



HIBRIDIZAÇÃO DE HEURÍSTICAS, COM PROGRAMAÇÃO INTEIRA E MINERAÇÃO DE DADOS, PARA A SOLUÇÃO DE DUAS VARIANTES DO PROBLEMA DO CAIXEIRO VIAJANTE

Francisco Glaubos Nunes Clímaco

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientador: Luidi Gelabert Simonetti

Rio de Janeiro
Setembro de 2020

HIBRIDIZAÇÃO DE HEURÍSTICAS, COM PROGRAMAÇÃO INTEIRA E
MINERAÇÃO DE DADOS, PARA A SOLUÇÃO DE DUAS VARIANTES DO
PROBLEMA DO CAIXEIRO VIAJANTE

Francisco Glaubos Nunes Clímaco

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ
COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE)
DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR
EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientador: Luidi Gelabert Simonetti

Aprovada por:

Prof. Luidi Gelabert Simonetti, D.Sc.

Prof^a. Isabel Cristina Mello Rosseti, D.Sc.

Prof. Yuri Abitbol De Menezes Frota , D.Sc.

Prof. Abilio Pereira de Lucena Filho , Ph.D.

Prof. Glaydston Mattos Ribeiro, D.Sc.

Prof^a. Adria Ramos de Lyra , D.Sc.

RIO DE JANEIRO, RJ – BRASIL

SETEMBRO DE 2020

Glaubos Nunes Clímaco, Francisco

Hibridização de heurísticas, com programação inteira e mineração de dados, para a solução de duas variantes do problema do caixeiro viajante/Francisco Glaubos Nunes Clímaco. – Rio de Janeiro: UFRJ/COPPE, 2020.

XV, 147 p.: il.; 29, 7cm.

Orientador: Luidi Gelabert Simonetti

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2020.

Referências Bibliográficas: p. 140 – 147.

1. Programação inteira.
 2. Mineração de dados.
 3. Hibridização de heurísticas.
 4. PRRCP.
 5. PCVCP.
- I. Gelabert Simonetti, Luidi. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

À minha família.

Agradecimentos

Agradeço à minha família, principalmente à minha esposa Letycia e filha Júlia, pelo suporte incondicional durante todo o período de doutorado. Sem a presença de vocês, a realização deste trabalho não seria possível.

Aos meus orientadores Ludi e Isabel, que me acompanharam desde o início do mestrado, e sempre estiveram dispostos a me ajudar quando precisei.

Aos membros da banca, pelas críticas e sugestões construtivas.

Aos meus colegas do Labotim, Pedro, Sávio, Daniela, Marcos, Hugo e todos os demais da COPPE que, direta e indiretamente, participaram dessa conquista.

Ao Departamento de Informática da Universidade Federal do Maranhão, pelo apoio.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

**HIBRIDIZAÇÃO DE HEURÍSTICAS, COM PROGRAMAÇÃO INTEIRA E
MINERAÇÃO DE DADOS, PARA A SOLUÇÃO DE DUAS VARIANTES DO
PROBLEMA DO CAIXEIRO VIAJANTE**

Francisco Glaubos Nunes Clímaco

Setembro/2020

Orientador: Luidi Gelabert Simonetti

Programa: Engenharia de Sistemas e Computação

Neste trabalho, são apresentadas estratégias para resolver, de maneira eficiente, duas generalizações do problema do caixeiro viajante. O primeiro, o problema de recobrimento de rotas com coleta de prêmios (PRRCP), é motivado pela necessidade de se obter uma rota de custo mínimo para equipe itinerantes, que prestam assistência a comunidades distantes de grandes centros urbanos. Enquanto o segundo, o problema do caixeiro viajante com coleta de prêmios (PCVCP), possui várias aplicações reais, incluindo o planejamento de rotas de helicópteros para plataformas de petróleo no mar, a criação de rotas para turistas, a distribuição de produtos para revendedores, entre outras.

Para a resolução de tais problemas, propõem-se estratégias heurísticas e exatas. Para o PRRCP, é proposta uma nova formulação matemática, novas desigualdades válidas, sete novas regras de redução, e quatro heurísticas híbridas, que combinam uma heurística estado-da-arte com programação linear inteira (PLI) e mineração de dados. Para o PCVCP, é proposto um procedimento de separação para as restrições *cut-sets* de uma formulação da literatura, uma heurística híbrida que combina GRASP e RVND com PLI, e um novo conjunto de instâncias.

No intuito de validar as estratégias propostas, foram realizados experimentos computacionais em instâncias já conhecidas da literatura e em novas apresentadas neste trabalho. Os resultados indicam o benefício da nova formulação e das combinações com PLI e mineração de dados, alcançando soluções de melhor qualidade em menor tempo de processamento para a maioria dos testes, e provando optimalidade inédita para vários casos.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

HYBRIDIZATION OF HEURISTICS WITH INTEGER PROGRAMMING AND
DATA MINING, TO SOLVE TWO VARIANTS OF THE TRAVELING
SALESMAN PROBLEM

Francisco Glaubos Nunes Clímaco

September/2020

Advisor: Luidi Gelabert Simonetti

Department: Systems Engineering and Computer Science

In this work, strategies are presented to solve, in an efficient way, two generalizations of the traveling salesman problem. The first, prize-collecting covering tour problem (PCCTP), it is motivated by the need to obtain a minimum cost route for itinerant staff, who assist communities far from large urban centers. While the second, the prize-collecting traveling salesman problem (PCTSP), has several real applications such as planning helicopter routes for offshore oil platforms, creating routes for tourists, and distributing products to resellers.

To solve such problems, heuristic and exact strategies were proposed. For the PCCTP, we proposed a new mathematical formulation, new valid inequalities, seven new reduction rules, and four hybrid heuristics, which combine a state-of-the-art heuristic with integer linear programming (ILP) and data mining. For the PCTSP, a separation procedure is proposed for the *cut-sets* constraints of a literature's formulation, a hybrid heuristic that combines GRASP and RVND with PLI, and a new set of instances.

In order to validate the proposed strategies, computational experiments were carried out on instances already known in the literature and new ones presented in this work. The results indicate the benefit of the new formulation and the combinations with ILP and data mining, achieving better quality solutions in less processing time for most tests, and proving unprecedented optimality for several cases.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xiii
1 Introdução	1
1.1 Organização da tese	2
2 O problema de recobrimento de rotas com coletas de prêmios	3
2.1 Formulações matemáticas	5
2.2 Regras de Redução	8
2.3 Heurísticas	10
2.3.1 GRASP	11
2.3.2 GENIUS	13
2.3.3 VND	16
2.3.4 Heurística estado-da-arte	18
3 Novas abordagens para o PRRCP	20
3.1 Formulação proposta	20
3.2 Desigualdades válidas	22
3.2.1 Desigualdades de cobertura	22
3.2.2 Desigualdades <i>2-matching</i>	23
3.2.3 Desigualdades <i>cutsets</i>	24
3.3 Algoritmos de separação	24
3.3.1 Restrições <i>cutset</i>	25
3.3.2 Desigualdades de cobertura	26
3.3.3 Desigualdades <i>2-matching</i>	26
3.4 Novas regras de redução para o PRRCP	27
3.5 Heurísticas	33
3.5.1 Heurísticas híbridas	34
3.5.2 Análise do conjunto E^0	35
3.5.3 MPO com E^0 estável	36
3.5.4 Local branching	37

3.5.5	Mineração de dados	38
4	Resultados para o PRRCP	45
4.1	Instâncias utilizadas	45
4.2	Resultado da aplicação das formulações matemáticas	46
4.2.1	Impacto das desigualdades válidas	50
4.3	MPO_S	51
4.4	Local branching	56
4.5	Regras de redução	60
4.5.1	Redução de grafos completos	61
4.5.2	Redução de grafos esparsos	62
4.5.3	Impacto das regras de redução nas formulações matemáticas .	63
4.6	Mineração de dados	64
4.6.1	O impacto da utilização de padrões minerados	71
4.7	Análise das melhores estratégias	75
4.7.1	Testes estatísticos	77
4.8	Considerações finais	79
5	O problema do caixeiro viajante com coleta de prêmios	81
5.1	Revisão da literatura	82
5.2	Clustering Search	85
5.3	CS aplicada ao PCVCP	87
5.3.1	A metaheurística GRASP/VNS	87
5.3.2	O processo de clusterização	89
5.4	Formulações matemáticas	92
5.4.1	Direcionadas	93
5.4.2	Não direcionadas	96
6	Contribuições para a resolução do PCVCP	98
6.1	Heurística	99
7	Resultados para o PCVCP	101
7.1	Instâncias de Chaves <i>et al.</i>	101
7.1.1	Formulações matemáticas	101
7.1.2	Heurísticas	103
7.2	Instâncias de Chaves e Lorena	105
7.2.1	Formulações matemáticas	106
7.2.2	Heurísticas	108
7.3	Novas instâncias	112
7.4	Considerações finais	114

8 Apêndice A **118**

Referências Bibliográficas **140**

Listas de Figuras

2.1	Solução para uma instância do PRRCP.	4
2.2	GENI - Inserção Tipo 1 do vértice v entre os vértices v_i e v_j	13
2.3	GENI - Inserção Tipo 2 do vértice v entre os vértices v_i e v_j	14
2.4	US - Remoção Tipo 1 do vértice v_i .	15
2.5	US - Remoção Tipo 2 do vértice v_i .	16
3.1	<i>Comb</i> com três <i>teeth</i> .	23
3.2	<i>2-matching</i> com três <i>teeth</i> .	24
3.3	Instância associada ao PRRCP, antes da aplicação da regra R8 .	29
3.4	Instância da Figura 3.3, após a aplicação da regra R8 .	29
3.5	Instância associada ao PRRCP, antes e depois da aplicação da regra R9 .	30
3.6	Instância associada ao PRRCP, antes e depois da aplicação da regra R10	30
3.7	Instância associada ao PRRCP, antes da aplicação da regra R11 .	31
3.8	Instância da Figura 3.7 , após a remoção de r_1 , r_2 , $\delta(r_1)$ e $\delta(r_2)$.	31
3.9	Instância da Figura 3.7, após mover r_1 e r_2 para o subconjunto T .	32
3.10	Instância do PRRCP com <i>PRIZE</i> de valor igual 40.	32
3.11	Instância da Figura 3.10, após aplicação da regra R12 .	32
3.12	Instância associada ao PRRCP, antes e depois da aplicação da regra R13 .	33
3.13	Instância associada ao PRRCP, antes da aplicação da regra R14 .	33
3.14	Instância da Figura 3.13, após a aplicação da regra R14 .	34
4.1	Resultados do MPO com a variação de γ .	52
4.2	Impacto da aplicação das sequências em grafos completos.	61
4.3	Impacto da aplicação das sequências em grafos esparsos.	63
4.4	Fase de construção	72
4.5	Fase de busca local	72
4.6	Fase de construção	73
4.7	Fase de busca local	73

4.8	Fase de construção	74
4.9	Fase de busca local	75
4.10	TTTplots para a instância kroA150_VT30_T30_W90_50, alvo: 12962.	76
4.11	TTTplots para a instância kroA150_VT30_T30_W90_50, alvo: 12918.	76
7.1	Função de crescimento do tempo das heurísticas CS* e MPO.	105
7.2	gap médio das heurísticas CS e MPO, em relação à solução ótima. . .	113
7.3	Funções de crescimento das heurísticas CS e MPO, para as instâncias de CHAVES e LORENA [19].	114

Lista de Tabelas

4.1	Resultados obtidos pelas formulações Fluxo, Multi-fluxo e Não-dir para o PRRCP	46
4.1	Resultados obtidos pelas formulações Fluxo, Multi-fluxo e Não-dir para o PRRCP	47
4.1	Resultados obtidos pelas formulações Fluxo, Multi-fluxo e Não-dir para o PRRCP	48
4.1	Resultados obtidos pelas formulações Fluxo, Multi-fluxo e Não-dir para o PRRCP	49
4.2	Resultados dos cortes propostos aplicados na formulação Não-dir. . .	50
4.2	Resultados dos cortes propostos aplicados na formulação Não-dir. . .	51
4.3	Comparação de resultados entre o GRASP-GENIUS-RVND e MPO. .	53
4.3	Comparação de resultados entre o GRASP-GENIUS-RVND e MPO. .	54
4.3	Comparação de resultados entre o GRASP-GENIUS-RVND e MPO. .	55
4.3	Comparação de resultados entre o GRASP-GENIUS-RVND e MPO. .	56
4.4	Resultados da estratégia Local branching com $\delta = 4$	57
4.4	Resultados da estratégia Local branching com $\delta = 4$	58
4.4	Resultados da estratégia Local branching com $\delta = 4$	59
4.5	Resultados em termos de qualidade de solução, da aplicação das regras de redução com as formulações.	64
4.6	Comparação entre o GRASP-GENIUS-RVND e o DM-GRASP.	65
4.6	Comparação entre o GRASP-GENIUS-RVND e o DM-GRASP.	66
4.6	Comparação entre o GRASP-GENIUS-RVND e o DM-GRASP.	67
4.6	Comparação entre o GRASP-GENIUS-RVND e o DM-GRASP.	68
4.7	Comparação entre DM-GRASP e MDM-GRASP.	68
4.7	Comparação entre DM-GRASP e MDM-GRASP.	69
4.7	Comparação entre DM-GRASP e MDM-GRASP.	70
4.8	Resumo dos resultados e teste de significância estatística entre GRASP-GENIUS-RVND e MDM-GRASP.	77
4.9	Resumo dos resultados e do teste de significância estatística entre GRASP-GENIUS-RVND e MPO_S.	78

7.1	Resultados das formulações matemáticas para as instâncias de CHAVES <i>et al.</i> [17].	102
7.2	Resultados das heurísticas para as instâncias de CHAVES <i>et al.</i> [17] (Tempos de execução originais do CS* multiplicados por 0.214).	103
7.3	Resultados das formulações para instâncias de CHAVES e LORENA [19], $c_{ij} \in [1, 1000]$ e $\gamma_i \in [1, 100]$	106
7.4	Resultados das formulações para as instâncias de CHAVES e LORENA [19], $c_{ij} \in [1, 10000]$ e $\gamma_i \in [1, 1000]$	107
7.5	Resultado das formulações para as instâncias de CHAVES e LORENA [19], $c_{ij} \in [1, 10000]$ e $\gamma_i \in [1, 100]$	107
7.6	Resultado das formulações para as instâncias de CHAVES e LORENA [19], $c_{ij} \in [1, 1000]$ e $\gamma_i \in [1, 10000]$	108
7.7	Resultados para o modelo de CHAVES e LORENA [16], em que a literatura apresenta melhores soluções que nesta tese.	109
7.8	Resultados das heurísticas para instâncias de CHAVES e LORENA [19], $c_{ij} \in [1, 1000]$ e $\gamma_i \in [1, 100]$ (Tempos de execução originais do CS multiplicados por 0,3).	109
7.9	Resultados das heurísticas para instâncias de CHAVES e LORENA [19], $c_{ij} \in [1, 10000]$ e $\gamma_i \in [1, 1000]$ (Tempos de execução originais do CS multiplicados por 0.3).	110
7.10	Resultados das heurísticas para instâncias de CHAVES e LORENA [19], $c_{ij} \in [1, 10000]$ e $\gamma_i \in [1, 100]$ (Tempos de execução originais do CS multiplicados por 0.3).	111
7.11	Resultados das heurísticas para instâncias de CHAVES e LORENA [19], $c_{ij} \in [1, 1000]$ e $\gamma_i \in [1, 10000]$ (Tempos de execução originais do CS multiplicados por 0.3).	112
7.12	Resultados do MPO para a novas instâncias do PCVCP.	115
8.1	Resultados do local branching com $\delta = 3$	118
8.2	Resultados do local branching com $\delta = 4$	119
8.4	Resultados da aplicação das sequências A e B em grafos completos.	119
8.4	Resultados da aplicação das sequências A e B em grafos completos.	120
8.5	Resultados da aplicação das sequências C e D em grafos completos.	120
8.5	Resultados da aplicação das sequências C e D em grafos completos.	121
8.6	Resultados da aplicação das sequências E e F em grafos completos.	121
8.3	Resultados do local branching com $\delta = 5$	122
8.7	Resultados da aplicação das sequências A e B em grafos esparsos.	122
8.7	Resultados da aplicação das sequências A e B em grafos esparsos.	123
8.8	Resultados da aplicação das sequências C e D em grafos esparsos.	123

8.8	Resultados da aplicação das sequências C e D em grafos esparsos. . .	124
8.9	Resultados da aplicação das sequências E e F em grafos esparsos. . .	124
8.10	Resultados da aplicação das sequências AA e BB antes e após a remoção de E^0	125
8.13	Resultados das formulações matemáticas após a aplicação das regras de redução.	125
8.13	Resultados das formulações matemáticas após a aplicação das regras de redução.	126
8.13	Resultados das formulações matemáticas após a aplicação das regras de redução.	127
8.13	Resultados das formulações matemáticas após a aplicação das regras de redução.	128
8.13	Resultados das formulações matemáticas após a aplicação das regras de redução.	129
8.11	Resultados da aplicação das sequências CC e DD antes e após a remoção de E^0	130
8.12	Resultados da aplicação das sequências EE e FF antes e após a remoção de E^0	131
8.14	Resultados do DM-GRASP com $sup_{min} = 20\%$	132
8.15	Resultados do DM-GRASP com $sup_{min} = 30\%$	133
8.16	Resultados do DM-GRASP com $sup_{min} = 40\%$	134
8.17	Resultados do DM-GRASP com $sup_{min} = 50\%$	135
8.18	Resultados do MDM-GRASP com $sup_{min} = 20\%$	136
8.19	Resultados do MDM-GRASP com $sup_{min} = 30\%$	137
8.20	Resultados do MDM-GRASP com $sup_{min} = 40\%$	138
8.21	Resultados do MDM-GRASP com $sup_{min} = 50\%$	139

Capítulo 1

Introdução

A prestação de assistência social, médica ou jurídica a comunidades geograficamente distantes de grandes centros urbanos tem sido uma preocupação de diversas entidades públicas, filantrópicas ou privadas. Contudo, a implantação de unidades fixas nesses locais envolve altos investimentos, e ainda há poucos profissionais com interesse em trabalhar nessas regiões.

Uma das soluções mais comuns para atender pessoas que moram em lugares distantes dos centros de atendimento é o uso de equipes itinerantes. Como exemplo, no Brasil, o Tribunal de Justiça do Maranhão (TJ-MA) tem um projeto chamado Justiça Itinerante, no qual um ônibus adaptado com uma estrutura completa de fórum é usado para executar a maioria dos serviços legais. Esse veículo cruza periodicamente cidades distantes das unidades do TJ-MA, ou bairros nas periferias das principais capitais, cobrindo assim um número maior de cidadãos, que sem esse serviço não poderiam ter assistência legal.

Nesse contexto, surge o problema de recobrimento de rotas com coleta de prêmios (PRRCP). O objetivo desse problema é encontrar uma rota de custo mínimo que passe por algumas cidades obrigatórias e, eventualmente, por outras não obrigatórias. Cada cidade possui um prêmio associado, que representa o número de pessoas atendidas caso a rota passe por ela. Algumas cidades podem ser inacessíveis, e seu habitantes devem se deslocar até a cidade mais próxima da rota para receber atendimento. Por fim, a rota deve garantir um atendimento mínimo de pessoas, para que valha a pena a viagem de um ônibus do TJ. Mais detalhes e aplicações do PRRCP são apresentados no início do Capítulo 2.

Outro problema que envolve a determinação de uma rota com custo mínimo é o problema do caixeiro viajante com coleta de prêmios (PCVCP). Nesse problema, cada cidade possui um prêmio e uma penalidade associada. O objetivo é minimizar o custo da rota mais as penalidades pagas, por cada cidade não visitada, enquanto inclui na rota, cidades suficientes para coletar um prêmio mínimo, definido a priori.

Ambos os PRRCP e PCVCP são variantes do problema do caixeiro viajante (PCV)

clássico, e são classificados como problemas NP-Difíceis. Portanto, para resolver essa classe de problemas, o desenvolvimento de estratégias eficientes é desafiador. Neste trabalho, são apresentadas propostas de soluções para esses dois problemas.

As propostas para o PRRCP são as seguintes: i) um nova formulação matemática, que reduza o esforço computacional requerido pelas formulações da literatura; ii) novas estratégias de pré-processamento baseadas em características ainda não exploradas do problema, e um estudo inédito acerca da melhor maneira de aplicá-las; iii) desigualdades válidas que fortaleçam a relaxação linear do problema, juntamente com procedimentos de separação para estas desigualdades; e iii) heurísticas híbridas, combinando a heurística estado-da-arte com programação linear inteira (PLI) e mineração de dados.

Já para o PCVCP, as propostas desta tese são: i) aplicação inédita de uma formulação não direcionada para instâncias da literatura; ii) comparação de desempenhos entre diversas formulações da literatura; iii) novas instâncias, mais difíceis que as instâncias encontradas na literatura; e iv) um método híbrido, que combina PLI com uma heurística baseada em GRASP [88].

1.1 Organização da tese

O restante deste trabalho está dividido em dois grupos de capítulos. O primeiro grupo aborda o problema de recobrimento de rotas com coletas de prêmios, enquanto o segundo grupo de capítulos discorre sobre o problema do caixeiro viajante com coleta de prêmios. Neste primeiro grupo, no Capítulo 2 é realizado um levantamento bibliográfico acerca do PRRCP, apresentando as formulações presentes na literatura, as regras de redução e a heurística estado-da-arte para o problema. No Capítulo 3 é apresentada com detalhes a proposta deste trabalho. Por fim, no Capítulo 4 são reportados os resultados dos experimentos computacionais realizados, e algumas considerações finais são feitas.

No segundo grupo de capítulos, é tratado o problema do caixeiro viajante com coleta de prêmios. No Capítulo 5 é apresentada uma revisão bibliográfica dos principais métodos de solução para o PCVCP. As propostas de resolução do problema são introduzidas no Capítulo 6. Por último, no Capítulo 7 são reportados os resultados dos experimentos realizados, e algumas considerações finais são feitas.

Capítulo 2

O problema de recobrimento de rotas com coletas de prêmios

O PRRCP foi inicialmente proposto por LYRA [71]. Esse problema é uma variação do problema de recobrimento de rotas (PRR) [43], que por sua vez é uma variação do problema do caixeiro viajante (PCV) [67]. Considere um grafo não direcionado $G = (N, E)$, com o conjunto de vértices $N = V \cup W$ e $V = R \cup T$, uma distância de cobertura D , um custo $c_e \geq 0$ associado a cada aresta $e \in E$ e um prêmio p_i associado a cada vértice $i \in V$. O subconjunto T contém vértices obrigatórios, que necessariamente devem ser visitados. Vértices pertencentes a R são opcionais e, portanto, podem ou não ser visitados. Por fim, o subconjunto W contém vértices que devem ser cobertos por outros vértices que não pertençam a W . Entende-se que um vértice $i \in V$ cobre um vértice $j \in W$ se $c_{ij} \leq D$.

O objetivo do PRRCP é encontrar um ciclo simples de custo mínimo que visite todos os vértices em T , coletando um prêmio mínimo (*PRIZE*), e que cubra todos os vértices do subconjunto W . A Figura 2 ilustra uma possível solução para uma instância desse problema, onde os círculos representam a área de cobertura de um vértice $j \in W$, os rótulos nos vértices indicam o seu respectivo prêmio, e o *PRIZE* mínimo a ser coletado é dez.

Uma aplicação real para o PRRCP pode ser observado no caso de fornecimento de equipes itinerantes mencionado no Capítulo 1. Os vértices de T representariam cidades de uma determinada região que são importantes e, portanto, são definidas como pontos de visita obrigatória, enquanto os vértices em R representariam cidades nas quais as visitas são opcionais. Há também um conjunto de cidades que devem ser assistidas, mas, por algum motivo, essa visita é impraticável e sua população deve ser atendida por algumas cidades vizinhas, cuja distância é de, no máximo, D unidades, evitando um grande deslocamento de pessoas. Esse último conjunto de cidades seriam representadas pelos vértices em W . É necessário que uma quantidade mínima de pessoas seja atendida, para que a liberação de uma equipe itinerante valha

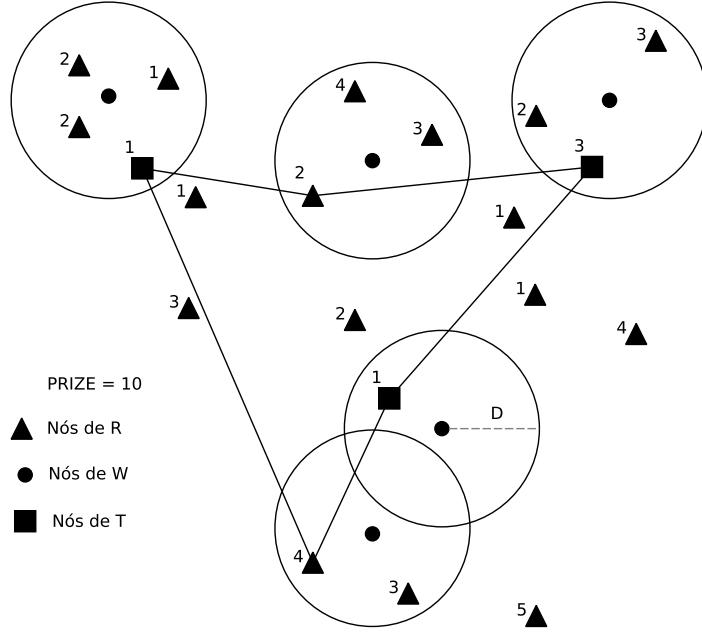


Figura 2.1: Solução para uma instância do PRRCP.

a pena. Portanto, a expectativa de atendimento de cada cidade em V é representada pelo prêmio de cada respectivo vértice, e o *PRIZE* a ser coletado corresponde ao total mínimo de cidadãos a serem atendidos.

Outra aplicação para o PRRCP pode ser investigada com base na situação descrita em CURRENT e SCHILLING [25], na qual é necessário projetar uma rota aérea de distribuição de suprimentos. Nesse caso, cidades maiores são servidas pelo ar, algumas servidas obrigatoriamente (T) e outras não (R). As cidades menores que não estão na rota (W) são servidas (cobertas) por caminhões, que saem das cidades mais próximas que estão na rota aérea.

Na aplicação do PRRCP, cada cidade servida pelo ar possui um prêmio associado, que significa o seu número de habitantes. Análogo ao exemplo anterior, uma aeronave somente pode ser liberada, se o planejamento da rota garantir que um número mínimo de pessoas seja atendido (*PRIZE*).

O PRRCP é um problema academicamente interessante, pois abrange subproblemas como o problema da mochila binária, o problema da cobertura de rotas, e o PCV. Nesta teste, esta abrangência permitiu o estudo das características de cada um destes problemas, e a elaboração de estratégias que podem ser utilizadas para resolver separadamente, cada um deles ou problemas semelhantes, como o Quota TSP [12] ou o *orienteering problem* [96].

2.1 Formulações matemáticas

Na literatura há duas formulações matemáticas para o PRRCP. A primeira formulação, proposta por LYRA [71], utiliza variáveis de fluxo para evitar formação de ciclos desconexos. A outra formulação da literatura, proposta por SILVA [93], utiliza variáveis de multi-fluxo para evitar subciclos na rota.

Nesta seção, considere $G = (V, A)$ como sendo um grafo completo e direcionado no qual V e A são, respectivamente, os conjuntos de vértices e arcos, e u um vértice raiz pertencente a T , escolhido como origem da rota. Segundo LYRA [71], o PRRCP pode ser formulado como um modelo de programação linear inteira (PLI) utilizando as seguintes variáveis:

- z_{ij} : variável inteira não-negativa que representa a quantidade de fluxo escoado no arco (i, j) ;
- y_k : variável binária para todo $k \in V$. É atribuído valor igual a um a y_k , se o vértice k estiver na rota, e valor igual a zero, caso contrário; e
- x_{ij} : variável binária para todo arco $(i, j) \in A$. Ela assume o valor igual a um, se o arco (i, j) pertence à rota, e valor igual a zero, caso contrário.

