresentados os resultados das formulações **AL** e **EPN'**, combinadas com as famílias de desigualdades válidas  $W_t$  e  $(L, S)$  (Seções 4.4.3.1 e 4.4.3.2, respectivamente), nas colunas indicadas por **+cuts**. O *gap* é calculado segundo a Equação (5.1), assim como na seção reservada para os resultados do PCTSP (Seção 3.2). Ele representa o desvio relativo da relaxação linear *LB*, em relação à solução ótima *opt*, obtida por meio de **SCF**, resolvida pelo Gurobi em até uma hora. Para os casos nos quais não foi possível encontrar a solução ótima em até uma hora, utiliza-se a melhor solução conhecida. Esta ou é obtida pelo Algoritmo 4, ou é a melhor solução encontrada pelo Gurobi em até uma hora, ou, para os casos nos quais nenhuma solução inteira foi encontrada em até uma hora, é a primeira solução inteira encontrada pelo Gurobi, sem limite de tempo.

$$gap = (Opt - LB)/Opt \quad (5.1)$$

Na parte superior da Tabela 14, para cada formulação, é apresentado na primeira linha (*AVG\_gap*), o *gap* médio para as instâncias. Na segunda linha são relatados os tempos computacionais médios (*AVG\_tempo*), em segundos, para obter o *LB* sobre essas mesmas instâncias. Cada linha dessa tabela apresenta a média para os 10 problemas-teste caracterizados pelos elementos  $P, T, \theta, cut$ . Os resultados detalhados para cada problema-teste são apresentados no Apêndice C.

Ao contrário do PCTSP, todas as 4 formulações conseguem resolver todas as relaxações lineares para as 240 instâncias em tempo razoável (menos de uma hora). Ao se considerar as formulações sem as desigualdades válidas, a formulação **EPN'** mostrou-se a mais forte entre as consideradas neste trabalho. A formulação **EPN'+cuts** possui o melhor valor do *gap* médio, destacado em negrito. Além disso, **EPN'+cuts** apresenta o *gap* menor ou igual para todas as instâncias. Tal fato pode ser observado na Tabela 25 do Apêndice C, pág. 146. Ou seja, isto indica que ela pode ser dominante (ou mais forte, ou mais apertada), em relação à demais formulações.

Tabela 14 – Resultados da comparação da relaxação linear dos 24 grupos ( $P, T, \theta, cut$ ) de 10 instâncias cada (LSSP).

		Formulações					SCF	N. de sol ótimas
		AL	AL+cuts	CSC	EPN'	EPN'+cuts		
	AVG_gap	67,08	1,37	1,45	1,36	<b>1,33</b>	1,45	
	AVG_tempo	1	1	14	78	81	6	
Data15-10-0.6-100	gap	69,39	<b>1,89</b>	2,10	1,97	<b>1,89</b>	2,10	2
	tempo	1	1	4	12	12	2	-
Data15-10-0.6-50	gap	74,13	0,25	0,44	<b>0,23</b>	<b>0,23</b>	0,40	10
	tempo	1	1	4	12	12	2	-
Data15-10-0.8-100	gap	70,02	<b>2,62</b>	2,78	2,66	<b>2,62</b>	2,78	0
	tempo	0	1	5	13	13	2	-
Data15-10-0.8-50	gap	73,89	0,40	0,49	<b>0,38</b>	<b>0,38</b>	0,53	10
	tempo	0	0	3	9	9	2	-
Data15-15-0.6-100	gap	79,19	<b>3,60</b>	3,83	3,71	<b>3,60</b>	3,79	0
	tempo	1	1	10	30	32	4	-
Data15-15-0.6-50	gap	82,23	0,32	0,48	<b>0,30</b>	<b>0,30</b>	0,36	0
	tempo	1	1	6	14	14	3	-
Data15-15-0.8-100	gap	79,48	<b>4,30</b>	4,41	4,37	<b>4,30</b>	4,45	0
	tempo	1	1	12	38	38	4	-
Data15-15-0.8-50	gap	82,10	0,49	0,58	<b>0,48</b>	<b>0,48</b>	0,55	0
	tempo	1	1	7	16	17	3	-
Data15-5-0.6-100	gap	48,04	1,35	1,40	1,31	<b>1,30</b>	1,50	10
	tempo	0	0	1	2	2	0	-
Data15-5-0.6-50	gap	55,65	0,22	0,37	<b>0,14</b>	<b>0,14</b>	0,36	10
	tempo	0	0	1	3	4	0	-
Data15-5-0.8-100	gap	51,45	1,70	1,75	<b>1,64</b>	<b>1,64</b>	1,88	10
	tempo	0	0	1	3	3	1	-
Data15-5-0.8-50	gap	55,88	0,40	0,45	0,36	<b>0,35</b>	0,52	10
	tempo	0	0	1	3	3	1	-
Data25-10-0.6-100	gap	66,46	<b>1,87</b>	1,92	1,88	<b>1,87</b>	1,90	0
	tempo	2	2	16	121	118	9	-
Data25-10-0.6-50	gap	71,11	<b>0,12</b>	0,18	<b>0,12</b>	<b>0,12</b>	0,14	0
	tempo	2	2	11	78	73	7	-
Data25-10-0.8-100	gap	67,48	<b>2,68</b>	2,73	2,69	<b>2,68</b>	2,72	0
	tempo	1	2	27	134	140	10	-
Data25-10-0.8-50	gap	71,26	0,47	0,50	<b>0,46</b>	<b>0,46</b>	0,48	0
	tempo	1	1	14	102	112	8	-
Data25-15-0.6-100	gap	76,15	<b>3,35</b>	3,41	3,36	<b>3,35</b>	3,39	0
	tempo	4	5	44	373	400	25	-
Data25-15-0.6-50	gap	79,40	<b>0,40</b>	0,44	<b>0,40</b>	<b>0,40</b>	0,42	0
	tempo	2	3	26	174	177	13	-
Data25-15-0.8-100	gap	76,45	3,61	3,64	3,61	<b>3,60</b>	3,63	0
	tempo	4	5	89	404	412	28	-
Data25-15-0.8-50	gap	79,44	1,03	1,06	<b>1,02</b>	<b>1,02</b>	1,04	0
	tempo	2	3	31	216	241	15	-
Data25-5-0.6-100	gap	45,65	0,61	0,66	0,60	<b>0,59</b>	0,67	10
	tempo	1	1	3	21	23	2	-
Data25-5-0.6-50	gap	52,87	0,09	0,14	<b>0,08</b>	<b>0,08</b>	0,13	10
	tempo	1	1	4	39	33	1	-
Data25-5-0.8-100	gap	48,92	0,78	0,81	<b>0,76</b>	<b>0,76</b>	0,80	10
	tempo	0	1	5	25	26	2	-
Data25-5-0.8-50	gap	53,26	0,27	0,30	<b>0,26</b>	<b>0,26</b>	0,28	10
	tempo	0	1	4	31	30	2	-



As formulações **CSC** e **SCF** são incomparáveis. Ou seja, não há como determinar qual é a mais forte entre elas. Observe a primeira linha da Tabela 25 (Apêndice C, pág. 146, instância Data15-5-0.6-100-0), na qual **CSC** é possui um LB melhor do que **SCF**. No entanto, para a instância Data15-5-0.6-100-7, **SCF** possui um LB melhor do que **CSC**.

Ainda em relação ao *gap*, a formulação **AL** apresentou um valor muito pior que as demais. Porém, ao ser combinada com as desigualdades válidas  $W_i$  e  $(L, S)$ , coluna **AL+cuts**, seu desempenho é ligeiramente pior que **EPN'**, (1,37% versus 1,36%). As desigualdades válidas também apresentaram uma melhora para o valor do *gap* médio de **EPN'**, de 1,36% para 1,33% (**EPN'+cuts**). Na verdade, **EPN'+cuts** encontrou um LB melhor ou igual para todas as instâncias.

Quanto ao tempo de execução, medido em seg., **EPN'** e **EPN'+cuts** apresentam um desempenho pior ou igual para todas as instâncias, especialmente para  $|I| = 25$ . As formulações **AL** e **AL+cuts** apresentam sempre o melhor desempenho. As formulações **CSC** e **SCF** possuem um desempenho semelhante, com uma vantagem de 8 seg., em média, para **SCF**.

Os resultados obtidos para o PCTSP, por meio da formulação **EPN**, são confirmados para o LSSP, formulação **EPN'**. Logo, optou-se por adotar, para o LSSP, a mesma abordagem aplicada ao PCTSP. Ou seja, resolver o problema por meio da mesma estrutura do algoritmo híbrido, que utiliza uma abordagem Lagrangeana combinada com heurísticas de busca local. Os resultados são apresentados a seguir.

### 5.3 Resultados da abordagem Lagrangeana

Aqui são relatados os resultados obtidos com a estrutura do Algoritmo 4, proposto na Seção 4.5, adaptado para lidar com a abordagem Lagrangeana do LSSP (descrita na Seção 4.4).

O Algoritmo 4 é executado até que o mesmo critério de parada do VOL seja atendido: ou  $\|v\| \leq 0,0001$ ; ou se  $\frac{|LB-UB|}{|LB|} < 0,0001$  ou se  $UB - LB < 1$ . Novamente, todos os custos e penalidades associados às instâncias utilizadas neste experimento são inteiros.

Quanto aos parâmetros que controlam o comportamento do VOL, o processo de calibração foi semelhante ao PCTSP, adotando os mesmos conjuntos de valores. Para mais detalhes sobre esta etapa, veja o Apêndice E, no qual a calibração é devidamente detalhada para o LSSP. Para maior transparência, os valores usados em comum por todas as instâncias para obter os resultados apresentados são:

- $grp = 0,0001$ ;  $pap = 0,0001$ ;

- `greentestinvl = 1; yellowtestinvl = 30; redtestinvl = 40`
- `lambdainit = 0,1; alphainit = 0,1; alphamin = 0,0001; alphafactor = 0,5; alphaint = 150.`
- $h_{it} = 10$ , ou seja, a cada 10 iterações (ao contrário do PCTSP que adota 100) o procedimento de busca local é chamado.

Das 240 instâncias utilizadas neste experimento, 102 foram originalmente propostas por Almada-Lobo et al. (2007). Para estas instâncias a solução ótima pode ser obtida em tempo moderado, a maioria em poucos segundos, por meio da formulação **SCF** resolvida pelo Gurobi. Por outro lado, não são conhecidos os valores ótimos para as 138 instâncias restantes. Assim, são apresentados os resultados da abordagem por relaxação Lagrangeana, proposta neste trabalho, para estas 138 instâncias cujos valores ótimos não são conhecidos. Neste caso, são comparados os *gaps* de otimalidade obtidos a partir da formulação **SCF** no Gurobi com um tempo limite de 1800 segundos. Este valor foi utilizado pois, para nenhuma instância, o tempo necessário para rodar a abordagem Lagrangeana ultrapassou 1800 segundos.

### 5.3.1 Comparação dos LB's e *gaps* de otimalidade

Nesta seção são apresentados os resultados para a comparação entre os LB's obtidos pela relaxação Lagrangeana com a relaxação Linear de **EPN'+cuts** (formulação **EPN'** combinada com as desigualdades válidas). Também são comparados os *gaps* de otimalidade do Algoritmo 4 e da **SCF** resolvida pelo Gurobi (tempo limite de 1800 segundos). Os critérios para a comparação são:

- *gap LP*: o desvio relativo entre o *LB* da relaxação Lagrangeana e o valor da relaxação linear *LP* de **EPN'+cuts** ( $100 \times (LP - LB)/LP$ ).
- *gap Δ* intervalo no qual a solução ótima reside ( $100 \times (UB - LB)/UB$ ).
- tempo: tempo total de execução, em segundos, do Algoritmo 4.
- *gap SCF*: tempo total de execução, em segundos, do método exato através da formulação **SCF** em até meia hora.

A Tabela 15 apresenta os resultados para as 138 instâncias, dentre as 240 consideradas, nas quais a solução ótima não é conhecida. O símbolo “-” (coluna *gap SCF*) indica os casos nos quais nenhuma solução viável é encontrada pela formulação **SCF**, dentro do limite de tempo de meia hora (ou 1800 seg.).

Tabela 15 – Resultado do Algoritmo 4 para as 138 instâncias nas quais a solução ótima não é conhecida.

Instância	gap LP	gap $\Delta$	tempo Alg 4	gap SCF
$ I  = 15$ $ T  = 10$				
Data15-10-0.6-100-1	0,17	2,41	476	11,53
Data15-10-0.6-100-2	0,80	3,56	478	8,26
Data15-10-0.6-100-4	0,80	2,85	479	4,35
Data15-10-0.6-100-5	1,80	3,65	481	3,04
Data15-10-0.6-100-6	0,12	2,33	480	6,23
Data15-10-0.6-100-7	0,10	1,67	479	3,12
Data15-10-0.6-100-8	0,08	1,82	478	4,11
Data15-10-0.6-100-9	1,30	2,92	480	2,03
Data15-10-0.8-100-0	0,10	2,65	490	8,22
Data15-10-0.8-100-1	0,04	1,31	494	8,57
Data15-10-0.8-100-2	0,05	2,68	488	15,93
Data15-10-0.8-100-3	0,40	3,40	492	9,56
Data15-10-0.8-100-4	0,20	2,59	492	6,99
Data15-10-0.8-100-5	1,30	5,68	489	9,47
Data15-10-0.8-100-6	0,06	4,08	490	12,82
Data15-10-0.8-100-7	0,60	3,57	489	5,70
Data15-10-0.8-100-8	0,90	3,22	491	10,62
Data15-10-0.8-100-9	0,70	1,85	495	7,02
$ I  = 15$ $ T  = 15$				
Data15-15-0.6-100-0	0,17	2,84	771	13,64
Data15-15-0.6-100-1	0,03	2,64	774	21,17
Data15-15-0.6-100-2	0,03	3,22	771	15,93
Data15-15-0.6-100-3	0,70	4,74	771	13,88
Data15-15-0.6-100-4	0,70	4,98	770	24,13
Data15-15-0.6-100-5	1,30	5,59	767	15,63
Data15-15-0.6-100-6	1,60	5,56	769	14,97
Data15-15-0.6-100-7	1,10	5,01	768	19,23
Data15-15-0.6-100-8	0,18	2,83	764	19,64
Data15-15-0.6-100-9	1,20	3,59	775	3,15
Data15-15-0.6-50-0	1,30	1,66	711	0,12
Data15-15-0.6-50-1	0,01	0,84	713	1,30
Data15-15-0.6-50-2	1,00	3,53	717	0,03
Data15-15-0.6-50-3	1,00	1,90	714	1,77
Data15-15-0.6-50-4	1,00	3,08	719	0,08
Data15-15-0.6-50-5	0,02	1,74	712	0,01
Data15-15-0.6-50-6	1,50	2,98	718	0,02
Data15-15-0.6-50-7	0,02	0,95	718	0,15
Data15-15-0.6-50-8	0,30	1,59	715	0,04
Data15-15-0.6-50-9	0,70	3,18	719	0,10
Data15-15-0.8-100-0	1,10	5,29	779	19,82
Data15-15-0.8-100-1	0,70	8,63	781	21,15
Data15-15-0.8-100-2	0,50	4,91	785	23,88
Data15-15-0.8-100-3	0,10	5,63	778	–
Data15-15-0.8-100-4	1,70	4,37	784	17,46
Data15-15-0.8-100-5	0,40	5,68	784	25,14

Continua na próxima página

Tabela 15 – Continuação da página anterior

Instância	gap LP	gap $\Delta$	tempo Alg 4	gap SCF
Data15-15-0.8-100-6	1,00	4,38	781	22,51
Data15-15-0.8-100-7	1,20	4,93	784	17,88
Data15-15-0.8-100-8	0,60	3,89	788	14,43
Data15-15-0.8-100-9	1,20	3,45	789	7,80
Data15-15-0.8-50-0	0,70	1,46	732	0,13
Data15-15-0.8-50-1	0,02	1,34	726	0,88
Data15-15-0.8-50-2	1,10	2,04	731	2,03
Data15-15-0.8-50-3	0,03	1,49	727	2,41
Data15-15-0.8-50-4	1,40	2,69	728	0,09
Data15-15-0.8-50-5	0,60	1,80	724	0,21
Data15-15-0.8-50-6	0,40	0,91	732	2,19
Data15-15-0.8-50-7	1,10	2,20	733	0,14
Data15-15-0.8-50-8	1,50	3,27	723	0,05
Data15-15-0.8-50-9	0,10	1,14	728	0,04
$ I  = 25 \quad  T  = 10$				
Data25-10-0.6-100-0	0,90	2,33	1336	7,63
Data25-10-0.6-100-1	0,52	2,28	1348	10,15
Data25-10-0.6-100-2	1,30	2,29	1349	4,29
Data25-10-0.6-100-3	0,02	1,96	1343	8,94
Data25-10-0.6-100-4	0,40	2,50	1347	6,58
Data25-10-0.6-100-5	0,70	2,18	1354	8,95
Data25-10-0.6-100-6	0,01	1,57	1343	6,02
Data25-10-0.6-100-7	0,13	2,66	1337	6,26
Data25-10-0.6-100-8	0,04	2,43	1342	13,61
Data25-10-0.6-100-9	0,59	2,27	1346	5,55
Data25-10-0.6-50-0	0,30	2,05	1290	7,13
Data25-10-0.6-50-1	1,90	2,44	1286	4,11
Data25-10-0.6-50-2	1,10	2,73	1298	4,01
Data25-10-0.6-50-3	0,02	1,79	1283	6,02
Data25-10-0.6-50-4	1,60	2,73	1293	5,13
Data25-10-0.6-50-5	1,40	1,53	1291	2,13
Data25-10-0.6-50-6	0,10	2,79	1283	2,11
Data25-10-0.6-50-7	0,01	1,98	1288	0,05
Data25-10-0.6-50-8	0,01	0,68	1300	0,93
Data25-10-0.6-50-9	0,10	2,06	1287	0,31
Data25-10-0.8-100-0	1,50	4,32	1347	–
Data25-10-0.8-100-1	0,60	4,67	1365	–
Data25-10-0.8-100-2	0,20	3,21	1363	–
Data25-10-0.8-100-3	0,90	2,81	1350	–
Data25-10-0.8-100-4	0,70	2,62	1351	–
Data25-10-0.8-100-5	0,02	3,25	1364	–
Data25-10-0.8-100-6	0,02	1,16	1356	7,41
Data25-10-0.8-100-7	1,70	3,61	1363	10,65
Data25-10-0.8-100-8	0,80	3,69	1349	–
Data25-10-0.8-100-9	0,02	1,96	1350	–
Data25-10-0.8-50-0	0,03	1,26	1298	2,95
Data25-10-0.8-50-1	0,40	2,51	1311	1,14

Continua na próxima página

Tabela 15 – Continuação da página anterior

Instância	gap LP	gap $\Delta$	tempo Alg 4	gap SCF
Data25-10-0.8-50-2	0,60	3,43	1299	0,14
Data25-10-0.8-50-3	0,02	2,56	1312	1,22
Data25-10-0.8-50-4	0,02	1,41	1312	3,09
Data25-10-0.8-50-5	0,02	1,82	1301	1,15
Data25-10-0.8-50-6	1,80	4,60	1311	0,10
Data25-10-0.8-50-7	0,80	1,09	1304	1,51
Data25-10-0.8-50-8	0,02	0,51	1300	1,41
Data25-10-0.8-50-9	1,40	2,22	1306	1,10
$ I  = 25 \quad  T  = 15$				
Data25-15-0.6-100-0	0,70	4,73	1630	–
Data25-15-0.6-100-1	0,30	4,11	1639	41,29
Data25-15-0.6-100-2	0,01	2,36	1631	–
Data25-15-0.6-100-3	1,00	5,02	1636	43,70
Data25-15-0.6-100-4	0,03	2,67	1646	–
Data25-15-0.6-100-5	0,02	3,85	1641	52,58
Data25-15-0.6-100-6	0,03	2,85	1645	40,68
Data25-15-0.6-100-7	0,20	3,70	1628	–
Data25-15-0.6-100-8	0,05	3,48	1637	–
Data25-15-0.6-100-9	1,60	3,66	1636	34,62
Data25-15-0.6-50-0	0,02	2,99	1579	14,67
Data25-15-0.6-50-1	0,03	3,13	1577	13,76
Data25-15-0.6-50-2	0,90	4,01	1584	–
Data25-15-0.6-50-3	0,02	2,12	1568	13,66
Data25-15-0.6-50-4	0,03	2,95	1569	13,52
Data25-15-0.6-50-5	0,02	3,34	1593	–
Data25-15-0.6-50-6	0,02	2,95	1586	14,28
Data25-15-0.6-50-7	0,02	2,96	1577	13,95
Data25-15-0.6-50-8	1,60	4,57	1585	–
Data25-15-0.6-50-9	0,80	4,41	1585	13,17
Data25-15-0.8-100-0	0,03	2,66	1648	–
Data25-15-0.8-100-1	1,60	5,99	1656	–
Data25-15-0.8-100-2	0,20	3,06	1659	–
Data25-15-0.8-100-3	0,02	2,17	1652	–
Data25-15-0.8-100-4	0,01	3,68	1645	–
Data25-15-0.8-100-5	0,90	4,21	1647	–
Data25-15-0.8-100-6	1,30	4,39	1646	–
Data25-15-0.8-100-7	0,02	3,29	1653	–
Data25-15-0.8-100-8	1,70	4,73	1645	–
Data25-15-0.8-100-9	0,02	3,04	1650	–
Data25-15-0.8-50-0	0,03	2,91	1590	–
Data25-15-0.8-50-1	0,03	2,82	1594	56,21
Data25-15-0.8-50-2	0,02	2,72	1587	–
Data25-15-0.8-50-3	0,80	3,96	1582	–
Data25-15-0.8-50-4	1,70	4,64	1585	–
Data25-15-0.8-50-5	0,03	2,28	1581	–
Data25-15-0.8-50-6	0,02	2,61	1607	–
Data25-15-0.8-50-7	1,40	4,71	1586	–

Continua na próxima página

Tabela 15 – Continuação da página anterior

Instância	gap LP	gap Δ	tempo Alg 4	gap SCF
Data25-15-0.8-50-8	1,90	5,71	1586	–
Data25-15-0.8-50-9	1,40	3,83	1595	21,12

Quanto ao LB reportado, observa-se que seu desvio médio, em relação à relaxação linear de **EPN'+cuts**, foi, em média, de 0,53% (*versus* 0,06% para o PCTSP, média para todos os grupos). Novamente, o Alg. 4 foi capaz de retornar um intervalo pequeno para a solução ótima, indicado pela coluna *gap Δ*. Seu valor médio de 3,05% é menor do que o *gap SCF* médio de 9,89%, isso quando é possível encontrar uma solução viável com a formulação **SCF** dentro do limite de tempo de 1800 seg. Ou seja, para um tempo médio de execução de 1043 seg., diante de 1800 seg. (meia hora) da formulação **SCF**, o Alg. 4 gerou um intervalo para a solução ótima menor do que o Gurobi (por meio de **SCF**). Destaca-se ainda que para 27 instâncias, a formulação **SCF** não encontrou nenhuma solução viável, enquanto o Alg. 4 encontrou, no pior caso (Data15-15-0.8-100-1), uma solução com um *gap Δ* de 8,63%, sendo que para apenas 10 das 138 instâncias o *gap Δ* foi maior que 5%. Para esta mesma instância, o *gap SCF* é de 21,15% e para outras 47 instâncias o *gap SCF* é maior que 10%.

### 5.3.2 Comparação dos UB's

Nesta seção são apresentados os resultados para os UB's obtidos pelo Algoritmo 4. Eles são comparados com os trabalhos de James e Almada-Lobo (2011) e Guimarães, Klabjan e Almada-Lobo (2013). Estes trabalhos possuem os melhores UB's para as instâncias consideradas. James e Almada-Lobo (2011) utilizam uma metaheurística, nomeada naquele trabalho como *INSRF*, baseada no *Iterative Neighborhood Search* combinada com um procedimento de construção *relax-and-fix*. Guimarães, Klabjan e Almada-Lobo (2013) adotam duas abordagens diferentes baseadas em metaheurísticas (*matheuristics*) que combinam os conceitos de *pricing* com heurísticas de construção e busca local baseadas em MIP's, nomeadas como  $P\&MIP^{rnd}$  e  $P\&MIP^{eval}$ . Elas diferem entre si no processo de seleção da estrutura de vizinhança da busca local. O critério para a comparação é o desvio relativo (*gap*) entre o LB obtido por James e Almada-Lobo (2011) e o valor do UB médio obtido pelo método considerado.

A Tabela 16 apresenta os resultados médios para as 240 instâncias. Cada linha da tabela apresenta o valor médio agrupado por elemento  $P, T, \theta, cut$ .

O Alg. 4 consegue gerar bons UB's, porém ele não encontrou resultados melhores que o algoritmo  $P\&MIP^{eval}$ , que obteve o melhor resultado médio 1,12%

Tabela 16 – Resultados da comparação da qualidade dos UB’s.

		$P\&MIP^{rnd}$	$P\&MIP^{eval}$	$INSRF$	Alg. 4
$P$	15	1,34	1,33	1,33	1,81
	25	0,94	0,90	0,99	1,91
$T$	5	0,79	0,79	0,67	1,11
	10	1,15	1,14	1,41	1,82
	15	1,47	1,42	1,41	2,66
$cut$	0,6	1,04	1,01	1,04	1,69
	0,8	1,23	1,22	1,29	2,04
$\theta$	50	0,35	0,30	0,31	1,10
	100	1,93	1,93	2,02	2,62
	média	1,14	<b>1,12</b>	1,16	1,86

(versus 1,86% do Alg. 4). Entretanto, ele é capaz de encontrar bons *gaps* de otimalidade, enquanto os demais algoritmos encontram apenas UB’s.

### 5.4 Resultados das desigualdades válidas

Analogamente à Seção 3.4, a seguir são apresentados os resultados para o impacto das desigualdades válidas, apresentadas na Seção 4.4.3, no resultado final do Algoritmo 4, para o LSSP.

A Tabela 17, assim como a Tabela 12, mostra o impacto do uso das desigualdades válidas para o LSSP. Nela também é mostrado o *gap*  $\Delta$  e o número de iterações. A primeira coluna indica as instâncias. Novamente, foram escolhidas somente as que mostraram-se mais difíceis de se resolver. As quatro colunas relativas ao *gap*  $\Delta$  mostram, respectivamente, os resultados para os casos nos quais: sem as desigualdades válidas; apenas uma família; e ambas as famílias de desigualdades; tal que:

- $\Delta_{no}$  indica o caso no qual nenhuma desigualdade válida é utilizada.
- $\Delta_{ls}$  indica o caso no qual apenas a família de desigualdades ( $L, S$ );
- $\Delta_w$  indica o caso no qual apenas a família de desigualdades  $W_t$ ;
- $\Delta_{wls}$  indica o caso no qual ambas as famílias de desigualdades foram utilizadas (versão final);

As demais colunas apresentam o número de iterações na mesma ordem do *gap*  $\Delta$ . Os valores destacados com negrito indicam o melhor resultado.

Tabela 17 – Impacto das desigualdades válidas no Algoritmo 4 – LSSP

Instância	<i>gap</i> $\Delta$				Iterações			
	$\Delta_{no}$	$\Delta_{ls}$	$\Delta_w$	$\Delta_{wls}$	$\Delta_{no}$	$\Delta_{ls}$	$\Delta_w$	$\Delta_{wls}$
$ I  = 15,  T  = 15$								
Data15-15-0.6-100-0	10,98	3,24	9,81	<b>2,84</b>	2802	2282	2293	1647

Continua

Tabela 17 – Continuação

Instância	gap $\Delta$				Iterations			
	$\Delta_{no}$	$\Delta_{ls}$	$\Delta_w$	$\Delta_{wls}$	$\Delta_{no}$	$\Delta_{ls}$	$\Delta_w$	$\Delta_{wls}$
Data15-15-0.6-50-0	10,83	3,92	9,49	<b>1,66</b>	2575	2108	2643	2100
Data15-15-0.8-100-0	11,23	5,79	9,99	<b>5,29</b>	2788	2317	2582	2301
Data15-15-0.8-50-0	6,88	3,53	5,66	<b>1,46</b>	2636	2137	2580	2120
Valor Médio	9,98	4,12	8,74	<b>2,81</b>	2700	2211	2525	2042
$ I  = 25,  T  = 10$								
Data25-10-0.6-100-0	11,81	3,50	10,57	<b>2,33</b>	1385	1270	1346	1267
Data25-10-0.6-50-0	13,51	3,03	12,30	<b>2,05</b>	1335	1227	1306	1227
Data25-10-0.8-100-0	10,83	6,04	9,30	<b>4,32</b>	1397	1255	1357	1249
Data25-10-0.8-50-0	7,68	2,34	6,01	<b>1,26</b>	1336	1202	1310	1191
Valor Médio	10,95	3,73	9,54	<b>2,49</b>	1363	1239	1330	1234
$ I  = 25,  T  = 15$								
Data25-15-0.6-100-0	12,46	5,18	11,32	<b>4,22</b>	1650	1535	1561	1533
Data25-15-0.6-50-0	11,20	3,79	9,86	<b>1,22</b>	1594	1471	1573	1469
Data25-15-0.8-100-0	10,76	7,10	9,24	<b>4,16</b>	1662	1556	1603	1543
Data25-15-0.8-50-0	9,08	2,88	7,71	<b>0,60</b>	1613	1478	1489	1477
Valor Médio	10,87	4,74	9,53	<b>2,55</b>	1630	1510	1557	1506

Novamente, é possível observar que cada corte foi individualmente útil para melhorar os resultados, bem como a convergência do algoritmo. A redução média do *gap*  $\Delta$  foi, aproximadamente, de 11% para 3%. Ou seja, uma redução menor do que para o PCTSP (17% para 7% e 17% para 2%). Bem como no PCTSP, a convergência do algoritmo também é aperfeiçoada. De fato, o número de iterações necessário foi sempre menor para a versão do algoritmo com o uso dos planos de corte. Portanto, o impacto positivo no desempenho, do uso de desigualdades válidas, de acordo com o esquema *relax-and-fix* embutido dentro do Algoritmo 4, também foi confirmado para o LSSP.



## 6 Considerações Finais

Este trabalho trata do Problema do Caixeiro Viajante com Coleta Seletiva de Prêmios, conhecido na literatura inglesa como *Prize Collecting Traveling Salesman Problem* – PCTSP. Este problema é uma generalização do amplamente estudado Problema do Caixeiro Viajante (*Traveling Salesman Problem* – TSP). A diferença entre eles reside no fato de que, para o PCTSP, nem todos os vértices precisam ser visitados, neste caso, uma penalidade é incorrida. O objetivo do PCTSP é encontrar uma rota de custo mínimo, que inicie no vértice raiz  $r$ , e visite cada vértice  $i$ , no máximo, uma única vez. A rota termina no vértice raiz depois de coletar um prêmio mínimo  $g$ . O custo da rota é composto pelo custo de percorrer o arco  $(i, j)$  e pela penalidade incorrida para todo vértice não visitado. Ainda, o trabalho foca a versão assimétrica, ou seja, o custo do arco  $(i, j)$  é diferente do arco  $(j, i)$ .

O PCTSP pertence a um grupo de problemas de Otimização Combinatória conhecido como Problemas de Sub-Rotas – StP. Os problemas deste grupo são caracterizados por dois tipos de decisão: na seleção de quais itens estarão ativos na solução (programação); e seu respectivo sequenciamento (roteamento). Diversos problemas pertencem a este grupo, dentre os quais destaca-se o Problema de Dimensionamento de Lotes com Sequenciamento, conhecido na literatura inglesa como *Lot Sizing Scheduling Problem* – LSSP. Assim como os StP, ele consiste em determinar o tamanho do lote de produção de cada produto (programação), bem como a sequência na qual os produtos serão produzidos (roteamento). O objetivo é minimizar os custos envolvidos, sujeito às restrições de demanda e capacidade produtiva, em um horizonte de planejamento finito e conhecido.

Ambos os problemas, o PCTSP e o LSSP, bem como os demais StP são classificados como NP-difícil. O que significa que não se conhece um algoritmo capaz de encontrar a solução ótima em tempo hábil para instâncias maiores. Logo, uma abordagem diferente do método exato pode ser uma opção melhor para resolver estes problemas.

Inicialmente, o TSP é utilizado como fonte de ideias para resolver o PCTSP. Assim, as principais formulações (ou as mais fortes) para o TSP foram adaptadas, neste trabalho, para o PCTSP (Seções 2.3.1 a 2.3.4). Em seguida são propostas novas formulações para o PCTSP (Seções 2.3.5 e 2.3.6). Então, mostra-se o problema dual Lagrangeano baseado na formulação **EPN**, apresentada na Seção 4.4, bem como seus subproblemas Lagrangeanos e métodos de resolução. A partir deste problema dual Lagrangeano, propõe-se uma abordagem que incorpora técnicas heurísticas para obter

soluções ótimas ou quase ótimas e famílias de cortes de otimalidade e desigualdades válidas da literatura em um esquema *relax-and-cut*. Na Seção 2.4.3, tais famílias válidas para o PCTSP são apresentadas. Estes procedimentos são embutidos no Algoritmo Volume, o qual é utilizado para resolver o problema dual Lagrangeano resultante. O Algoritmo 4, derivado destes procedimentos, é apresentado na Seção 2.5.

Para realizar os experimentos computacionais são adaptadas instâncias da literatura, inicialmente, propostas para o TSP assimétrico. A Seção 3.1 apresenta estas instâncias, bem como as suas adaptações para o PCTSP.