Além disso, c_{ij} representa o custo de se utilizar o arco $(i, j) \in A$, p_k indica o prêmio associado a cada vértice $k \in V$, $PRIZE$ é o prêmio mínimo a ser coletado pela rota, e R_w é o conjunto de todos os vértices $k \in V$ que cobrem $w \in W$. Dessa forma, o PPLI pode ser definido da seguinte maneira:

$$\text{Minimizar} \sum_{(i,j) \in A} c_{ij}x_{ij} \quad (2.1)$$

$$\text{s.a : } \sum_{k \in V} p_k y_k \geq PRIZE \quad (2.2)$$

$$\sum_{k \in R_w} y_k \geq 1, \quad \forall w \in W \quad (2.3)$$

$$\sum_{(i,k) \in A} x_{ik} + \sum_{(k,j) \in A} x_{kj} = 2y_k, \quad \forall k \in V \quad (2.4)$$

$$\sum_{j \in V} z_{kj} = \sum_{i \in V} z_{ik} + y_k, \quad \forall k \in V \setminus \{u\} \quad (2.5)$$

$$\sum_{j \in V} z_{uj} = 1 \quad (2.6)$$

$$\sum_{j \in V} z_{ju} = \sum_{j \in V} y_j, \quad (2.7)$$

$$x_{ij} \leq z_{ij}, \quad \forall (i, j) \in A \quad (2.8)$$

$$x_{ij} \geq z_{ij}/(|V| + 1), \quad \forall (i, j) \in A \quad (2.9)$$

$$y_k = 1, \quad \forall k \in T \quad (2.10)$$

$$y_k \in \{0, 1\} \quad \forall k \in R \quad (2.11)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (2.12)$$

$$z_{ij} \in \mathbb{Z}^+ \quad \forall (i, j) \in A \quad (2.13)$$

Nessa formulação, a função objetivo (2.1) minimiza o custo da rota. A restrição (2.2) garante que o prêmio mínimo, $PRIZE$, seja coletado. As restrições (2.3) asseguram que cada vértice de W seja coberto por pelo menos um vértice da rota. As restrições (2.4) são responsáveis pela conservação de fluxo, enquanto as restrições (2.5), (2.6) e (2.7) evitam que sejam formados ciclos desconexos da origem. As restrições (2.8) e (2.9) asseguram que as rotas geradas a partir das variáveis de fluxo z_{ij} e das variáveis binárias x_{ij} coincidam. As restrições (2.10) garantem que todos os vértices de T estejam na rota. Por fim, as restrições (2.11), (2.12) e (2.13) determinam o domínio das variáveis y_k , x_{ij} e z_{ij} .

Como alternativa à formulação proposta por LYRA [71], SILVA [93] propôs uma nova modelagem matemática, que utiliza variáveis multi-fluxo, em vez de fluxo, para evitar ciclos desconexos do vértice origem. Essa nova formulação é composta pelas seguintes variáveis:

- x_{ij} : variável binária que assume valor igual a um se a rota contém o arco (i, j) , e valor igual a zero, caso contrário;
- y_k : variável binária que indica se o vértice k está na rota ou não. Portanto, para todo vértice $k \in T$, $y_k = 1$;
- z_{ij}^k : variável inteira não-negativa que representa a quantidade de fluxo do produto k escoado no arco $(i, j) \in A$.

Dadas as variáveis supracitadas, a formulação de SILVA [93] pode ser apresentada da seguinte maneira:

$$\text{Minimizar} \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (2.14)$$

$$s.a : \sum_{j \in V : j \neq i} x_{ij} = y_i, \quad \forall i \in V \quad (2.15)$$

$$\sum_{i \in V : i \neq j} x_{ji} = y_j, \quad \forall j \in V \quad (2.16)$$

$$z_{ij}^k \leq x_{ij}, \quad \forall (i, j) \in A, k \in V \quad (2.17)$$

$$\sum_{i \in V \setminus \{u\}} z_{ui}^k = y_k, \quad \forall k \in V \setminus \{u\} \quad (2.18)$$

$$\sum_{i \in V \setminus \{u\}} z_{iu}^k = 0, \quad \forall k \in V \setminus \{u\} \quad (2.19)$$

$$\sum_{i \in V \setminus \{k\}} z_{ik}^k = y_k, \quad \forall k \in V \setminus \{u\} \quad (2.20)$$

$$\sum_{j \in V \setminus \{k\}} z_{kj}^k = 0, \quad \forall k \in V \setminus \{u\} \quad (2.21)$$

$$\sum_{i \in V : i \neq j} z_{ij}^k - \sum_{i \in V : i \neq j} z_{ji}^k = 0, \quad \forall k, j \in V \setminus \{u\}, j \neq k \quad (2.22)$$

$$\sum_{i \in V} p_i y_i \geq PRIZE \quad (2.23)$$

$$\sum_{i \in R_w} y_i \geq 1 \quad \forall w \in W \quad (2.24)$$

$$y_i = 1 \quad \forall i \in T \quad (2.25)$$

$$y_i \in \{0, 1\} \quad \forall i \in R \quad (2.26)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (2.27)$$

$$z_{ij}^k \in \mathbb{Z}^+ \quad \forall (i, j) \in A, k \in V \quad (2.28)$$

Nessa formulação, a função objetivo (2.14) minimiza o custo total da rota. As restrições (2.15) e (2.16) garantem que se um vértice está na rota, então esse vértice possui grau igual a dois. As restrições (2.17) limitam a passagem de fluxo de produtos à aresta da rota. As restrições (2.18) asseguram que apenas um produto é escoado da origem para cada cliente presente na rota. As restrições (2.19) não permitem que nenhum produto retorne à origem. As restrições (2.20) e (2.21) asseguram que todo vértice receberá seu produto correspondente, que por sua vez não deverá ser enviado a nenhum outro vértice. As restrições (2.22) garantem a conservação do

fluxo dos produtos que não alcançaram seus vértices de destino. A restrição (2.23) assegura que o *PRIZE* será coletado. As restrições (2.24) garantem que todo vértice de W é coberto por pelo menos um vértice da rota. As restrições (2.25) impõem que todos os vértices de T estejam na rota. Por fim, as restrições (2.26), (2.27) e (2.28) representam o domínio das variáveis y_i , x_{ij} e z_{ij}^k .

Na formulação de SILVA [93], obtém-se o problema de recobrimento de rotas (PRR) ao se remover as restrições (2.23). Seja $n = |V \cup W|$, $p = |V|$ e $m = |E|$, tem-se que o número de restrições da formulação de LYRA [71] é da ordem de $O(n^2)$ e de variáveis geradas é $O(2m + p)$. Já na formulação de SILVA [93], o número de restrições e variáveis é da ordem de $O(n^3)$.

2.2 Regras de Redução

Um dos principais gargalos dos problemas de Otimização Combinatória é o tamanho de suas entradas [71]. Ao se limitar o espaço de busca, o esforço computacional para resolver um problema torna-se menor. Dessa forma, as regras de redução surgiram como uma maneira de reduzir o espaço de busca explorado, não afetando o encontro de soluções ótimas para um problema, removendo-se vértices e arestas.

As regras de redução para o PRRCP foram inicialmente apresentadas por LYRA [71]. São duas adaptações das regras criadas por GENDREAU *et al.* [43] para o PRR, e uma nova regra relacionando o subconjunto R com o *PRIZE*. Além disso, em LYRA [71] foi proposta uma ordem para a aplicação dessas regras, de modo que haja uma maior redução do grafo, sem comprometer o encontro da solução ótima.

Posteriormente, três novas regras de redução foram propostas por SILVA [93], direcionadas principalmente à redução do subconjunto de vértices opcionais. Assim como em LYRA [71], foi realizada uma análise acerca da ordem de aplicação dessas novas regras, considerando as regras já existentes na literatura.

As regras apresentadas por LYRA [71] e SILVA [93] foram propostas considerando a hipótese que os custos c_{ij} satisfazem a desigualdade triangular, ou seja, $\forall(i, j) \in E$ e $k \in V \setminus \{i, j\}$, $c_{ij} \leq c_{ik} + c_{kj}$. Portanto, neste trabalho, toda regra aplicada a partir dessa hipótese possuirá o símbolo Δ no início.

A seguir, são apresentadas em ordem de aplicação de acordo com SILVA [93], as regras presentes na literatura. Lembre que $R_w = \{i \in R \cup T \mid c_{iw} \leq D\}$, e considere $R_W = \bigcup_{w \in W} R_w$, e um conjunto de vértices em W cobertos por um vértice $i \in V$ como $R_i = \{w \in W \mid c_{iw} \leq D\}$.

- **R1** - Se $\exists w \in W$, onde $|R_w| = 1$, então $T = T \cup R_w$.
- **R2** - Se $\exists j \in R$, onde $p_j > 0$ e $\sum_{i \in V \setminus \{j\}} p_i < PRIZE$, então $T = T \cup \{j\}$.

- **R3** - Se Δ e $\sum_{i \in T} p_i \geq PRIZE$, então todos os vértices de $R \setminus R_W$ podem ser removidos de G , desde que o grafo resultante dessa remoção seja completo em relação a T .
- **R4** - Se Δ , $\sum_{i \in T} p_i \geq PRIZE$ e $\exists i \in R$ onde $R_i \subseteq \bigcup_{t \in T} R_t$, então pode-se remover o vértice i de G , desde que o grafo resultante seja completo em relação a T .
- **R5** - Se Δ , $\exists i, j \in R$, $i \neq j$, $\sum_{k \in V \setminus \{i\}} p_k \geq PRIZE$, $R_i \subseteq R_j$, $c_{ik} > c_{jk} \forall k$ adjacente a i , e $\exists (v_1, v_2) \in E \forall (i, v_1)$ e $(i, v_2) \in E$, então pode-se remover o vértice i de G .
- **R6** - Se $\exists t \in T$, tal que $R_t \neq \emptyset$, então $W = W \setminus R_t$.
- **R7** - Se $\exists w \in W$ tal que $R_w = R$ então $W = W \setminus \{w\}$ e pelo menos um vértice de R deve estar na solução.

Para a aplicação das regras de redução, considera-se que primeiramente o subconjunto de vértices obrigatórios deve ser definido. Isto é, devem ser buscados vértices em R que possam ser movidos para o subconjunto T . Portanto, na regra **R1** proposta por LYRA [71], caso $j \in R$ seja o único vértice que possa cobrir algum $i \in W$, j é movido para T . Na regra **R2**, proposta por SILVA [93], um vértice $j \in R$ torna-se obrigatório, caso o $PRIZE$ não possa ser coletado sem p_j .

Se aplicadas logo no início, as regras **R1** e **R2** poderão reduzir o tamanho do subconjunto R . Em consequência disso, o tamanho do conjunto T aumentará, incrementando o prêmio coletado por vértices que obrigatoriamente estarão na rota, e contribuindo também para a aplicação das regras **R3**, **R4**, **R5** e **R6**. Se o prêmio coletado por todos os vértices em T já satisfazer o $PRIZE$, o PRRCP pode ser tratado como sendo o PPR, uma vez que não há mais a necessidade de se coletar um prêmio mínimo [93].

As próximas regras a serem aplicadas **R3**, **R4** e **R5**, são aquelas que poderão remover vértices restantes em R . Além disso, para essas regras, o grafo de entrada deve satisfazer a desigualdade triangular. Caso contrário, soluções ótimas para o problema podem não ser encontradas. Para as regras **R3** e **R4**, exige-se que o grafo resultante após a redução seja completo em relação ao subconjunto T . Na regra **R3** de LYRA [71], uma vez que o $PRIZE$ já é atingido pela coleta dos vértices obrigatórios, pode-se remover todo vértice $r \in R$ que não cubra nenhum vértice de W .

A segunda regra proposta por SILVA [93], **R4**, afirma que, se o $PRIZE$ já for alcançado com a soma apenas dos vértices obrigatórios, e que se os vértices cobertos por $r \in R$ também forem cobertos pelo subconjunto T , r pode ser removido. A regra

R5, também proposta por SILVA [93], é aplicada a dois vértices $i, j \in R$ quaisquer, tal que $i \neq j$. Nessa regra, o vértice i pode ser removido se as seguintes condições forem atendidas: i) o *PRIZE* puder ser coletado sem a necessidade do prêmio do vértice i ; ii) todos os vértices cobertos por i também serem cobertos por j ; iii) a distância do vértice i para todos seus vértices adjacentes, ser maior que a distância do vértice j para todos esses mesmos vértices; e iv) existência de uma aresta entre qualquer dois vértices adjacentes a i .

Neste trabalho, para as regras **R3**, **R4** e **R5** supracitadas, adiciona-se a observação de que o grafo resultante, após a remoção de um vértice $r \in R$, deve ser completo em relação a $T \cup R_W$, isto é, a remoção de r não deve implicar na remoção de uma aresta incidente a T ou R_W , que poderia ser escolhida para reduzir o custo da rota.

Segundo a ordem de aplicação de SILVA [93], a próxima é a regra **R6**. Essa regra foi proposta por GENDREAU *et al.* [43] para o PRR, porém, também pode ser aplicada ao PRRCP sem modificações. Trata-se de uma regra que remove todos os vértices em W que são cobertos por algum vértice em T . Por fim, a última regra a ser aplicada é a **R7**, proposta por LYRA [71]. Essa regra consiste em remover um vértice $w \in W$, caso w seja coberto por todos os vértices do subconjunto R , e garantir a presença de pelo menos um vértice de R na solução. Além disso, antes de sua aplicação, deve-se verificar se o subconjunto R não está vazio.

2.3 Heurísticas

Para a resolução do PRRCP, algumas heurísticas presentes na literatura foram propostas, e as principais são descritas a seguir. A primeira abordagem foi realizada por LYRA [71], e trata-se de um algoritmo de construção por adição de vértices, denominado ADD. Nesse método, uma solução parcial é gerada a partir dos vértices obrigatórios T , e, ao longo das iterações, por meio de um critério de inserção, vértices que ainda não pertencem a rota são inseridos.

Além do ADD, LYRA [71] desenvolveu duas buscas locais, no intuito de aprimorar a solução encontrada pelo seu algoritmo de construção. Na primeira busca local (BL1), é realizada a troca de um vértice $r_1 \in R$ que está na rota, por outro vértice $r_2 \in R$ que não está na rota, e verifica-se se a solução resultante da troca é melhor que a gerada na construção. A BL1 foi aplicada seguindo o critério *First Improvement* [62].

A segunda busca local (BL2) é baseada na metaheurística VNS (*Variable Neighbourhood Search*) [60], e explora a vizinhança de uma solução da seguinte maneira. Enquanto um contador $k \leftarrow 1$ for menor que um valor k_{max} , uma solução x' é obtida aleatoriamente a partir da k -ésima vizinhança de x ($x' \in N_k(x)$); e então aplica-se uma busca local em x' , obtendo-se uma solução x'' ; se esse ótimo local for melhor

que a melhor solução corrente, então x é atualizada como x'' ($x \leftarrow x''$), e o processo de busca continua com $N_1(k \leftarrow 1)$; caso contrário, $k \leftarrow k + 1$. Em BL2, LYRA [71] define uma vizinhança como sendo a troca de n vértices da rota por n vértices fora da rota, na qual n varia de 1 a $|R|$.

Em seguida, SILVA [93] propôs cinco versões de um algoritmo heurístico baseado em *Iterated Local Search* [70] e um Algoritmo Evolutivo (AE) [30], que gerou as melhores soluções para a maioria dos problemas-teste, mas demandando um tempo computacional elevado. Segundo SILVA [93], o AE sobressaiu-se em qualidade de solução devido à característica do algoritmo de ser bem-sucedido com a hibridização de dois tipos de buscas, local e populacional. A busca populacional é realizada a partir de uma população de indivíduos do AE, na qual garante exploração de boa parte do espaço de busca, enquanto a busca local identifica indivíduos de boa qualidade em regiões consideradas promissoras.

Os métodos supracitados são estratégias bem sucedidas na abordagem do PRRCP. Entretanto, neste trabalho, será apresentada detalhadamente apenas a estratégia estado-da-arte para esse problema, o GRASP-GENIUS-RVND, proposto por DA SILVA [26]. Na próxima seção, são apresentadas separadamente as heurísticas utilizadas no trabalho de DA SILVA [26], e, posteriormente, como essas heurísticas foram combinadas na construção do método estado-da-arte para o PRRCP.

2.3.1 GRASP

A metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*) é um processo multipartida, que obtém soluções suficientemente boas para problemas computacionalmente difíceis. Esse método tem sido aplicado com êxito na resolução de vários problemas de otimização, em várias áreas como problemas de escalonamentos, roteamento, particionamento, alocação e atribuição [33, 34].

Essa metaheurística é executada iterativamente, e cada iteração é composta por duas fases: construção e busca local. Basicamente, uma solução viável é gerada na fase de construção, e então sua vizinhança é explorada pela busca local, com o objetivo de encontrar uma solução de melhor qualidade. Esse processo iterativo é repetido até que um critério de parada seja alcançado, que pode ser um número limite de iterações permitidas, um número de iterações sem melhoria de solução, entre outros. Uma vez atingido esse critério de parada, a melhor solução alcançada ao longo de todas as iterações, é retornada como o resultado final do processo.

O pseudocódigo da metaheurística GRASP para um problema de minimização é apresentado no Algoritmo 1. Na linha 1, a variável que armazena a melhor solução é iniciada como vazia. O bloco de instruções entre as linhas 2 e 8 é executado iterativamente. A fase de construção é executada na linha 3 e, na linha 4, a busca

local é executada a partir da solução gerada na fase anterior. Nas linhas 5 e 6, a qualidade da solução atual é comparada à melhor solução encontrada até o momento, e, se necessário, a melhor solução é atualizada. Finalmente, na linha 9, a melhor solução é retornada.

Algoritmo 1: Pseudocódigo básico da metaheurística GRASP.

```

1:  $S^* \leftarrow \emptyset; f^* \leftarrow \infty;$ 
2: Enquanto Criterio de Parada não for satisfeito faca
3:    $S \leftarrow \text{FaseDeConstrucao}();$ 
4:    $S_{bl} \leftarrow \text{BuscaLocal}(S);$ 
5:   Se  $f(S_{bl}) < f^*$  então
6:      $S^* \leftarrow S_{bl}; f^* \leftarrow f(S_{bl});$ 
7:   Fim-se
8: Fim-enquanto
9: Retorne  $S^*;$ 
```

Na fase de construção, os componentes da solução são selecionados individualmente e agrupados a uma solução parcial, até que essa seja completamente gerada. Esse processo é ilustrado no Algoritmo 2. Na linha 1, a solução é inicializada como um conjunto vazio, e da linha 2 até a linha 6, as componentes que ainda não estão na solução, são classificadas de acordo com uma função gulosa. As componentes melhores classificadas formam uma lista restrita de candidatos (LRC), na linha 3. Na linha 4, uma componente é escolhida aleatoriamente a partir dessa lista e inserida na solução parcial atual na linha 5. E, por fim, na linha 7, a solução gerada nessa fase é retornada.

Algoritmo 2: Pseudocódigo da fase de construção básica do GRASP.

```

1:  $S \leftarrow \emptyset;$ 
2: Enquanto Solucao Incompleta faca
3:    $LRC \leftarrow \text{ConstruirLRC}(S);$ 
4:    $s \leftarrow \text{SelecionarAleatoriamente}(LRC);$ 
5:    $S \leftarrow S \cup \{s\};$ 
6: Fim-enquanto
7: Retorne  $S;$ 
```

Após passar por uma fase de construção, não há garantia que a solução construída seja um ótimo local. Nesse sentido, é aplicada uma busca local, um processo que explora a vizinhança de uma solução, buscando por soluções de melhor qualidade. A vizinhança de uma solução é definida por uma função que relaciona a similaridade entre essa solução e um subconjunto de soluções. Durante esse processo, se uma solução melhor x' é encontrada, uma nova busca local é realizada a partir de x' . Caso contrário, a busca local termina. Para a heurística estado-da-arte, DA SILVA [26] utilizou a heurística GENIUS [44] na fase de construção do GRASP.

2.3.2 GENIUS

Proposta por Gendreau *et al.* [42], a heurística GENIUS é composta por duas fases: um procedimento de construção da solução, GENI (*Generalized Insertion*) e um de melhoria dessa solução, US (*Unstringing and Stringing*). A fase GENI inicia-se, a partir de uma solução parcial, uma rota S contendo três vértices de uma lista de candidatos. Posteriormente, a cada iteração, tenta-se inserir um novo vértice v na solução S , e então é executada uma re-otimização dessa rota. Essa re-otimização é dada pela combinação dos p vizinhos mais próximos de v , a fim de reduzir o custo de S .

Na fase GENI, há duas maneiras diferentes para inserir um vértice v . Suponha que v será inserido entre outros dois vértices v_i e v_j da rota. Dado que a rota possui um sentido, considere v_k um vértice no caminho de v_j até v_i , e v_l um vértice no caminho de v_i a v_j . Além disso, considere também que dado um vértice v_h da rota, v_{h-1} é seu antecessor e v_{h+1} é seu sucessor. Os dois tipos de inserção são explicados a seguir, e ilustrados nas Figuras 2.2 e 2.3:

- *Inserir Tipo 1*: antes de v ser inserido, são removidas as arestas (v_i, v_{i+1}) , (v_j, v_{j+1}) e (v_k, v_{k+1}) . Essas últimas arestas, serão substituídas por quatro novas arestas (v_i, v) , (v, v_j) , (v_{i+1}, v_k) e (v_{j+1}, v_{k+1}) , tal que $k \neq i$ e $k \neq j$. Para a preservação da orientação da rota, o sentido dos caminhos (v_{i+1}, \dots, v_j) e (v_{j+1}, \dots, v_k) são invertidos.

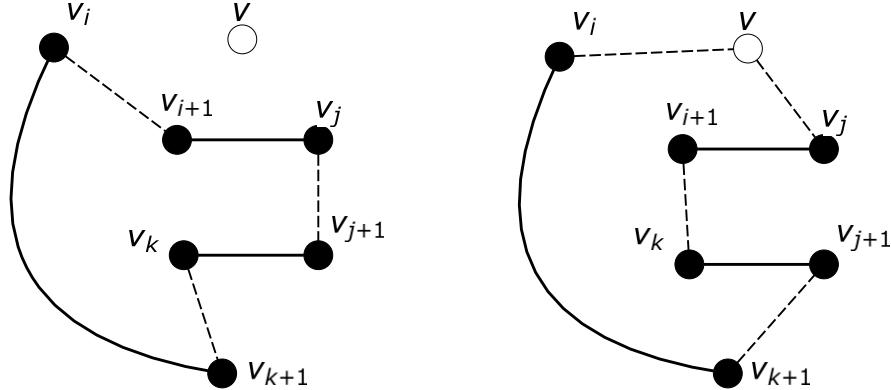


Figura 2.2: GENI - Inserção Tipo 1 do vértice v entre os vértices v_i e v_j

- *Inserir Tipo 2*: antes da inserção, verifica-se se $v_k \neq v_j$, $v_k \neq v_{j+1}$, $v_l \neq v_i$ e $v_l \neq v_{i+1}$. Caso verdadeiro, ocorre a remoção das arestas (v_i, v_{i+1}) , (v_{l-1}, v_l) , (v_j, v_{j+1}) e (v_{k-1}, v_k) , seguido da inserção das arestas (v_i, v) , (v, v_j) , (v_l, v_{j+1}) , (v_{k-1}, v_{l-1}) e (v_{i+1}, v_k) . Assim, como ocorre na inserção anterior, o sentido dos caminhos $(v_{i+1}, \dots, v_{l-1})$ e (v_l, \dots, v_j) também devem ser invertidos para manter o sentido da rota.

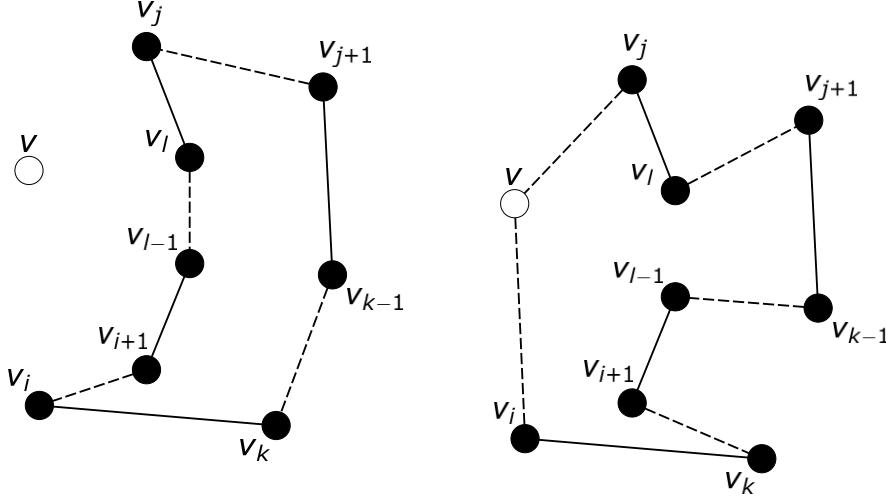


Figura 2.3: GENI - Inserção Tipo 2 do vértice v entre os vértices v_i e v_j

A principal característica dessa fase, segundo GENDREAU *et al.* [43], é que v não necessariamente é inserido entre dois vértices v_i e v_j adjacentes. Além disso, após a inserção, v_i e v_j passam a ser adjacentes a v . No Algoritmo 3, é ilustrado o pseudocódigo da fase GENI, no qual o método recebe uma rota parcial S e um vértice aleatório v a ser inserido.

Na linha 1, é criada a vizinhança $N_p(v)$ de cada vértice v . $N_p(v)$ é constituída dos p vértices da rota que estão mais próximos de v , de acordo com a matriz de distâncias do problema. Das linhas 3 a 16, para cada dois vizinhos de v e para cada sentido da rota (horário e anti-horário), os dois tipos de inserção são testados. Por fim, a melhor solução é retornada na linha 17.

Algoritmo 3: GENI(S, v)

```

1:  $N_p \leftarrow \text{DefinirVerticesProximos}(V);$ 
2:  $f^* \leftarrow \infty; S^* \leftarrow S;$ 
3: Para  $v_i, v_j \in N_p(v), v_i \neq v_j$  faça
4:   Para  $\text{sentido} \in \{\text{horario,anti-horario}\}$  faça
5:      $S' \leftarrow \text{inserir\_tipo1}(v_i, v_j, S, v, \text{sentido});$ 
6:     Se  $f(S') < f^*$  então
7:        $S^* \leftarrow S';$ 
8:        $f^* \leftarrow f(S^*);$ 
9:     Fim-se
10:     $S'' \leftarrow \text{inserir\_tipo2}(v_i, v_j, S, v, \text{sentido});$ 
11:    Se  $f(S'') < f^*$  então
12:       $S^* \leftarrow S'';$ 
13:       $f^* \leftarrow f(S^*);$ 
14:    Fim-se
15:  Fim-para
16: Fim-para
17: Retorne  $S^*$ 

```

Após a primeira fase, é iniciada a fase US, aplicada no intuito de aprimorar a rota construída. Nessa fase, iterativamente, cada um dos vértices de S passa por uma tentativa de remoção e reinserção em S , por meio dos métodos de inserção da fase GENI. Analogamente à fase de construção do GENIUS, a fase US possui dois tipos de remoção para um vértice v_i . Esses dois tipos de remoção são detalhados a seguir e ilustrados nas Figuras 2.4 e 2.5:

- *US - Remover Tipo 1*: seja $v_j \in N_p(v_{i+1})$, e seja $v_k \in N_p(v_{i-1})$ um vértice no caminho $(v_{i+1}, \dots, v_{j-1})$ para um dado sentido. Ao se remover o vértice v_i , as arestas (v_{i-1}, v_i) , (v_i, v_{i+1}) , (v_k, v_{k+1}) e (v_j, v_{j+1}) são removidas. Em seguida, a rota é reconectada utilizando as seguintes arestas, (v_{i-1}, v_k) , (v_{i+1}, v_j) e (v_{k+1}, v_{j+1}) . Além disso, o sentido dos caminhos (v_{i+1}, \dots, v_k) e (v_{k+1}, \dots, v_j) deve ser invertido.