Na sequência, as formulações consideradas são comparadas em termos da relaxação linear (Seção 3.2). A formulação **EPN**, proposta neste trabalho, dominou as demais formulações adaptadas da literatura do TSP assimétrico. Isto pode indicar que ela é mais forte que as demais. No entanto, tais formulações requerem grande esforço computacional para resolver o problema. A memória computacional e o tempo demandados, para rodar estas formulações em um *solver*, mostraram-se o maior desafio para a utilização das mesmas. De fato, para algumas instâncias não foi possível resolver nem mesmo a relaxação linear.

Na Seção 3.3 são mostrados os resultados obtidos com a abordagem Lagrangeana (Algoritmo 4), para o PCTSP. Ele mostrou-se bem-sucedido para o PCTSP, pois obtém soluções viáveis dentro de 1% da otimalidade na maioria dos casos. Ressalta-se que é possível obter soluções ótimas para boa parte das instâncias consideradas por meio da formulação **PSC** (a qual é mais fraca em termos da relaxação linear). Para tanto, utiliza-se o *solver* Gurobi sem limite de tempo, embora os tempos computacionais serem muito altos para instâncias maiores. Por outro lado, a abordagem de relaxação Lagrangeana proposta foi muito boa. Ele foi capaz de limitar o valor da solução ótima dessas grandes instâncias em pequenos intervalos de otimalidade, com um tempo computacional moderado. Especialmente, se comparado ao método exato por meio da formulação **PSC**.

Finalmente, a Seção 3.4 apresenta o impacto na abordagem proposta por meio do uso de famílias de cortes de otimalidade e de desigualdades válidas em um esquema *relax-and-cut*. Elas mostraram-se satisfatórias, pois como pode ser observado, elas foram capazes, não só de incrementar o LB do Algoritmo 4, como também em melhorar a convergência do mesmo. De fato, com o uso dessas famílias foi possível reduzir o número de iterações necessárias.

Com base no sucesso do algoritmo proposto para o PCTSP, a abordagem Lagrangeana foi novamente adotada para o LSSP. Dessa forma, as mesmas etapas da pesquisa são consideradas para o LSSP. Assim, depois de uma revisão da literatura, as principais formulações foram selecionadas e implementadas (Seções 4.3.1 e 4.3.2). As formulações, propostas originalmente neste trabalho para o PCTSP, são, então,

adaptadas ao LSSP (Seções 4.3.3 e 4.3.4). Em seguida, o problema dual Lagrangeano baseado na formulação **EPN'**, apresentada na Seção 4.3.3, a qual é uma adaptação de **EPN** para o LSSP, é apresentado. Este é sucedido pelas famílias de desigualdades válidas, retiradas da literatura, Seção 4.4.3. Finalmente, as modificações na estrutura do Algoritmo 4, adaptado ao LSSP, são mostradas na Seção 4.5.

O Cap. 5, apresenta os resultados dos experimentos computacionais para o LSSP. Inicialmente, a Seção 5.1 apresenta as instâncias retiradas de James e Almada-Lobo (2011), seguida pela comparação das formulações em termos da relaxação linear, na Seção 5.2. Como esperado, a formulação **EPN'** (a qual é uma adaptação de **EPN** para o LSSP), dominou mais uma vez as demais formulações da literatura. O que também confirma que ela pode ser mais forte que as demais. Analogamente ao PCTSP, tais formulações também requerem um esforço computacional muito alto para resolver o problema. Ou seja, demandam memória e tempo além do razoável, o que torna impraticável seu uso para resolver estes problemas por meio de *solvers* comerciais como o Gurobi.

Na Seção 5.3 os resultados provenientes do esquema heurístico proposto, adaptado ao LSSP, são reportados. Estes resultados, para o LSSP, repetiram o bom desempenho também observado para o PCTSP. Sobretudo em termos do limite inferior (LB) gerado pelo mesmo. O algoritmo foi muito bom para obter soluções viáveis em um tempo razoável. Ele destaca-se, especialmente, para as instâncias mais difíceis, nas quais o método exato não foi sequer capaz de encontrar soluções viáveis, em até uma hora. O que indica que ele é uma boa alternativa para gerar soluções razoáveis dentro de um pequeno intervalo de tempo para o LSSP.

Analogamente à Seção 3.4, a Seção 5.4 apresenta o impacto na abordagem proposta por meio do uso de desigualdades válidas em um esquema *relax-and-cut*. Novamente, elas mostraram-se muito boas em incrementar o LB do Algoritmo 4 e em melhorar a convergência do mesmo.

Os próximos passos desta pesquisa visam a adaptação e aplicação da formulação forte **EPN** aos demais Problemas de Subrota, bem como do TSP e seus problemas correlatos. Dessa forma, espera-se explorar a relaxação Lagrangeana por meio do Algoritmo 4 para estes problemas. Ainda, desenvolver metaheurísticas que explorem as informações do subproblema Lagrangeano para melhorar ainda mais a qualidade dos UB's obtidos, principalmente no LSSP.

Além disso, espera-se embutir a relaxação Lagrangeana, ao invés da Relaxação Linear, em esquemas *branch-and-cut* como, por exemplo, em Barahona e Ladányi (2006). Também pretende-se melhorar o desempenho do LB por meio de novos planos de corte. Isto pode viabilizar a aplicação destas formulações para problemas mais complexos e contribuir com metodologias que atendam problemas do mundo real.

## Referências

- AHUJA, R. K.; MEHLHORN, K.; ORLIN, J.; TARJAN, R. E. Faster algorithms for the shortest path problem. *Journal of the Association for Computing Machinery*, v. 37, n. 2, p. 213–223, 1990.
- ALMADA-LOBO, B.; KLABJAN, D.; CARRAVILLA, M. A.; OLIVEIRA, J. F. Single machine multi-product capacitated lot sizing with sequence-dependent setups. *International Journal of Production Research*, v. 45, n. 20, p. 4873–4894, 2007.
- ALMADA-LOBO, B.; OLIVEIRA, J. F.; CARRAVILLA, M. A. Production planning and scheduling in the glass container industry: A VNS approach. *International Journal of Production Economics*, v. 114, n. 1, p. 363 – 375, 2008.
- AMORIM, P.; PINTO-VARELA, T.; ALMADA-LOBO, B.; BARBÓSA-PÓVOA, A. P. F. D. Comparing models for lot-sizing and scheduling of single-stage continuous processes: Operations research and process systems engineering approaches. *Computers & Chemical Engineering*, v. 52, p. 177–192, 2013.
- ARAÚJO, S. A. de; ARENALES, M. N.; CLARK, A. R. Lot sizing and furnace scheduling in small foundries. *Computers & Operations Research*, v. 35, n. 3, p. 916–932, 2008.
- AUSIELLO, G.; BONIFACI, V.; LAURA, L. The online prize-collecting traveling salesman problem. *Information Processing Letters*, v. 107, n. 6, p. 199–204, 2008.
- AUSIELLO, G.; DEMANGE, M.; LAURA, L.; PASCHOS, V. Algorithms for the on-line quota traveling salesman problem. *Information Processing Letters*, v. 92, n. 2, p. 89–94, 2004.
- BAHIENSE, L.; BARAHONA, F.; PORTO, O. Solving steiner tree problems in graphs with Lagrangian relaxation. *Journal of Combinatorial Optimization*, v. 7, p. 259–282, 2003.
- BAHIENSE, L.; MACULAN, N.; SAGASTIZÁBAL, C. The volume algorithm revisited: relation with bundle methods. *Mathematical Programming*, v. 94, p. 41–69, 2002.
- BALAS, E. The prize collecting traveling salesman problem. *Networks*, v. 19, p. 621–636, 1989.
- BALAS, E. The prize collecting traveling salesman problem: II polyhedral results. *Networks*, v. 25, p. 199–216, 1995.
- BALAS, E.; MARTIN, G. *ROLL-A-ROUND: software package for scheduling the rounds of a rolling mill*. 104, Maples Heights Road, Pittsburgh, 1985.
- BALDO, T. A.; SANTOS, M. O.; ALMADA-LOBO, B.; MORABITO, R. An optimization approach for the lot sizing and scheduling problem in the brewery industry. *Computers & Industrial Engineering*, v. 72, p. 58–71, 2014.

- BAR-YEHUDA, R.; EVEN, G.; SHAHAR, S. M. On approximating a geometric prize-collecting traveling salesman problem with time windows. *Journal of Algorithms*, v. 55, n. 1, p. 76–92, 2005.
- BARAHONA, F.; ANBIL, R. The volume algorithm: producing primal solutions with a subgradient method. *Mathematical Programming*, v. 87, p. 385–399, 2000.
- BARAHONA, F.; ANBIL, R. On some difficult linear programs coming from set partitioning. *Discrete Applied Mathematics*, v. 118, n. 1, p. 3 – 11, 2002.
- BARAHONA, F.; LADÁNYI, L. Branch and cut based on the volume algorithm: Steiner trees in graphs and max-cut. *RAIRO - Operations Research*, v. 40, n. 1, p. 53–73, 2006.
- BARANY, I.; ROY, T. J. V.; WOLSEY, L. A. Strong formulations for multi-item capacitated lot sizing. *Management Science*, v. 30, n. 10, p. 1255–1261, 1984.
- BARTAL, Y.; GOTTLIEB, L.-A.; KRAUTHGAMER, R. The traveling salesman problem: Low-dimensionality implies a polynomial time approximation scheme. *SIAM Journal on Computing*, v. 45, n. 4, p. 1563–1581, 2016.
- BEASLEY, J. E. Lagrangian relaxation. In: REEVES, C. R. (Ed.). *Modern Heuristic Techniques for Combinatorial Problems*. New York, USA: John Wiley & Sons, 1993. p. 243–303.
- BERNARDES, E. D.; ARAUJO, S. A. d.; RANGEL, S. Reformulação para um problema integrado de dimensionamento e sequenciamento de lotes. *Pesquisa Operacional*, v. 30, p. 637–655, 2010.
- BÉRUBÉ, J.-F.; GENDREAU, M.; POTVIN, J.-Y. A branch-and-cut algorithm for the undirected prize collecting traveling salesman problem. *Networks*, v. 54, p. 56–67, 2009.
- BIENSTOCK, D.; GOEMANS, M. X.; SIMCHI-LEVI, D.; WILLIAMSON, D. A note on the prize collecting traveling salesman problem. *Mathematical Programming*, v. 59, p. 413–420, 1993.
- BITRAN, G. R.; YANASSE, H. H. Computational complexity of the capacitated lot size problem. *Management Science*, v. 28, n. 10, p. 1174–1186, 1982.
- BURKE, E.; KENDALL, G.; NEWALL, J.; HART, E.; ROSS, P.; SCHULENBURG, S. Hyper-heuristics: An emerging direction in modern search technology. In: GLOVER, F.; KOCHENBERGER, G. A. (Ed.). *Handbook of Metaheuristics: International Series in Operations Research & Management Science*. : Springer, 2003. v. 57, cap. 16, p. 457–474.
- CARVALHO, D. M.; NASCIMENTO, M. C. Lagrangian heuristics for the capacitated multi-plant lot sizing problem with multiple periods and items. *Computers & Operations Research*, v. 71, p. 137–148, 2016.
- CHAN, T.-H. H.; JIANG, H.; JIANG, S. H.-C. A unified PTAS for prize collecting TSP and steiner tree problem in doubling metrics. *ACM Transactions on Algorithms*, v. 16, n. 2, Article No. 24, p. 1–23, 2020.

- CHAVES, A. A.; BIAJOLI, F. A.; MINE, O. M.; SOUZA, M. J. F. Metaheurísticas híbridas para resolução do problema do caixeiro viajante com coleta de prêmios. *Revista Produção*, v. 17, p. 263–272, 2007.
- CIRASELLA, J.; JOHNSON, D.; MCGEOCH, L. A.; ZHANG, W. The asymmetric traveling salesman problem: Algorithms, instance generators, and tests. In: BUCHSBAUM, A.; SNOEYINK, J. (Ed.). *Algorithm Engineering and Experimentation. ALENEX 2001, Lecture Notes in Computer Science.* : Springer, 2001. v. 2153, p. 32–59.
- CLAUS, A. A new formulation for the travelling salesman problem. *SIAM Journal on Algebraic Discrete Methods*, v. 5, p. 21–25, 1984.
- COPIE, K.; WÖRBELAUER, M.; MEYER, H.; TEMPELMEIER, H. Simultaneous lot sizing and scheduling problems: a classification and review of models. *OR Spectrum*, v. 39, n. 1, p. 1–64, 2017.
- CROES, G. A. A method for solving traveling-salesman problems. *Operations Research*, v. 6, p. 791–812, 1958.
- DANTZIG, G. B.; FULKERSON, D. R.; JOHNSON, S. M. Solution of a large scale traveling salesman problem. *Journal of the Operations Research Society of America*, v. 2, n. 4, p. 393–410, 1954.
- DELL'AMICO, M.; MAFFIOLI, F.; SCIOMACHEN, A. A Lagrangian heuristic for the prize collecting travelling salesman problem. *Annals of Operations Research*, v. 81, p. 289–306, 1998.
- DELL'AMICO, M.; MAFFIOLI, F.; VARBRAND, P. On prize-collecting tours and the asymmetric travelling salesman problem. *International Transactions in Operational Research*, v. 2, p. 297–308, 1995.
- DESROCHERS, M.; LAPORTE, G. Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, v. 10, p. 27–36, 1991.
- DIAZ-MADROÑERO, M.; MULA, J.; ANDRES, B.; POLER, R.; SANCHIS, S. Capacitated lot-sizing and scheduling problem for second-tier suppliers in the automotive sector. In: VILES, E.; ORMAZÁBAL, M.; LLEÓ, A. (Ed.). *Closing the Gap Between Practice and Research in Industrial Engineering.* : Springer International Publishing, 2018. p. 121–129.
- DINITZ, Y. Dinitz' algorithm: The original version and even's version. In: GOLDREICH, O.; ROSENBERG, A. L.; SELMAN, A. L. (Ed.). *Theoretical Computer Science. Lecture Notes in Computer Science.* : Springer, 2006. v. 3895, p. 218–240.
- DREXL, A.; KIMMS, A. Lot sizing and scheduling — survey and extensions. *European Journal of Operational Research*, v. 99, n. 2, p. 221 – 235, 1997.
- DUECK, G.; SCHEUER, T. Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics*, v. 90, n. 1, p. 161 – 175, 1990.

- EPPEN, G. D.; MARTIN, R. K. Solving multi-item capacitated lot-sizing problems using variable redefinition. *Operations Research*, v. 35, n. 6, p. 832–848, 1987.
- FEILLET, D.; DEJAX, P.; GENDREAU, M. Traveling salesman problems with profits. *Transportation Science*, v. 39, p. 188–205, 2005.
- FEO, T. A.; RESENDE, M. G. C. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, v. 6, p. 109–133, 1995.
- 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 & Operations Research*, v. 37, n. 4, p. 684 – 691, 2010.
- FISCHETTI, M.; TOTH, P. An additive approach for the optimal solution of the prize-collecting travelling salesman problem. In: GOLDEN, B. L. (Ed.). *Vehicle Routing: Methods and Studies. Studies in Management Science and Systems.* : Elsevier, 1988. v. 16, p. 319–343.
- FISCHETTI, M.; TOTH, P. An additive bounding procedure for the asymmetric travelling salesman problem. *Mathematical Programming*, v. 53, p. 173–197, 1992.
- FISCHETTI, M.; TOTH, P. A polyhedral approach to the asymmetric traveling salesman problem. *Management Science*, v. 43, n. 11, p. 1520–1536, 1997.
- FISCHETTI, M.; TOTH, P.; VIGO, D. A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs. *Operations Research*, v. 42, p. 846–859, 1994.
- FISHER, M. L. An applications oriented guide to Lagrangian relaxation. *Interfaces*, v. 15, n. 2, p. 10–21, 1985.
- FLEISCHMANN, B. The discrete lot-sizing and scheduling problem. *European Journal of Operational Research*, v. 44, n. 3, p. 337–348, 1990.
- FOX, K. R.; GAVISH, B.; GRAVES, S. C. Technical note-an n-constraint formulation of the time-dependent traveling salesman problem. *Operations Research*, v. 28, n. 4, p. 1018–1021, 1980.
- FRANGIONI, A.; GENDRON, B.; GORGONE, E. On the computational efficiency of subgradient methods: a case study with Lagrangian bounds. *Mathematical Programming Computation*, v. 9, p. 573–604, 2017.
- GAUDREAU, J.; FRAYRET, J.-M.; ROUSSEAU, A.; D'AMOURS, S. Combined planning and scheduling in a divergent production system with co-production: A case study in the lumber industry. *Computers & Operations Research*, v. 38, n. 9, p. 1238 – 1250, 2011.
- GAVISH, B.; GRAVES, S. C. *The travelling salesman problem and related problems.* Massachusetts, USA: Massachusetts Institute of Technology, 1978.
- GENDREAU, M.; HERTZ, A.; LAPORTE, G. New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, v. 40, p. 1086–1094, 1992.

- GIMADI, E.; TSIDULKO, O. Asymptotically optimal algorithms for the prize-collecting traveling salesman problem on random inputs. In: MATSATSINIS, N. F.; MARINAKIS, Y.; PARDALOS, P. M. (Ed.). *Learning and Intelligent Optimization. LION 2019.* : Springer, 2020. v. 11968, p. 201–207.
- GODINHO, M.; GOUVEIA, L.; MAGNANTI, T. Combined route capacity and route length models for unit demand vehicle routing problems. *Discrete Optimization*, v. 5, p. 350–372, 2008.
- GODINHO, M. T.; GOUVEIA, L.; PESNEAU, P. *Natural and extended formulations for the time-dependent traveling salesman problem.* 2010. CIO Working Paper 8-2010, Universidade de Lisboa.
- GODINHO, M. T.; GOUVEIA, L.; PESNEAU, P. On a time-dependent formulation and an updated classification of atsp formulations. In: MAHJOUB, A. R. (Ed.). *Progress in Combinatorial Optimization.* : Wiley, 2011. p. 223–254.
- GOEMANS, M. X. Combining approximation algorithms for the prize-collecting TSP. *CoRR*, abs/0910.0553, 2009.
- GOEMANS, M. X.; WILLIAMSON, D. P. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, v. 24, n. 2, p. 296–317, 1995.
- GOMES, L.; DINIZ, V.; MARTINHON, C. A. An hybrid grasp+vnd metaheuristic for the prize-collecting traveling salesman problem. In: *XXXII Simposio Brasileiro de Pesquisa Operacional, SBPO.* Viçosa, Brazil: , 2000. p. 1657–1665.
- GOUVEIA, L.; PIRES, J. M. The asymmetric travelling salesman problem: on generalizations of disaggregated miller-tucker-zemlin constraints. *Discrete Applied Mathematics*, v. 112, p. 129–145, 2001.
- GUIGNARD, M. Lagrangian relaxation. *Top*, v. 11, p. 151–200, 2003.
- GUIMARÃES, L.; KLABJAN, D.; ALMADA-LOBO, B. Pricing, relaxing and fixing under lot sizing and scheduling. *European Journal of Operational Research*, v. 230, n. 2, p. 399–411, 2013.
- GUIMARÃES, L.; KLABJAN, D.; ALMADA-LOBO, B. Modeling lotsizing and scheduling problems with sequence dependent setups. *European Journal of Operational Research*, v. 239, n. 3, p. 644–662, 2014.
- GUNAWAN, A.; LAU, H. C.; VANSTEENWEGEN, P. Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, v. 255, n. 2, p. 315–332, 2016.
- GUPTA, D.; MAGNUSSON, T. The capacitated lot-sizing and scheduling problem with sequence-dependent setup costs and setup times. *Computers & Operations Research*, v. 32, n. 4, p. 727 – 747, 2005.
- HAASE, K.; KIMMS, A. Lot sizing and scheduling with sequence-dependent setup costs and times and efficient rescheduling opportunities. *International Journal of Production Economics*, v. 66, n. 2, p. 159 – 169, 2000.



- JAMES, R. J.; ALMADA-LOBO, B. Single and parallel machine capacitated lotsizing and scheduling: New iterative MIP-based neighborhood search heuristics. *Computers & Operations Research*, v. 38, n. 12, p. 1816 – 1825, 2011.
- KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. *Science*, v. 220, n. 4598, p. 671–680, 1983.
- KLEMENT, N.; ABDELJAOUAD, M. A.; PORTO, L.; SILVA, C. Lot-sizing and scheduling for the plastic injection molding industry - a hybrid optimization approach. *Applied Sciences*, v. 11, n. 3: 1202, 2021.
- KRARUP, J.; BILDE, O. Plant location, set covering and economic lot size: an  $O(mn)$ -algorithm for structured problems. *International Series of Numerical Mathematics*, v. 36, p. 155–180, 1977.
- KUBO, M.; KASUGAI, H. On symmetric subtour problems. *Japan Journal of Industrial and Applied Mathematics*, v. 9, p. 383–396, 1992.
- KUMAR, R.; LI, H. *On Asymmetric TSP: Transformation to Symmetric TSP and Performance Bound*. 2000. KY 40506-0046, University of Kentucky.
- LANGEVIN, A.; SOUMIS, F.; DESROSIERS, J. Classification of travelling salesman problem formulations. *Operational Research Letters*, v. 9, p. 127–132, 1990.
- LAPORTE, G.; MARTELLO, S. The selective travelling salesman problem. *Discrete Applied Mathematics*, v. 26, p. 193–207, 1990.
- LEIFER, A. C.; ROSENWEIN, M. B. Strong linear programming relaxations for the orienteering problem. *European Journal of Operational Research*, v. 73, p. 517–523, 1994.
- LIN, S. Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, Blackwell Publishing Ltd, v. 44, p. 2245–2269, 1965.
- LIU, L.; WANG, Q.; GOLDEN, B. L. A multifaceted heuristic for the orienteering problem. *Naval Research Logistics*, v. 35, p. 359–366, 1988.
- LOPEZ, L.; CARTER, M. W.; GENDREAU, M. The hot strip mill production scheduling problem: A tabu search approach. *European Journal of Operational Research*, v. 106, p. 317–335, 1998.
- LUCENA, A. Non delayed relax-and-cut algorithms. *Annals of Operations Research*, v. 140, p. 375–410, 2005.
- MAHDIEH, M.; CLARK, A.; BIJARI, M. A novel flexible model for lot sizing and scheduling with non-triangular, period overlapping and carryover setups in different machine configurations. *Flexible Services and Manufacturing Journal*, v. 30, n. 4, p. 884–923, 2018.
- MANNE, A. S. Capacity expansion and probabilistic growth. *Econometrica*, v. 29, n. 4, p. 632–649, 1961.
- MARINELLI, F.; NENNI, M.; SFORZA, A. Capacitated lot sizing and scheduling with parallel machines and shared buffers: A case study in a packaging company. *Annals of Operations Research*, v. 150, p. 177–192, 2007.

- MARTÍNEZ, K. Y.; TOSO, E. A.; MORABITO, R. Production planning in the molded pulp packaging industry. *Computers & Industrial Engineering*, v. 98, p. 554 – 566, 2016.
- MELEGA, G.; ARAUJO, S. D.; MORABITO, R. Mathematical model and solution approaches for integrated lot-sizing, scheduling and cutting stock problems. *Annals of Operations Research*, v. 295, p. 1–42, 2020.
- MELO, V. A.; MARTINHON, C. A. Metaheurísticas híbridas para o problema do caixeiro viajante com coleta de prêmios. In: *Anais do XXXVI Simpósio Brasileiro de Pesquisa Operacional, SBPO*. São João Del Rei, MG: , 2004. p. 1295–1306.
- MENEZES, A. A.; CLARK, A.; ALMADA-LOBO, B. Capacitated lot-sizing and scheduling with sequence-dependent, period-overlapping and non-triangular setups. *Journal of Scheduling*, v. 14, n. 2, p. 209–219, 2011.
- MEYR, H. Simultaneous lotsizing and scheduling by combining local search with dual reoptimization. *European Journal of Operational Research*, v. 120, n. 2, p. 311 – 326, 2000.
- MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. Integer programming formulation of traveling salesman problems. *Journal of the Association for Computing Machinery*, v. 7, p. 326–329, 1960.
- MLADENOVIC, N.; HANSEN, P. Variable neighborhood search. *Computers & Operations Research*, v. 24, p. 1097–1100, 1997.
- OLIVEIRA, W. d.; SANTOS, M. A new branching rule to solve the capacitated lot sizing and scheduling problem with sequence dependent setups. *TEMA*, v. 18, p. 515–529, 2017.
- ONCAN, T.; ALTINEL, I. K.; LAPORTE, G. A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research*, v. 36, p. 637–654, 2009.
- PEDRO, O.; SALDANHA, R.; CAMARGO, R. A tabu search approach for the prize collecting traveling salesman problem. *Electronic Notes in Discrete Mathematics*, v. 41, p. 261–268, 2013.
- PEKKNY, J.; MILLER, D. An exact parallel algorithm for the resource constrained travelling salesman problem with application to scheduling with an aggregate deadline. In: *Anais do ACM Annual Computer Science Conference, CSC*. Washington, USA: , 1990. p. 208–214.
- PERRAUDAT, A.; DAUZÈRE-PÉRÈS, S.; MASON, S. J. Stochastic programming approaches for an energy-aware lot-sizing and sequencing problem with incentive. *International Journal of Production Research*, v. 0, n. 0, p. 1–23, 2021.
- POCHET, Y.; WOLSEY, L. A. *Production Planning by Mixed Integer Programming*. 1st. ed. : Springer Publishing Company, Incorporated, 2010.
- PORKKA, P.; VEPSÄLÄINEN, A. P. J.; KUULA, M. Multiperiod production planning carrying over set-up time. *International Journal of Production Research*, v. 41, n. 6, p. 1133–1148, 2003.

- REINELT, G. TSPLIB, a traveling salesman problem library. *ORSA Journal on Computing*, v. 3, p. 376–384, 1991.
- ROGERS, J. A computational approach to the economic lot scheduling problem. *Management Science*, v. 4, n. 3, p. 264–291, 1958.
- RÍOS-SOLÍS, Y. ; IBARRA-ROJAS, O. J.; CABO, M.; POSSANI, E. A heuristic based on mathematical programming for a lot-sizing and scheduling problem in mold-injection production. *European Journal of Operational Research*, v. 284, n. 3, p. 861 – 873, 2020.
- SARIN, S.; SHERALI, H.; BHOOTRA, A. New tighter polynomial length formulations for the asymmetric traveling salesman problem with and without precedence constraint. *Operations Research Letters*, v. 33, p. 62–70, 2005.
- SHERALI, H. D.; ADAMS, W. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal of Discrete Mathematics*, v. 3, p. 411–430, 1990.
- 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.
- SHERALI, H. D.; SARIN, S. C.; TSAI, P.-F. A class of lifted path and flow-based formulations for the asymmetric traveling salesman problem with and without precedence constraints. *Discrete Optimization*, v. 3, p. 20–32, 2006.
- SILVA, C.; MAGALHAES, J. M. Heuristic lot size scheduling on unrelated parallel machines with applications in the textile industry. *Computers & Industrial Engineering*, v. 50, n. 1, p. 76–89, 2006.
- SÛRIE, C.; STADTLER, H. The capacitated lot-sizing problem with linked lot sizes. *Management Science*, v. 49, p. 1039–1054, 2003.
- TALWAR, K. Bypassing the embedding: Algorithms for low dimensional metrics. In: *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*. 2004. (STOC '04), p. 281–290.
- TANG, L.; WANG, X. An iterated local search heuristic for the capacitated prize-collecting travelling salesman problem. *Journal of the Operational Research Society*, v. 59, n. 5, p. 590–599, 2008.
- TOSCANO, A.; FERREIRA, D.; MORABITO, R. Formulation and MIP-heuristics for the lot sizing and scheduling problem with temporal cleanings. *Computers & Chemical Engineering*, v. 142, n. 1: 107038, 2020.
- TOSO, E. A.; MORABITO, R.; CLARK, A. R. Lot sizing and sequencing optimisation at an animal-feed plant. *Computers & Industrial Engineering*, v. 57, n. 3, p. 813–821, 2009.
- TOTH, P. Dynamic programming algorithms for the zero-one knapsack problem. *Computing*, v. 25, p. 29–45, 1980.

VAEZ, P.; SABOUHI, F.; JABALAMELI, M. S. Sustainability in a lot-sizing and scheduling problem with delivery time window and sequence-dependent setup cost consideration. *Sustainable Cities and Society*, v. 51, p. 101718, 2019.

VERWEIJ, B.; AARDAL, K. The merchant subtour problem. *Mathematical Programming*, v. 94, p. 295–322, 2003.

VOHRA, R.; LEVY, L.; GOLDEN, B. L. The orienteering problem. *Naval Research Logistics*, v. 34, p. 307–318, 1987.

WONG, R. T. Integer programming formulations of the traveling salesman problems. In: *IEEE international conference of circuits and computers, (ICCC80)*. New York, USA: , 1980. p. 149–152.

YOUNG, C.; JOHNSON, D. S.; SMITH, M. D.; KARGER, D. R. Near-optimal intraprocedural branch alignment. In: *Proceedings of the ACM SIGPLAN 1997 Conference on Programming Language Design and Implementation*. 1997. (PLDI '97), p. 183–193.

YU, Q.; FANG, K.; ZHU, N.; MA, S. A matheuristic approach to the orienteering problem with service time dependent profits. *European Journal of Operational Research*, v. 273, n. 2, p. 488–503, 2019.

ZHANG, C.; GAO, Y.; YANG, L.; GAO, Z.; QI, J. Joint optimization of train scheduling and maintenance planning in a railway network: A heuristic algorithm using Lagrangian relaxation. *Transportation Research Part B: Methodological*, v. 134, p. 64–92, 2020.

## Apêndices

# APÊNDICE A – Resultados dos modelos inteiros e da Relaxação Linear de **EPN**

Este apêndice apresenta alguns resultados para a resolução do problema inteiro até a otimalidade e da relaxação linear de **EPN**. Os experimentos foram executados em um PC Intel® Core™ i7-2600, 3,40 GHz, 16 GB de RAM, sob uma distribuição Linux 64 bits. As formulações foram implementadas com a linguagem C, por meio da interface do *solver* Gurobi versão 7.5.2, 64 bits, o qual também é utilizado para tentar resolver as relaxações lineares.

Para o modelo inteiro, nesta seção foram consideradas algumas instâncias do grupo *realworld*, a saber: as *ft*, *krol24*, *ry48p*, as *ftv* e as *rbg*; totalizando 69 instâncias testadas.

A Tabela 18 apresenta os resultados para as instâncias do grupo *realworld* nas quais todas as 6 formulações resolveram até a otimalidade em até 3 horas. A primeira linha apresenta o número de problemas resolvidos (**#SOLVED**), seguida do tempo médio de execução (**AVG\_tempo**), calculado sobre as 25 instâncias apresentadas. As demais linhas detalham os tempos médios de execução.

Tabela 18 – Resultados das instâncias do grupo *realworld* resolvidas por todas as 6 formulações com tempo limite de 10800 segundos (PCTSP).

		Formulações					
		MCF	EnC	GSEC	PSC	EPN	OPB
		Seção 2.3.2	Seção 2.3.4	Seção 2.3.1	Seção 2.3.6	Seção 2.3.5	Seção 2.3.3
#SOLVED		44	25	69	69	44	34
AVG_tempo		53	3231	424	9	70	1112
Instância							
$\tau = 20$							
br17	tempo	0	104	6	0	0	2
ftv33	tempo	3	414	2	1	3	11
ftv35	tempo	6	932	32	1	9	28
ftv38	tempo	16	2138	39	5	42	45
ftv44	tempo	15	1082	370	2	23	40
ftv47	tempo	22	1702	66	3	54	113
ftv55	tempo	115	9824	117	8	115	5798
ftv64	tempo	96	8785	35	14	201	3567
p43	tempo	15	817	592	13	50	160
$\tau = 50$							
br17	tempo	0	141	5	0	0	2
ftv33	tempo	3	423	5	1	2	70

Continua na próxima página

Tabela 18 – Continuação

		Formulações					
		MCF	EnC	GSEC	PSC	EPN	OPB
		Seção 2.3.2	Seção 2.3.4	Seção 2.3.1	Seção 2.3.6	Seção 2.3.5	Seção 2.3.3
ftv35	tempo	6	412	17	1	6	18
ftv38	tempo	11	2953	21	3	14	43
ftv44	tempo	14	1452	200	2	12	44
ftv47	tempo	21	1690	159	3	35	114
ftv55	tempo	215	9858	117	9	203	10800
ftv64	tempo	97	8799	44	12	109	3550
p43	tempo	14	1007	625	17	30	148
ry48p	tempo	121	9703	2561	8	119	181
$\tau = 80$							
br17	tempo	0	241	7	1	0	1
ft53	tempo	135	1191	2804	64	168	262
ft70	tempo	339	8702	2502	56	299	2942
ftv33	tempo	5	1401	62	1	15	405
ftv35	tempo	26	3276	10	2	34	125
ftv38	tempo	52	5953	26	3	216	299
p43	tempo	15	1011	593	6	49	159

A Tabela 19 apresenta os resultados detalhados para as instâncias nas quais pelo menos uma das formulações resolveu até a otimalidade em até 3 horas. O desempenho é comparado por meio do tempo gasto para resolver as formulações. Nos casos para os quais não foi possível resolver até a otimalidade, o *gap* reportado pelo Gurobi é apresentado. Note que para as instâncias resolvidas, o *gap* é igual a zero.

Tabela 19 – Resultados da instâncias do grupo *realworld* resolvidas por pelo menos uma formulação com tempo limite de 10800 segundos (PCTSP).