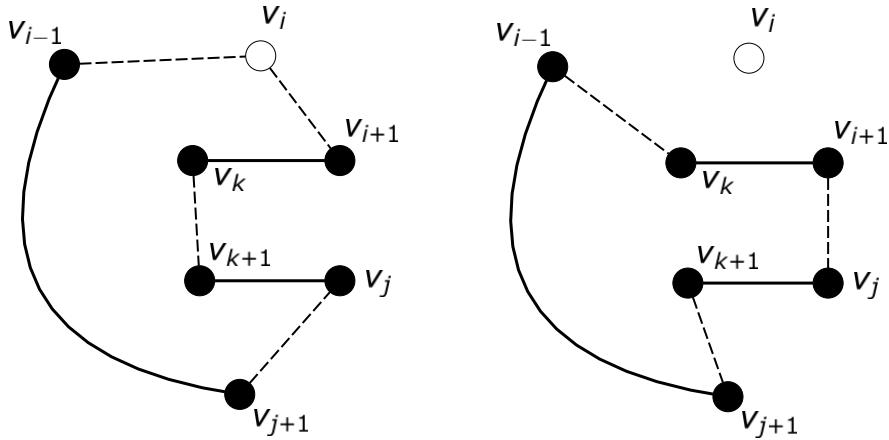


Figura 2.4: US - Remoção Tipo 1 do vértice v_i .

- *US - Remover Tipo 2*: dada uma orientação da rota, considere $v_j \in N_p(v_{i+1})$ e v_k um vértice do caminho $(v_{j+1}, \dots, v_{i-2})$, no qual $v_k \in N_p(v_{i-1})$, e $v_l \in N_p(v_{k+1})$ pertence ao caminho (v_j, \dots, v_{k-1}) . Ao se remover o vértice v_i da rota, remove-se também as arestas (v_{i-1}, v_i) , (v_i, v_{i+1}) , (v_{j-1}, v_j) , (v_l, v_{l+1}) e (v_k, v_{k+1}) , e em seguida, a conectividade da rota é restabelecida incluindo-se as arestas (v_{i-1}, v_k) , (v_{l+1}, v_{j-1}) , (v_{i+1}, v_j) e (v_l, v_{k+1}) . Por fim, o sentido dos caminhos $(v_{i+1}, \dots, v_{j-1})$ e (v_{l+1}, \dots, v_k) são invertidos para garantir a orientação da rota.

O Algoritmo 4 apresenta com mais detalhes como é realizada a fase US. Das linhas 3 a 16, para cada vértice em S e sentido da rota, os dois tipos de remoção supracitados são testados. Para cada tipo de remoção, a exclusão de um vértice v_i é seguida de dois tipos de inserção da fase GENI, para a reinserção de v_i na rota. Finalmente, na linha 17, a rota de menor custo encontrada pela fase US é retornada.

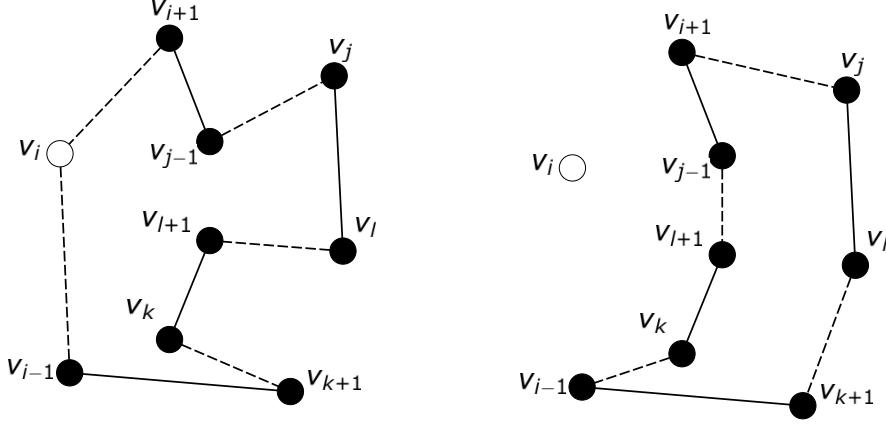


Figura 2.5: US - Remoção Tipo 2 do vértice v_i .

Algoritmo 4: $\text{US}(S)$

```

1:  $S^* \leftarrow S$ 
2:  $f^* \leftarrow f(S^*)$ 
3: Para  $v_i \in S$  faça
4:   Para  $\text{sentido} \in \{\text{horario, anti-horario}\}$  faça
5:      $S' \leftarrow \text{remover\_tipo1}(v_i, S, \text{sentido})$ 
6:     Se  $(f(S') < f^*)$  então
7:        $S^* \leftarrow S'$ 
8:        $f^* \leftarrow f(S^*)$ 
9:     Fim-se
10:     $S'' \leftarrow \text{remover\_tipo2}(v_i, S, \text{sentido})$ 
11:    Se  $(f(S'') < f^*)$  então
12:       $S^* \leftarrow S''$ 
13:       $f^* \leftarrow f(S^*)$ 
14:    Fim-se
15:  Fim-pará
16: Fim-pará
17: Retorne  $S^*$ 

```

2.3.3 VND

Finalizada a fase de construção do GRASP modificado, a melhor solução atual é repassada para a fase busca local, guiada pelo método de descida em vizinhança variável (VND). Proposto por Mladenović [73], VND é uma metaheurística de busca local que utiliza diferentes estruturas de vizinhanças. Dada uma lista ordenada de vizinhanças, VND inicia explorando a primeira vizinhança de S , $N^{(x)}S$, e se uma melhor solução não for encontrada, a próxima vizinhança $N^{(x+1)}S$ é explorada. Caso

contrário, retorna-se para a primeira vizinhança da lista. O algoritmo para quando todas as estruturas de vizinhança forem exploradas, retornando a melhor solução encontrada.

A efetividade desse método está relacionada principalmente às estruturas de vizinhanças utilizadas, ou seja, à qualidade dos movimentos realizados no espaço de busca. A seguir, são apresentadas as estruturas de vizinhanças utilizadas por DA SILVA [26]:

1. *shift*: troca um vértice de posição dentro da rota;
2. *swap*: consiste na inversão de posição de um vértice i com outro vértice j dentro da rota;
3. *or-opt*: movimento similar ao *shift*, porém é feita a troca de posição de n vértices consecutivos dentro da rota;
4. *2-opt*: remove duas arestas não adjacentes da solução e insere duas novas para manter o ciclo único;
5. *3-opt*: remove três arestas não adjacentes e insere três novas rotas, semelhante ao *2-opt*;
6. *remover_simples_re_inserir_mais_barata*: remove um vértice e o reinsere via inserção mais barata [87];
7. *remover_simples_re_inserir_genius*: remove um vértice e o reinsere via um dos métodos de inserção de GENIUS;
8. *remover_genius_re_inserir_mais_barata*: remove um vértice usando um dos métodos de remoção de GENIUS e faz a inserção usando critério de menor custo entre vértices adjacentes;
9. *remover_genius_re_genius*: remove e reinsere um vértice de S usando as técnicas de GENIUS;
10. *remover_mais_barata_re_inserir_genius*: remove usando critério de inserção mais barata e reinsere via um dos procedimentos de inserção do GENIUS.
11. *dupla_remocao_simples_inserir_mais_barata*: substitui dois vértices pertencentes a R , por um único que não pertença à solução;
12. *remover_simples_inserir_mais_barata*: substituir um vértice pertencente a R da rota, inserindo via inserção mais barata;
13. *remover_simples_inserir_genius*: substituir um vértice pertencente a R da rota, inserindo via GENIUS;
14. *remover_genius_inserir_genius*: substituir um vértice pertencente a R da rota, removendo e inserindo via GENIUS; e
15. *remover_genius_inserir_mais_barata*: retirar um vértice pertencente a R da solução via GENIUS, e inserir um novo via inserção mais barata.

No Algoritmo 5, é apresentado com mais detalhes o pseudocódigo do VND. Nas linhas 1 e 2 é definido o número de vizinhanças a serem exploradas. Das linhas 3 a 11, iterativamente, seleciona-se o melhor vizinho S' em cada vizinhança de S (linha 4); é verificado se S' é melhor que S (linha 5); se verdadeiro, S' passa a ser a melhor

solução atual do VND (linha 6) e o algoritmo retorna à primeira vizinhança (linha 7); caso contrário, o algoritmo avança para a próxima vizinhança (linha 9). Após explorar todas as vizinhanças, sem encontrar uma solução melhor que S , retorna-se S como solução final do VND.

Algoritmo 5: VND($S, N^{(t)}$).

```

1:  $k \leftarrow 1$ ;
2:  $k\_max \leftarrow t$ ;
3: Enquanto  $k \leq k\_max$  faça
4:    $S' \leftarrow MelhorVizinho(N^{(k)}(S))$ ;
5:   Se  $f(S') < f(S)$  então
6:      $S \leftarrow S'$ ;
7:      $k \leftarrow 1$ ;
8:   Senão
9:      $k \leftarrow k + 1$ ;
10:  Fim-se
11: Fim-enquanto
12: Retorne  $S$ 
```

2.3.4 Heurística estado-da-arte

Como mencionado anteriormente, O GRASP-GENIUS-RVND [26] é um método baseado na metaheurística GRASP [31], no qual utiliza a heurística GENIUS [42] para a construção de uma solução inicial, e o método RVND (*Random Variable Neighborhood Descent*) [61] para exploração das vizinhanças dessa solução. O RVND é uma variação do VND na qual possui uma ordem aleatória de utilização das vizinhanças.

Para resolver o PRRCP, as heurísticas GENIUS e RVND foram adaptadas. No GRASP-GENIUS-RVND, a lista restrita de candidatos (LRC) é inicialmente restrita ao subconjunto dos vértices obrigatórios T , e então os métodos de inserção GENIUS são aplicados a partir desta lista. Em seguida, uma nova LRC é gerada, restrita aos vértices opcionais pertencentes a R que cobrem pelo menos um vértice em W , e, novamente, as inserções GENIUS são executadas. Por fim, para atender a restrição de *PRIZE*, as mesmas inserções são aplicadas aos vértices restantes.

Em relação à busca local do GRASP, especificamente na implementação das estruturas de vizinhança, os movimentos foram restringidos de maneira a não interferir na viabilidade da rota obtida. Outra adaptação foi a utilização da técnica *first improvement* [62], na qual o RVND retorna à primeira vizinhança assim que encontra uma primeira rota de custo menor que a atual. Além disso, com o intuito de evitar que a técnica *first improvement* tornasse a busca local em um procedimento determinístico, a cada chamada, as estruturas de vizinhança são re-organizadas de

maneira aleatória.

No Algoritmo 6, é ilustrado como o GRASP faz uso das heurísticas GENIUS e RVND em seu procedimento. Na heurística proposta em DA SILVA [26], os critérios de parada desse GRASP são os seguintes: ou o encontro de um valor de solução alvo, ou um tempo máximo de execução pré-estabelecidos.

Algoritmo 6: GRASP-GENIUS-RVND($tempo_max$, $alvo$, $seed$, α)

```

1:  $S^* \leftarrow \emptyset$ ;  $f^* \leftarrow \infty$ ;  $tempo\_execucao \leftarrow 0$ ;
2: Enquanto ( $tempo\_execucao < tempo\_max$ ) faz
3:    $S \leftarrow GENIUS(seed, \alpha)$ ;
4:    $S' \leftarrow RVND(seed)$ ;
5:   Se ( $f(S') < f^*$ ) então
6:      $S^* \leftarrow S'$ ;
7:      $f^* \leftarrow f(S^*)$ ;
8:   Se ( $f^* \leq alvo$ ) então
9:     Retorne  $S^*$ ;
10:    Fim-se
11:  Fim-se
12:   $tempo\_execucao \leftarrow obter\_tempo\_execucao()$ ;
13: Fim-enquanto
14: Retorne  $S^*$ ;
```

Nas linhas 3 e 4, a cada iteração, os procedimentos de construção da solução e busca local são executados. A solução inicial S é construída por meio do método GENIUS, adaptado para o PRRCP e, em seguida, a busca local é realizada pelo método RVND. Ao final da busca local, o RVND retorna uma solução S' que é comparada à melhor solução atual (linha 5). Se S' possuir custo menor, a melhor solução é atualizada. Na linha 8, é verificado se foi alcançada uma solução com custo menor ou igual ao custo da solução alvo. Se verdadeiro, o método é encerrado e a melhor solução é retornada. Na linha 12, o tempo de execução é atualizado, e, por fim, na linha 14, a melhor solução é retornada.

No próximo capítulo, serão apresentadas novas abordagens para a resolução do PRRCP, tais como um novo conjunto de regras de redução, uma nova formulação matemática, e novas heurísticas híbridas.

Capítulo 3

Novas abordagens para o PRRCP

Neste capítulo, são apresentados os métodos propostos para a solução do PRRCP. Inicialmente apresenta-se o modelo matemático, seguido das desigualdades válidas para o fortalecimento da sua relaxação linear. Em seguida, sete novas regras de redução são descritas; e, por fim, a última seção mostra toda a abordagem heurística deste trabalho. São exibidas quatro heurísticas híbridas que combinam a heurística estado-da-arte com PLI e mineração de dados.

3.1 Formulação proposta

A nova formulação proposta nesta tese é um modelo não direcionado que utiliza restrições do tipo *cutset*, em vez de variáveis de fluxo [71] ou multifluxo [93], para eliminação de subciclos na rota. As variáveis desta nova formulação são as seguintes:

- x_e : variável binária que recebe valor igual a um, caso a aresta $e \in E$ esteja na rota, e valor igual a zero, caso contrário.
- y_i : variável binária para todo $i \in R$. É atribuído valor um a y_i , se o vértice i estiver na rota e zero, caso contrário.

Como nas formulações anteriores [71, 93], c_e representa o custo de utilizar a aresta e , p_i o prêmio associado a cada vértice $i \in V$, e *PRIZE*, o prêmio mínimo a ser coletado pela rota. Para facilitar o entendimento e a escrita do modelo proposto, considere $\delta(S) = \{(i, j) \in E \mid i \in S, j \in V \setminus S\}$, $\delta(i)$ como sendo todas as arestas incidentes ao vértice i , e um vértice raiz $u \in T$. Além disto, lembre que $R_w = \{i \in V \mid c_{iw} \leq D\}$ para $w \in W$, e $R_W = \bigcup_{w \in W} R_w$.

Para a formulação proposta, consideramos o *PRIZE* como sendo o prêmio mínimo a ser coletado menos os prêmios dos vértices em T , e W como sendo o conjunto de vértices a serem cobertos menos os vértices cobertos por algum vértice

de T . Desta forma, o novo modelo de programação inteira pode ser definido da seguinte maneira:

$$\min \sum_{e \in E} c_e x_e \quad (3.1)$$

$$s.a : \sum_{e \in E} x_e = |T| + \sum_{i \in R} y_i \quad (3.2)$$

$$\sum_{i \in R} p_i y_i \geq PRIZE \quad (3.3)$$

$$\sum_{i \in R_w} y_i \geq 1, \quad \forall w \in W \quad (3.4)$$

$$\sum_{e \in \delta(i)} x_e = 2, \quad \forall i \in T \quad (3.5)$$

$$\sum_{e \in \delta(i)} x_e = 2y_i, \quad \forall i \in R \quad (3.6)$$

$$\sum_{e \in \delta(S)} x_e \geq 2, \quad \forall S \subseteq V \setminus \{u\}, S \cap T \neq \emptyset \quad (3.7)$$

$$\sum_{e \in \delta(S)} x_e \geq 2, \quad \forall S \subseteq R, \forall w \in W, R_w \subseteq S \quad (3.8)$$

$$\sum_{e \in \delta(S)} x_e \geq 2y_t, \quad \forall S \subseteq R, t \in S \quad (3.9)$$

$$x_e \leq y_v, \quad \forall e = (i, j) \in E, v \in \{i, j\}, v \in R \quad (3.10)$$

$$x_e \in \{0, 1\}, \quad \forall e \in E \quad (3.11)$$

$$y_i \in \{0, 1\}, \quad \forall i \in R \quad (3.12)$$

Nesta formulação, a função objetivo (3.1) minimiza o custo da rota. As restrições (3.2) asseguram que o número de vértices na rota, será igual ao número de vértices obrigatórios somado ao número de vértices escolhidos de R . A restrição (3.3) garante a coleta do prêmio mínimo. As restrições (3.4) demandam que todos os vértices em W devem ser cobertos por pelo menos um vértice em V . Dado que a solução do PRRCP possui a forma de um ciclo simples, as restrições (3.5) restringem para dois o grau de todo vértice obrigatório. Análoga a estas últimas restrições, as restrições (3.6) limitam para dois, o grau de qualquer $i \in R$ que esteja na rota.

O conjunto de restrições do tipo *cutset*, (3.7), (3.8) e (3.9), garantem a conectividade da rota, e unido às restrições (3.2), (3.5) e (3.6) são responsáveis pela eliminação de subciclos nessa rota. Os procedimentos de separação destas *cutsets* são descritos na Seção 3.2.

As restrições (3.10) associam as variáveis de cada aresta na rota aos seus respectivos vértices, isto é, se uma aresta estiver na rota, seus vértices extremos farão parte

dela. Por fim, as restrições (3.11) e (3.12) asseguram a integralidade das variáveis de decisão.

3.2 Desigualdades válidas

A relaxação linear da formulação proposta nesta tese para o PRRCP, pode ser aprimorada pela introdução de desigualdades válidas. Nesta seção, propõem-se desigualdades obtidas a partir dos politopos associados aos problemas da mochila [8] e caixeiro viajante [12], e desigualdades do tipo *cutset* para conjuntos de vértices que não contém uma solução para o problema.

3.2.1 Desigualdades de cobertura

Desigualdades de cobertura podem ser aplicadas a problemas cujas formulações possuam restrições associadas ao politopo do problema da mochila [8]. Se uma restrição assume a forma de uma restrição de mochila, isto é, uma soma de variáveis binárias com coeficientes não negativos menores ou iguais a um valor não negativo no lado direito, então existe uma cobertura minimal associada a esta restrição [97].

A partir das restrições (3.3), referentes à coleta de prêmio mínimo, pode-se definir o politopo associado ao problema da mochila como $\text{conv}\{y \in \{0, 1\}^R : y \text{ satisfaz } (3.3)\}$ [12]. Assim, um corte de cobertura para o PRRCP pode ser definido como segue:

$$\sum_{i \in C} y_i \geq 1 \quad , \forall C \subset R, \sum_{i \in R \setminus C} p_i < PRIZE. \quad (3.13)$$

Para toda cobertura encontrada no corte supracitado, garante-se que pelo menos um vértice da cobertura esteja na rota, ou seja, não há como definir todas as variáveis de C com valor igual a zero, e simultaneamente atender o *PRIZE*. Para que a desigualdade (3.13) seja válida, a cobertura C deve ser minimal, isto é, $C \setminus \{j\}$ não é uma cobertura para qualquer $j \in C$.

Analogamente ao proposto por BALAS [8] para o problema da mochila, os cortes apresentados nesta seção podem ser fortalecidos por uma extensão de C . Seja o conjunto $\bar{C} = \{i \in R \setminus C : p_i \geq \max_{j \in C} p_j\}$, o corte estendido pode ser definido como segue:

$$\sum_{i \in C} y_i + \sum_{i \in \bar{C}} y_i \geq 1 + |\bar{C}| \quad , \forall C \subset R, \sum_{i \in R \setminus C} p_i < PRIZE. \quad (3.14)$$

3.2.2 Desigualdades *2-matching*

Na literatura, há vários estudos sobre desigualdades válidas para o problema do Caixeiro Viajante (PCV) [67, 74]. Uma das desigualdades mais conhecidas, são as desigualdades *comb* de Grötschel & Padberg [53, 54]. Tais desigualdades podem ser definidas como: sejam $\tau \geq 3$ um número inteiro ímpar, $H \subset V$ e $\mathcal{T}_j \subset V$, para $j = 1, \dots, \tau$, tal que $\mathcal{T}_j \cap H \neq \emptyset$ e $\mathcal{T}_j \setminus H \neq \emptyset$, para $j = 1, \dots, \tau$, e \mathcal{T}_j conjuntos disjuntos em vértices (ver Figura 3.1). A desigualdade *comb* é descrita como:

$$\sum_{e \in E(H)} x_e + \sum_{j=0}^{\tau} \sum_{e \in E(\mathcal{T}_j)} x_e \leq |H| + \sum_{j=0}^{\tau} |\mathcal{T}_j| - \left\lceil \frac{3\tau}{2} \right\rceil \quad (3.15)$$

O conjunto H é conhecido como *handle* e \mathcal{T}_j como *teeth* da *comb*.

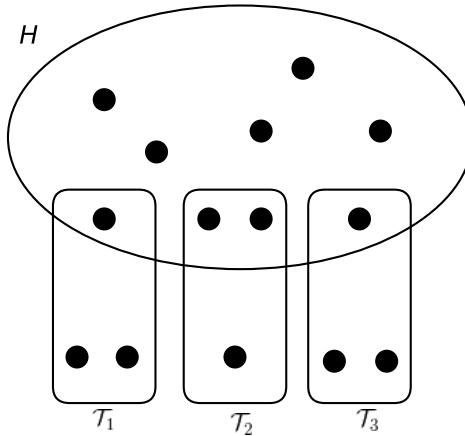


Figura 3.1: *Comb* com três *teeth*.

As desigualdades *comb* induzem facetas para o PCV(n) $n \geq 6$ [63, 65]. Para o caso especial em que $|\mathcal{T}_j \cap H| = 1$, CHVÁTAL [20] provou que as desigualdades *comb* permanecem válidas. Se, além disso, $|\mathcal{T}_j \setminus H| = 1$ para todo $j = 1, \dots, \tau$, então tem-se as clássicas desigualdades *2-matching* propostas por EDMONDS [29] (ver Figura 3.2). Nesta tese, serão utilizadas as desigualdades *2-matching*:

$$\sum_{e \in E(H)} x_e + \sum_{e \in E(\mathcal{T})} x_e \leq |H| + \left\lfloor \frac{\tau}{2} \right\rfloor \quad (3.16)$$

Uma vez que o PRRCP possui vértices obrigatórios, nos quais suas respectivas variáveis podem ser consideradas como constantes, as desigualdades *2-matching* foram adaptadas em (3.17):

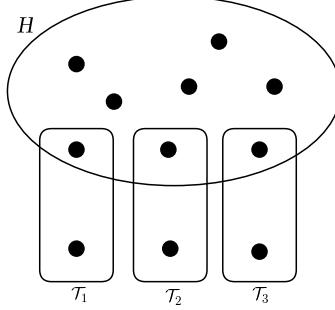


Figura 3.2: *2-matching* com três *teeth*.

$$\sum_{e \in E(H)} x_e + \sum_{e \in E(T)} x_e \leq |H \cap T| + \sum_{i \in H \setminus T} y_i + \left\lfloor \frac{\tau}{2} \right\rfloor. \quad (3.17)$$

3.2.3 Desigualdades *cutsets*

Os cortes supracitados nesta seção são baseados em desigualdades para o *Undirected prize collecting traveling salesman problem* [12]. Neste trabalho, é apresentada uma nova desigualdade válida para o PRRCP, que é um fortalecimento para as restrições (3.9), em casos específicos. Dado um conjunto de arestas, e seu respectivo conjunto de vértices S , tal que $\{u\} \notin S$, este novo corte é aplicado quando não há possibilidade do conjunto $V \setminus S$ conter a solução ótima. Por exemplo, se a soma dos prêmios associados aos vértices contidos no subconjunto $V \setminus S$ for menor que o *PRIZE*, ou se o custo do ciclo contido no subgrafo induzido por $V \setminus S$, for maior que o custo de uma solução já conhecida.

No cenário em que não houver possibilidade do conjunto $V \setminus S$ conter a solução ótima, então é necessário que haja pelo menos duas arestas incidentes em S . Desta forma, as novas desigualdades *cutset* podem ser descritas como:

$$\sum_{e \in \delta(S)} x_e \geq 2 \quad , \forall S \subseteq R, S^* \not\subset V \setminus S, \text{ e } S^* \not\subset S \quad (3.18)$$

no qual S^* é uma solução ótima para o PRRCP.

3.3 Algoritmos de separação

Em muitos problemas que envolvem rotas como o PCV e suas variantes, uma solução viável deve ser um grafo conectado. Em formulações de programação inteira para tais problemas, restrições *cutsets* são utilizadas para garantir a conectividade

das soluções. Existem diferentes tipos de *cutsets*, e para alguns desses tipos, o número de *cutsets* é exponencial no tamanho da instância do problema.

Quando formulações utilizando restrições classificadas como *cutsets* são resolvidas por algoritmos *branch-and-cuts*, *cutsets* violadas devem ser identificadas dinamicamente ao longo da execução do algoritmo. Esse processo de identificação é geralmente referenciado na literatura como “separação” [97]. A separação de *cutsets* violadas é usualmente executada por meio de algoritmos de corte mínimo/fluxo máximo, tanto para problemas em grafos não-direcionados [66] ou direcionados [35].

3.3.1 Restrições *cutset*

O modelo matemático proposto nesta tese contém um número exponencial de restrições (3.7), (3.8) e (3.9). Contudo, pode-se separá-las e checar se há ou não uma restrição violada, em tempo polinomial, fazendo com que um algoritmo de plano de corte seja uma abordagem interessante. Nesta seção, são descritos os procedimentos de separação utilizados para identificar as *cutsets* violadas. Para os algoritmos a seguir, considere $G^* = (V^*, E^*)$ como sendo o grafo associado à solução $S = (x^*, y^*)$ obtida por meio de uma relaxação linear, no qual $V^* = \{i \in V : y_i^* > 0\}$ e $E^* = \{e \in E : x_e^* > 0\}$.

Para as restrições (3.7), considere x_e^* como o valor da capacidade de uma aresta $e \in E^*$. Para todo $i \in T \setminus \{u\}$ um corte mínimo (u, i) pode ser encontrado em G^* por meio de um algoritmo de corte mínimo/fluxo máximo, e se a saída da aplicação deste procedimento for menor que dois, então tem-se uma restrição violada. Neste caso, o próximo passo é adicionar esta restrição ao modelo e resolvê-lo novamente. Como algoritmo de fluxo-máximo, nesta tese foi utilizado o algoritmo *preflow push* de CIUPALA e CIUREA [21], que é executado em $O(|V|^2 \sqrt{|E|})$ para cada vértice. Portanto, o processo de separação das restrições (3.7) possui uma complexidade geral de $O(|T||V|^2 \sqrt{|E|})$.

As separações das restrições (3.8) e (3.9) foram realizadas de maneira semelhante à separação supracitada, porém com diferentes subconjuntos de vértices que participam do algoritmo de fluxo máximo. Nas restrições (3.8), para todo $w \in W$ foi criado um vértice especial \bar{v} , e para todo $i \in R_w$ foi criada uma aresta (i, \bar{v}) com capacidade $c_{i,\bar{v}} = 2$. A ideia de inserir \bar{v} é encontrar um corte de capacidade mínima em G^* , entre todo par (u, \bar{v}) . Desta forma, este algoritmo de separação possui complexidade de $O(|W||V|^2 \sqrt{|E|})$.

Por fim, nas restrições (3.9), para todo $i \in R^* : y_i > \epsilon$ encontra-se um corte de capacidade mínima (u, i) em G^* , e se a saída for menor que $2y_i$, tem-se uma restrição violada. Desta forma, este algoritmo de separação possui complexidade de $O(|R||V|^2 \sqrt{|E|})$.