		Formulações					
		MCF	EnC	GSEC	PSC	EPN	OPB
		Seção 2.3.2	Seção 2.3.4	Seção 2.3.1	Seção 2.3.6	Seção 2.3.5	Seção 2.3.3
Instância							
$\tau = 20$							
ft53	gap	0,0	0,1	0,0	0,0	0,0	0,0
	cpu	30	10901	38	46	29	46
ft70	gap	0,0	1,1	0,0	0,0	0,0	0,0
	cpu	72	10891	106	63	203	3335
ftv100	gap	1,1	–	0,0	0,0	0,0	–
	cpu	10834	–	6902	974	10803	–
ftv110	gap	0,1	–	0,0	0,0	0,1	–
	cpu	10851	–	7234	2132	10924	–
ftv120	gap	3,2	–	0,0	0,0	–	–
	cpu	10892	–	8330	2744	–	–
ftv130	gap	6,9	–	0,0	0,0	–	–
	cpu	10812	–	8923	3179	–	–
ftv140	gap	–	–	0,0	0,0	–	–
	cpu	–	–	10801	8262	–	–
ftv150	gap	–	–	0,0	0,0	–	–
	cpu	–	–	10209	9342	–	–
ftv160	gap	–	–	0,0	0,0	–	–

Continua na próxima página

Tabela 19 – Continuação

		Formulações					
		MCF	EnC	GSEC	PSC	EPN	OPB
		Seção 2.3.2	Seção 2.3.4	Seção 2.3.1	Seção 2.3.6	Seção 2.3.5	Seção 2.3.3
ftv170	cpu	–	–	8771	8199	–	–
	gap	–	–	0,0	0,0	–	–
ftv70	cpu	–	–	8807	10164	–	–
	gap	0,0	–	0,0	0,0	0,0	0,1
ftv80	cpu	375	–	150	15	731	10800
	gap	0,0	–	0,0	0,0	0,0	0,7
ftv90	cpu	2922	–	3118	723	3619	10912
	gap	0,7	–	0,0	0,0	0,0	0,9
kro124	cpu	10832	–	2120	825	10809	10812
	gap	0,0	–	0,0	0,0	0,1	–
ry48p	cpu	8425	–	7507	136	10823	–
	gap	0,0	0,1	0,0	0,0	0,0	0,0
ft53	cpu	187	10895	2703	14	142	3346
	gap	0,0	1,0	0,0	0,0	0,0	0,0
ft70	cpu	89	10806	2229	28	72	528
	gap	0,0	0,8	0,0	0,0	0,0	0,0
ftv100	cpu	134	10862	2406	38	256	4289
	gap	0,7	–	0,0	0,0	0,0	–
ftv110	cpu	10802	–	7007	995	10838	–
	gap	0,1	–	0,0	0,0	0,0	–
ftv120	cpu	10828	–	7012	2105	10879	–
	gap	1,3	–	0,0	0,0	–	–
ftv130	cpu	10953	–	8465	3278	–	–
	gap	–	–	0,0	0,0	–	–
ftv140	cpu	–	–	9523	3154	–	–
	gap	–	–	0,0	0,0	0,6	–
ftv150	cpu	–	–	10800	8231	11052	–
	gap	–	–	0,0	0,0	1,0	–
ftv160	cpu	–	–	10021	9124	11780	–
	gap	–	–	0,0	0,0	–	–
ftv170	cpu	–	–	8006	8192	–	–
	gap	–	–	0,0	0,0	–	–
ftv70	cpu	–	–	8114	10800	–	–
	gap	0,0	–	0,0	0,0	0,0	0,1
ftv80	cpu	362	–	148	13	745	10800
	gap	0,0	–	0,0	0,0	0,0	0,8
ftv90	cpu	3115	–	3236	734	5499	10920
	gap	0,1	–	0,0	0,0	0,0	0,9
kro124	cpu	10870	–	5972	827	10819	10892
	gap	0,0	–	0,0	0,0	0,0	1,0
ftv100	cpu	8399	–	8614	206	9464	10866
	gap	0,1	–	0,0	0,0	0,0	–
ftv110	cpu	10819	–	7091	1026	10823	–
	gap	1,1	–	0,0	0,0	0,0	–
ftv120	cpu	10873	–	7564	2295	10874	–
	gap	16,1	–	0,0	0,0	–	–
ftv130	cpu	10801	–	9022	2735	–	–
	gap	–	–	0,0	0,0	–	–
ftv140	cpu	–	–	9932	2973	–	–
	gap	–	–	0,0	0,0	–	–
ftv150	cpu	–	–	10798	8386	–	–
	gap	–	–	0,9	0,0	–	–
ftv160	cpu	–	–	10804	8192	–	–
	gap	–	–	1,1	0,0	–	–
	cpu	–	–	8806	8208	–	–

Continua na próxima página



Tabela 19 – Continuação

		Formulações					
		MCF	EnC	GSEC	PSC	EPN	OPB
		Seção 2.3.2	Seção 2.3.4	Seção 2.3.1	Seção 2.3.6	Seção 2.3.5	Seção 2.3.3
ftv170	gap	–	–	0,9	0,0	–	–
	cpu	–	–	8259	9912	–	–
ftv44	gap	0,0	0,3	0,0	0,0	0,0	0,0
	cpu	150	10832	132	6	880	2134
ftv47	gap	0,0	0,2	0,0	0,0	0,0	0,0
	cpu	179	10832	106	6	710	6221
ftv55	gap	0,0	0,9	0,0	0,0	0,0	0,1
	cpu	163	10847	118	10	1035	10800
ftv64	gap	0,0	–	0,0	0,0	0,0	0,0
	cpu	77	–	135	60	294	3591
ftv70	gap	0,0	–	0,0	0,0	0,0	0,1
	cpu	947	–	147	101	1128	10800
ftv80	gap	0,0	–	0,0	0,0	0,0	0,8
	cpu	2209	–	3657	726	5023	10911
ftv90	gap	0,8	–	0,0	0,0	0,0	–
	cpu	10819	–	6991	828	10836	–
krol24	gap	0,1	–	0,0	0,0	0,0	1,0
	cpu	10818	–	9563	537	10002	10840
ry48p	gap	0,0	0,1	0,0	0,0	0,0	0,0
	cpu	121	10816	7562	13	132	936

As Tabelas 20, 21 e 22 apresentam, respectivamente, os resultados da formulação EPN para as instâncias dos grupos *realword*, *stilt* e *crane* nas quais foi possível resolver a relaxação linear, em até três horas. A primeira coluna mostra o nome da instância; a segunda o *gap* (desvio relativo em relação à solução ótima); a terceira o tempo, em segundos, para resolver a relaxação linear; e *opt* a solução ótima. O *gap* é calculado segundo a Eq. (3.1), (Seção 3.2, pág. 60). A solução *opt* foi obtida por meio da formulação PSC em até 3 horas de execução.

Tabela 20 – Resultados da relaxação linear da formulação EPN das instâncias do grupo *realword*

Instância	gap	tempo	opt
$\tau = 20$			
atex1	0,00	0	951
atex3	0,00	1	1258
atex4	0,00	17	2371
atex5	1,17	136	3610
dc112	0,00	1457	4221
dc126	0,05	481	6021
dc134	0,03	4523	3680

Continua

Tabela 20 – Continuação

Instância	gap	tempo	opt
td100_1	0,65	38	33650
br17	0,00	0	39
p43	0,00	49	371
ry48p	8,67	35	4426
kro124	2,84	1968	9619
ft53	0,65	18	2990
ft70	1,42	55	8161
ftv33	0,00	3	1083
ftv35	0,49	5	1161
ftv38	0,80	14	1185
ftv44	0,00	22	1269
ftv47	0,00	53	1410
ftv55	1,09	106	1553
ftv64	0,00	199	1545
ftv70	1,45	380	1669
ftv80	2,03	488	1376
ftv90	0,90	686	1452
ftv100	1,68	1306	1659
ftv110	0,00	2724	1892
$\tau = 50$			
atex1	21,00	0	1564
atex3	3,79	2	1729
atex4	0,00	16	2371
atex5	0,42	134	3631
dc112	0,46	1975	4460
dc126	0,50	3247	10292
dc134	0,03	4678	3680
td100_1	0,37	228	88033
br17	0,00	0	39
p43	0,00	30	371
ry48p	2,95	20	5960
kro124	1,55	1326	16595
ft53	1,71	38	3667
ft70	0,56	103	15699
ftv33	0,00	2	1083
ftv35	0,49	6	1161
ftv38	0,80	9	1185

Continua

Tabela 20 – Continuação

Instância	gap	tempo	opt
ftv44	0,00	12	1269
ftv47	0,00	33	1410
ftv55	1,09	71	1553
ftv64	0,00	173	1545
ftv70	1,45	360	1669
ftv80	2,03	460	1376
ftv90	0,90	690	1452
ftv100	1,68	1380	1659
ftv110	0,00	2533	1892
$\tau = 80$			
atex1	2,58	0	1606
atex3	1,50	2	2415
atex4	4,49	19	2552
atex5	3,11	163	4250
dc112	0,27	1750	7005
dc126	0,17	4797	40379
dc134	0,10	5303	4123
td100_1	0,18	644	159927
br17	0,00	0	39
p43	0,00	48	371
ry48p	1,95	79	9461
kro124	0,99	1876	25036
ft53	0,44	165	4898
ft70	0,21	294	25195
ftv33	1,36	8	1148
ftv35	0,98	11	1189
ftv38	1,83	18	1219
ftv44	1,59	54	1312
ftv47	1,45	93	1449
ftv55	1,09	102	1553
ftv64	0,07	390	1548
ftv70	3,88	327	1716
ftv80	2,03	461	1376
ftv90	0,90	718	1452
ftv100	1,68	1413	1659
ftv110	0,00	2689	1892

Tabela 21 – Resultados da relaxação linear da formulação EPN das instâncias do grupo *stilt*

Instância	<i>gap</i>	tempo	<i>opt</i>
$\tau = 20$			
stilt100_0	3,71	760	6517954
stilt100_1	7,24	894	7008825
stilt100_2	4,32	1020	6571508
stilt100_3	8,66	796	7679518
stilt100_4	2,32	844	7069503
stilt100_5	1,33	836	6293280
stilt100_6	7,84	878	6917872
stilt100_7	5,25	851	6908021
stilt100_8	5,21	881	6583375
stilt100_9	2,28	811	6770781
$\tau = 50$			
stilt100_0	1,03	746	8360358
stilt100_1	0,31	778	8666227
stilt100_2	0,74	797	8853774
stilt100_3	1,64	739	9285634
stilt100_4	2,30	687	9058936
stilt100_5	3,38	782	8823165
stilt100_6	2,06	839	8723479
stilt100_7	1,12	773	8709580
stilt100_8	0,98	792	8110343
stilt100_9	2,28	840	8977213
$\tau = 80$			
stilt100_0	2,43	1950	11164021
stilt100_1	1,65	726	11387079
stilt100_2	0,98	748	11688589
stilt100_3	1,80	662	11740098
stilt100_4	0,58	731	11330701
stilt100_5	2,15	693	11689627
stilt100_6	0,97	800	11566314
stilt100_7	0,81	731	11301530
stilt100_8	2,79	671	10710112
stilt100_9	2,31	678	11520413

Tabela 22 – Resultados da relaxação linear da formulação EPN das instâncias do grupo crane

Instância	gap	tempo	opt
$\tau = 20$			
crane66_0	3,68	72	3719919
crane66_1	2,13	40	3559758
crane66_2	1,42	45	3816506
crane100_0	1,14	696	5210562
crane100_1	0,33	862	5111261
crane100_2	1,59	836	5254378
crane100_3	0,42	824	5278826
crane100_4	0,11	858	5043187
crane100_5	0,22	702	4758933
crane100_6	0,00	587	4743189
crane100_7	0,00	466	4758371
crane100_8	0,38	753	5432053
crane100_9	0,44	790	5153290
$\tau = 50$			
crane66_0	2,50	77	4203049
crane66_1	0,64	81	4080929
crane66_2	0,74	83	4255651
crane100_0	0,66	764	5558390
crane100_1	0,50	663	5318679
crane100_2	0,50	775	5365325
crane100_3	0,26	731	5481511
crane100_4	0,82	813	5439456
crane100_5	0,08	756	5046106
crane100_6	0,39	634	4777667
crane100_7	0,05	481	4766693
crane100_8	0,24	722	5602752
crane100_9	1,91	693	5431151
$\tau = 80$			
crane66_0	1,23	71	4964035
crane66_1	2,00	88	5089489
crane66_2	0,99	72	4901931
crane100_0	0,40	869	6210493
crane100_1	0,91	754	6068276
crane100_2	3,09	781	6198067
crane100_3	0,32	721	5915857
crane100_4	1,17	762	6100475
crane100_5	0,77	664	6167998
crane100_6	0,69	738	5334800
crane100_7	0,25	575	5628337
crane100_8	0,11	677	6057102
crane100_9	0,60	681	5877785

## APÊNDICE B – Resultados dos modelos inteiros – LSSP

Este apêndice apresenta os resultados para a resolução do problema inteiro até a otimalidade. Os experimentos foram executados em um PC Intel® Core™ i7-2600, 3,40 GHz, 16 GB de RAM, sob uma distribuição Linux 64 bits. As formulações foram implementadas com a linguagem C, por meio da interface do *solver* Gurobi versão 8.0.1, 64 bits, o qual também é utilizado para tentar resolver as relaxações lineares.

A Tabela 23 apresenta os resultados detalhados para as instâncias nas quais todas as formulações resolveram até a otimalidade em até 1 hora (3600 seg.). A primeira linha apresenta o número de problemas resolvidos (#SOLVED), seguida do tempo médio de execução (AVG\_tempo). Este tempo médio é calculado sobre as 66 instâncias apresentadas. As demais linhas detalham os tempos médios de execução.

Tabela 23 – Resultados da instâncias resolvidas por todas as formulações com tempo limite de 3600 segundos (LSSP).

	Formulações				
	AL+cuts Seção 4.3.1	CSC Seção 4.3.4	EPN' Seção 4.3.3	SCF Seção 4.3.2	EPN'+cuts Seção 4.3.3
#SOLVED	66	100	70	102	81
AVG_tempo	204	33	225	25	193
Instância					
$ I  = 15$ $ T  = 5$					
Data15-5-0.6-100-0	12	19	82	13	66
Data15-5-0.6-100-1	73	35	115	26	157
Data15-5-0.6-100-2	35	4	39	10	69
Data15-5-0.6-100-3	6	2	84	3	17
Data15-5-0.6-100-4	67	24	65	13	64
Data15-5-0.6-100-5	19	4	38	5	17
Data15-5-0.6-100-6	13	3	88	9	16
Data15-5-0.6-100-7	18	14	74	16	40
Data15-5-0.6-100-8	10	4	12	4	18
Data15-5-0.6-100-9	15	3	11	2	10
Data15-5-0.6-50-0	10	4	4	2	7
Data15-5-0.6-50-1	12	7	13	3	12
Data15-5-0.6-50-2	5	2	5	1	4
Data15-5-0.6-50-3	5	4	4	2	11
Data15-5-0.6-50-4	3	2	7	1	9
Data15-5-0.6-50-5	4	3	5	7	5
Data15-5-0.6-50-6	2	1	1	1	2

*Continua na próxima página*

Tabela 23 – Continuação

	Formulações				
	AL+cuts	CSC	EPN'	SCF	EPN'+cuts
	Seção 4.3.1	Seção 4.3.4	Seção 4.3.3	Seção 4.3.2	Seção 4.3.3
Data15-5-0.6-50-7	3	4	3	1	4
Data15-5-0.6-50-8	4	2	6	1	7
Data15-5-0.6-50-9	2	1	1	0	1
Data15-5-0.8-100-2	192	32	138	17	97
Data15-5-0.8-100-3	949	57	147	31	133
Data15-5-0.8-100-5	2843	106	140	46	254
Data15-5-0.8-100-6	275	22	130	26	84
Data15-5-0.8-100-7	1450	69	231	100	224
Data15-5-0.8-100-8	243	24	96	26	118
Data15-5-0.8-100-9	18	9	24	3	18
Data15-5-0.8-50-0	13	3	59	9	15
Data15-5-0.8-50-1	9	6	10	8	13
Data15-5-0.8-50-2	7	3	9	1	9
Data15-5-0.8-50-3	7	3	17	7	19
Data15-5-0.8-50-4	16	3	14	7	10
Data15-5-0.8-50-5	10	2	14	7	12
Data15-5-0.8-50-6	11	2	40	6	4
Data15-5-0.8-50-7	2	2	3	1	5
Data15-5-0.8-50-8	5	1	8	1	5
Data15-5-0.8-50-9	3	1	4	1	4
$ I  = 15 \quad  T  = 5$					
Data25-5-0.6-100-0	112	103	748	70	421
Data25-5-0.6-100-1	1405	109	1131	123	1132
Data25-5-0.6-100-2	82	118	671	132	823
Data25-5-0.6-100-3	100	50	624	102	450
Data25-5-0.6-100-4	482	101	985	71	767
Data25-5-0.6-100-5	211	165	1504	73	1273
Data25-5-0.6-100-6	742	199	1631	154	1749
Data25-5-0.6-100-7	77	46	611	31	374
Data25-5-0.6-100-8	484	235	639	75	1379
Data25-5-0.6-100-9	32	9	253	23	104
Data25-5-0.6-50-0	29	42	95	40	97
Data25-5-0.6-50-1	20	12	28	3	20
Data25-5-0.6-50-2	90	64	77	27	89
Data25-5-0.6-50-3	26	16	62	12	44
Data25-5-0.6-50-4	32	10	33	7	28
Data25-5-0.6-50-5	24	7	84	14	40
Data25-5-0.6-50-6	38	63	141	22	111
Data25-5-0.6-50-7	37	30	38	14	49
Data25-5-0.6-50-8	28	26	42	4	47
Data25-5-0.6-50-9	17	6	36	3	42
Data25-5-0.8-50-1	957	57	885	38	505
Data25-5-0.8-50-2	23	23	246	7	138
Data25-5-0.8-50-3	1121	90	569	42	307
Data25-5-0.8-50-4	107	19	333	35	112

Continua na próxima página

Tabela 23 – Continuação

	Formulações				
	<b>AL+cuts</b>	<b>CSC</b>	<b>EPN'</b>	<b>SCF</b>	<b>EPN'+cuts</b>
	Seção 4.3.1	Seção 4.3.4	Seção 4.3.3	Seção 4.3.2	Seção 4.3.3
Data25-5-0.8-50-5	565	31	536	49	469
Data25-5-0.8-50-6	12	11	43	3	43
Data25-5-0.8-50-7	79	27	463	22	295
Data25-5-0.8-50-8	165	28	479	34	140
Data25-5-0.8-50-9	26	13	91	5	132

A Tabela 24 apresenta os resultados para as instâncias nas quais pelo menos uma das formulações resolveu até a otimalidade em até 1 hora. O desempenho é comparado por meio do tempo gasto para resolver as formulações. O símbolo “–” indica que a formulação não encontrou a solução ótima em até uma hora, entretanto ela pode ter encontrado uma solução inteira viável.

Tabela 24 – Resultados da instâncias resolvidas por pelo menos uma formulação com tempo limite de 3600 segundos (LSSP).

Instância	Formulações				
	<b>AL+cuts</b>	<b>CSC</b>	<b>EPN'</b>	<b>EPN'+cuts</b>	<b>SCF</b>
	Seção 4.3.1	Seção 4.3.4	Seção 4.3.3	Seção 4.3.3	Seção 4.3.2
$ I  = 15  T  = 5$					
Data15-5-0.8-100-0	–	185	844	635	137
Data15-5-0.8-100-1	–	247	257	588	206
Data15-5-0.8-100-4	–	100	483	237	120
$ I  = 15  T  = 10$					
Data15-10-0.6-100-0	–	–	–	–	1987
Data15-10-0.6-100-3	–	–	–	–	911
Data15-10-0.6-50-0	–	152	–	157	122
Data15-10-0.6-50-1	–	243	–	336	45
Data15-10-0.6-50-2	–	8	–	6	2
Data15-10-0.6-50-3	–	272	–	108	73
Data15-10-0.6-50-4	–	34	–	52	18
Data15-10-0.6-50-5	–	47	–	40	23
Data15-10-0.6-50-6	–	27	–	25	16
Data15-10-0.6-50-7	–	26	–	21	18
Data15-10-0.6-50-8	–	14	–	11	14
Data15-10-0.6-50-9	–	6	–	18	1
Data15-10-0.8-50-0	–	31	–	–	42
Data15-10-0.8-50-1	–	350	–	–	189
Data15-10-0.8-50-2	–	32	–	–	39
Data15-10-0.8-50-3	–	264	–	–	125
Data15-10-0.8-50-4	–	26	–	–	22
Data15-10-0.8-50-5	–	176	–	–	33

Continua na próxima página



Tabela 24 – Continuação

Instância	Formulações				
	AL+cuts Seção 4.3.1	CSC Seção 4.3.4	EPN' Seção 4.3.3	EPN'+cuts Seção 4.3.3	SCF Seção 4.3.2
Data15-10-0.8-50-6	–	33	–	–	23
Data15-10-0.8-50-7	–	21	–	–	20
Data15-10-0.8-50-8	–	21	–	–	20
Data15-10-0.8-50-9	–	4	–	–	2
$ I  = 25$ $ T  = 5$					
Data25-5-0.8-100-0	–	2412	–	–	1365
Data25-5-0.8-100-1	–	1010	–	–	387
Data25-5-0.8-100-2	–	2043	–	–	1398
Data25-5-0.8-100-3	–	382	–	2802	236
Data25-5-0.8-100-4	–	476	–	–	556
Data25-5-0.8-100-5	–	866	–	–	383
Data25-5-0.8-100-6	–	437	–	–	307
Data25-5-0.8-100-7	–	1268	–	–	1531
Data25-5-0.8-100-8	–	933	–	–	3464
Data25-5-0.8-100-9	–	374	–	–	221
Data25-5-0.8-50-0	–	76	1005	583	25

# APÊNDICE C – Resultados adicionais da relaxação linear – LSSP

Os resultados apresentados nesta seção foram obtidos com a mesma máquina descrita no Apêndice B.

A Tabela 25 apresenta os resultados para a relaxação linear, detalhados para cada problema-teste. Novamente, para calcular o *gap* – Eq. 5.1, a solução ótima *opt*, quando não é conhecida, é substituída pela melhor solução. A coluna “UB” mostra a melhor solução e o valor em negrito indica que a solução é ótima.

Tabela 25 – Resultados detalhados da comparação da relaxação linear das 240 instâncias para as formulações – LSSP.

		Formulações						
		AL	AL+cuts	CSC	EPN'	EPN'+cuts	SCF	UB
I  = 15  T  = 10								
Data15-10-0.6-100-0	gap	68,81	1,84	2,00	1,94	1,84	2,06	<b>57384</b>
	tempo	1	1	4	10	10	2	
Data15-10-0.6-100-1	gap	69,56	2,20	2,30	2,23	2,20	2,33	59037
	tempo	1	1	5	14	14	2	
Data15-10-0.6-100-2	gap	69,50	2,42	2,75	2,46	2,42	2,69	56297
	tempo	1	1	5	14	15	2	
Data15-10-0.6-100-3	gap	69,25	1,85	2,17	1,97	1,85	1,97	<b>54957</b>
	tempo	1	1	5	15	16	2	
Data15-10-0.6-100-4	gap	68,78	2,07	2,22	2,10	2,07	2,45	57671
	tempo	0	1	4	9	8	2	
Data15-10-0.6-100-5	gap	69,60	1,88	2,28	2,13	1,88	2,25	56577
	tempo	1	1	4	18	18	2	
Data15-10-0.6-100-6	gap	68,79	2,07	2,16	2,12	2,07	2,16	57231
	tempo	0	1	4	7	7	2	
Data15-10-0.6-100-7	gap	69,95	1,57	1,68	1,57	1,57	1,83	59655
	tempo	1	1	4	13	13	2	
Data15-10-0.6-100-8	gap	69,03	2,02	2,26	2,10	2,02	2,15	57635
	tempo	1	1	4	10	9	2	
Data15-10-0.6-100-9	gap	70,61	1,00	1,14	1,10	1,00	1,14	60658
	tempo	1	1	3	9	10	2	
Data15-10-0.6-50-0	gap	73,61	0,45	0,55	0,43	0,43	0,59	<b>35303</b>
	tempo	0	1	3	6	5	1	
Data15-10-0.6-50-1	gap	74,61	0,31	0,38	0,28	0,28	0,66	<b>36328</b>
	tempo	0	0	4	10	10	1	
Data15-10-0.6-50-2	gap	73,83	0,09	0,25	0,00	0,00	0,17	<b>33931</b>
	tempo	1	1	1	6	9	1	
Data15-10-0.6-50-3	gap	73,54	0,48	0,89	0,45	0,45	0,64	<b>33188</b>
	tempo	0	0	3	8	7	1	
Data15-10-0.6-50-4	gap	73,77	0,17	0,34	0,16	0,16	0,27	<b>35488</b>

*Continua*

Tabela 25 – Continuação

		Formulações						
		AL	AL+cuts	CSC	EPN'	EPN'+cuts	SCF	UB
	tempo	0	1	3	7	7	2	
Data15-10-0.6-50-5	gap	74,59	0,50	0,98	0,48	0,48	0,57	<b>34807</b>
	tempo	0	1	2	5	5	2	
Data15-10-0.6-50-6	gap	73,71	0,18	0,28	0,18	0,18	0,20	<b>34859</b>
	tempo	0	0	3	5	5	1	
Data15-10-0.6-50-7	gap	74,61	0,17	0,35	0,15	0,14	0,60	<b>35817</b>
	tempo	0	0	3	8	8	1	
Data15-10-0.6-50-8	gap	73,77	0,13	0,22	0,13	0,13	0,24	<b>35090</b>
	tempo	0	0	3	5	5	1	
Data15-10-0.6-50-9	gap	75,26	0,03	0,11	0,02	0,02	0,04	<b>36197</b>
	tempo	0	1	2	7	7	1	
Data15-10-0.8-100-0	gap	69,41	2,58	2,55	2,55	2,55	2,66	60146
	tempo	1	1	6	12	12	3	
Data15-10-0.8-100-1	gap	69,84	2,10	2,20	2,13	2,10	2,21	60941
	tempo	1	1	6	20	20	3	
Data15-10-0.8-100-2	gap	70,21	3,52	3,66	3,57	3,52	3,68	58872
	tempo	1	1	5	11	11	2	
Data15-10-0.8-100-3	gap	70,00	2,95	3,03	3,03	2,95	3,13	57653
	tempo	1	1	7	14	14	3	
Data15-10-0.8-100-4	gap	69,37	2,40	2,52	2,46	2,40	2,79	60019
	tempo	0	1	6	11	12	2	
Data15-10-0.8-100-5	gap	70,38	2,94	3,79	3,05	2,94	3,05	59268
	tempo	1	1	6	21	21	3	
Data15-10-0.8-100-6	gap	69,74	3,22	3,31	3,23	3,22	3,30	59989
	tempo	0	1	6	12	12	2	
Data15-10-0.8-100-7	gap	70,68	2,98	3,03	3,00	2,98	3,18	62116
	tempo	0	1	6	12	12	2	
Data15-10-0.8-100-8	gap	69,70	2,34	2,39	2,38	2,34	2,53	59946
	tempo	1	1	4	10	10	2	
Data15-10-0.8-100-9	gap	70,83	1,16	1,27	1,23	1,16	1,28	62087
	tempo	0	1	4	10	10	2	
Data15-10-0.8-50-0	gap	73,32	0,40	0,45	0,37	0,37	0,45	<b>35455</b>
	tempo	0	0	3	6	7	2	
Data15-10-0.8-50-1	gap	74,30	0,44	0,52	0,41	0,41	0,66	<b>36407</b>
	tempo	0	1	4	13	14	2	
Data15-10-0.8-50-2	gap	73,74	0,56	0,63	0,54	0,54	0,59	<b>34333</b>
	tempo	0	0	2	14	14	1	
Data15-10-0.8-50-3	gap	73,44	0,74	0,94	0,69	0,69	0,77	<b>33598</b>
	tempo	0	1	4	10	10	2	
Data15-10-0.8-50-4	gap	73,51	0,29	0,38	0,27	0,27	0,63	<b>35664</b>
	tempo	0	0	3	10	9	2	
Data15-10-0.8-50-5	gap	74,38	0,67	0,90	0,65	0,65	0,75	<b>35070</b>
	tempo	0	0	3	8	8	2	
Data15-10-0.8-50-6	gap	73,51	0,38	0,41	0,37	0,37	0,40	<b>35083</b>
	tempo	0	0	3	7	7	2	
Data15-10-0.8-50-7	gap	74,25	0,21	0,26	0,18	0,18	0,58	<b>35859</b>
	tempo	0	0	3	10	10	1	
Data15-10-0.8-50-8	gap	73,54	0,27	0,33	0,27	0,27	0,40	<b>35328</b>
	tempo	0	0	3	6	5	1	
Data15-10-0.8-50-9	gap	74,89	0,07	0,12	0,06	0,06	0,09	<b>36220</b>
	tempo	1	1	2	7	7	1	
I  = 15  T  = 15								
Data15-15-0.6-100-0	gap	78,77	3,42	3,73	3,58	3,42	3,58	86867
	tempo	1	1	8	37	38	4	
Data15-15-0.6-100-1	gap	79,60	3,41	3,56	3,44	3,41	3,57	89350
	tempo	1	1	14	39	39	4	

Continua

Tabela 25 – Continuação

		Formulações						
		AL	AL+cuts	CSC	EPN'	EPN'+cuts	SCF	UB
Data15-15-0.6-100-2	gap	79,12	3,80	3,93	3,83	3,80	3,91	84980
	tempo	1	1	8	32	35	3	
Data15-15-0.6-100-3	gap	78,36	4,07	4,39	4,24	4,07	4,31	82794
	tempo	1	2	10	37	37	4	
Data15-15-0.6-100-4	gap	79,34	4,31	4,46	4,40	4,31	4,48	88337
	tempo	1	1	12	25	26	4	
Data15-15-0.6-100-5	gap	78,98	4,06	4,66	4,38	4,06	4,61	85879
	tempo	1	1	9	31	35	5	
Data15-15-0.6-100-6	gap	79,31	4,02	4,12	4,07	4,02	4,07	87529
	tempo	1	1	9	22	22	4	
Data15-15-0.6-100-7	gap	79,66	3,96	4,11	4,03	3,96	4,12	91166
	tempo	1	1	12	34	37	4	
Data15-15-0.6-100-8	gap	78,85	3,70	4,01	3,87	3,70	3,93	87057
	tempo	1	1	9	25	25	4	
Data15-15-0.6-100-9	gap	79,88	1,24	1,32	1,30	1,24	1,30	91039
	tempo	1	1	9	22	23	3	
Data15-15-0.6-50-0	gap	81,93	0,37	0,50	0,35	0,35	0,44	52782
	tempo	1	1	5	15	16	3	
Data15-15-0.6-50-1	gap	82,84	0,36	0,45	0,35	0,35	0,43	54515
	tempo	3	4	7	16	17	3	
Data15-15-0.6-50-2	gap	82,03	0,12	0,24	0,12	0,12	0,19	51109
	tempo	1	1	5	8	9	2	
Data15-15-0.6-50-3	gap	81,36	0,79	1,26	0,76	0,76	0,83	49829
	tempo	1	2	7	24	23	5	
Data15-15-0.6-50-4	gap	82,41	0,17	0,24	0,16	0,16	0,21	53150
	tempo	2	2	8	12	13	4	
Data15-15-0.6-50-5	gap	82,37	0,62	1,03	0,60	0,60	0,68	52329
	tempo	1	1	6	11	12	3	
Data15-15-0.6-50-6	gap	82,12	0,18	0,26	0,18	0,18	0,20	52305
	tempo	1	1	5	11	13	3	
Data15-15-0.6-50-7	gap	82,42	0,31	0,45	0,30	0,30	0,33	53752
	tempo	1	1	8	17	16	4	
Data15-15-0.6-50-8	gap	81,95	0,22	0,32	0,19	0,19	0,28	52531
	tempo	1	1	4	10	12	2	
Data15-15-0.6-50-9	gap	82,86	0,02	0,10	0,00	0,00	0,02	54165
	tempo	1	1	4	10	12	2	
Data15-15-0.8-100-0	gap	79,13	4,24	4,31	4,24	4,24	4,39	89958
	tempo	1	1	10	40	36	4	
Data15-15-0.8-100-1	gap	80,72	7,99	8,06	8,03	7,99	8,06	95955
	tempo	1	1	14	50	52	5	
Data15-15-0.8-100-2	gap	79,25	4,43	4,56	4,47	4,43	4,55	87623
	tempo	1	1	12	40	41	4	
Data15-15-0.8-100-3	gap	79,02	5,56	5,53	5,53	5,53	5,75	86485
	tempo	1	1	13	39	35	5	
Data15-15-0.8-100-4	gap	79,16	2,72	2,83	2,80	2,72	2,84	89055
	tempo	1	1	13	33	34	5	
Data15-15-0.8-100-5	gap	79,43	5,30	5,56	5,56	5,30	5,66	89313
	tempo	1	1	14	52	66	5	
Data15-15-0.8-100-6	gap	79,29	3,41	3,44	3,44	3,41	3,52	88898
	tempo	1	1	10	29	28	4	
Data15-15-0.8-100-7	gap	79,73	3,78	3,89	3,83	3,78	3,92	92662
	tempo	1	1	17	35	36	4	
Data15-15-0.8-100-8	gap	78,97	3,31	3,47	3,44	3,31	3,48	88886
	tempo	1	1	10	23	24	4	
Data15-15-0.8-100-9	gap	80,04	2,28	2,41	2,34	2,28	2,36	93109
	tempo	1	1	8	36	31	4	