3.3.2 Desigualdades de cobertura

Dado que o problema de separação das desigualdades de cobertura (DC) é NP-Difícil [32, 55], várias separações exatas e heurísticas têm sido propostas. CROWDER *et al.* [24] propuseram uma nova forma de escrita para essas desigualdades, $\sum_{j \in C} (1 - y_j) \geq 1$, na qual permite que o problema de separação possa ser formulado como um problema da mochila 0-1. Considere $y^* \in [0, 1]^n$ como sendo a solução fracionária a ser separada. Seja N o conjunto de itens do problema. Para todo item j , a variável z_j assume o valor igual a um, se e somente se j estiver na cobertura. Assim, o problema de separação pode ser definido como segue:

$$\min \sum_{j \in N} (1 - y_j^*) z_j \quad (3.19)$$

$$s.a : \sum_{j \in N} a_j z_j \geq b + 1 \quad (3.20)$$

$$z_j \in \{0, 1\}, \quad \forall j \in N \quad (3.21)$$

Para a resolução desse problema de separação, CROWDER *et al.* [24] propuseram uma heurística gulosa para construção de C , inserindo os itens na cobertura em ordem não-decrescente de acordo com a função gulosa $g(j) = (1 - y_j)/a_j$. O objetivo dessa heurística é colocar itens de melhor custo/benefício em C .

Para a separação das DC desta tese (3.13), será utilizado o mesmo procedimento de CROWDER *et al.* [24] para construção de uma cobertura, com apenas a modificação da função gulosa para $g(j) = -y_j/p_j$, no qual p_j é o prêmio associado ao vértice j . Assim que uma cobertura minimal for obtida, o processo de extensão será realizado. Este processo consta da formação de uma extensão \bar{C} , contendo todos os vértices de $R \setminus C$ com prêmios iguais ou maiores que $\max\{p_j : j \in C\}$.

3.3.3 Desigualdades 2-matching

Estas desigualdades, quando aplicadas ao PRRCP, podem ser separadas em tempo polinomial, por meio de uma simples modificação do algoritmo exato de separação proposto por PADBERG e RAO [77]. Contudo, a fim de reduzir o esforço computacional demandado por esse algoritmo exato, nesta tese, foi utilizada a heurística proposta por FISCHETTI *et al.* [36].

Considere x_e^* como pesos associados às arestas. O método de FISCHETTI *et al.* [36] consiste na aplicação do algoritmo guloso de Kruskal [51] para encontrar uma árvore geradora de custo mínimo em G^* . A cada iteração na qual esse algoritmo seleciona uma nova aresta e , é determinada (no subgrafo de G^* induzido por todas as arestas selecionadas até o momento) a componente conexa que contém e , chamada

de *handle* (H). Para cada H , as arestas *teeth* são determinadas de maneira ótima pelo seguinte procedimento. Seja $\delta(H) = \{e_1, \dots, e_\tau\}$ tal que $x_{e1}^* \geq x_{e2}^* \geq \dots \geq x_{e\tau}^*$. Inicialmente, é relaxada a restrição de que os *teeth* devem ser disjuntos em vértices e, para qualquer $|\mathcal{T}| \geq 3$ e ímpar, a melhor escolha para \mathcal{T} consiste das arestas e_1, \dots, e_τ . Portanto, uma desigualdade *2-matching* violada consiste na escolha de um número ímpar de *teeths*, $|\mathcal{T}| \geq 3$, no qual maximize:

$$x_{e1}^* + (x_{e2}^* + x_{e3}^* - 1) + \dots + (x_{e\tau-1}^* + x_{e\tau}^* - 1) \quad (3.22)$$

Se nenhum corte violado puder ser produzido dessa maneira, então tem-se que nenhuma restrição *2-matching* violada existe para o dado H . Caso contrário, tem-se uma restrição *2-matching* violada e, caso duas arestas *teeth*, e e f , se sobreponem em um nó v , então, simplifica-se a desigualdade definindo um novo par *handle-teeth* (H', \mathcal{T}') com $\mathcal{T}' \leftarrow \mathcal{T} \setminus \{e, f\}$ e, $H' \leftarrow H \setminus \{v\}$ (se $v \in H$) ou $H' \leftarrow H \cup \{v\}$ (se $v \notin H$). De acordo com FISCHETTI *et al.* [36], esse passo de simplificação sempre resulta em uma restrição *2-matching* violada.

Em alguns casos, esse procedimento pode até levar a uma restrição *2-matching* com $|\mathcal{T}| = 1$; se isso ocorrer, rejeitamos a desigualdade em favor de uma *cutset* associada ao *handle*.

3.4 Novas regras de redução para o PRRCP

Nesta seção são propostas novas regras de redução para o PRRCP. Assim como as regras presentes na literatura, estas novas regras reduzem o espaço de busca, sem comprometer a obtenção de soluções ótimas. Nestas novas regras, vértices dos subconjuntos R e W são removidos do grafo, e vértices de R são movidos para o subconjunto T . Além disto, há uma regra que facilita a redução nas regras seguintes, e uma regra que reduz o prêmio mínimo a ser coletado. A seguir, são apresentadas as novas regras de redução:

- **R8** - Se $\Delta, r \in R \setminus R_W$ e $(PRIZE \leq 0 \text{ ou } p_r = 0)$, então $R = R \setminus \{r\}$ e $\forall(i, r), (j, r) \in \delta(r)$ $c_{ij} = \min(c_{ij}, (c_{ir} + c_{rj}))$, caso $(i, j) \notin E$, adiciona-se (i, j) em E com custo $c_{ir} + c_{rj}$;
- **R9** - Generalização de **R3** e **R4** - Se Δ , e $\exists r \in R$ tal que $p_r = 0$ ou $PRIZE \leq 0$, $\exists v \in R$ onde $R_r \subseteq R_v$ e $\forall(r, i), (r, j) \in \delta(r)$ tal que $i, j \in V \setminus \{v\}$ e $i \neq j$ ($c_{ir} + c_{rj} \geq c_{iv} + c_{vj}$) e ($c_{ij} \leq c_{ir} + c_{rj}$), então $R = R \setminus \{r\}$, desde que o grafo resultante dessa remoção seja completo em relação a T ;
- **R10** - Se $\exists r \in R$ tal que em $G' = (V', E')$, onde $V = V \setminus \{r\}$ e $E = E \setminus \delta(r)$, não exista dois caminhos disjuntos em vértices entre dois vértices terminais, um terminal

e nenhum vértice de R_w , $w \in W$, ou entre nenhum par de vértice i, j onde $i \in R_w$ e $j \in R_{\bar{w}}$, $w \neq \bar{w}$ e $w, \bar{w} \in W$, então $T = T \cup \{r\}$;

- **R11** - Se $\exists r_1, r_2 \in R$ tal que $G' = (V', E')$, no qual $V' = V \setminus \{r_1, r_2\}$ e $E' = E \setminus (\delta(r_1) \cup \delta(r_2))$, é desconexo; Define-se cada componente conexa de G' por $G'_i = (V'_i, E'_i)$, $i \in \{1, \dots, nc\}$. Se $\nexists i \in \{1, \dots, nc\}$ tal que $T \subseteq V'_i$, $V'_i \cap R_w \neq \emptyset$ $\forall w \in W$, e $\sum_{i \in V'_i} p_i \geq PRIZE$, então $T = T \cup \{r_1, r_2\}$.
- **R12** - Se $\exists i \in T$, onde $p_i > 0$ então $PRIZE = PRIZE - p_i$ e $p_i = 0$.
- **R13** - Se $\exists i \in V$ tal que $|\delta(i)| \leq 1$, então $V = V \setminus \{i\}$;
- **R14** - Se $\exists w, w' \in W$, $w \neq w'$, tal que $R_{w'} \subseteq R_w$, então $W = W \setminus \{w\}$.

Na regra **R8**, é removido um vértice que não contribui para cobertura de W e para a coleta do prêmio mínimo. Portanto, se um vértice $r \in R$ não cobrir nenhum vértice $w \in W$ e se o $PRIZE$ já for atingido pela coleta dos prêmios de T , r é removido de R . Contudo, se a desigualdade triangular não for satisfeita, um caminho entre dois vértices i e j pode ter custo menor utilizando r como vértice intermediário. Neste caso, o custo da aresta (i, j) recebe o valor mínimo entre c_{ij} e $c_{ir} + c_{rj}$. Caso a aresta (i, j) não exista, uma nova aresta (i, j) é criada e recebe um custo de valor $c_{ir} + c_{rj}$.

As Figuras 3.3 e 3.4 ilustram a aplicação da regra **R8** em uma instância do PRRCP, no qual existe um vértice $r \in R$ que possui prêmio com valor igual a zero, e não cobre nenhum vértice $w \in W$ (como mostrado na Figura 3.3). Nas figuras desta seção, considere os valores nas linhas pontilhadas como sendo o custo das respectivas arestas, e os valores na região superior do vértice, como sendo seu prêmio.

Após a aplicação da regra **R8**, o vértice r é removido, juntamente com as arestas (r, i) e (r, j) . Além disto, a aresta (i, j) passa a possuir um novo custo, com valor igual a sete (como mostrado na Figura 3.4).

A regra **R8** pode aumentar o número de arestas do grafo, ao se criar uma aresta entre dois vértices adjacentes ao vértice removido. Entretanto, ela diminui o número de vértices e pode facilitar a redução realizada por outras regras. Portanto, a aplicação desta regra foi estabelecida logo após a aplicação das regras existentes na literatura, e no início das novas regras propostas. A regra **R9** é uma generalização das regras **R3** e **R4**. Remove-se um vértice $r \in R$, caso as seguintes condições sejam satisfeitas: i) o prêmio de r ser igual a zero ou o $PRIZE$ igual a zero; ii) exista um $v \in R$ tal que todo $w \in W$ coberto por r , também seja coberto v ; iii) o custo de i para j , passando por um vértice intermediário, seja menor utilizando v em vez de r ; iv) o custo direto de i para j , seja menor que utilizando r como vértice intermediário, ou se existir um vértice v' com as mesmas características de v . Se não houver aresta

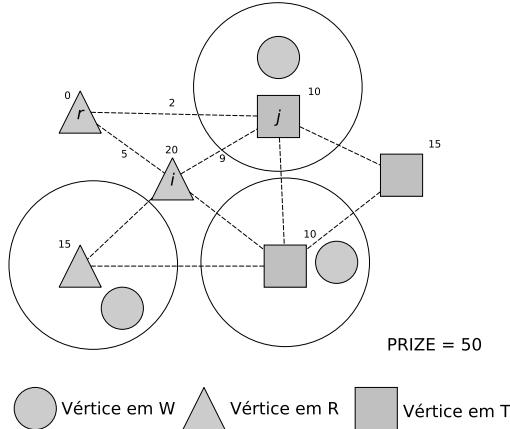


Figura 3.3: Instância associada ao PRRCP, antes da aplicação da regra **R8**.

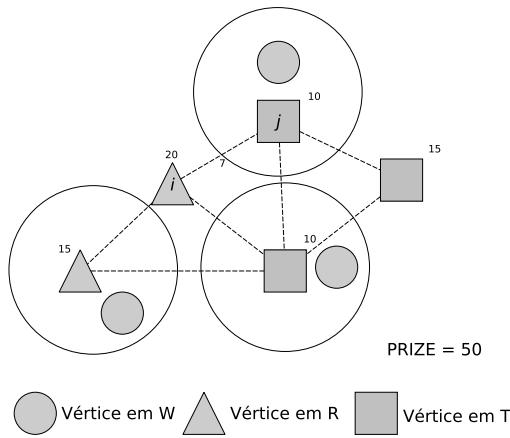


Figura 3.4: Instância da Figura 3.3, após a aplicação da regra **R8**.

(i, j) , então o valor infinito é atribuído ao custo dessa aresta para o teste desta última condição.

A Figura 3.5 ilustra a aplicação da regra **R9** em uma instância do PRRCP, no qual os círculos representam a área de cobertura de um vértice $w \in W$, e as linhas pontilhadas representam as arestas entre os vértices, com seus respectivos custos.

As regras **R10** e **R11** movem um vértice $r \in R$ para o subconjunto T , implicando que r faça parte da rota. Considere G' uma cópia do grafo original do problema, com exceção do vértice r e das arestas incidentes a r . Na regra **R10**, move-se r para T se não houver dois caminhos disjuntos em vértices para algum dos seguintes casos: i) entre dois vértices pertencentes a T ; ii) entre um vértice em T e nenhum vértice que cubra $w \in W$; iii) entre nenhum par de vértices em V que cubram diferentes vértices em W .

A Figura 3.6 mostra a aplicação da regra **R10**. Nesta ilustração, a única rota possível que possui dois caminhos disjuntos em vértices, entre i e j , inclui o vértice r . Portanto, r é movido para o subconjunto T e sua presença torna-se obrigatória na

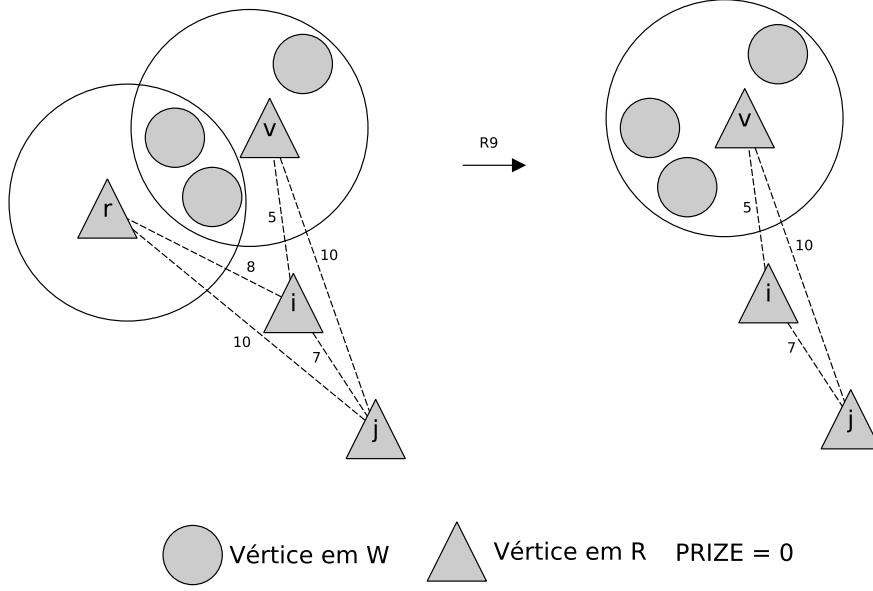


Figura 3.5: Instância associada ao PRRCP, antes e depois da aplicação da regra **R9**.

rota.

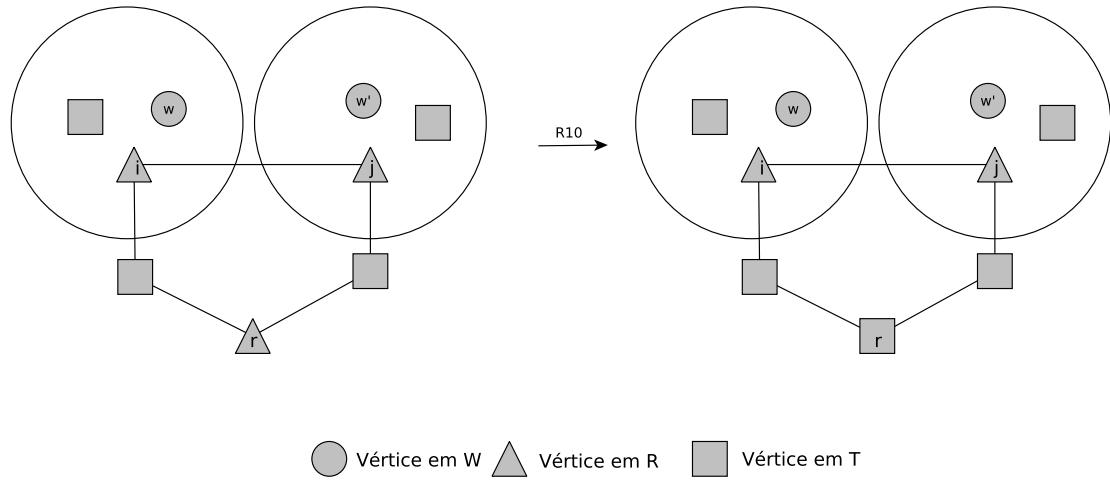


Figura 3.6: Instância associada ao PRRCP, antes e depois da aplicação da regra **R10**

A regra **R11** busca por pares de vértices em R que sejam essenciais para uma solução viável do PRRCP. Assim, dois vértices r_1 e r_2 são movidos para o subconjunto T , se as seguintes condições forem satisfeitas: i) a remoção de r_1 e r_2 do grafo original $G = (V, E)$, juntamente com suas respectivas arestas incidentes, resulta em um grafo desconexo $G' = (V', E')$; e ii) não existe uma única componente conexa de $G' = (V', E')$, que contenha todos os vértices obrigatórios, cubra todos os vértices de W e contenha o *PRIZE*.

As Figuras 3.7, 3.8 e 3.9 ilustram a aplicação da regra supracitada, em uma instância do PRRCP. Inicialmente, dois vértices $r_1, r_2 \in R$ e suas arestas incidentes

são removidos, transformando o grafo original em duas componentes conexas (como mostrado nas Figuras 3.7 e 3.8). Como nenhuma destas componentes atende as condições da regra, no grafo original definem-se os vértices r_1 e r_2 como obrigatórios (como mostrado Figura 3.9).

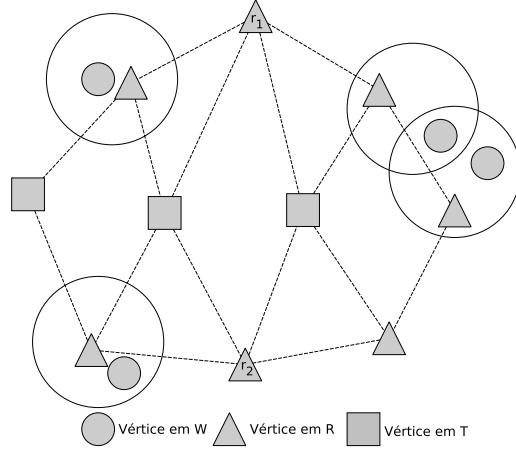


Figura 3.7: Instância associada ao PRRCP, antes da aplicação da regra **R11**.

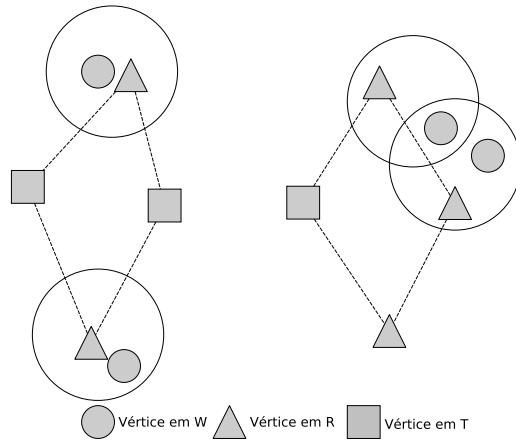


Figura 3.8: Instância da Figura 3.7 , após a remoção de r_1 , r_2 , $\delta(r_1)$ e $\delta(r_2)$.

Após a aplicação da regra **R11** o subconjunto T torna-se estável, e nenhum vértice de R é movido para T . Portanto, aplica-se a regra **R12** após a aplicação de **R11**. Na regra **R12**, o prêmio mínimo é reduzido. Para todo $i \in T$, se p_i for maior que zero, subtrai-se p_i de $PRIZE$ e atribui-se valor igual a zero para p_i (ver Figuras 3.10 e 3.11).

A regra **R13** remove um vértice de V , isto é, um vértice que pertence ou a R , ou T . Dado que a solução do PRRCP é um ciclo, se um vértice $i \in V$ possui grau igual a um, i é removido de V (como mostrado na Figura 3.12). Se o vértice i pertencer ao subconjunto T , tem-se que o problema é inviável.

Por fim, a última regra remove um vértice $w \in W$, que já é coberto pela cobertura

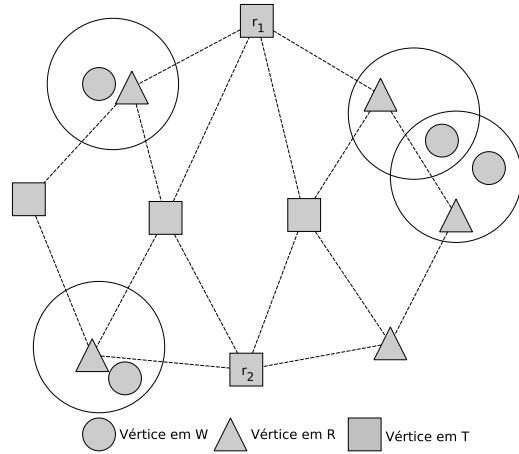


Figura 3.9: Instância da Figura 3.7, após mover r_1 e r_2 para o subconjunto T .

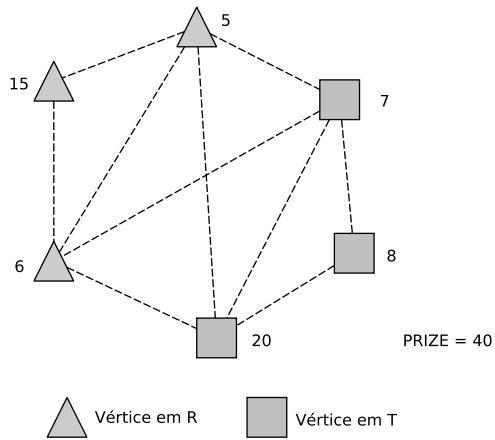


Figura 3.10: Instância do PRRCP com $PRIZE$ de valor igual 40.

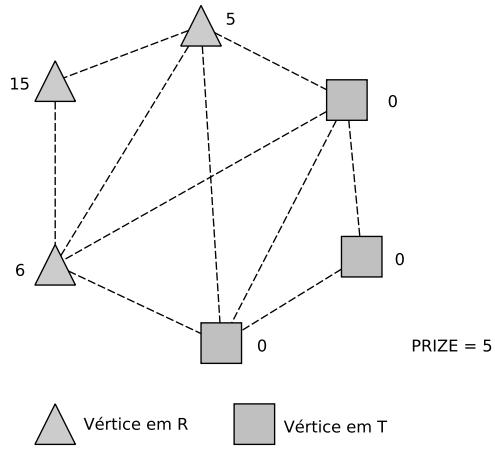


Figura 3.11: Instância da Figura 3.10, após aplicação da regra **R12**.

de outro $w' \in W$. Na regra **R14**, se a cobertura de w' está contida na cobertura de w , então w será automaticamente coberto quando w' for, portanto, remove-se w de

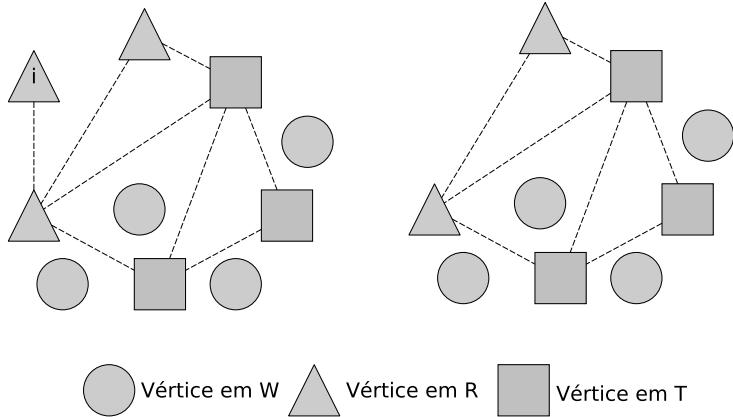


Figura 3.12: Instância associada ao PRRCP, antes e depois da aplicação da regra **R13**.

W (como mostrado nas Figuras 3.13 e 3.14).

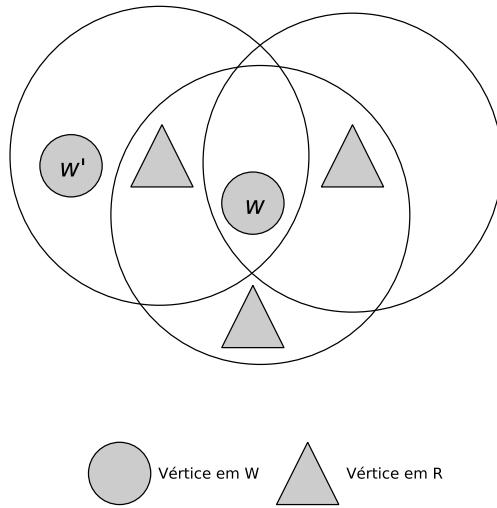


Figura 3.13: Instância associada ao PRRCP, antes da aplicação da regra **R14**.

3.5 Heurísticas

Como será mostrado no Capítulo 4, apesar da nova formulação proposta permitir a resolução inédita de várias instâncias da literatura, ainda há aquelas que não são possíveis de serem resolvidas de maneira exata, seja pela necessidade de um tempo elevado de execução ou por problemas relacionados à falta de memória do computador. Portanto, neste trabalho também são propostas quatro novas heurísticas híbridas: duas hibridizações com PLI (MPO_S e LB) e duas hibridizações com mineração de dados (DM-GRASP e MDM-GRASP).

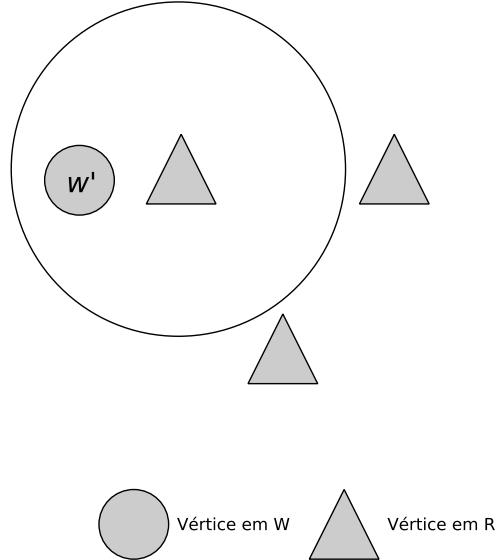


Figura 3.14: Instância da Figura 3.13, após a aplicação da regra **R14**.

3.5.1 Heurísticas híbridas

Metaheurísticas representam uma importante classe de técnicas para lidar com problemas combinatórios difíceis. Contudo, a concentração em uma única metaheurística é bastante restritiva para o avanço do estado-da-arte ao abordar problemas práticos de otimização [14]. Nas últimas décadas, têm surgido vários relatos de aplicações bem-sucedidas da combinação de componentes metaheurísticos clássicos e outras técnicas, como programação linear inteira (PLI) e mineração de dados, para fornecer heurísticas híbridas robustas [94].

Algumas heurísticas clássicas baseadas em PLI são: a) *Local branching* (LB) [38], que é usada como uma maneira de melhorar uma dada solução viável. Essa técnica pode ser comparada com uma busca local, mas as soluções vizinhas são obtidas pela introdução de desigualdades lineares no modelo de PLI; b) *Relax and Fix* [97], que encontra uma solução viável a partir da fixação de variáveis, guiada pela solução de programação linear (PL) do problema original. Sua ideia geral é, iterativamente, relaxar um grupo de variáveis inteiras, resolver o modelo relaxado, fixar um subconjunto de variáveis, e então resolver o submodelo; c) *Feasibility pump* [37], que busca por uma solução inteira viável por meio da minimização da fracionaldade de variáveis inteiras, a partir da solução ótima de um PL.

Apesar de resolvedores matemáticos terem dificuldade em lidar com problemas reais, eles possuem desempenho satisfatório na resolução de grandes problemas de PL ou problemas de PLI com poucas variáveis. Uma abordagem amplamente utilizada é realizar uma fixação de variáveis antes de resolver um modelo de PLI [3, 64]. Geralmente o problema é decomposto em subproblemas menores, e então

uma heurística é encarregada de resolver os subproblemas mais difíceis, enquanto que o resolvedor lida com os mais fáceis. Por exemplo, para resolver o *multilevel capacitated lot-sizing problem*, ALMEDER [3] fez uso de uma estratégia baseada em colônia de formigas para decidir para qual período de produção um item deveria ser escalonado. Em seguida, baseado nessas decisões, as quantidades de produções são determinadas pela resolução do problema de PL restante, a partir de um resolvedor de PLI.