Continua

Tabela 25 – Continuação

		Formulações						
		AL	AL+cuts	CSC	EPN'	EPN'+cuts	SCF	UB
Data15-15-0.8-50-0	gap	81,77	0,39	0,47	0,37	0,37	0,46	52956
	tempo	1	1	7	14	15	3	
Data15-15-0.8-50-1	gap	82,64	0,38	0,45	0,37	0,37	0,61	54538
	tempo	1	1	8	20	22	3	
Data15-15-0.8-50-2	gap	81,97	0,56	0,60	0,55	0,55	0,55	51538
	tempo	1	1	7	23	24	4	
Data15-15-0.8-50-3	gap	81,36	1,44	1,62	1,41	1,41	1,46	50385
	tempo	1	1	9	24	24	5	
Data15-15-0.8-50-4	gap	82,25	0,29	0,38	0,29	0,29	0,35	53297
	tempo	1	1	8	18	18	4	
Data15-15-0.8-50-5	gap	82,24	0,74	0,99	0,74	0,74	0,84	52575
	tempo	1	1	7	16	16	4	
Data15-15-0.8-50-6	gap	82,00	0,45	0,47	0,44	0,43	0,46	52600
	tempo	1	1	6	15	15	3	
Data15-15-0.8-50-7	gap	82,23	0,29	0,34	0,29	0,29	0,30	53752
	tempo	1	1	8	12	13	3	
Data15-15-0.8-50-8	gap	81,82	0,34	0,37	0,32	0,32	0,38	52753
	tempo	1	1	6	9	9	3	
Data15-15-0.8-50-9	gap	82,68	0,05	0,10	0,05	0,05	0,06	54195
	tempo	1	1	4	14	14	2	
$ T  = 15 \quad  T'  = 5$								
Data15-5-0.6-100-0	gap	47,52	1,54	1,37	1,37	1,37	1,53	<b>29348</b>
	tempo	0	0	1	2	2	1	
Data15-5-0.6-100-1	gap	48,38	1,73	1,75	1,65	1,65	1,93	<b>30155</b>
	tempo	0	0	1	2	2	1	
Data15-5-0.6-100-2	gap	46,72	1,40	1,25	1,25	1,25	1,39	<b>28166</b>
	tempo	0	0	1	2	2	1	
Data15-5-0.6-100-3	gap	46,99	1,49	1,43	1,43	1,43	1,64	<b>27827</b>
	tempo	0	0	1	2	2	1	
Data15-5-0.6-100-4	gap	48,04	1,51	1,60	1,56	1,51	1,77	<b>29390</b>
	tempo	0	1	1	1	1	1	
Data15-5-0.6-100-5	gap	47,74	1,46	1,79	1,39	1,39	1,93	<b>28731</b>
	tempo	0	0	1	3	3	0	
Data15-5-0.6-100-6	gap	47,80	1,10	1,19	1,10	1,10	1,20	<b>29053</b>
	tempo	0	0	1	2	2	0	
Data15-5-0.6-100-7	gap	48,93	1,42	1,47	1,42	1,42	1,42	<b>30350</b>
	tempo	0	0	1	2	2	0	
Data15-5-0.6-100-8	gap	47,22	1,26	1,38	1,28	1,26	1,38	<b>29245</b>
	tempo	0	0	1	2	2	0	
Data15-5-0.6-100-9	gap	51,04	0,63	0,74	0,67	0,63	0,76	<b>31057</b>
	tempo	0	0	1	2	2	1	
Data15-5-0.6-50-0	gap	55,19	0,26	0,37	0,22	0,22	0,38	<b>17725</b>
	tempo	0	0	1	2	2	0	
Data15-5-0.6-50-1	gap	56,71	0,27	0,38	0,25	0,24	0,44	<b>18359</b>
	tempo	0	0	1	3	3	1	
Data15-5-0.6-50-2	gap	55,07	0,14	0,32	0,12	0,12	0,26	<b>17131</b>
	tempo	0	0	1	2	3	0	
Data15-5-0.6-50-3	gap	54,40	0,41	0,80	0,34	0,34	0,63	<b>16827</b>
	tempo	0	0	1	3	7	0	
Data15-5-0.6-50-4	gap	55,75	0,18	0,28	0,16	0,15	0,44	<b>17834</b>
	tempo	0	0	1	5	7	0	
Data15-5-0.6-50-5	gap	55,84	0,42	0,70	0,05	0,05	0,67	<b>17527</b>
	tempo	0	0	1	5	5	0	
Data15-5-0.6-50-6	gap	55,24	0,08	0,19	0,03	0,03	0,10	<b>17482</b>
	tempo	0	0	1	1	1	1	
Data15-5-0.6-50-7	gap	55,75	0,17	0,29	0,09	0,09	0,26	<b>17967</b>

Continua

Tabela 25 – Continuação

		Formulações						
		AL	AL+cuts	CSC	EPN'	EPN'+cuts	SCF	UB
Data15-5-0.6-50-8	tempo	0	0	1	3	3	0	
	gap	54,54	0,25	0,39	0,14	0,14	0,38	<b>17560</b>
Data15-5-0.6-50-9	tempo	0	0	1	6	6	0	
	gap	57,98	0,00	0,02	0,00	0,00	0,00	<b>18202</b>
Data15-5-0.8-100-0	tempo	0	0	0	1	4	0	
	gap	51,48	1,67	1,69	1,61	1,61	1,76	<b>31852</b>
Data15-5-0.8-100-1	tempo	0	0	1	3	3	1	
	gap	51,53	1,84	1,87	1,82	1,81	1,99	<b>32302</b>
Data15-5-0.8-100-2	tempo	0	0	1	3	4	1	
	gap	50,58	1,75	1,82	1,75	1,75	1,85	<b>30506</b>
Data15-5-0.8-100-3	tempo	0	0	1	3	3	1	
	gap	51,77	1,99	1,94	1,89	1,88	2,32	<b>30643</b>
Data15-5-0.8-100-4	tempo	0	1	1	3	3	1	
	gap	51,60	2,15	2,08	1,88	1,88	2,58	<b>31804</b>
Data15-5-0.8-100-5	tempo	0	0	1	6	5	1	
	gap	51,26	2,13	2,22	2,09	2,09	2,50	<b>31133</b>
Data15-5-0.8-100-6	tempo	0	0	1	3	3	1	
	gap	51,09	1,40	1,42	1,36	1,36	1,46	<b>31094</b>
Data15-5-0.8-100-7	tempo	0	0	2	3	3	1	
	gap	51,50	1,69	1,86	1,64	1,64	1,75	<b>32106</b>
Data15-5-0.8-100-8	tempo	0	0	1	2	2	1	
	gap	51,03	1,64	1,66	1,62	1,62	1,71	<b>31672</b>
Data15-5-0.8-100-9	tempo	0	0	1	2	2	1	
	gap	52,66	0,78	0,97	0,77	0,77	0,88	<b>32211</b>
Data15-5-0.8-50-0	tempo	0	0	1	2	2	1	
	gap	55,56	0,53	0,59	0,49	0,49	0,62	<b>17891</b>
Data15-5-0.8-50-1	tempo	0	0	1	2	2	1	
	gap	56,66	0,29	0,36	0,28	0,28	0,42	<b>18374</b>
Data15-5-0.8-50-2	tempo	0	0	1	3	3	1	
	gap	55,42	0,29	0,37	0,28	0,26	0,31	<b>17282</b>
Data15-5-0.8-50-3	tempo	0	0	1	6	7	1	
	gap	54,82	0,77	0,84	0,70	0,70	0,98	<b>17021</b>
Data15-5-0.8-50-4	tempo	0	0	1	4	4	1	
	gap	55,94	0,48	0,50	0,44	0,44	0,80	<b>17944</b>
Data15-5-0.8-50-5	tempo	0	0	1	3	3	1	
	gap	56,23	0,61	0,76	0,54	0,54	0,88	<b>17691</b>
Data15-5-0.8-50-6	tempo	0	0	1	3	4	1	
	gap	55,72	0,45	0,44	0,41	0,41	0,45	<b>17676</b>
Data15-5-0.8-50-7	tempo	0	0	1	2	2	1	
	gap	55,64	0,15	0,20	0,12	0,12	0,25	<b>17967</b>
Data15-5-0.8-50-8	tempo	0	0	1	2	3	1	
	gap	54,92	0,33	0,37	0,32	0,32	0,44	<b>17736</b>
Data15-5-0.8-50-9	tempo	0	0	1	3	3	1	
	gap	57,94	0,07	0,11	0,00	0,00	0,08	<b>18227</b>
Data25-10-0.6-100-0	tempo	0	0	1	4	3	0	
	gap	65,31	1,46	1,48	1,45	1,45	1,46	91963
Data25-10-0.6-100-1	tempo	2	2	16	164	156	8	
	gap	65,91	1,80	1,78	1,77	1,77	1,77	94005
Data25-10-0.6-100-2	tempo	2	2	21	104	93	10	
	gap	65,60	1,00	1,03	1,00	1,00	1,03	93309
Data25-10-0.6-100-3	tempo	2	2	14	63	69	9	
	gap	66,10	1,96	2,03	1,98	1,96	2,02	93824
Data25-10-0.6-100-4	tempo	2	2	17	120	123	7	
	gap	67,31	2,11	2,17	2,12	2,11	2,15	99019
	tempo	2	2	17	154	146	10	
	gap							

Continua

Tabela 25 – Continuação

		Formulações						
		AL	AL+cuts	CSC	EPN'	EPN'+cuts	SCF	UB
Data25-10-0.6-100-5	gap	65,40	1,50	1,55	1,49	1,49	1,51	92106
	tempo	2	2	16	119	122	10	
Data25-10-0.6-100-6	gap	67,74	1,76	1,80	1,77	1,76	1,78	100887
	tempo	2	2	15	117	117	8	
Data25-10-0.6-100-7	gap	66,53	2,53	2,61	2,54	2,53	2,57	96280
	tempo	1	2	14	134	123	8	
Data25-10-0.6-100-8	gap	66,90	2,92	3,01	2,96	2,92	2,99	98149
	tempo	2	2	19	130	126	9	
Data25-10-0.6-100-9	gap	67,81	1,69	1,74	1,69	1,69	1,72	100497
	tempo	2	2	13	105	100	7	
Data25-10-0.6-50-0	gap	70,17	0,37	0,44	0,37	0,37	0,38	55712
	tempo	1	1	11	190	131	8	
Data25-10-0.6-50-1	gap	70,57	0,09	0,13	0,09	0,09	0,12	56183
	tempo	1	2	9	72	66	6	
Data25-10-0.6-50-2	gap	70,85	0,04	0,09	0,04	0,04	0,06	56782
	tempo	1	2	12	44	36	6	
Data25-10-0.6-50-3	gap	70,70	0,05	0,07	0,05	0,05	0,06	55984
	tempo	1	1	12	57	51	6	
Data25-10-0.6-50-4	gap	71,76	0,11	0,16	0,11	0,11	0,12	58373
	tempo	1	2	13	55	65	7	
Data25-10-0.6-50-5	gap	70,67	0,16	0,18	0,13	0,13	0,16	56221
	tempo	2	2	11	97	118	6	
Data25-10-0.6-50-6	gap	71,95	0,08	0,15	0,07	0,07	0,08	59226
	tempo	2	2	11	80	61	6	
Data25-10-0.6-50-7	gap	71,14	0,07	0,13	0,06	0,06	0,06	57442
	tempo	2	2	10	57	65	6	
Data25-10-0.6-50-8	gap	71,65	0,19	0,27	0,19	0,19	0,22	58549
	tempo	2	2	12	76	83	8	
Data25-10-0.6-50-9	gap	71,67	0,10	0,14	0,09	0,09	0,10	58501
	tempo	3	4	12	53	59	7	
Data25-10-0.8-100-0	gap	66,60	2,86	2,94	2,87	2,86	2,90	96647
	tempo	2	2	25	126	125	12	
Data25-10-0.8-100-1	gap	67,55	4,10	4,12	4,12	4,10	4,14	99763
	tempo	1	2	29	105	110	9	
Data25-10-0.8-100-2	gap	67,21	3,01	3,06	3,03	3,01	3,04	98579
	tempo	2	2	24	112	114	10	
Data25-10-0.8-100-3	gap	67,01	1,93	1,99	1,92	1,92	1,98	96969
	tempo	1	2	28	145	151	9	
Data25-10-0.8-100-4	gap	68,04	1,94	1,95	1,95	1,94	1,97	101983
	tempo	1	2	39	174	200	10	
Data25-10-0.8-100-5	gap	67,04	4,12	4,13	4,12	4,12	4,16	98014
	tempo	2	2	29	136	142	10	
Data25-10-0.8-100-6	gap	68,42	1,94	1,96	1,94	1,94	1,95	103619
	tempo	1	2	28	127	139	10	
Data25-10-0.8-100-7	gap	67,15	1,95	2,02	1,99	1,95	2,02	98740
	tempo	1	2	29	165	160	10	
Data25-10-0.8-100-8	gap	67,44	2,92	2,97	2,92	2,92	2,92	100903
	tempo	1	2	19	112	129	10	
Data25-10-0.8-100-9	gap	68,31	2,06	2,11	2,05	2,05	2,09	103012
	tempo	1	2	17	134	134	10	
Data25-10-0.8-50-0	gap	70,54	1,25	1,25	1,24	1,24	1,26	56664
	tempo	1	1	16	134	151	9	
Data25-10-0.8-50-1	gap	70,91	0,67	0,70	0,67	0,67	0,68	57076
	tempo	1	1	11	67	90	7	
Data25-10-0.8-50-2	gap	70,93	0,15	0,19	0,14	0,13	0,17	57166
	tempo	1	2	12	122	148	8	

Continua

Tabela 25 – Continuação

		Formulações						
		AL	AL+cuts	CSC	EPN'	EPN'+cuts	SCF	UB
Data25-10-0.8-50-3	gap	70,95	0,63	0,65	0,60	0,60	0,64	56721
	tempo	1	1	14	125	149	6	
Data25-10-0.8-50-4	gap	71,91	0,61	0,63	0,60	0,60	0,62	58964
	tempo	1	1	17	107	107	7	
Data25-10-0.8-50-5	gap	70,86	0,30	0,34	0,30	0,30	0,31	56826
	tempo	1	2	17	99	103	8	
Data25-10-0.8-50-6	gap	71,95	0,19	0,22	0,17	0,17	0,19	59479
	tempo	1	2	14	97	86	7	
Data25-10-0.8-50-7	gap	71,21	0,30	0,33	0,30	0,30	0,31	57791
	tempo	1	1	11	103	95	8	
Data25-10-0.8-50-8	gap	71,67	0,38	0,41	0,37	0,37	0,39	58840
	tempo	1	2	14	65	72	8	
Data25-10-0.8-50-9	gap	71,69	0,20	0,23	0,20	0,20	0,22	58769
	tempo	1	1	16	101	119	8	
$ I  = 25 \quad  T  = 15$								
Data25-15-0.6-100-0	gap	75,42	3,56	3,60	3,55	3,55	3,58	139894
	tempo	5	6	41	369	401	27	
Data25-15-0.6-100-1	gap	75,89	3,74	3,79	3,75	3,74	3,77	142555
	tempo	4	6	38	269	264	22	
Data25-15-0.6-100-2	gap	75,78	3,68	3,73	3,69	3,68	3,72	142816
	tempo	4	5	34	335	304	24	
Data25-15-0.6-100-3	gap	76,06	4,43	4,46	4,43	4,43	4,45	143041
	tempo	3	4	50	440	456	24	
Data25-15-0.6-100-4	gap	76,56	2,59	2,66	2,61	2,59	2,64	148443
	tempo	4	5	53	397	459	25	
Data25-15-0.6-100-5	gap	75,52	4,02	4,06	4,01	4,01	4,01	140274
	tempo	4	5	45	403	424	27	
Data25-15-0.6-100-6	gap	76,94	2,81	2,85	2,81	2,81	2,84	152183
	tempo	3	4	43	417	438	25	
Data25-15-0.6-100-7	gap	76,41	3,91	3,98	3,93	3,91	3,95	145813
	tempo	4	5	42	377	461	25	
Data25-15-0.6-100-8	gap	76,42	3,56	3,62	3,60	3,56	3,64	147413
	tempo	4	5	56	463	508	24	
Data25-15-0.6-100-9	gap	76,47	1,24	1,30	1,26	1,24	1,27	148658
	tempo	4	5	33	261	281	24	
Data25-15-0.6-50-0	gap	78,81	0,97	1,04	0,97	0,97	0,98	83873
	tempo	2	3	28	272	344	14	
Data25-15-0.6-50-1	gap	79,07	0,42	0,44	0,42	0,42	0,43	84834
	tempo	2	2	23	136	119	12	
Data25-15-0.6-50-2	gap	79,02	0,11	0,17	0,10	0,10	0,13	85258
	tempo	2	3	31	150	165	14	
Data25-15-0.6-50-3	gap	79,04	0,17	0,21	0,17	0,17	0,18	84025
	tempo	3	4	25	166	187	12	
Data25-15-0.6-50-4	gap	79,74	0,24	0,29	0,24	0,24	0,26	87741
	tempo	3	4	30	189	205	10	
Data25-15-0.6-50-5	gap	79,09	0,65	0,70	0,65	0,65	0,67	84599
	tempo	2	3	27	171	216	14	
Data25-15-0.6-50-6	gap	80,01	0,42	0,45	0,41	0,41	0,43	89105
	tempo	3	3	27	190	150	15	
Data25-15-0.6-50-7	gap	79,59	0,59	0,62	0,58	0,58	0,60	86725
	tempo	3	3	21	246	169	14	
Data25-15-0.6-50-8	gap	79,94	0,40	0,43	0,40	0,40	0,43	88050
	tempo	3	4	25	140	134	12	
Data25-15-0.6-50-9	gap	79,65	0,05	0,09	0,05	0,05	0,06	87744
	tempo	2	3	22	84	86	15	
Data25-15-0.8-100-0	gap	76,06	4,74	4,80	4,73	4,73	4,74	145197