A ideia de hibridização proposta nesta tese difere da estratégia híbrida supracitada, pois no PRRCP não foi possível identificar um subproblema fácil o suficiente para ser solucionado pelo resolvedor. Em vez disto, adotou-se a mesma estratégia de CLÍMACO *et al.* [23]. Essa estratégia consiste em realizar uma fixação de variáveis que envolva o problema como um todo, resultando em um único problema restrito. Portanto, se o resolvedor é chamado após esse procedimento de fixação, ele está, de fato, resolvendo um problema reduzido e não uma parte mais fácil dele. Nesse caso, é importante ressaltar que não há garantia que a solução ótima do problema original seja alcançada.

De acordo com BLUM *et al.* [14], um ponto crítico em métodos de fixação de variáveis é a escolha das variáveis a serem fixadas em cada passo. Decisões precipitadas de fixação podem levar o algoritmo a um espaço de soluções que não seja interessante. Dessa forma, uma boa direção é desejável antes de qualquer fixação de variáveis.

Assim como em CLÍMACO *et al.* [23], a proposta de hibridização deste trabalho tem como base a hipótese de que, uma vez que soluções de boa qualidade ao longo das iterações de uma heurística eficiente compartilham das mesmas componentes, encontrar características comuns é uma potencial boa direção para a fixação de variáveis. Dentre essas características, podem ser encontrados elementos frequentes, padrões, ou até mesmo elementos de impacto negativos, e cada um desses pode ser utilizado para reduzir o problema original.

3.5.2 Análise do conjunto E^0

A proposta híbrida de CLÍMACO *et al.* [23] partiu de uma observação acerca do espaço de solução explorado por uma heurística eficiente estado-da-arte para o *2-path network design problem* (2-PNDP). Nos experimentos em CLÍMACO *et al.* [23], foi observado por seus autores que grande parte das arestas do espaço de busca, pelo menos 70%, nunca eram consideradas como parte de uma solução ótima local. Assim, CLÍMACO *et al.* [23] denominaram esse conjunto de arestas como E^0 , e decidiram remover as suas respectivas variáveis do modelo de PLI.

A estratégia de hibridização de CLÍMACO *et al.* [23] (MPO) funciona da seguinte

maneira: primeiramente, uma heurística base é executada, e então o conjunto E^0 é construído; em seguida, são selecionadas as arestas mais promissoras da solução final obtida pela heurística (S_{fix}); são removidas do modelo todas as variáveis que representam as arestas de E^0 ; fixam-se no modelo as respectivas variáveis de cada aresta em S_{fix} ; e por fim, é aplicado o resolvedor matemático no modelo modificado.

Neste trabalho, foi reproduzido o mesmo experimento de CLÍMACO *et al.* [23] com a heurística GRASP-GENIUS-RVND, e percebeu-se que, em média, o tamanho do conjunto E^0 é em torno de 88% do tamanho E . O resultado deste experimento foi a motivação para a construção de abordagens híbridas semelhantes à estratégia de CLÍMACO *et al.* [23].

3.5.3 MPO com E^0 estável

Em análises preliminares, também percebeu-se que a partir de certas iterações, o conjunto E^0 não estava sendo alterado. A partir desta observação, foi determinado que o processo heurístico poderia ser interrompido assim que o conjunto E^0 se tornasse estável - não fosse alterado após um certo número de iterações. Então, foi criada a hipótese de que esta interrupção poderia economizar tempo de processamento, e a qualidade do conjunto E^0 construído seria semelhante à qualidade deste mesmo conjunto, caso o número máximo de iterações fosse executado.

CLÍMACO *et al.* [23] fixaram em seu modelo parte da solução da heurística base, no intuito de reduzir ainda mais o espaço de busca, sob a hipótese de que a heurística e o modelo de PLI estavam tomando decisões similares, uma vez que suas soluções compartilhavam várias arestas em comum. Como, em experimentos preliminares, observou-se que o tamanho de E^0 já poderia fornecer uma aceleração satisfatória ao modelo proposto neste trabalho, e dado que quanto maior a redução do espaço de busca, menores são as chances de se encontrar soluções de boa qualidade, esta última fixação de CLÍMACO *et al.* [23] não foi adotada.

O pseudocódigo da abordagem híbrida proposta (denotada como MPO_S) é apresentado no Algoritmo 7. Na linha 1, são inicializados a solução final do método, um contador auxiliar e o conjunto E^0 como sendo o próprio E . Na linha 2, a constante *stable* representa o número de iterações sem modificações que o E^0 precisa ter, para ser considerado como estável; esta constante é inicializada como γ vezes o número máximo de iterações do processo heurístico (*maxIt*), no qual $\gamma \in \mathbb{R}$ tal que $0 \leq \gamma \leq 1$. Das linhas 9 a 19, a heurística GRASP-GENIUS-RVND é descrita com algumas instruções extras, para a construção do conjunto E^0 e para a sua interrupção caso E^0 se torne estável. Se E^0 não ficar estável em nenhum momento, a heurística será executada por *maxIt* iterações.

Na linha 4, o procedimento GENIUS gera uma solução viável e então, na linha 5,

a vizinhança desta solução é explorada pela heurística RVND. Das linhas 6 a 8, se S'' for melhor que a melhor solução S , então a melhor solução é atualizada. Na linha 9 a função `update()` remove de E^0 toda aresta $e \in S''$. Como retorno, é atribuído valor *false* à variável *flag*, se nenhuma aresta foi removida de E^0 , e *true*, caso contrário. Entre as linhas 10 e 13, um contador é incrementado, se na iteração atual o conjunto E^0 não foi alterado. Das linhas 15 a 17, o processo heurístico é interrompido se o conjunto E^0 tornar-se estável.

Uma vez que a heurística é interrompida ou atinge o número máximo de iterações, o modelo de PLI é modificado na linha 19. Por fim, na linha 20 o modelo reduzido é resolvido, e a solução final é retornada na linha 21.

Algoritmo 7: MPO_S($maxIt, seed, \gamma, model$)

```

1:  $S \leftarrow \emptyset; f^* \leftarrow \infty; E^0 \leftarrow E; count \leftarrow 0; flag \leftarrow false;$ 
2:  $stable \leftarrow \gamma \times maxIt;$ 
3: Para  $k = 1$  até  $k \leq maxIt$  faça
4:    $S' \leftarrow \text{GENIUS}(seed);$  {Fase de construção}
5:    $S'' \leftarrow \text{RVND}(S', seed);$  {Fase de busca local}
6:   Se  $f(S'') \leq f^*$  então
7:      $S \leftarrow S''; f^* \leftarrow f(s);$ 
8:   Fim-se
9:    $flag \leftarrow \text{update}(S'', E^0);$ 
10:  Se  $flag \neq true$  então
11:     $count \leftarrow count + 1;$ 
12:  Senão
13:     $count \leftarrow 0;$ 
14:  Fim-se
15:  Se  $count = stable$  então
16:    Vá para a linha 19; {aqui o processo heurístico é interrompido}
17:  Fim-se
18: Fim-pará
19: remove( $E^0, model$ );
20:  $S \leftarrow \text{solve}(model);$  {solver matemático resolve o modelo reduzido}
21: Retorne  $S;$ 

```

3.5.4 Local branching

Ao perceber que para alguns casos o MPO_S proposto não alcançava uma solução ótima, foi investigado qual a diferença entre as soluções encontradas pelo MPO_S (S) e as soluções ótimas atingidas pelo *solver* (S^*). Observou-se que a aresta mais custosa da solução ótima ($e^+ \in S^*$), nunca pertencia a nenhuma solução do MPO_S. Então foi criada a hipótese de que os vértices (i, j) de e^+ devam ser visitados para que a solução ótima seja alcançada.

Em todos os casos investigados, o MPO_S não utiliza e^+ para visitar i e j , mas

sim duas outras arestas (i, k) e (k, j) , que individualmente são mais baratas que e^+ , mas que somadas possuem um custo maior que o da aresta e^+ . Esta decisão do MPO_S pode ser entendida como um comportamento guloso e iterativo do GENIUS. Verificou-se também que para as instâncias testadas, a diferença de custo entre S e S^* é exatamente a diferença de custo entre (i, j) e $[(i, k) + (k, j)]$, o que fortalece esta hipótese.

Assim, elaborou-se a seguinte estratégia. Em S , para cada par de arestas incidentes em um mesmo vértice (i, k) e (k, j) , se (i, j) estiver em E^0 e possuir custo menor que $[(i, k) + (k, j)]$, então (i, j) será movida de E^0 para E_l , um novo conjunto contendo as arestas liberadas de E^0 . De acordo com esta investigação sobre a vizinhança de S , que pode conter a solução ótima, decidiu-se aplicar a técnica de *local branching* (LB). Após a chamada da heurística base, da liberação de arestas do conjunto E^0 , e da redução do modelo com a remoção das arestas em E^0 , foi aplicado o LB. Esta abordagem consiste em adicionar desigualdades que restringem o espaço de busca do *solver* à vizinhança $\delta - opt$ de uma dada solução.

Neste trabalho, o LB foi aplicado a partir de S (definida neste contexto como \bar{x}), e adicionou-se a seguinte restrição ao modelo:

$$\sum_{e \in E \setminus E^0 | \bar{x}_e = 1} (1 - x_e) + \sum_{e \in E_l | \bar{x}_e = 0} x_e \leq \delta, \quad (3.23)$$

onde δ é um valor inteiro positivo fornecido como parâmetro de entrada, indicando o número de variáveis $\bar{x}_e \in E_l$ que são permitidas mudar de um para zero, e vice-versa, isto é, o resolvedor pode trocar até δ arestas da solução por arestas do conjunto E_l . O pseudocódigo do *local branching* proposto neste trabalho é apresentado no Algoritmo 8. Na linha 1, os mesmos passos do MPO_S são realizados até o momento da remoção do conjunto E^0 do modelo (até a linha 19 de MPO_S). A partir desta linha, no *local branching*, a restrição (3.23) é adicionada ao modelo de PLI. Por fim, o modelo modificado é resolvido (linha 3) e a sua solução é retornada na linha 4.

Algoritmo 8: Local_branching($maxIt, seed, \gamma, model, \delta$)

- 1: (Linhas 1 a 19 do Algoritmo 7);
 - 2: Adicionar a desigualdade (3.23) ao *model*;
 - 3: $S \leftarrow \text{solve}(model, \delta)$;
 - 4: **Retorne** S ;
-

3.5.5 Mineração de dados

Mineração de dados (MD) é um processo de descoberta de conhecimento a partir de bancos de dados, que pode ser descrito em termos de regras ou padrões [58].

Minerar tais conhecimentos requer técnicas específicas para transformar registros de um banco de dados em informações úteis. Uma aplicação comum de técnicas de MD pode ser observada em um banco de dados de um supermercado, por exemplo, no qual essas técnicas podem ser usadas para extrair conjuntos de itens frequentes, representando a preferência de clientes, de tal forma a auxiliar na criação de alguma estratégia de negócios para melhorar as vendas.

Em um banco de dados composto por transações, onde cada transação corresponde a um conjunto de elementos do domínio da aplicação, diferentes relacionamentos entre dados podem ser extraídos em forma de conjuntos de itens frequentes (padrões). Um conjunto de itens, que é um subconjunto de itens do domínio da aplicação, é extraído se o seu suporte, um indicador percentual de sua ocorrência no conjunto de dados, for maior ou igual a um determinado suporte mínimo. A esse conjunto chamamos de conjunto de itens frequentes.

Dessa forma, o problema de mineração de conjuntos de itens frequentes consiste em identificar todos os conjuntos de itens frequentes na base em relação a um determinado suporte mínimo. Além disso, conjuntos de itens frequentes maximais podem ser também minerados, onde um conjunto de itens frequentes maximal corresponde a um conjunto de itens frequentes que não é subconjunto de nenhum outro conjunto de itens frequentes.

Estabelecidos os conceitos de MD supracitados, considere $E = \{e_1, e_2, e_3, \dots, e_n\}$ como sendo um conjunto de elementos da aplicação. Uma transação é um subconjunto de E e um banco de dados D é um conjunto de transações. Um conjunto de itens frequentes F , com suporte sup , é um subconjunto de E que ocorre em pelo menos $(sup \times 100)\%$ das transações em D . Nesse sentido, a identificação de todos os conjuntos de itens frequentes em D , com um suporte mínimo (sup_{min}) definido como um parâmetro de entrada, é o problema da mineração de conjuntos frequentes (MCF), no qual vários algoritmos eficientes foram propostos na literatura para resolvê-lo, incluindo-se o Apriori [1] e o FP-Growth [59]. Há também o algoritmo FP-max* [48], que foi proposto para minerar, de maneira eficiente, conjuntos de itens frequentes maximais [52].

Hibridização de GRASP com mineração de dados

O GRASP, em sua forma original, não possui qualquer tipo de memória de longo prazo. Esse fato deve-se à sua natureza iterativa e multipartida, com iterações totalmente independentes que, a cada partida, inicia um novo processo de busca aleatório [4].

Na literatura, há propostas de estratégias que mantêm informações de iterações anteriores, para auxiliar o processo de busca em iterações futuras [40, 69, 83, 84, 89, 90]. Em relação à metaheurística GRASP, uma combinação com mineração de

dados foi proposta por RIBEIRO *et al.* [89] para a resolução do *Set Packing Problem*. Essa abordagem obteve sucesso em alcançar resultados promissores não apenas em termos de qualidade de solução, mas também em tempo de execução. A ideia básica de incorporar mineração de dados consiste na extração de padrões encontrados em soluções sub-ótimas, para então eficientemente guiar a busca por soluções melhores.

A estratégia adotada por RIBEIRO *et al.* [89] foi dividir o GRASP original em duas partes. A primeira parte é responsável pela obtenção de um conjunto de soluções de boa qualidade (conjunto elite). Em seguida, o processo de mineração de dados é aplicado nesse conjunto elite, no intuito de extrair um conjunto de padrões. A segunda parte utiliza, em cada iteração, um padrão do conjunto de padrões extraídos para gerar uma solução inicial, por meio de um método construtivo adaptado. Uma vez que essa estratégia obteve resultados satisfatórios na resolução do problema abordado em RIBEIRO *et al.* [89] e, em outros da literatura [10, 56, 57, 91], decidiu-se adotá-la como uma proposta para a resolução do PRRCP.

Inicialmente, o GRASP com mineração foi aplicado com sucesso a problemas de otimização que representam soluções como conjuntos de elementos. A mineração de conjuntos de itens para tal categoria de problemas foi direta, uma vez que conjuntos de itens são conjuntos de elementos. Contudo, essa técnica não pode ser diretamente empregada, quando a ordem dos elementos é relevante na solução.

Em GUERINE *et al.* [57], foi proposta uma estratégia baseada no GRASP com MD para uma variante do problema do caixeiro viajante (PCV), para lidar com o problema da ordem dos elementos da seguinte forma: em vez da mineração intuitiva de vértices, que são os elementos da solução, essa técnica considerou a mineração de arcos, que naturalmente mantém a ordem dos vértices. Basicamente, cada par de vértices (i, j) , isto é, um arco de uma solução, é mapeado para um identificador único que o representa (a_{ij}) . Esse mapeamento é realizado para todas as soluções do conjunto elite, ou seja, transformando as soluções em conjuntos de elementos, que são facilmente tratados por uma técnica MCF.

Em GUERINE *et al.* [57], o mapeamento das soluções foi realizado da seguinte forma. Dado um par de vértices sequenciais, representado pela aresta (i, j) , o identificador único que representa essa aresta é calculado por $a_{ij} = (i \times |V|) + j$, onde $|V|$ é o número de vértices do problema. Uma vez que o padrão é extraído, um processo inverso ao de mapeamento é realizado da seguinte maneira. A partir do identificador a_{ij} , temos que $i = (a_{ij} \text{ div } |V|)$ e $j = (a_{ij} \text{ mod } |V|)$, onde div e mod são os operadores matemáticos utilizado para se obter, respectivamente, o quociente e o resto da divisão.

Como as soluções do PRRCP são também influenciadas pela ordem dos vértices, a estratégia de mapeamento de GUERINE *et al.* [57] foi adotada neste trabalho. Na próxima seção, é apresentada mais uma heurística híbrida proposta nesta tese,

que combina o GRASP-GENIUS-RVND de DA SILVA [26] com mineração de dados (DM-GRASP).

DM-GRASP

O DM-GRASP, proposto neste trabalho, é baseado na estratégia de GUERINE *et al.* [57], sendo dividido em duas fases, com metade do número total de iterações cada, separadas pelo processo de mineração de dados. Na primeira fase, que é idêntica à heurística original GRASP-GENIUS-RVND, as melhores soluções encontradas são armazenadas em um conjunto elite M de tamanho cap . Uma solução S é elegível para ser inserida em M se, ou S é melhor que a pior solução de M , ou se M não estiver cheio. Ademais, M não permite soluções idênticas.

Após a primeira fase, o processo de mineração de dados é executado e gera um conjunto de padrões com suporte maior ou igual a um determinado suporte mínimo sup_{min} . Em seguida, a segunda fase utiliza, a cada iteração, um padrão do conjunto de padrões para construir uma solução inicial por meio de um procedimento construtivo adaptado. No final, a melhor solução encontrada é retornada. O pseudocódigo do DM-GRASP para o PRRCP é apresentado no Algoritmo 9.

Na linha 1 são inicializados todos os conjuntos e as variáveis do algoritmo. Na primeira fase da estratégia (linhas 2 a 10), que corresponde à primeira metade das iterações, o algoritmo permanece igual ao GRASP-GENIUS-RVND original, exceto pela presença da função `AtualizarConjuntoElite(M, S'', cap)` (linha 8). Esta função insere S'' , que é uma solução obtida pelo RVND, em M se S'' é única em M e o seu custo é menor que o da pior solução em M . Após o fim das k primeiras iterações, na linha 11 o procedimento de mineração extrai t padrões do conjunto elite, e tais padrões são inseridos em ordem decrescente de tamanho (quantidade de arcos) no conjunto de padrões. Neste trabalho, a extração de padrões é realizada pelo algoritmo FP-max* [48], que é uma implementação eficiente para a extração de conjuntos de itens frequentes maximais.

Na segunda fase do algoritmo (linhas 12 a 20), que executa a segunda metade das iterações, um procedimento construtivo adaptado substitui o procedimento construtivo original (linha 14). Este procedimento adaptado, que é explicado posteriormente, é executado após selecionar-se o próximo maior padrão por meio de uma seleção *round-robin* [41] a partir de `conjunto_padroes`, e os identificadores contidos nos padrões são mapeados de volta em forma de arcos, como explicado anteriormente. Em seguida, as linhas restantes da segunda parte são iguais às suas respectivas partes do GRASP-GENIUS-RVND original. Ao final de todas as iterações, a melhor solução S é retornada (linha 21).

Para a geração de soluções utilizando padrões, utilizou-se uma versão adaptada da heurística GENIUS proposta por DA SILVA [26], na qual cada construção do

Algoritmo 9: DM-GRASP($maxIt, seed, t, cap, \alpha, sup_{min}$)

```
1:  $S \leftarrow \emptyset; f(S) \leftarrow \infty; M \leftarrow \emptyset; k \leftarrow 1;$ 
2: Enquanto  $k \leq maxIt/2$  faça
3:    $S' \leftarrow \text{GENIUS}(seed, \alpha); \{\text{Fase de construção}\}$ 
4:    $S'' \leftarrow \text{RVND}(S', seed); \{\text{Fase de busca local}\}$ 
5:   Se  $f(S'') \leq f(S)$  então
6:      $S \leftarrow S'';$ 
7:   Fim-se
8:   AtualizarConjuntoElite( $M, S'', cap$ );
9:    $k \leftarrow k + 1;$ 
10:  Fim-enquanto
11:   $\text{conjunto\_padroes} \leftarrow \text{minerar}(M, cap, t, sup_{min})$ 
12:  Enquanto  $k \leq maxIt$  faça
13:     $padrao \leftarrow \text{selecionarProximoMaiorPadrao}(\text{conjunto\_padroes});$ 
14:     $S' \leftarrow \text{GENIUS\_adaptado}(seed, \alpha, padrao);$ 
15:     $S'' \leftarrow \text{RVND}(S', seed);$ 
16:    Se  $f(S'') \leq f(S)$  então
17:       $S \leftarrow S'';$ 
18:    Fim-se
19:     $k \leftarrow k + 1;$ 
20:  Fim-enquanto
21:  Retorne  $S;$ 
```

GENIUS não mais se inicia a partir de uma sub-rota com três vértices aleatórios, e sim a partir de um padrão minerado. O Algoritmo 10 apresenta com mais detalhes esta construção.

Na linha 1, a solução S é inicializada contendo todos os arcos do padrão. Na linha 2, a lista de candidatos é inicializada com todos os vértices do problema que ainda não estão na rota. Em seguida, enquanto S não estiver completa, um vértice é escolhido aleatoriamente de uma lista restrita de candidatos, e inserido na rota utilizando a heurística GENI (linhas 4 a 10). Por fim, a heurística de refinamento US tenta melhorar S (linha 11), que é retornada na linha 12.

Algoritmo 10: GENIUS _ adaptado($seed, \alpha, padrao$)

```
1:  $S \leftarrow padrao;$ 
2:  $LC \leftarrow V \setminus S$ 
3: Avaliar( $LC$ );
4: Enquanto  $\text{!SolucaoCompleta}(S)$  faça
5:    $LRC \leftarrow \text{ConstroiLRC}(\alpha, LC);$ 
6:    $v \leftarrow \text{EscolhaAleatoria}(LRC);$ 
7:    $\text{GENI}(S, v);$ 
8:    $LC \leftarrow LC \setminus \{v\};$ 
9:   Avaliar( $LC$ );
10:  Fim-enquanto
11:   $\text{US}(S);$ 
12:  Retorne  $S;$ 
```

MDM-GRASP

Na proposta DM-GRASP o procedimento de mineração de dados é executado apenas uma vez, exatamente após metade das iterações. Além desta estratégia, este trabalho também propõe e examina uma outra abordagem, que aplica a mineração mais de uma vez durante a execução do algoritmo, baseada em investigações da literatura [10, 82, 84].

Segundo BARBALHO *et al.* [10], executar a mineração de dados mais de uma vez explorará a evolução gradual do conjunto elite, e permitirá a extração de padrões mais refinados. Neste novo método, a mineração é aplicada assim que M se tornar estável e sempre que este conjunto for alterado e novamente se tornar estável. Este último método é chamado de *Multi Data-Mining GRASP* (MDM-GRASP).

No contexto deste trabalho, M é dito estável se o seu conjunto de soluções permanece inalterado por um determinando número de iterações ω consecutivas, isto é, ao longo destas iterações, não são encontradas soluções de boa qualidade que possam ser inseridas neste conjunto.

O pseudocódigo do MDM-GRASP é apresentado no Algoritmo 11. Das linhas 2 a 24, enquanto um número máximo de interações $maxIt$ não for atingido, as seguintes instruções são executadas. Das linhas 3 a 7, se já foi realizado algum procedimento de mineração de dados, então S' é obtida por uma construção adaptada (linha 7), caso contrário, a construção do GRASP-GENIUS-RVND é aplicada (linha 4). Na linha 9, uma solução S'' é obtida a partir de uma busca local aplicada em S' . Das linhas 10 a 12, se o custo de S'' for menor que o custo da melhor solução corrente S , então S é atualizada. Das linhas 13 a 17, se M for atualizado com a solução S'' , então o contador de iterações sem atualização de M ($Iter_{sa}$) é definido como zero (linha 14), caso contrário, esse contador é incrementado (linha 16). Das linhas 18 a 22, se M estiver estável, isto é, se $Iter_{sa}$ for igual a ω , então o procedimento de mineração é aplicado (linha 19), e esse contador é definido como zero (linha 20). Por fim, a melhor solução é retornada na linha 25.

Algoritmo 11: MDM-GRASP($maxIt, seed, t, cap, \alpha, sup_{min}, \omega$)

```
1:  $S \leftarrow \emptyset; f^* \leftarrow \infty; M \leftarrow \emptyset; k \leftarrow 0; fez\_mineracao \leftarrow false; Iter_{sa} \leftarrow 0;$ 
2: Enquanto  $k < maxIt$  faça
3:   Se  $fez\_mineracao = false$  então
4:      $S' \leftarrow \text{GENIUS}(seed, \alpha);$ 
5:   Senão
6:      $padrao \leftarrow \text{selecionarProximoMaiorPadrao}(conjunto\_padroes);$ 
7:      $S' \leftarrow \text{GENIUS\_adaptado}(seed, \alpha, padrao);$ 
8:   Fim-se
9:    $S'' \leftarrow \text{RVND}(S', seed);$ 
10:  Se  $f(S'') \leq f^*$  então
11:     $S \leftarrow S''; f^* \leftarrow f(S'');$ 
12:  Fim-se
13:  Se  $\text{AtualizarConjuntoElite}(M, S'', cap)$  então
14:     $Iter_{sa} \leftarrow 0;$ 
15:  Senão
16:     $Iter_{sa} \leftarrow Iter_{sa} + 1;$ 
17:  Fim-se
18:  Se  $Iter_{sa} = \omega$  então
19:     $conjunto\_padroes \leftarrow \text{minerar}(M, cap, t, sup_{min});$ 
20:     $Iter_{sa} \leftarrow 0;$ 
21:     $fez\_mineracao \leftarrow true;$ 
22:  Fim-se
23:   $k \leftarrow k + 1;$ 
24: Fim-enquanto
25: Retorne  $S;$ 
```

Capítulo 4

Resultados para o PRRCP

Neste capítulo, são apresentados os experimentos computacionais realizados com as abordagens propostas no Capítulo 3: formulação matemática, desigualdades válidas, regras de redução e heurísticas. Em seguida, estas novas propostas são comparadas com as melhores estratégias da literatura.

Todos os métodos propostos nesta tese foram implementados na linguagem *C++* e compiladas com *g++ 5.4.0*. Os testes foram executados em uma máquina Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz com 16GB de memória RAM. Para a implementação e execução das formulações foi utilizado o *solver* Gurobi 6.5.2 [76], com suas heurísticas, cortes e pré-processamento desativados. Além disto, foi determinado um tempo limite de uma hora de processamento e a capacidade de paralelização do processador não foi utilizada.

4.1 Instâncias utilizadas

As instâncias utilizadas neste trabalho foram propostas por Silva [93], e consistem em 144 problemas-teste adaptados a partir do problema do caixeiro viajante (PCV) disponíveis no repositório TSPLIB [86]. Para evitar que alguma instância do PRRCP se transformasse no problema de recobrimento de rotas (PRR), o *PRIZE* nunca é satisfeito com a coleta apenas dos prêmios associados aos vértices obrigatórios.

Essas instâncias foram nomeadas de acordo com seu nome original na TSPLIB, agregando-se informações sobre os subconjuntos R , T , W e a porcentagem de prêmio a ser coletado, em relação ao prêmio total disponível. Por exemplo, a instância *brazil58_R18_T20_W20_25* possui as seguintes características: 58 vértices no total; 18 vértices opcionais R ; 20 vértices obrigatórios T ; 20 vértices a serem cobertos W ; e 25% do prêmio total da instância deve ser coletado.

4.2 Resultado da aplicação das formulações matemáticas

Esta seção apresenta comparações entre os resultados obtidos pelas duas formulações presentes na literatura, Fluxo [71] e Multi-fluxo [93], e a nova formulação proposta (Não-dir). O objetivo destas comparações é verificar qual abordagem é mais propensa a resolver um número maior de instâncias do PRRCP, com menor esforço computacional.

Na Tabela 4.1, são apresentados os resultados obtidos pela execução destas formulações. A primeira coluna refere-se ao nome das 144 instâncias e a segunda, o valor da melhor solução conhecida na literatura. Esse valor não necessariamente é o ótimo, mas quando o for, é destacado com o símbolo “*”. A partir da terceira coluna, são apresentadas a solução e o tempo despendido por cada formulação. Na apresentação dos resultados da formulação proposta, há uma coluna extra indicando a diferença percentual entre os tempos dependidos entre a formulação multi-fluxo (melhor da literatura) e a não-dir. Esta diferença foi calculada a partir da seguinte fórmula:

$$Dif_T = 100 \times \frac{(T_{multi-fluxo} - T_{nao-dir})}{T_{multi-fluxo}} \quad (4.1)$$

Os campos com “-” indicam que, devido ao tempo excedido de uma hora de processamento, ou a problemas relacionados a falta memória, não foi possível resolver a respectiva instância. Os valores em negrito indicam os melhores resultados em tempo de processamento e melhor solução encontrada.

Tabela 4.1: Resultados obtidos pelas formulações Fluxo, Multi-fluxo e Não-dir para o PRRCP.

Instância	MSC	Fluxo [71]		Multi-Fluxo [93]		Não-dir		Dif_T
		Sol.	T (s)	Sol.	T (s)	Sol.	T (s)	
att48_R9_T10_W29_25	2291*	2291	47.68	2291	0.39	2291	0.04	89.47
att48_R9_T29_W10_50	2824*	3040	3600.00	2824	52.85	2824	0.35	99.34
att48_R16_T16_W16_75	2650*	3012	3600.00	2650	30.59	2650	0.59	98.06
berlin52_R9_T11_W32_25	4861*	4861	08.07	4861	1.56	4861	0.12	92.31
berlin52_R16_T18_W18_50	5186*	5484	3600.00	5186	463.36	5186	2.98	99.36
berlin52_R9_T32_W11_75	6916	6994	3600.00	6916	92.68	6916	0.29	99.69
bier127_R24_T77_W26_25	98453	-	3600.00	-	3600.00	98453	140.19	96.11
bier127_R24_T26_W77_50	74745	125855	3600.00	74745	495.19	74745	8.90	98.20
bier127_R43_T42_W42_75	98311	251062	3600.00	-	3600.00	98311	140.21	96.11
brazil58_R18_T20_W20_25	19417*	21198	3600.00	19417	798.92	19417	801.36	-0.31
brazil58_R11_T35_W12_50	23312*	32029	3600.00	23312	109.56	23312	1.78	98.38
brazil58_R11_T12_W35_75	20466*	20466	3599.85	20466	53.70	20466	0.16	99.69
brg180_R36_T36_W108_25	550	142850	3600.00	-	3600.00	550	4.94	99.86
brg180_R60_T60_W60_50	2350	-	3600.00	-	3600.00	2350	37.61	98.96
brg180_R36_T108_W36_75	2190	-	3600.00	-	3600.00	2190	123.43	96.57
ch130_R44_T43_W43_25	4383	-	3600.00	-	3600.00	4383	27.99	99.22
ch130_R26_T26_W78_50	4139*	-	3600.00	4139	296.02	4139	2.27	99.23

Tabela 4.1: Resultados obtidos pelas formulações Fluxo, Multi-fluxo e Não-dir para o PRRCP.

Instância	MSC	Fluxo [71]		Multi-Fluxo [93]		Não-dir		Dif_T
		Sol.	T (s)	Sol.	T (s)	Sol.	T (s)	
ch130_R26_T78_W26_75	5287	5443	3600.00	-	3600.00	5287	210.71	94.15
ch150_R50_T50_W50_25	4778	-	3600.00	-	3600.00	4778	511.80	85.78
ch150_R30_T90_W30_50	5239	-	3600.00	-	3600.00	5239	1113.60	69.07
ch150_R30_T30_W90_75	4292*	10981	3600.00	4292	513.73	4292	2.55	99.50
d198_R39_T40_W119_25	10916*	14140	3600.00	-	3600.00	11255	3600.00	0.00
d198_R39_T119_W40_50	13598	-	3600.00	-	3600.00	17023	3600.00	0.00
d198_R66_T66_W66_75	12905*	-	3600.00	-	3600.00	13653	3600.00	0.00
eil51_R17_T17_W17_25	273*	273	2666.75	273	94.59	273	1.29	98.64
eil51_R9_T11_W31_50	248*	248	187.56	248	1.46	248	0.12	91.48
eil51_R9_T31_W11_75	348*	484	3600.00	348	909.62	348	0.37	99.96
eil76_R14_T46_W16_25	387*	612	3600.00	387	1135.69	387	0.61	99.95
eil76_R24_T26_W26_50	338*	775	3600.00	338	252.28	338	0.95	99.62
eil76_R14_T16_W46_75	332*	332	1525.17	332	24.46	332	0.44	98.19
eil101_R19_T21_W61_25	356*	433	3600.00	356	285.60	356	0.77	99.73
eil101_R19_T61_W21_50	506*	1547	3600.00	-	3600.00	506	4.40	99.88
eil101_R33_T34_W34_75	462*	1367	3600.00	462	3602.99	462	8.65	99.76
gr48_R16_T16_W16_25	3571*	3571	1607.97	3571	10.92	3571	0.49	95.47
gr48_R9_T10_W29_50	3524*	3524	17.90	3524	01.02	3524	0.05	94.66
gr48_R9_T29_W10_75	4415*	5135	3600.00	4415	0.40	4415	0.18	54.72
gr96_R18_T58_W20_25	42746*	-	3600.00	42746	1670.91	42746	418.02	74.98
gr96_R32_T32_W32_50	35461*	74785	3600.00	35461	993.11	35461	945.19	4.82
gr96_R18_T20_W58_75	30015*	34379	3600.00	30015	467.11	30015	5.14	98.90
gr120_R40_T40_W40_25	4301*	17438	3600.00	4301	1929.79	4301	2.85	99.85
gr120_R24_T72_W24_50	5159*	-	3600.00	-	3600.00	5159	6.55	99.82
gr120_R24_T24_W72_75	4275*	9702	3600.00	4275	139.75	4275	2.27	98.38
gr137_R26_T28_W83_25	39305*	59114	3600.00	60630	3600.00	39305	270.55	92.48
gr137_R45_T46_W46_50	46122*	-	3600.00	-	3600.00	50905	3600.00	0.00
gr137_R26_T83_W28_75	58720*	-	3600.00	-	3600.00	59093	3600.00	0.00
hk48_R9_T29_W10_25	9708*	10557	3600.00	9708	50.47	9708	0.40	99.21
hk48_R16_T16_W16_50	7593*	7593	3381.67	7593	17.11	7593	0.52	96.97
hk48_R9_T10_W29_75	7803*	7803	10.92	7803	0.39	7803	0.04	88.57
kroA100_R20_T20_W60_25	10848*	11980	3600.00	10848	104.59	10848	12.51	88.04
kroA100_R20_T60_W20_50	16544*	-	3600.00	-	3600.00	16544	96.52	97.32
kroA100_R34_T33_W33_75	14606*	47550	3600.00	-	3600.00	14641	3600.00	0.00
kroA150_R50_T50_W50_25	14382	-	3600.00	-	3600.00	16465	3600.00	0.00
kroA150_R30_T30_W90_50	12846	19405	3600.00	19898	3600.00	12919	3600.00	0.00
kroA150_R30_T90_W30_75	21974	-	3600.00	-	3600.00	21976	3600.00	0.00
kroA200_R40_T120_W40_25	23495	-	3600.00	-	3600.00	25550	3600.00	0.00
kroA200_R68_T66_W66_50	17297	-	3600.00	-	3600.00	20143	3600.00	0.00
kroA200_R40_T40_W120_75	15197*	-	3600.00	-	3600.00	15197	2188.88	39.20
kroB100_R20_T60_W20_25	18406*	-	3600.00	-	3600.00	18544	3600.00	0.00
kroB100_R34_T33_W33_50	14160	24846	3600.00	-	3600.00	14360	3600.00	0.00
kroB100_R34_T33_W33_75	14768*	25197	3600.00	-	3600.00	14768	150.70	95.81
kroB150_R30_T30_W90_25	11467	18533	3600.00	-	3600.00	23422	3600.00	0.00
kroB150_R50_T50_W50_50	15388	-	3600.00	-	3600.00	17653	3600.00	0.00
kroB150_R30_T90_W30_75	21489	-	3600.00	-	3600.00	21489	325.52	90.96
kroB200_R40_T120_W40_25	23946	-	3600.00	-	3600.00	28081	3600.00	0.00
kroB200_R40_T40_W120_50	14125	45214	3600.00	-	3600.00	14515	3600.00	0.00
kroB200_R68_T66_W66_75	19458	-	3600.00	-	3600.00	22931	3600.00	0.00
kroC100_R20_T60_W20_25	16785*	45872	3600.00	16785	3369.75	16785	278.65	91.73
kroC100_R34_T33_W33_50	13235*	40281	3600.00	13235	1796.34	13563	3600.00	-100.41
kroC100_R20_T20_W60_75	12582*	15900	3600.00	12582	78.43	12582	3.60	95.41
kroE100_R20_T60_W20_25	17199*	-	3600.00	17199	3091.27	17199	370.25	88.02
kroE100_R20_T20_W60_50	10586*	12260	3600.00	10586	551.71	10586	2.79	99.49
kroE100_R34_T33_W33_75	14622*	27385	3600.00	-	3600.00	14622	690.34	80.82

Tabela 4.1: Resultados obtidos pelas formulações Fluxo, Multi-fluxo e Não-dir para o PRRCP.

Instância	MSC	Fluxo [71]		Multi-Fluxo [93]		Não-dir		
		Sol.	T (s)	Sol.	T (s)	Sol.	T (s)	Dif _T
lin105_R21_T21_W63_25	8893*	11121	3600.00	8893	90.91	8893	21.98	75.83
lin105_R21_T63_W21_50	11583*	58498	3600.00	-	3600.00	11583	15.03	99.58
lin105_R35_T35_W35_75	10566*	-	3600.00	11090	3600.00	11135	3600.00	0.00
pr76_R14_T16_W46_25	66007*	66533	3600.00	66007	10.40	66007	2.21	78.77
pr76_R24_T26_W26_50	74539*	116859	3600.00	74539	1012.28	75594	3600.00	-255.63
pr76_R14_T46_W16_75	90862*	219818	3600.00	90862	523.25	90862	12.61	97.59
pr107_R20_T65_W22_25	40994	-	3600.00	-	3600.00	40994	42.83	98.81
pr107_R20_T22_W65_50	35869*	47333	3600.00	35869	65.72	35869	0.45	99.31
pr107_R35_T36_W36_75	37292*	57482	3600.00	-	3600.00	37292	291.66	91.90
pr124_R42_T41_W41_25	35465	-	3600.00	-	3600.00	41123	3600.00	0.00
pr124_R24_T75_W25_50	49160	-	3600.00	-	3600.00	49184	3600.00	0.00
pr124_R24_T25_W75_75	34359*	62995	3600.00	36719	3600.00	34359	11.07	99.69
pr136_R26_T28_W82_25	47620*	76646	3600.00	47620	243.84	47620	24.25	90.05
pr136_R46_T45_W45_50	59445	-	3600.00	-	3600.00	63947	3600.00	0.00
pr136_R26_T82_W28_75	78624	-	3600.00	-	3600.00	78624	3600.00	0.00
pr144_R28_T87_W29_25	46940	-	3600.00	-	3600.00	47261	3600.00	0.00
pr144_R28_T29_W87_50	30294*	82642	3600.00	30294	407.97	33433	3600.00	-782.42
pr144_R48_T48_W48_75	43713	-	3600.00	-	3600.00	48998	3600.00	0.00
pr152_R50_T51_W51_25	50667	-	3600.00	-	3600.00	62092	3600.00	0.00
pr152_R29_T92_W31_50	61068	-	3600.00	-	3600.00	78156	3600.00	0.00
pr152_R29_T31_W92_75	49110*	70971	3600.00	49167	3600.00	49110	39.03	98.92
rat99_R19_T20_W60_25	792*	1362	3600.00	792	12.79	792	0.12	99.09
rat99_R33_T33_W33_50	847*	1196	3600.00	863	3600.00	847	3.64	99.90
rat99_R19_T60_W20_75	1017*	-	3600.00	-	3600.00	1017	1.02	99.97
rat195_R65_T65_W65_25	1605	-	3600.00	-	3600.00	1605	356.43	90.10
rat195_R39_T39_W117_50	1199	2345	3600.00	-	3600.00	1199	8.83	99.75
rat195_R39_T117_W39_75	1928	-	3600.00	-	3600.00	1928	413.66	88.51
rd100_R34_T33_W33_25	5552	15716	3600.00	5970	3600.00	5579	3600.00	0.00
rd100_R20_T60_W20_50	6619	-	3600.00	-	3600.00	6619	40.51	98.87
rd100_R20_T20_W60_75	5421*	6705	3600.00	5421	143.36	5421	1.10	99.23
si175_R35_T35_W105_25	4845*	9941	3600.00	4845	1960.30	4845	0.31	99.98
si175_R35_T105_W35_50	6871	-	3600.00	-	3600.00	6871	0.34	99.99
si175_R59_T58_W58_75	8402*	-	3600.00	-	3600.00	8402	96.16	97.33
st70_R14_T14_W42_25	419*	474	3600.00	419	42.59	419	0.33	99.21
st70_R14_T42_W14_50	554	1020	3600.00	554	1582.12	554	3.56	99.78
st70_R22_T24_W24_75	505*	826	3600.00	540	3600.00	505	1.57	99.96
u159_R31_T96_W32_25	35442	-	3600.00	-	3600.00	35573	3600.00	0.00
u159_R31_T32_W96_50	28613*	47566	3600.00	28613	606.42	28613	1.31	99.78
u159_R53_T53_W53_75	33015	-	3600.00	-	3600.00	33015	290.24	91.94
a280_R94_T93_W93_25	1717	-	3600.00	-	3600.00	2196	3600.00	0.00
a280_R56_T168_W56_50	1985	-	3600.00	-	3600.00	2001	3600.00	0.00
a280_R56_T56_W168_75	1647	-	3600.00	-	3600.00	1652	3600.00	0.00
gil262_R51_T53_W158_25	1465	-	3600.00	-	3600.00	2109	3600.00	0.00
gil262_R88_T87_W87_50	1678	-	3600.00	-	3600.00	2377	3600.00	0.00
gil262_R51_T158_W53_75	1981	-	3600.00	-	3600.00	-	3600.00	0.00
gr202_R39_T122_W41_25	32162	-	3600.00	-	3600.00	32215	3600.00	0.00
gr202_R68_T67_W67_50	28706	-	3600.00	-	3600.00	31524	3600.00	0.00
gr202_R39_T41_W122_75	25905	-	3600.00	-	3600.00	26202	3600.00	0.00
gr229_R45_T46_W138_25	61278	-	3600.00	-	3600.00	65650	3600.00	0.00
gr229_R45_T138_W46_50	96707	-	3600.00	-	3600.00	118544	3600.00	0.00
gr229_R77_T76_W76_75	90701	-	3600.00	-	3600.00	-	3600.00	0.00
lin318_R63_T64_W191_25	16297	-	3600.00	-	3600.00	-	3600.00	0.00
lin318_R108_T105_W105_50	23177	-	3600.00	-	3600.00	-	3600.00	0.00
lin318_R63_T191_W64_75	32790	-	3600.00	-	3600.00	77789	3600.00	0.00
pr226_R76_T75_W75_25	55774	-	3600.00	-	3600.00	51848	3600.00	0.00

Tabela 4.1: Resultados obtidos pelas formulações Fluxo, Multi-fluxo e Não-dir para o PRRCP.

Instância	MSC	Fluxo [71]		Multi-Fluxo [93]		Não-dir		Dif_T
		Sol.	T (s)	Sol.	T (s)	Sol.	T (s)	
pr226_R44_T46_W136_75	67286	-	3600.00	-	3600.00	-	3600.00	0.00
pr226_R44_T136_W46_50	50701	-	3600.00	-	3600.00	51848	3600.00	0.00
pr264_R52_T53_W159_25	27693	-	3600.00	-	3600.00	-	3600.00	0.00
pr264_R88_T88_W88_50	34057	-	3600.00	-	3600.00	38553	3600.00	0.00
pr264_R52_T159_W53_75	45020	-	3600.00	-	3600.00	63158	3600.00	0.00
pr299_R59_T180_W60_25	34671	-	3600.00	-	3600.00	-	3600.00	0.00
pr299_R59_T60_W180_50	20181	-	3600.00	-	3600.00	26956	3600.00	0.00
pr299_R101_T99_W99_75	29697	-	3600.00	-	3600.00	38250	3600.00	0.00
rd400_R80_T240_W80_25	10523	-	3600.00	-	3600.00	-	3600.00	0.00
rd400_R80_T80_W240_50	5747	-	3600.00	-	3600.00	-	3600.00	0.00
rd400_R136_T132_W132_75	8849	-	3600.00	-	3600.00	-	3600.00	0.00
ts225_R45_T45_W135_25	87889	-	3600.00	-	3600.00	87889	51.38	98.57
ts225_R45_T135_W45_50	108805	-	3600.00	-	3600.00	116968	3600.00	0.00
ts225_R75_T75_W75_75	96959	-	3600.00	-	3600.00	100697	3600.00	0.00
tsp225_R75_T75_W75_25	2676	-	3600.00	-	3600.00	-	3600.00	0.00
tsp225_R75_T75_W75_50	2732	-	3600.00	-	3600.00	4681	3600.00	0.00
tsp225_R45_T45_W135_75	2447	-	3600.00	-	3600.00	2447	658.31	81.71

Na literatura são apresentados resultados das formulações de LYRA [71] e SILVA [93], porém com instâncias diferentes. Portanto, na Tabela 4.1, são apresentados resultados inéditos da formulação Fluxo [71] para as instâncias propostas no trabalho de SILVA [93]. Com um tempo limite de uma hora de processamento, a formulação Fluxo [71] foi capaz de atingir soluções ótimas em apenas 6.94% das instâncias, enquanto que a formulação de SILVA [93], baseada em restrições multi-fluxo, encontra soluções ótimas em 34.72% dos casos.

Mesmo sem estabelecer um limite de tempo de processamento, a formulação Multi-fluxo [93] não consegue provar optimalidade para pelo menos metade dos casos, incluindo quatro instâncias em que o processamento é encerrado por falta de memória [93]. Portanto, a segunda coluna da Tabela 4.1 apresenta valores de soluções ótimas obtidos pelas formulações, e valores das melhores soluções obtidos por heurísticas da literatura.

Na Tabela 4.1 percebe-se que, em relação ao número de soluções ótimas encontradas, a nova formulação proposta neste trabalho foi superior às demais, atingindo solução ótima em 56% dos casos e provando optimalidade para 12 instâncias, até então resolvidas apenas por heurísticas da literatura. Vale ressaltar que os valores ótimos obtidos para essas novas instâncias são os mesmos valores já obtidos pelas heurísticas. Em relação ao tempo de processamento despendido pelo *solver*, a superioridade da formulação proposta se manteve. Em média, o novo modelo reduziu o tempo de CPU da melhor formulação da literatura (Multi-fluxo) em 43%, obtendo um tempo pior em apenas quatro casos.

4.2.1 Impacto das desigualdades válidas

Após o experimento com as três formulações, foi realizado um experimento para verificar qual o impacto de se aplicar os novos cortes propostos na formulação Não-dir. Os resultados deste experimento são apresentados na Tabela 4.2. A primeira coluna representa as instâncias utilizadas; a segunda e a terceira colunas indicam, respectivamente, a solução e o tempo computacional despendido pela formulação Não-dir, utilizando as desigualdades propostas. Por fim, a última coluna “Dif. (%) T” mostra a diferença percentual entre os tempos de resolução do modelo, com e sem a aplicação dos cortes.

Para o cálculo dessa diferença foi utilizada uma fórmula análoga à Fórmula 4.1. O valor positivo na coluna indica que houve redução no tempo de processamento do *solver*; caso o valor seja negativo, a aplicação dos cortes deixou o modelo mais lento.

Tabela 4.2: Resultados dos cortes propostos aplicados na formulação Não-dir.

Instância	Sol.	T (s)	Dif _T	Instância	Sol.	T (s)	Dif _T
att48_R9_T10_W29_25	2291	0.06	-33.33	lin105_R21_T21_W63_25	8893	3.13	602.24
att48_R9_T29_W10_50	2824	0.69	-49.28	lin105_R21_T63_W21_50	11583	25.17	-40.29
att48_R16_T16_W16_75	2650	0.81	-27.16	lin105_R35_T35_W35_75	10661	36000.00	-90.00
berlin52_R9_T11_W32_25	4861	0.07	71.43	pr76_R14_T16_W46_25	66007	5.48	-59.67
berlin52_R16_T18_W18_50	5186	1.01	195.05	pr76_R24_T26_W26_50	74539	134.40	2578.57
berlin52_R9_T32_W11_75	6916	0.32	-9.38	pr76_R14_T46_W16_75	90862	13.67	-7.75
bier127_R24_T77_W26_25	98453	161.46	-13.17	pr107_R20_T65_W22_25	40994	72.08	-40.58
bier127_R24_T26_W77_50	74745	17.05	-47.80	pr107_R20_T22_W65_50	35869	0.51	-11.76
bier127_R43_T42_W42_75	98311	261.14	-46.31	pr107_R35_T36_W36_75	37292	18.43	1482.53
brazil58_R18_T20_W20_25	19417	8.26	9601.69	pr124_R42_T41_W41_25	41891	3600.00	0.00
brazil58_R11_T35_W12_50	23312	1.59	11.95	pr124_R24_T75_W25_50	49205	3600.00	0.00
brazil58_R11_T12_W35_75	20466	0.22	-27.27	pr124_R24_T25_W75_75	34359	11.97	-7.52
brg180_R36_T36_W108_25	550	30.79	-83.96	pr136_R26_T28_W82_25	47620	17.58	37.94
brg180_R60_T60_W60_50	2350	51.79	-27.38	pr136_R46_T45_W45_50	59445	2396.73	50.20
brg180_R36_T108_W36_75	2190	81.66	51.15	pr136_R26_T82_W28_75	78624	742.79	384.66
ch130_R44_T43_W43_25	4383	55.11	-49.21	pr144_R28_T87_W29_25	54064	3600.00	0.00
ch130_R26_T26_W78_50	4139	3.24	-29.94	pr144_R28_T29_W87_50	32419	3600.00	0.00
ch130_R26_T78_W26_75	5287	546.19	-61.42	pr144_R48_T48_W48_75	45366	3600.00	0.00
ch150_R50_T50_W50_25	4778	211.03	142.52	pr152_R50_T51_W51_25	51027	3600.00	0.00
ch150_R30_T90_W30_50	5239	1417.21	-21.42	pr152_R29_T92_W31_50	74873	3600.00	0.00
ch150_R30_T30_W90_75	4292	8.79	-70.99	pr152_R29_T31_W92_75	49110	18.67	109.05
d198_R39_T40_W119_25	10916	47.22	7523.89	rat99_R19_T20_W60_25	792	0.24	-50.00
d198_R39_T119_W40_50	13877	3600.00	0.00	rat99_R33_T33_W33_50	847	5.41	-32.72
d198_R66_T66_W66_75	12986	3600.00	0.00	rat99_R19_T60_W20_75	1017	3.24	-68.52
eil51_R17_T17_W17_25	273	1.58	-18.35	rat195_R65_T65_W65_25	1605	471.69	-24.44
eil51_R9_T11_W31_50	248	0.20	-40.00	rat195_R39_T39_W117_50	1199	18.76	-52.93
eil51_R9_T31_W11_75	348	0.82	-54.88	rat195_R39_T117_W39_75	1928	827.78	-50.03
eil76_R14_T46_W16_25	387	1.45	-57.93	rd100_R34_T33_W33_25	5552	89.61	3917.41
eil76_R24_T26_W26_50	338	1.85	-48.65	rd100_R20_T60_W20_50	6619	73.37	-44.79
eil76_R14_T16_W46_75	332	0.19	131.58	rd100_R20_T20_W60_75	5421	2.04	-46.08
eil101_R19_T21_W61_25	356	2.36	-67.37	si175_R35_T35_W105_25	4845	0.39	-20.51
eil101_R19_T61_W21_50	506	30.77	-85.70	si175_R35_T105_W35_50	6871	0.97	-64.95
eil101_R33_T34_W34_75	462	39.06	-77.85	si175_R59_T58_W58_75	8402	40.32	138.49
gr48_R16_T16_W16_25	3571	0.78	-37.18	st70_R14_T14_W42_25	419	0.53	-37.74
gr48_R9_T10_W29_50	3524	0.06	-16.67	st70_R14_T42_W14_50	554	9.65	-63.11
gr48_R9_T29_W10_75	4415	0.40	-55.00	st70_R22_T24_W24_75	505	1.91	-17.80
gr96_R18_T58_W20_25	42746	12.86	3150.54	u159_R31_T96_W32_25	35442	2275.40	58.21

Tabela 4.2: Resultados dos cortes propostos aplicados na formulação Não-dir.

Instância	Sol.	T (s)	Dif_T	Instância	Sol.	T (s)	Dif_T
gr96_R32_T32_W32_50	35461	17.44	5319.67	u159_R31_T32_W96_50	28613	3.70	-64.59
gr96_R18_T20_W58_75	30015	4.03	27.54	u159_R53_T53_W53_75	33015	187.78	54.56
gr120_R40_T40_W40_25	4301	14.37	-80.17	a280_R94_T93_W93_25	2700	3600.00	0.00
gr120_R24_T72_W24_50	5159	12.74	-48.59	a280_R56_T168_W56_50	2000	3600.00	0.00
gr120_R24_T24_W72_75	4275	1.84	23.37	a280_R56_T56_W168_75	1661	3600.00	0.00
gr137_R26_T28_W83_25	39305	13.64	1883.50	gil262_R51_T53_W158_25	1465	68.27	5173.18
gr137_R45_T46_W46_50	46733	3600.00	0.00	gil262_R88_T87_W87_50	2782	3600.00	0.00
gr137_R26_T83_W28_75	58720	185.90	1836.53	gil262_R51_T158_W53_75	-	3600.00	0.00
hk48_R9_T29_W10_25	9708	2.52	-84.13	gr202_R39_T122_W41_25	-	3600.00	0.00
hk48_R16_T16_W16_50	7593	1.82	-71.43	gr202_R68_T67_W67_50	30292	3600.00	0.00
hk48_R9_T10_W29_75	7803	0.03	33.33	gr202_R39_T41_W122_75	26085	3600.00	0.00
kroA100_R20_T20_W60_25	10848	7.53	66.14	gr229_R45_T46_W138_25	61596	3600.00	0.00
kroA100_R20_T60_W20_50	16544	62.26	55.03	gr229_R45_T138_W46_50	-	3600.00	0.00
kroA100_R34_T33_W33_75	14606	51.77	6853.83	gr229_R77_T76_W76_75	95025	3600.00	0.00
kroA150_R50_T50_W50_25	16399	3600.00	0.00	lin318_R63_T64_W191_25	29187	3600.00	0.00
kroA150_R30_T30_W90_50	13157	3600.00	0.00	lin318_R108_T105_W105_50	-	3600.00	0.00
kroA150_R30_T90_W30_75	21974	1502.38	139.62	lin318_R63_T191_W64_75	75393	3600.00	0.00
kroA200_R40_T120_W40_25	39469	3600.00	0.00	pr226_R76_T75_W75_25	101345	3600.00	0.00
kroA200_R68_T66_W66_50	22377	3600.00	0.00	pr226_R44_T46_W136_75	51385	3600.00	0.00
kroA200_R40_T40_W120_75	15197	300.28	628.95	pr264_R52_T53_W159_25	28947	3600.00	0.00
kroB100_R20_T60_W20_25	18406	81.17	4335.14	pr264_R88_T88_W88_50	51014	3600.00	0.00
kroB100_R34_T33_W33_50	14160	3062.08	17.57	pr264_R52_T159_W53_75	46468	3600.00	0.00
kroB100_R34_T33_W33_75	14768	114.85	31.21	pr299_R59_T180_W60_25	46893	3600.00	0.00
kroB150_R30_T30_W90_25	11467	74.28	4746.53	pr299_R59_T60_W180_50	24216	3600.00	0.00
kroB150_R50_T50_W50_50	16049	3600.00	0.00	pr299_R101_T99_W99_75	61818	3600.00	0.00
kroB150_R30_T90_W30_75	21489	1289.45	-74.76	rd400_R80_T240_W80_25	-	3600.00	0.00
kroB200_R40_T120_W40_25	45352	3600.00	0.00	rd400_R80_T80_W240_50	10298	3600.00	0.00
kroB200_R40_T40_W120_50	14318	3600.00	0.00	rd400_R136_T132_W132_75	12776	3600.00	0.00
kroB200_R68_T66_W66_75	22633	3600.00	0.00	ts225_R45_T45_W135_25	87889	151.92	0.00
kroC100_R20_T60_W20_25	16785	33.20	739.31	ts225_R45_T135_W45_50	111606	3600.00	-66.18
kroC100_R34_T33_W33_50	13235	64.85	5451.27	ts225_R75_T75_W75_75	100234	3600.00	0.00
kroC100_R20_T20_W60_75	12582	4.95	-27.27	tsp225_R75_T75_W75_25	2693	3600.00	0.00
kroE100_R20_T60_W20_25	17199	36.41	916.89	tsp225_R75_T75_W75_50	2768	3600.00	0.00
kroE100_R20_T20_W60_50	10586	5.51	-49.36	tsp225_R45_T45_W135_75	2447	75.07	776.93

A partir da Tabela 4.2, observa-se que a aplicação dos cortes válidos apresentam benefícios à resolução da formulação, principalmente para instâncias de maior porte como a d198_VT39_T40_W119_25, kroA150_VT30_T90_W30_75 e gil262_VT51_T53_W158_25. Em termos de qualidade de solução, foi possível resolver mais 14 instâncias, em relação à formulação sem os cortes. Em termos de CPU, o desempenho do *solver* foi prejudicado ou beneficiado dependendo da instância, mas quando beneficiado o ganho em tempo foi bastante significativo, havendo um caso em que a resolução do modelo foi acelerada em 793.10 segundos.