Continua



Tabela 25 – Continuação

		Formulações						
		AL	AL+cuts	CSC	EPN'	EPN'+cuts	SCF	UB
Data25-15-0.8-100-1	tempo	4	5	71	386	389	29	
	gap	76,24	4,40	4,42	4,42	4,40	4,42	147162
Data25-15-0.8-100-2	tempo	4	5	79	265	287	24	
	gap	75,81	2,83	2,86	2,84	2,83	2,84	144789
Data25-15-0.8-100-3	tempo	4	5	73	331	319	30	
	gap	76,12	3,25	3,26	3,22	3,22	3,23	144568
Data25-15-0.8-100-4	tempo	4	5	104	476	473	28	
	gap	77,26	4,53	4,59	4,54	4,53	4,56	154443
Data25-15-0.8-100-5	tempo	3	4	107	623	599	26	
	gap	75,78	3,62	3,64	3,63	3,62	3,64	143246
Data25-15-0.8-100-6	tempo	4	5	98	372	385	32	
	gap	77,22	3,27	3,30	3,27	3,27	3,28	155283
Data25-15-0.8-100-7	tempo	4	5	78	376	414	28	
	gap	76,25	2,58	2,61	2,60	2,58	2,63	147000
Data25-15-0.8-100-8	tempo	4	5	88	372	382	29	
	gap	76,64	3,63	3,70	3,65	3,63	3,69	150327
Data25-15-0.8-100-9	tempo	4	5	100	535	564	27	
	gap	77,08	3,23	3,24	3,23	3,23	3,23	154110
Data25-15-0.8-50-0	tempo	3	4	90	305	302	27	
	gap	78,72	0,81	0,85	0,80	0,80	0,82	84133
Data25-15-0.8-50-1	tempo	2	3	39	301	320	15	
	gap	79,07	0,62	0,64	0,62	0,62	0,63	85499
Data25-15-0.8-50-2	tempo	2	2	24	177	172	16	
	gap	79,17	1,27	1,31	1,25	1,25	1,28	86506
Data25-15-0.8-50-3	tempo	2	3	31	262	354	16	
	gap	79,13	0,94	0,97	0,93	0,93	0,95	85003
Data25-15-0.8-50-4	tempo	2	2	32	218	251	14	
	gap	79,72	0,63	0,69	0,63	0,63	0,65	88343
Data25-15-0.8-50-5	tempo	2	3	35	217	207	11	
	gap	79,49	2,80	2,83	2,79	2,79	2,81	86920
Data25-15-0.8-50-6	tempo	2	2	42	322	318	22	
	gap	79,87	0,38	0,41	0,37	0,37	0,38	89170
Data25-15-0.8-50-7	tempo	3	4	23	165	175	15	
	gap	79,52	0,85	0,87	0,85	0,85	0,86	87063
Data25-15-0.8-50-8	tempo	2	3	29	216	225	16	
	gap	80,10	1,81	1,83	1,81	1,81	1,81	89427
Data25-15-0.8-50-9	tempo	2	2	30	130	180	14	
	gap	79,56	0,15	0,18	0,15	0,15	0,16	87988
Data25-5-0.6-100-0	tempo	3	4	27	157	205	15	
	gap	44,65	0,54	0,60	0,55	0,54	0,71	<b>46874</b>
Data25-5-0.6-100-1	tempo	1	1	3	22	20	2	
	gap	44,60	0,79	0,87	0,81	0,79	0,86	<b>47552</b>
Data25-5-0.6-100-2	tempo	1	1	3	17	19	1	
	gap	44,53	0,63	0,67	0,61	0,61	0,66	<b>47536</b>
Data25-5-0.6-100-3	tempo	1	1	2	20	21	1	
	gap	44,95	0,52	0,59	0,52	0,52	0,59	<b>47357</b>
Data25-5-0.6-100-4	tempo	1	1	3	20	20	2	
	gap	47,50	0,50	0,66	0,50	0,50	0,68	<b>50081</b>
Data25-5-0.6-100-5	tempo	1	1	4	26	27	2	
	gap	44,05	0,91	0,83	0,83	0,83	0,85	<b>46495</b>
Data25-5-0.6-100-6	tempo	1	1	4	29	41	2	
	gap	47,41	0,61	0,68	0,62	0,61	0,67	<b>50918</b>
Data25-5-0.6-100-7	tempo	1	1	3	29	30	2	
	gap	45,24	0,64	0,63	0,59	0,59	0,67	<b>48069</b>
	tempo	1	1	3	17	22	1	
	gap							

Continua

Tabela 25 – Continuação

		Formulações						
		AL	AL+cuts	CSC	EPN'	EPN'+cuts	SCF	UB
Data25-5-0.6-100-8	gap	46,16	0,62	0,70	0,63	0,62	0,67	<b>49158</b>
	tempo	1	1	4	19	19	2	
Data25-5-0.6-100-9	gap	47,38	0,32	0,37	0,34	0,32	0,34	<b>50416</b>
	tempo	1	1	2	12	13	1	
Data25-5-0.6-50-0	gap	51,37	0,17	0,23	0,17	0,17	0,26	<b>28020</b>
	tempo	1	1	3	19	21	2	
Data25-5-0.6-50-1	gap	51,94	0,05	0,07	0,05	0,05	0,06	<b>28456</b>
	tempo	1	1	4	12	10	1	
Data25-5-0.6-50-2	gap	52,30	0,15	0,19	0,13	0,13	0,17	<b>28732</b>
	tempo	1	1	3	49	67	1	
Data25-5-0.6-50-3	gap	52,26	0,08	0,10	0,07	0,07	0,13	<b>28259</b>
	tempo	1	1	4	18	22	1	
Data25-5-0.6-50-4	gap	54,25	0,02	0,09	0,00	0,00	0,14	<b>29314</b>
	tempo	1	1	3	33	28	1	
Data25-5-0.6-50-5	gap	51,99	0,08	0,17	0,06	0,06	0,09	<b>28261</b>
	tempo	1	1	4	70	27	2	
Data25-5-0.6-50-6	gap	53,93	0,14	0,21	0,12	0,12	0,15	<b>29750</b>
	tempo	0	0	4	99	71	2	
Data25-5-0.6-50-7	gap	52,65	0,10	0,16	0,08	0,08	0,12	<b>29047</b>
	tempo	1	1	3	32	30	1	
Data25-5-0.6-50-8	gap	54,15	0,06	0,09	0,06	0,06	0,07	<b>29508</b>
	tempo	0	1	4	28	28	2	
Data25-5-0.6-50-9	gap	53,85	0,05	0,10	0,05	0,05	0,07	<b>29472</b>
	tempo	1	1	3	27	27	2	
Data25-5-0.8-100-0	gap	48,78	1,01	1,05	0,99	0,99	1,14	<b>50869</b>
	tempo	1	1	5	24	23	2	
Data25-5-0.8-100-1	gap	48,50	0,66	0,71	0,66	0,66	0,71	<b>51445</b>
	tempo	0	1	5	19	19	2	
Data25-5-0.8-100-2	gap	47,91	0,78	0,85	0,77	0,77	0,81	<b>51098</b>
	tempo	1	1	5	27	26	2	
Data25-5-0.8-100-3	gap	48,78	0,85	0,89	0,84	0,84	0,87	<b>50994</b>
	tempo	1	1	5	31	31	2	
Data25-5-0.8-100-4	gap	50,57	0,71	0,79	0,71	0,71	0,74	<b>53207</b>
	tempo	0	1	7	25	28	2	
Data25-5-0.8-100-5	gap	48,11	0,78	0,72	0,69	0,69	0,69	<b>50441</b>
	tempo	0	1	6	31	32	2	
Data25-5-0.8-100-6	gap	49,62	0,58	0,59	0,57	0,56	0,59	<b>53183</b>
	tempo	0	1	5	19	19	2	
Data25-5-0.8-100-7	gap	48,42	0,79	0,82	0,79	0,79	0,81	<b>51666</b>
	tempo	1	1	6	26	30	2	
Data25-5-0.8-100-8	gap	48,97	0,98	1,07	0,97	0,97	1,02	<b>52192</b>
	tempo	0	1	6	26	24	2	
Data25-5-0.8-100-9	gap	49,57	0,61	0,65	0,61	0,61	0,63	<b>52813</b>
	tempo	0	1	5	25	25	2	
Data25-5-0.8-50-0	gap	52,02	0,53	0,58	0,50	0,50	0,54	<b>28427</b>
	tempo	1	1	5	53	53	2	
Data25-5-0.8-50-1	gap	52,63	0,37	0,37	0,36	0,36	0,38	<b>28869</b>
	tempo	1	1	3	20	20	2	
Data25-5-0.8-50-2	gap	52,58	0,22	0,24	0,20	0,20	0,24	<b>28902</b>
	tempo	1	1	3	18	16	2	
Data25-5-0.8-50-3	gap	52,83	0,38	0,39	0,37	0,37	0,41	<b>28603</b>
	tempo	0	1	4	28	33	2	
Data25-5-0.8-50-4	gap	54,69	0,20	0,22	0,19	0,19	0,21	<b>29597</b>
	tempo	0	1	4	17	20	2	
Data25-5-0.8-50-5	gap	52,57	0,36	0,42	0,36	0,36	0,37	<b>28609</b>
	tempo	0	1	4	22	22	2	

Continua

Tabela 25 – Continuação

		Formulações						
		AL	AL+cuts	CSC	EPN'	EPN'+cuts	SCF	UB
Data25-5-0.8-50-6	gap	53,99	0,06	0,07	0,02	0,02	0,08	<b>29791</b>
	tempo	0	1	3	43	32	2	
Data25-5-0.8-50-7	gap	52,83	0,22	0,27	0,21	0,21	0,25	<b>29157</b>
	tempo	0	1	4	45	44	2	
Data25-5-0.8-50-8	gap	54,37	0,26	0,29	0,25	0,25	0,27	<b>29653</b>
	tempo	0	0	5	23	22	2	
Data25-5-0.8-50-9	gap	54,04	0,09	0,15	0,09	0,09	0,10	<b>29590</b>
	tempo	0	1	4	43	40	2	

## APÊNDICE D – Processo de calibração – PCTSP

O procedimento de calibração mostrou-se uma parte crítica para uso do VOL. Esta é uma tarefa árdua com alto custo computacional e fundamental para encontrar bons resultados. Para selecionar os melhores valores para os parâmetros foram feitos extensos testes computacionais. No final dos experimentos foram escolhidos o conjunto de valores que apresentaram os melhores resultados médios. O procedimento é descrito a seguir:

Inicialmente, um subconjunto de instâncias do grupo *realworld* foi selecionado: *br17, ftv33, ftv35, ftv38, ftv47, ftv70, ry48p, p43, ft53, ftv80*. Para estas instâncias foram testadas todas as seguintes combinações:

- *greentestinvl*: 1, 2, 4, 8, 10.
- *yellowtestinvl*: 2 4 6 8 10 20 30 40 50 80 100 150 200 300 400 500
- *redtestinvl*: 2 4 6 8 10 20 30 40 50 80 100 150 200 300 400 500
- *lambdainit*: 0.01 0.1 1 1.5 1.99
- *alphainit*: 0.1 0.01
- *alphamin*: 10e-4 10e-5
- *alphafactor*: 0.01 0.03 0.1 0.5 0.9
- *alphaint*: 50 70 80 100 150 200 250 300 400 500

A partir dos resultados, os possíveis valores foram redefinidos, diminuindo, assim, o número de possibilidades. Assim, para as demais exceto RBG's, alguns valores foram desconsiderados, sendo testados todas as seguintes combinações:

- *greentestinvl*: 1
- *yellowtestinvl*: 2 4 6 8 10 30 50 100 200 500
- *redtestinvl*: 2 4 6 8 10 30 50 100 200 500
- *lambdainit*: 0.1
- *alphainit*: 0.1

- `alphamin`; 10e-4 10e-5
- `alphafactor`: 0.01 0.03 0.1 0.5 0.9
- `alphaint`: 50 70 80 100 150 200 250 300 400 500

Para as instâncias RBG, as mais difíceis de serem resolvidas (ou seja, tempo de execução muito alto), foram testadas um conjunto ainda menor de possibilidades:

- `greentestinvl`: 1
- `yellowtestinvl`: 2 4 6 8 10 30
- `redtestinvl`: 2 4 6 8 10 30
- `lambdainit`: 0.1
- `alphainit`: 0.1
- `alphamin`; 10e-5
- `alphafactor`: 0.01 0.03 0.1 0.5 0.9
- `alphaint`: 80 100 150 200

A partir destes resultados foi possível verificar que:

- Para o parâmetro `greentestinvl`, praticamente não há diferença nos resultados quando somente este parâmetro varia, exceto em alguns poucos casos. DE fato foi possível observar que não houve diferença nos resultados, para nenhuma instância, a partir do valor 10.
- O par `alphafactor` mostrou-se um dos mais difíceis de se calibrar, pois ele não apresentou um comportamento linear.
- Para o parâmetro `redtestinvl` partir do valor 20 não houve diferença nos resultados.
- Para a iteração `greentestinvl` o valor 1 apresentou sempre os melhores resultados.
- Para `alphainit` e `lambdainit`, o valor 0.1 apresentou sempre os melhores resultados.

A princípio, como ftv80 a ftv170 foram geradas a partir do grafo da ftv170, somente a ftv80 foi utilizada durante a calibração e as demais não foram consideradas. Como os resultados foram satisfatórios com os mesmos parâmetros, não foram feitos mais testes com estas instâncias.

Para as demais instâncias (ATEX, dc, tc, code) e os grupos crane e stilt foram testados, foram testados um conjunto de valores ainda menor:

- greentestinvl: 1
- yellowtestinvl: 2 4 6 8 10 30 100 400
- redtestinvl: 2 4 6 8 10 30 100 400
- lambdainit: 0.1
- alphainit: 0.1
- alphamin; 10e-5
- alphafactor: 0.01 0.03 0.1 0.5 0.9
- alphaint: 100 150

Ocasionalmente, outras configurações deram melhores resultados do que a escolhida para alguns problemas. Além disso, embora o número total de combinações possíveis fosse muito grande, descobriu-se que o parâmetro `alphafactor`, possui um impacto mais significativo no desempenho. Outra observação interessante é que, embora o VOL tenha vários parâmetros, parece que eles operam de forma bastante independente uns dos outros, já que alterar um deles sempre tem um efeito semelhante, independentemente dos demais.

## APÊNDICE E – Processo de calibração – LSSP

O processo de calibração para o LSSP é semelhante ao apresentado no Apêndice D. Primeiro definiu-se um subconjunto de instâncias: Data15-5-0.6-100-0, Data15-5-0.8-100-0, Data15-10-0.6-100-0, Data15-10-0.8-100-0, Data15-15-0.6-100-0, Data15-15-0.8-100-0, Data25-5-0.6-100-0, Data25-5-0.8-100-0, Data25-10-0.6-100-0, Data25-10-0.8-100-0, Data25-15-0.6-100-0 e Data25-15-0.8-100-0.

Em seguida, para essas instâncias, foram testados todas as possíveis combinações determinadas pelos valores abaixo descritos:

- greentestinvl: 1, 2, 4, 8, 10.
- yellowtestinvl: 2 4 6 8 10 20 30 40 50 80 100 150 200 300 400 500
- redtestinvl: 2 4 6 8 10 20 30 40 50 80 100 150 200 300 400 500
- lambdainit: 0.01 0.1 1 1.5 1.99
- alphainit: 0.1 0.01
- alphamin; 10e-4 10e-5
- alphafactor: 0.01 0.03 0.1 0.5 0.9
- alphaint: 50 70 80 100 150 200 250 300 400 500

Após esta fase, observa-se o mesmo comportamento do Algoritmo Volume. Dessa forma, são escolhidos os valores que fornecem o melhor resultado médio.