4.3 MPO_S

Originalmente, a heurística GRASP-GENIUS-RVND [26] foi desenvolvida para ser finalizada assim que um certo valor de solução alvo fosse atingido, ou em um tempo máximo de processamento pré-definido. Esta ideia foi adotada para uma

comparação direta entre o GRASP-GENIUS-RVND e a até então heurística estado-da-arte de SILVA [93]. Uma vez que o processo heurístico do MPO_S necessita de um número máximo de iterações, neste trabalho foram realizados experimentos com 37 instâncias-teste que mais representam as características do conjunto inteiro de instâncias. Considerando os critérios de parada originais do algoritmo, verificou-se que em média, o GRASP-GENIUS-RVND executa 70 iterações. Portanto, para o processo heurístico contido no MPO_S, o parâmetro $maxIt$ foi definido como 70. Além disto, o parâmetro $\alpha = 0.6$ foi mantido como reportado em DA SILVA [26].

Em um segundo experimento, foram testados diferentes valores de γ , e observou-se que, à medida que decrementava-se o valor de γ , o E^0 foi considerado estável mais rápido e a fase heurística foi interrompida em um número menor de iterações. Para estimar melhor o valor de γ , foi realizada uma análise empírica das execuções do MPO_S a partir das mesmas 37 instâncias-teste anteriores, e variando o valor de γ em 0.05, 0.10, 0.15 e 0.20. Para cada instância, dez execuções foram feitas com sementes distintas, variando-se de um a dez. O resultado resumido deste último experimento pode ser observado na Figura 4.1.

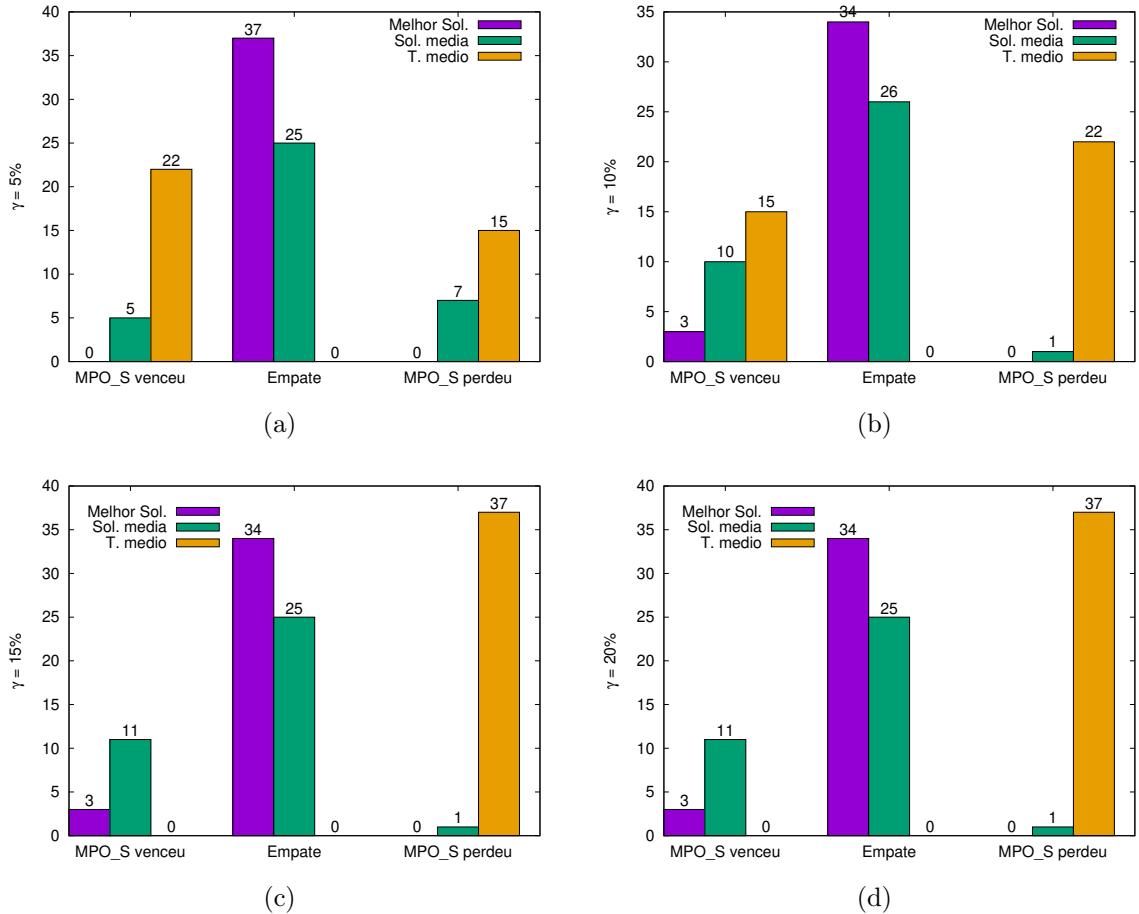


Figura 4.1: Resultados do MPO com a variação de γ .

Os resultados mostram que com $\gamma = 0.15$ e $\gamma = 0.20$, a parte heurística do MPO_-

S sempre executou o numero de iterações máximo, portanto não foram testados valores de γ maiores que 0.20, uma vez que, o conjunto E^0 não se estabilizaria nenhuma vez. Para ambos os valores, o MPO_S foi capaz de aprimorar as melhores soluções encontradas pelo GRASP-GENIUS-RVND em três instâncias, e sua solução média em 11 delas. Contudo, o tempo de processamento extra despendido pelo *solver* não compensou o ganho em qualidade de solução.

O MPO_S com $\gamma = 0.05$ foi executado em menor tempo que o GRASP-GENIUS-RVND em 22 instâncias. Porém, esta configuração não permitiu uma boa construção do conjunto E^0 e, como consequência, algumas arestas promissoras foram erroneamente removidas do modelo, e o *solver* não foi capaz de alcançar soluções de boa qualidade. Nenhuma das melhores soluções encontradas pelo GRASP-GENIUS-RVND foi melhorada, e suas médias de solução foram aprimoradas em apenas cinco casos. Apesar de demandar um pouco mais de tempo de processamento, o conjunto E^0 construído no MPO_S com $\gamma = 0.10$ foi o melhor, permitindo o *solver* aprimorar as melhores soluções do GRASP-GENIUS-RVND em três casos, e suas médias de solução em dez. Portanto, foi definido 0.10 como sendo o valor padrão de γ no MPO_S.

Para os experimentos com o MPO_S, o tempo de limite do *solver* foi reduzido para 1800 segundos, pois verificou-se que com a remoção de E^0 , o *solver* nunca atingia uma hora de processamento. Após definir os parâmetros para a estratégia proposta, foram realizados experimentos com o MPO_S e o GRASP-GENIUS-RVND utilizando as 107 instâncias restantes de SILVA [93], não utilizadas para experimentos de calibragem. A Tabela 4.3 apresenta os resultados desse experimento. A primeira coluna refere-se ao nome de todas as 144 instâncias e a segunda, o valor da melhor solução conhecida na literatura. A partir da terceira coluna, são apresentados a melhor solução, a média de solução e o tempo despendido por cada heurística. Os melhores resultados estão destacados em negrito, e a coluna “*Dif_{ms}*” apresenta o ganho percentual do MPO_S em termos de média de solução, a partir da fórmula:

$$Dif_{ms} = 100 \times \frac{(S_{grasp} - S_{mpo})}{S_{grasp}}. \quad (4.2)$$

Tabela 4.3: Comparação de resultados entre o GRASP-GENIUS-RVND e MPO.

Instância	MSC	GRASP-GENIUS-RVND				MPO_S		
		Melhor Sol.	Média Sol.	Média T. (s)	Melhor Sol.	Média Sol.	Média T. (s)	Dif _{ms}
att48_R9_T10_W29_25	2291	2291	2291.00	0.82	2291	2291.00	0.26	0.00
att48_R9_T29_W10_50	2824	2824	2824.00	4.59	2824	2824.00	2.23	0.00
att48_R16_T16_W16_75	2650	2650	2650.00	3.71	2650	2650.00	1.83	0.00
berlin52_R9_T11_W32_25	4861	4861	4861.00	0.86	4861	4861.00	0.01	0.00
berlin52_R16_T18_W18_50	5186	5186	5186.00	4.57	5186	5186.00	2.43	0.00

Tabela 4.3: Comparação de resultados entre o GRASP-GENIUS-RVND e MPO.

Instância	MSC	GRASP-GENIUS-RVND				MPO_S				Dif_{ms}
		Melhor Sol.	Média Sol.	Média T. (s)	Melhor Sol.	Média Sol.	Média T. (s)			
berlin52_R9_T32_W11_75	6916	6916	6916.00	5.99	6916	6917.00	2.81	-0.01		
bier127_R24_T77_W26_25	98453	98597	98700.70	79.45	98453	98453.00	94.17	0.25		
bier127_R24_T26_W77_50	74745	74745	74745.00	22.29	74745	74745.00	21.14	0.00		
bier127_R43_T42_W42_75	98311	98311	98390.90	101.08	98311	98329.60	142.90	0.06		
brazil58_R18_T20_W20_25	19417	19417	19417.00	4.02	19417	19417.00	6.41	0.00		
brazil58_R11_T35_W12_50	23312	23312	23312.00	8.12	23312	23312.00	7.54	0.00		
brazil58_R11_T12_W35_75	20466	20466	20466.00	1.75	20466	20466.00	0.92	0.00		
brg180_R36_T36_W108_25	550	550	550.00	22.64	550	550.00	21.19	0.00		
brg180_R60_T60_W60_50	2350	2360	2360.00	134.29	2360	2360.00	135.04	0.00		
brg180_R36_T108_W36_75	2190	2210	2211.00	185.08	2210	2210.00	234.37	0.05		
ch130_R44_T43_W43_25	4383	4383	4383.00	21.78	4383	4383.00	15.45	0.00		
ch130_R26_T26_W78_50	4139	4139	4139.00	16.65	4139	4139.00	10.69	0.00		
ch130_R26_T78_W26_75	5287	5287	5303.80	87.47	5287	5287.00	265.68	0.32		
ch150_R50_T50_W50_25	4778	4778	4800.20	58.63	4778	4790.00	50.23	0.21		
ch150_R30_T90_W30_50	5239	5239	5245.50	131.38	5239	5239.00	413.71	0.12		
ch150_R30_T30_W90_75	4292	4292	4292.00	20.33	4292	4292.00	11.38	0.00		
d198_R39_T40_W119_25	10916	10916	10916.00	53.19	10916	10916.00	1830.03	0.00		
d198_R39_T119_W40_50	13598	13598	13602.60	448.89	13598	13603.20	2247.81	0.00		
d198_R66_T66_W66_75	12905	12907	12924.40	534.73	12907	12920.40	2335.71	0.03		
eil51_R17_T17_W17_25	273	273	273.00	2.51	273	273.00	1.08	0.00		
eil51_R9_T11_W31_50	248	248	248.00	01.03	248	248.00	0.29	0.00		
eil51_R9_T31_W11_75	348	348	348.50	5.18	348	348.40	1.99	0.03		
eil76_R14_T46_W16_25	387	387	387.00	12.18	387	387.00	1.19	0.00		
eil76_R24_T26_W26_50	338	338	338.00	10.54	338	338.00	7.07	0.00		
eil76_R14_T16_W46_75	332	332	332.00	2.92	332	332.00	0.77	0.00		
eil101_R19_T21_W61_25	356	356	356.00	7.73	356	356.00	3.56	0.00		
eil101_R19_T61_W21_50	506	506	506.00	38.80	506	506.00	37.12	0.00		
eil101_R33_T34_W34_75	462	462	463.40	40.06	462	462.80	21.87	0.13		
gr48_R16_T16_W16_25	3571	3571	3571.00	1.82	3571	3571.00	0.46	0.00		
gr48_R9_T10_W29_50	3524	3524	3524.00	0.79	3524	3524.00	0.19	0.00		
gr48_R9_T29_W10_75	4415	4415	4415.00	4.16	4415	4415.00	0.92	0.00		
gr96_R18_T58_W20_25	42746	42746	42749.60	30.44	42746	42746.00	38.49	0.01		
gr96_R32_T32_W32_50	35461	35461	35461.00	58.09	35461	35461.00	874.39	0.00		
gr96_R18_T20_W58_75	30015	30015	30015.00	9.67	30015	30015.00	2.08	0.00		
gr120_R40_T40_W40_25	4301	4301	4301.00	27.80	4301	4301.00	16.92	0.00		
gr120_R24_T72_W24_50	5159	5159	5163.70	60.02	5159	5159.00	49.13	0.09		
gr120_R24_T24_W72_75	4275	4275	4275.00	13.73	4275	4275.00	6.12	0.00		
gr137_R26_T28_W83_25	39305	39305	39305.00	15.82	39305	39305.00	1815.91	0.00		
gr137_R45_T46_W46_50	46122	46122	46122.00	180.51	46122	46122.00	1847.09	0.00		
gr137_R26_T83_W28_75	58720	58720	58778.00	116.18	58720	58720.00	404.03	0.10		
hk48_R9_T29_W10_25	9708	9708	9708.60	3.82	9708	9708.60	2.34	0.00		
hk48_R16_T16_W16_50	7593	7593	7593.00	2.11	7593	7593.00	1.32	0.00		
hk48_R9_T10_W29_75	7803	7803	7803.00	0.94	7803	7803.00	0.21	0.00		
kroA100_R20_T20_W60_25	10848	10848	10848.00	5.39	10848	10848.00	5.38	0.00		
kroA100_R20_T60_W20_50	16544	16544	16544.00	34.10	16544	16544.00	29.64	0.00		
kroA100_R34_T33_W33_75	14606	14606	14606.00	43.86	14606	14606.00	62.29	0.00		
kroA150_R50_T50_W50_25	14382	14382	14453.20	115.63	14382	14382.20	204.65	0.49		
kroA150_R30_T30_W90_50	12846	12846	12964.60	41.69	12846	12969.40	255.08	-0.04		
kroA150_R30_T90_W30_75	21974	21974	21991.00	166.24	21974	21976.40	767.00	0.07		
kroA200_R40_T120_W40_25	23495	23495	23495.80	303.96	23495	23495.80	2102.80	0.00		
kroA200_R68_T66_W66_50	17297	17297	17323.90	727.36	17297	17297.60	2526.19	0.15		
kroA200_R40_T40_W120_75	15197	15197	15197.00	93.34	15197	15197.00	1765.06	0.00		
kroB100_R20_T60_W20_25	18406	18406	18406.00	29.93	18406	18406.00	1824.64	0.00		
kroB100_R34_T33_W33_50	14160	14160	14164.00	55.50	14160	14162.60	1856.18	0.01		
kroB100_R34_T33_W33_75	14768	14768	14768.00	52.13	14768	14768.00	379.05	0.00		

Tabela 4.3: Comparação de resultados entre o GRASP-GENIUS-RVND e MPO.

Instância	MSC	GRASP-GENIUS-RVND				MPO_S				Dif_{ms}
		Melhor Sol.	Média Sol.	Média T. (s)	Melhor Sol.	Média Sol.	Média T. (s)	Média T. (s)		
kroB150_R30_T30_W90_25	11467	11467	11477.80	23.54	11467	11477.80	268.80	0.00		
kroB150_R50_T50_W50_50	15388	15388	15414.80	181.73	15388	15429.40	1981.81	-0.09		
kroB150_R30_T90_W30_75	21489	21489	21541.70	138.07	21489	21489.00	297.69	0.24		
kroB200_R40_T120_W40_25	23946	24030	24057.80	320.21	24030	24057.80	2200.14	0.00		
kroB200_R40_T40_W120_50	14125	14103	14126.60	145.33	14103	14126.60	599.21	0.00		
kroB200_R68_T66_W66_75	19458	19459	19493.10	614.45	19459	19493.10	2415.18	0.00		
kroC100_R20_T60_W20_25	16785	16785	16785.00	31.50	16785	16785.00	209.09	0.00		
kroC100_R34_T33_W33_50	13235	13235	13235.00	50.06	13235	13235.00	76.23	0.00		
kroC100_R20_T20_W60_75	12582	12582	12582.00	11.39	12582	12582.00	5.90	0.00		
kroE100_R20_T60_W20_25	17199	17199	17206.50	27.83	17199	17199.00	33.63	0.04		
kroE100_R20_T20_W60_50	10586	10586	10586.00	10.97	10586	10586.00	6.96	0.00		
kroE100_R34_T33_W33_75	14622	14622	14622.00	47.92	14622	14622.00	52.61	0.00		
lin105_R21_T21_W63_25	8893	8893	8893.00	7.72	8893	8893.00	7.89	0.00		
lin105_R21_T63_W21_50	11583	11583	11583.00	102.19	11583	11583.00	6.96	0.00		
lin105_R35_T35_W35_75	10566	10566	10566.20	69.05	10566	10566.00	52.61	0.00		
pr76_R14_T16_W46_25	66007	66007	66007.00	2.55	66007	66007.00	1.32	0.00		
pr76_R24_T26_W26_50	74539	74539	74540.50	15.99	74539	74539.00	104.80	0.00		
pr76_R14_T46_W16_75	90862	90862	90862.00	16.04	90862	90862.00	10.20	0.00		
pr107_R20_T65_W22_25	40994	40994	40994.00	33.53	40994	40994.00	394.86	0.00		
pr107_R20_T22_W65_50	35869	35869	35869.00	4.80	35869	35869.00	1.40	0.00		
pr107_R35_T36_W36_75	37292	37292	37322.00	36.78	37292	37292.00	191.91	0.08		
pr124_R42_T41_W41_25	35465	35465	35465.00	61.79	35465	35465.00	1861.81	0.00		
pr124_R24_T75_W25_50	49160	49160	49160.00	115.84	49160	49160.00	2.05	0.00		
pr124_R24_T25_W75_75	34359	34359	34359.00	21.25	34359	34359.00	18.12	0.00		
pr136_R26_T28_W82_25	47620	47620	47620.00	11.09	47620	47620.00	8.62	0.00		
pr136_R46_T45_W45_50	59445	59445	59512.80	125.37	59445	59445.40	1922.40	0.11		
pr136_R26_T82_W28_75	78624	78624	78629.80	113.31	78624	78624.00	1904.90	0.01		
pr144_R28_T87_W29_25	46940	46940	46940.00	89.39	46940	46940.00	1634.82	0.00		
pr144_R28_T29_W87_50	30294	30294	30294.00	18.90	30294	30294.00	18.72	0.00		
pr144_R48_T48_W48_75	43713	43713	43726.50	155.46	43713	43713.00	561.48	0.03		
pr152_R50_T51_W51_25	50667	50667	50667.20	80.99	50667	50667.00	1876.32	0.00		
pr152_R29_T92_W31_50	61068	61068	61068.00	291.48	61068	61068.00	306.42	0.00		
pr152_R29_T31_W92_75	49110	49110	49110.00	49.12	49110	49112.20	54.18	0.00		
rat99_R19_T20_W60_25	792	792	792.00	6.35	792	792.00	6.29	0.00		
rat99_R33_T33_W33_50	847	847	847.00	24.65	847	847.00	20.60	0.00		
rat99_R19_T60_W20_75	1017	1017	1017.00	38.08	1017	1017.00	25.71	0.00		
rat195_R65_T65_W65_25	1605	1605	1609.10	344.38	1605	1606.00	469.67	0.19		
rat195_R39_T39_W117_50	1199	1199	1199.00	79.22	1199	1199.00	20.88	0.00		
rat195_R39_T117_W39_75	1928	1929	1938.00	378.31	1928	1929.00	36.44	0.46		
rd100_R34_T33_W33_25	5552	5552	5552.00	16.36	5552	5552.00	106.73	0.00		
rd100_R20_T60_W20_50	6619	6619	6644.80	32.46	6619	6626.50	34.63	0.28		
rd100_R20_T20_W60_75	5421	5421	5421.00	6.21	5421	5421.00	391.17	0.00		
si175_R35_T35_W105_25	4845	4845	4850.20	71.29	4845	4845.00	69.70	0.11		
si175_R35_T105_W35_50	6871	7011	7039.00	404.75	6871	6871.00	34.72	2.39		
si175_R59_T58_W58_75	8402	8430	8447.70	379.09	8402	8402.00	1.96	0.54		
st70_R14_T14_W42_25	419	419	419.00	2.27	419	419.00	2.37	0.00		
st70_R14_T42_W14_50	554	554	554.00	13.92	554	554.00	13.59	0.00		
st70_R22_T24_W24_75	505	505	505.00	9.39	505	505.00	3.72	0.00		
u159_R31_T96_W32_25	35442	35442	35442.00	86.73	35442	35442.00	85.97	0.00		
u159_R31_T32_W96_50	28613	28613	28613.00	27.23	28613	28613.00	23.46	0.00		
u159_R53_T53_W53_75	33015	33015	33072.00	132.21	33015	33015.00	388.41	0.17		
a280_R94_T93_W93_25	1717	1717	1724.60	747.75	1717	1724.40	2559.09	0.01		
a280_R56_T168_W56_50	1985	1996	2001.10	1208.35	1985	1985.00	1402.41	0.80		
a280_R56_T56_W168_75	1647	1647	1652.00	209.48	1647	1652.00	207.14	0.00		
gil262_R51_T53_W158_25	1465	1465	1471.30	226.15	1465	1466.60	227.63	0.32		

Tabela 4.3: Comparação de resultados entre o GRASP-GENIUS-RVND e MPO.

Instância	MSC	GRASP-GENIUS-RVND				MPO_S				Dif_{ms}
		Melhor Sol.	Média Sol.	Média T. (s)	Melhor Sol.	Média Sol.	Média T. (s)	Média T. (s)		
gil262_R88_T87_W87_50	1678	1678	1683.10	522.83	1678	1678.60	2337.61	0.27		
gil262_R51_T158_W53_75	1981	1988	1996.90	1383.03	1981	1981.00	1406.30	0.80		
gr202_R39_T122_W41_25	32162	32162	32185.40	404.52	32162	32162.00	373.51	0.07		
gr202_R68_T67_W67_50	28706	28706	28747.00	886.79	28706	28706.00	671.32	0.14		
gr202_R39_T41_W122_75	25905	25905	25905.00	147.29	25905	25905.00	150.40	0.00		
gr229_R45_T46_W138_25	61278	61278	61307.30	79.33	61278	61304.30	1878.20	0.00		
gr229_R45_T138_W46_50	96707	96707	97145.60	746.14	96707	96707.00	706.36	0.45		
gr229_R77_T76_W76_75	90701	90701	90838.20	871.34	90701	90701.00	745.01	0.15		
lin318_R63_T64_W191_25	16297	16297	16297.00	438.01	16297	16297.00	2235.68	0.00		
lin318_R108_T105_W105_50	23177	23215	23303.00	4425.67	23215	23303.00	6350.31	0.00		
lin318_R63_T191_W64_75	32790	32879	32920.90	2550.26	32879	32920.90	4409.86	0.00		
pr226_R76_T75_W75_25	55774	55812	56455.60	474.67	55812	56455.60	2280.05	0.00		
pr226_R44_T136_W46_50	67286	67323	67773.80	523.85	67323	67773.80	2547.29	0.00		
pr226_R44_T46_W136_75	50701	50768	50856.00	753.95	50768	50856.00	1592.19	0.00		
pr264_R52_T53_W159_25	27693	27693	27693.00	138.98	27693	27693.00	1939.45	0.00		
pr264_R88_T88_W88_50	34057	34057	34120.90	1597.63	34057	34119.20	3433.08	0.00		
pr264_R52_T159_W53_75	45020	45157	45398.40	1116.70	45157	45335.00	2924.32	0.14		
pr299_R59_T180_W60_25	34671	34671	34734.60	1518.55	34671	34734.60	3327.24	0.00		
pr299_R59_T60_W180_50	20181	20186	20195.70	403.78	20186	20193.80	2205.30	0.01		
pr299_R101_T99_W99_75	29697	29711	29855.80	3219.02	29711	29711.00	3184.93	0.48		
rd400_R80_T240_W80_25	10523	10557	10606.90	5892.05	10557	10606.90	7363.21	0.00		
rd400_R80_T80_W240_50	5747	5747	5747.00	2077.88	5747	5747.00	3800.37	0.00		
rd400_R136_T132_W132_75	8849	8900	8929.70	8816.49	8900	8929.70	10924.11	0.00		
ts225_R45_T45_W135_25	87889	87889	88052.10	48.37	87889	87889.00	0.25	0.19		
ts225_R45_T135_W45_50	108805	108805	109113.00	229.37	108805	109057.60	2027.41	0.05		
ts225_R75_T75_W75_75	96959	97236	97380.40	463.44	97.372	97456.15	4061.84	-0.08		
tsp225_R75_T75_W75_25	2676	2676	2676.90	251.12	2676	2676.80	2070.54	0.00		
tsp225_R75_T75_W75_50	2732	2733	2740.70	411.56	2733	2739.60	2247.29	0.04		
tsp225_R45_T45_W135_75	2447	2447	2447.50	103.88	2447	2447.00	124.04	0.02		

Considerando somente as 107 instâncias não utilizadas para a calibração do parâmetro γ , em relação à melhor solução atingida, o MPO_S foi superior em cinco e empatou em 101 instâncias, com uma redução no custo de até 2,39%. O desempenho do MPO_S foi ainda melhor em termos de solução média, vencendo em 43 e empatando em 58 casos.

Em relação ao tempo de processamento, a heurística estado-da-arte foi superior. O GRASP-GENIUS-RVND foi mais rápido em 59 casos, enquanto o MPO_S foi mais rápido em 48 casos. Contudo, verificamos que para vários casos, o MPO_S continua sendo executado mesmo já tendo encontrado a melhor solução do modelo modificado. Isso ocorre porque o *solver* é programado para tentar provar a otimalidade da solução, até que o tempo limite seja atingido.

4.4 Local branching

A técnica de local branching utilizada neste trabalho, explora a vizinhança δ -opt de uma dada solução S . Portanto, antes de aplicar o LB em todas as 144 instâncias

de SILVA [93], foi realizado um experimento de calibragem do parâmetro δ com as 37 instâncias utilizadas na calibragem do γ (Seção 4.3). Uma vez observado que, a diferença entre as soluções ótimas e as soluções obtidas pelo MPO_S é em torno de quatro arestas, então decidiu-se testar os valores 3,4 e 5 para o parâmetro δ .

Os resultados desta calibragem são apresentados nas Tabelas 8.1,8.2 e 8.3, localizadas no apêndice desta tese. Após a análise destes resultados, o parâmetro δ foi definido como 4, por apresentar o melhor custo-benefício entre custo de solução e tempo médio despendido. Os resultados com o LB para as 144 instâncias de SILVA [93] são apresentados na Tabela 4.4, na qual as duas últimas colunas representam a melhoria do LB em relação aos resultados do GRASP-GENIUS-RVND, em termos de solução média e tempo médio despendido.

Tabela 4.4: Resultados da estratégia Local branching com $\delta = 4$.

Instância	LB com $\delta = 4$				
	Melhor Sol.	Média Sol.	Média T. (s)	Dif_{ms}	Dif_T
att48_VT9_T10_W29_25	2291	2291.00	0.23	0.00	72.56
att48_VT9_T29_W10_50	2824	2824.00	2.26	0.00	50.65
att48_VT16_T16_W16_75	2650	2650.00	1.84	0.00	50.57
berlin52_VT9_T11_W32_25	4861	4861.00	0.31	0.00	63.67
berlin52_VT16_T18_W18_50	5186	5186.00	2.41	0.00	47.33
berlin52_VT9_T32_W11_75	6916	6917.67	2.68	-0.02	55.34
bier127_VT24_T77_W26_25	98531	98634.20	80.54	0.07	-1.37
bier127_VT24_T26_W77_50	74745	74745.00	21.46	0.00	3.72
bier127_VT43_T42_W42_75	98311	98358.40	101.51	0.03	-0.43
brazil58_VT18_T20_W20_25	19417	19417.00	4.18	0.00	-4.03
brazil58_VT11_T35_W12_50	23312	23312.00	7.52	0.00	7.32
brazil58_VT11_T12_W35_75	20466	20466.00	0.90	0.00	48.80
brg180_VT36_T36_W108_25	550	550.00	21.95	0.00	3.05
brg180_VT60_T60_W60_50	2360	2360.00	137.60	0.00	-2.46
brg180_VT36_T108_W36_75	2210	2210.00	188.80	0.05	-2.01
ch130_VT44_T43_W43_25	4383	4383.00	16.18	0.00	25.71
ch130_VT26_T26_W78_50	4139	4139.00	10.69	0.00	35.84
ch130_VT26_T78_W26_75	5297	5303.00	89.69	0.02	-2.53
ch150_VT50_T50_W50_25	4778	4797.60	50.37	0.05	14.08
ch150_VT30_T90_W30_50	5239	5242.40	125.89	0.06	4.18
ch150_VT30_T30_W90_75	4292	4292.00	11.50	0.00	43.42
d198_VT39_T40_W119_25	10916	10916.00	62.35	0.00	-17.22
d198_VT39_T119_W40_50	13598	13602.80	463.42	0.00	-3.24
d198_VT66_T66_W66_75	12907	12920.40	550.81	0.03	-3.01
eil51_VT17_T17_W17_25	273	273.00	1.04	0.00	58.60
eil51_VT9_T11_W31_50	248	248.00	0.26	0.00	74.27
eil51_VT9_T31_W11_75	348	348.80	2.02	-0.09	61.08
eil76_VT14_T46_W16_25	387	387.00	6.50	0.00	46.62
eil76_VT24_T26_W26_50	338	338.00	7.18	0.00	31.93
eil76_VT14_T16_W46_75	332	332.00	0.75	0.00	74.28
eil101_VT19_T21_W61_25	356	356.00	4.38	0.00	43.34
eil101_VT19_T61_W21_50	506	506.40	36.34	-0.08	6.32
eil101_VT33_T34_W34_75	462	462.80	21.03	0.13	47.50
gr48_VT16_T16_W16_25	3571	3571.00	0.44	0.00	75.85
gr48_VT9_T10_W29_50	3524	3524.00	0.19	0.00	76.08
gr48_VT9_T29_W10_75	4415	4415.00	0.91	0.00	78.07
gr96_VT18_T58_W20_25	42746	42746.00	31.10	0.01	-2.17

Tabela 4.4: Resultados da estratégia Local branching com $\delta = 4$.

Instância	LB com $\delta = 4$				
	Melhor Sol.	Média Sol.	Média T. (s)	Dif_{ms}	Dif_T
gr96_VT32_T32_W32_50	35461	35461.00	58.42	0.00	-0.57
gr96_VT18_T20_W58_75	30015	30015.00	2.05	0.00	78.76
gr120_VT40_T40_W40_25	4301	4301.00	14.97	0.00	46.15
gr120_VT24_T72_W24_50	5159	5164.60	49.54	-0.02	17.46
gr120_VT24_T24_W72_75	4275	4275.00	6.13	0.00	55.35
gr137_VT26_T28_W83_25	39305	39305.00	17.81	0.00	-12.63
gr137_VT45_T46_W46_50	46122	46122.00	188.10	0.00	-4.21
gr137_VT26_T83_W28_75	58767	58807.80	119.42	-0.05	-2.79
hk48_VT9_T29_W10_25	9708	9708.00	2.30	0.01	39.78
hk48_VT16_T16_W16_50	7593	7593.00	1.27	0.00	40.06
hk48_VT9_T10_W29_75	7803	7803.00	0.21	0.00	78.13
kroA100_VT20_T20_W60_25	10848	10848.00	5.17	0.00	4.12
kroA100_VT20_T60_W20_50	16544	16544.00	27.78	0.00	18.54
kroA100_VT34_T33_W33_75	14606	14606.00	45.14	0.00	-2.93
kroA150_VT50_T50_W50_25	14382	14429.20	119.73	0.17	-3.55
kroA150_VT30_T30_W90_50	12846	12995.60	42.50	-0.24	-1.95
kroA150_VT30_T90_W30_75	21974	21985.20	169.05	0.03	-1.69
kroA200_VT40_T120_W40_25	23495	23495.80	306.29	0.00	-0.77
kroA200_VT68_T66_W66_50	17323	17326.20	739.80	-0.01	-1.71
kroA200_VT40_T40_W120_75	15197	15197.00	98.21	0.00	-5.21
kroB100_VT20_T60_W20_25	18406	18406.00	24.47	0.00	18.25
kroB100_VT34_T33_W33_50	14160	14160.00	57.59	0.03	-3.77
kroB100_VT34_T33_W33_75	14768	14768.00	52.27	0.00	-0.26
kroB150_VT30_T30_W90_25	11467	11476.60	44.31	0.01	-88.25
kroB150_VT50_T50_W50_50	15388	15429.40	208.66	-0.09	-14.82
kroB150_VT30_T90_W30_75	21489	21549.00	139.96	-0.03	-1.37
kroB200_VT40_T120_W40_25	24030	24050.60	325.65	0.03	-1.70
kroB200_VT40_T40_W120_50	14103	14121.00	150.41	0.04	-3.50
kroB200_VT68_T66_W66_75	19459	19502.80	626.89	-0.05	-2.02
kroC100_VT20_T60_W20_25	16785	16785.00	26.58	0.00	15.60
kroC100_VT34_T33_W33_50	13235	13235.00	52.19	0.00	-4.25
kroC100_VT20_T20_W60_75	12582	12582.00	5.86	0.00	48.57
kroE100_VT20_T60_W20_25	17199	17204.00	23.63	0.01	15.07
kroE100_VT20_T20_W60_50	10586	10586.00	6.58	0.00	39.98
kroE100_VT34_T33_W33_75	14622	14622.00	49.08	0.00	-2.43
lin105_VT21_T21_W63_25	8893	8893.00	8.10	0.00	-4.92
lin105_VT21_T63_W21_50	11583	11583.00	100.81	0.00	1.35
lin105_VT35_T35_W35_75	10566	10571.80	68.96	-0.05	0.13
pr76_VT14_T16_W46_25	66007	66007.00	1.41	0.00	44.78
pr76_VT24_T26_W26_50	74539	74539.00	16.27	0.00	-1.73
pr76_VT14_T46_W16_75	90862	90862.00	13.30	0.00	17.08
pr107_VT20_T65_W22_25	40994	40994.00	34.25	0.00	-2.16
pr107_VT20_T22_W65_50	35869	35869.00	2.05	0.00	57.41
pr107_VT35_T36_W36_75	37292	37317.60	37.03	0.01	-0.67
pr124_VT42_T41_W41_25	35465	35465.00	62.18	0.00	-0.64
pr124_VT24_T75_W25_50	49160	49160.00	117.59	0.00	-1.51
pr124_VT24_T25_W75_75	34359	34359.00	20.14	0.00	5.22
pr136_VT26_T28_W82_25	47620	47620.00	8.50	0.00	23.33
pr136_VT46_T45_W45_50	59445	59468.80	128.14	0.07	-2.20
pr136_VT26_T82_W28_75	78624	78624.00	115.71	0.01	-2.11
pr144_VT28_T87_W29_25	46940	46940.00	89.72	0.00	-0.37
pr144_VT28_T29_W87_50	30294	30294.00	19.59	0.00	-3.65
pr144_VT48_T48_W48_75	43713	43713.00	154.06	0.03	0.90
pr152_VT50_T51_W51_25	50667	50667.00	120.86	0.00	-49.22
pr152_VT29_T92_W31_50	61068	61068.00	304.99	0.00	-4.64

Tabela 4.4: Resultados da estratégia Local branching com $\delta = 4$.

Instância	LB com $\delta = 4$				
	Melhor Sol.	Média Sol.	Média T. (s)	Dif_{ms}	Dif_T
pr152_VT29_T31_W92_75	49110	49110.60	25.39	0.00	48.31
rat99_VT19_T20_W60_25	792	792.00	2.23	0.00	64.84
rat99_VT33_T33_W33_50	847	847.00	20.99	0.00	14.82
rat99_VT19_T60_W20_75	1017	1017.00	36.84	0.00	3.26
rat195_VT65_T65_W65_25	1605	1607.80	352.45	0.08	-2.34
rat195_VT39_T39_W117_50	1199	1199.00	73.34	0.00	7.42
rat195_VT39_T117_W39_75	1929	1936.20	386.58	0.09	-2.18
rd100_VT34_T33_W33_25	5552	5552.00	16.68	0.00	-1.97
rd100_VT20_T60_W20_50	6619	6651.00	33.05	-0.09	-1.82
rd100_VT20_T20_W60_75	5421	5421.00	1.97	0.00	68.30
si175_VT35_T35_W105_25	4845	4845.00	73.26	0.11	-2.75
si175_VT35_T105_W35_50	6967	6991.40	412.11	0.68	-1.82
si175_VT59_T58_W58_75	8430	8447.80	389.79	0.00	-2.82
st70_VT14_T14_W42_25	419	419.00	1.78	0.00	21.45
st70_VT14_T42_W14_50	554	554.00	13.31	0.00	4.33
st70_VT22_T24_W24_75	505	505.00	5.54	0.00	41.07
u159_VT31_T96_W32_25	35442	35442.00	89.17	0.00	-2.82
u159_VT31_T32_W96_50	28613	28613.00	25.45	0.00	6.57
u159_VT53_T53_W53_75	33015	33083.40	134.37	-0.03	-1.63
a280_VT94_T93_W93_25	1717	1723.80	770.20	0.05	-3.00
a280_VT56_T168_W56_50	1996	2000.80	1236.24	0.01	-2.31
a280_VT56_T56_W168_75	1647	1652.00	212.04	0.00	-1.22
gil262_VT51_T53_W158_25	1465	1471.00	221.45	0.02	2.08
gil262_VT88_T87_W87_50	1678	1683.00	533.90	0.01	-2.12
gil262_VT51_T158_W53_75	1988	1997.20	1412.84	-0.02	-2.16
gr202_VT39_T122_W41_25	32162	32164.80	416.10	0.06	-2.86
gr202_VT68_T67_W67_50	28706	28745.60	903.07	0.00	-1.84
gr202_VT39_T41_W122_75	25905	25905.00	148.79	0.00	-1.02
gr229_VT45_T46_W138_25	61278	61304.20	191.51	0.01	-141.42
gr229_VT45_T138_W46_50	96707	97042.60	752.78	0.11	-0.89
gr229_VT77_T76_W76_75	90701	90789.40	878.96	0.05	-0.87
lin318_VT63_T64_W191_25	16297	16297.00	449.84	0.00	-2.70
lin318_VT108_T105_W105_50	23215	23309.40	4470.63	-0.03	-1.02
lin318_VT63_T191_W64_75	32878	32893.20	2611.52	0.08	-2.40
pr226_VT76_T75_W75_25	55812	56580.40	476.05	-0.22	-0.29
pr226_VT44_T46_W136_75	67323	67810.85	2535.77	-0.05	-384.07
pr226_VT44_T136_W46_50	50768	50880.00	123.62	-0.05	83.60
pr264_VT52_T53_W159_25	27693	27693.00	938.06	0.00	-574.97
pr264_VT88_T88_W88_50	34057	34119.20	3061.09	0.00	-91.60
pr264_VT52_T159_W53_75	45088	45321.20	1201.45	0.17	-7.59
pr299_VT59_T180_W60_25	34671	34734.60	1540.83	0.00	-1.47
pr299_VT59_T60_W180_50	20186	20193.60	412.81	0.01	-2.24
pr299_VT101_T99_W99_75	29711	29855.80	3258.56	0.00	-1.23
rd400_VT80_T240_W80_25	10557	10606.90	5969.60	0.00	-1.32
rd400_VT80_T80_W240_50	5747	5747.00	1987.21	0.00	4.36
rd400_VT136_T132_W132_75	8900	8929.70	8947.22	0.00	-1.48
ts225_VT45_T45_W135_25	87889	88227.10	38.79	-0.20	19.81
ts225_VT45_T135_W45_50	108805	109112.90	230.17	0.00	-0.35
ts225_VT75_T75_W75_75	97236	97380.40	472.99	0.00	-2.06
tsp225_VT75_T75_W75_25	2676	2676.90	266.38	0.00	-6.08
tsp225_VT75_T75_W75_50	2733	2740.70	439.04	0.00	-6.68
tsp225_VT45_T45_W135_75	2447	2447.50	105.00	0.00	-1.08

Considerando somente as 107 instâncias não utilizadas na calibragem do δ , a

partir da Tabela 4.4, pode ser observado que em termos de qualidade da melhor solução, o LB empatou com a heurística estado-da-arte, onde ambos venceram em três casos. Em relação à média de solução, a abordagem proposta obteve um desempenho melhor, vencendo em 31, empatando em 55 e perdendo em 21 instâncias. Em termos de tempo computacional despendido, o GRASP-GENIUS-RVND foi superior no número de vitórias, utilizando menos tempo em 65 casos, enquanto que o LB foi mais rápido em 42 instâncias.

Apesar de não ser mais rápido que a heurística estado-da-arte para 65 instâncias, o LB reduziu em média, 4.97% do tempo médio do GRASP-GENIUS-RVND. Foram realizadas reduções significativas de tempo para várias instâncias, havendo casos em que a redução foi de 83.6%. Assim como no MPO_S, o tempo de CPU do LB é também prejudicado pela necessidade do *solver* em provar a otimilidade da solução. Neste experimento foi verificado vários casos em que, a solução ótima do problema modificado já havia sido atingida, mas o *solver* gastou bastante tempo para provar sua otimalidade.

4.5 Regras de redução

SILVA [93] e LYRA [71] realizaram a aplicação de suas regras de redução, na sequência em que são apresentadas nesta tese. Porém, antes de verificar o seu impacto na redução do problema de entrada, foi realizada uma investigação sobre qual seria a melhor ordem de aplicação das regras da literatura, combinadas com as regras propostas neste trabalho.

O impacto de aplicação das regras definidas no Capítulo 3, dependem do grafo que representa o problema. Com a remoção do conjunto E^0 , tem-se uma versão mais esparsa do grafo que originalmente era completo. Portanto, com esta investigação, buscou-se também encontrar as sequências de aplicação que melhor se encaixassem em cada versão de grafo.

Para a criação das sequências, foram considerados os seguintes cenários: i) algumas regras terão uma ordem aleatória de aplicação, pois pela definição das instâncias é difícil estabelecer a posição ideal para a sua chamada; ii) algumas regras possuirão “antecessores fixos”, pois foi observado que essas regras provavelmente causarão um impacto maior se executadas após certas outras regras; e, metade das sequências serão elaboradas de modo a favorecer uma redução maior no grafo original das instâncias, e a outra metade em grafos mais esparsos (após a remoção de E^0). As sequências propostas foram as seguintes:

A: R6, R2, R13, R8, R5, R9, R7, R3, R11, R10, R4, R1, R14, R12;

B: R6, R5, R2, R13, R8, R7, R9, R4, R14, R1, R10, R11, R3, R12;

C: R10, R11, R12, R1, R3, R9, R2, R4, R6,R7,R13, R5, R14, R8;

D: R11, R3, R10, R12, R1, R4, R9, R6, R8, R2, R7, R5, R14, R13;

E: R2, R11, R3, R12, R1, R4, R10, R9, R6, R8, R7, R5, R14, R13;

F: R6, R10, R2, R11, R13, R12, R8, R1, R5, R3, R9, R4, R7, R14.

4.5.1 Redução de grafos completos

Após a definição das sequências de aplicação, foram realizados experimentos com as mesmas 37 instâncias utilizadas para a calibragem de γ no MPO_S (Seção 4.3). O primeiro experimento com as regras foi realizado a partir de um grafo completo, isto é, sem a remoção de um conjunto E^0 . Os resultados deste experimento são mostrados nas Tabelas 8.4, 8.5 e 8.6, localizadas no apêndice desta tese.

Como a aplicação das regras R10 e R11 é bem mais custosa que as demais e não são capazes de fazer redução alguma em grafos completos, ao início de cada regra R10 e R11, é verificado se o grafo de entrada é completo. Se for, então a regra não é executada. A Figura 4.2 apresenta uma visão mais resumida das tabelas supracitadas. Para cada conjunto de vértices R , T e W , é mostrada a diferença média (Δ) entre o número de vértices antes e depois da aplicação das regras, e o tempo médio gasto por cada sequência.

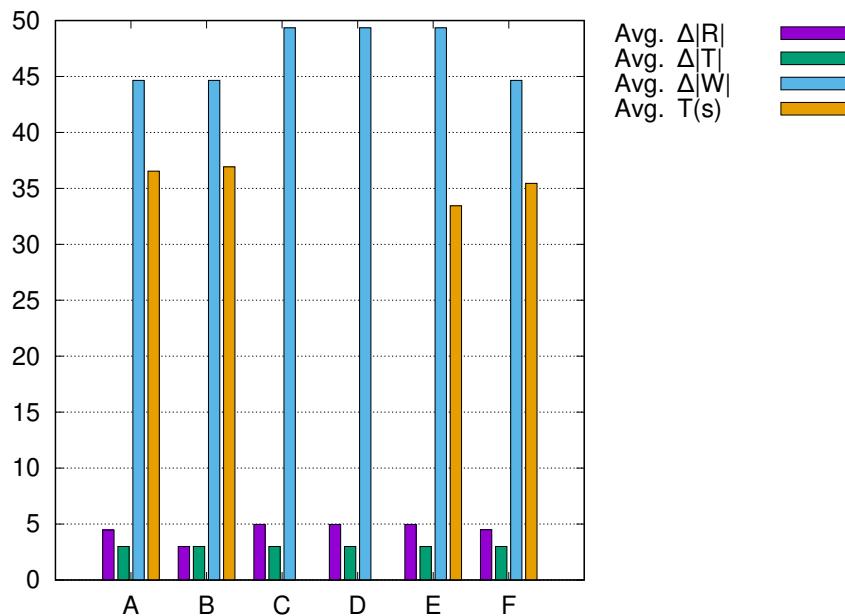


Figura 4.2: Impacto da aplicação das sequências em grafos completos.

A partir das Tabelas 8.4, 8.5, 8.6 e da Figura 4.2, observa-se que C, D e E são as regras que mais impactam na redução do grafo, empregando uma redução média

de 57.32 vértices por instância. Dentre estas regras, C e D são as mais rápidas, demandando em média menos que 0.01 segundos por instância.

Em termos de tempo de processamento, as sequências com melhor desempenho foram aquelas em que R10 e R11 foram umas das três primeiras regras aplicadas. Nas demais sequências, as regras que antecedem R10 e R11 reduzem alguns vértices, tornando o grafo incompleto e permitindo a aplicação de R10 e R11. Porém as reduções das regras que precedem R10 e R11, não deixam o grafo esparsa o suficiente para compensar o tempo gasto por estas duas últimas regras.

O conjunto W é com folga o mais reduzido entre os conjuntos de vértices, portanto, as sequências que mais reduziram vértices (C, D e E) foram aquelas que reduziram um maior número de vértices $w \in W$. A regra que, em média, mais reduziu vértices foi a R6 (remove $w \in W$ cobertos por $t \in T$) que, em todas as três sequências com mais reduções, foi precedida pela regra R1 (move $r \in R$ para T).

R1 é a única regra que precede R6 e faz algum tipo de redução. Então, criou-se a hipótese que esta precedência é o motivo das três sequências se destacarem em relação às demais. Considerando as observações supracitadas, definiu-se que as sequências C e D são as melhores quando se trata de aplicação das regras em grafos completos.

4.5.2 Redução de grafos esparsos

Após a escolha da melhor sequência a ser aplicada em grafos completos, foram realizados novos experimentos utilizando as mesmas sequências A, B, C, D e E, mas agora para grafos mais esparsos. Para tal experimento, o GRASP-GENIUS-RVND foi executado apenas uma vez para cada instância teste, com a semente de valor um e $\gamma = 0.10$. Após a chamada da heurística e a construção do conjunto E^0 , todas as arestas de E^0 foram removidas do grafo que representa a instância, originalmente completo.

Com esta remoção de arestas, tem-se um grafo mais esparsa que o original e portanto, mais suscetível a reduções de vértices a partir das regras. Assim como no caso de grafos completos, os resultados dos experimentos com grafos esparsos são apresentados em três tabelas (8.7, 8.8 e 8.9), no apêndice desta tese. A Figura 4.3 apresenta uma ilustração mais abstrata destes resultados.

Para grafos mais esparsos, o número médio de vértices reduzidos aumentou na maioria das sequências: A passou de 52.11 para 56.65; B passou de 50.65 para 56.49; C passou de 57.32 para 54.15; D passou de 57.32 para 59.21; E passou de 57.32 para 59.21; e F passou de 52.14 para 53.49. Neste último experimento, o desempenho das sequências foi mais equilibrado que o experimento em grafos completos e, novamente, obtiveram melhor desempenho as sequências que fizeram reduções no conjunto W . As sequências que mais reduziram vértices foram a D e a E, em média 59.21 vértices.

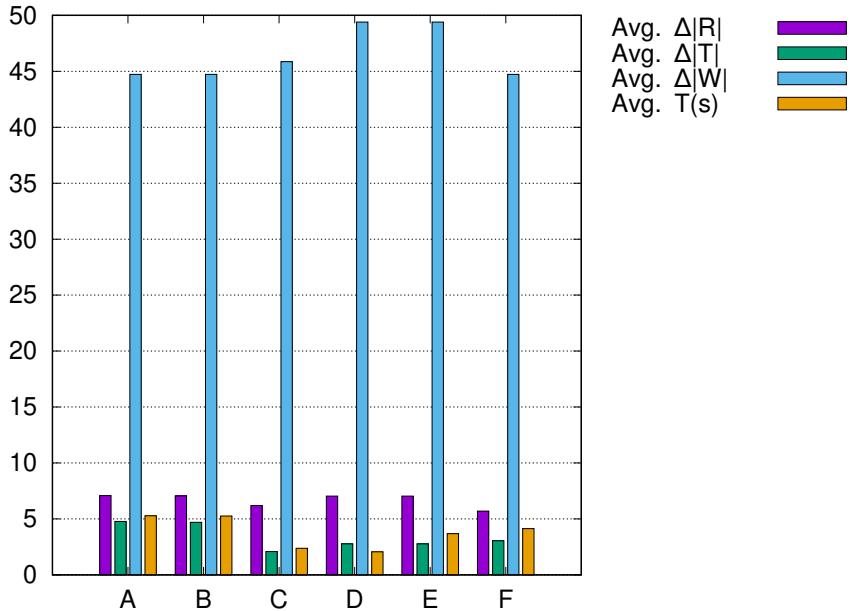


Figura 4.3: Impacto da aplicação das sequências em grafos esparsos.

Das duas sequências que mais reduziram, a sequência D foi a que gastou menor tempo de processamento, em média 2.60 segundos.

Um terceiro cenário foi utilizado para os experimentos com regras de redução. Neste cenário, cada sequência foi aplicada na mesma execução, antes da remoção de E^0 (em grafos completos), e depois da remoção de E^0 (em grafos esparsos). A partir deste experimento, foi considerado que o número de vértices reduzidos a mais, não compensou o tempo de processamento extra requerido. Por exemplo, a sequência D teve uma aumento de 6.46 em sua média de vértices reduzidos mas, por outro lado, houve um acréscimo em tempo superior a 11 segundos. Os resultados deste experimento podem ser vistos nas Tabelas 8.10, 8.11 e 8.12 localizadas no Apêndice desta tese.

4.5.3 Impacto das regras de redução nas formulações matemáticas

Nesta seção, analisa-se o impacto da aplicação das regras de redução R1 a R14 nos problemas de entrada das três formulações matemáticas apresentadas neste trabalho. Para tal, foram realizados experimentos aplicando-se a sequência D, e os resultados são apresentados na Tabela 8.13, localizada no apêndice desta tese. Nesta tabela é apresentada a solução obtida e o ganho percentual de tempo de CPU de cada modelo, em relação à execução sem aplicação das regras.

A seguir é apresentada a Tabela 4.5, que resume todas as melhorias de solução que ocorreram com a aplicação das regras. Na primeira coluna é identificada a

instância em que ocorreu a melhoria, e nas colunas restantes, para cada formulação, são apresentados o custo da solução sem aplicação das regras, seguido do novo custo obtido com a aplicação. O termo “n.h.” significa que não houve melhoria, o símbolo “+” indica que o valor obtido é ótimo e “-” aponta que não foi obtida nenhuma solução dentro do tempo limite de uma hora.

Tabela 4.5: Resultados em termos de qualidade de solução, da aplicação das regras de redução com as formulações.

Instâncias c/ Sol. melhoradas	Fluxo	Multi-fluxo	Não-dir
gr137_R26_T28_W83_25	n.h.	60630/39305 ⁺	n.h.
kroA150_R30_T90_W30_75	n.h.	n.h. 21976/21974 ⁺	
lin105_R21_T63_W21_50	n.h.	-/11583 ⁺	n.h.
lin105_R35_T35_W35_75	n.h.	11090/10566 ⁺	
pr107_R20_T65_W22_25	n.h.	-/40994 ⁺	n.h.
pr107_R35_T36_W36_75	n.h.	-/37292 ⁺	n.h.
pr152_R29_T31_W92_75	n.h.	49167/ 49110 ⁺	n.h.
rat99_R19_T60_W20_75	n.h.	-/1017 ⁺	n.h.
rat99_R19_T20_W60_25	1362/792 ⁺	n.h.	n.h.
st70_R22_T24_W24_75	n.h.	540/505 ⁺	n.h.
st70_R14_T14_W42_25	474/419 ⁺	n.h.	n.h.
kroA100_R20_T60_W20_50	n.h.	-/16544 ⁺	n.h.
eil101_R33_T34_W34_75	n.h.	462/462 ⁺	n.h.

A partir das Tabelas 4.5 e 8.13, nota-se que as regras de redução executadas antes da chamada do *solver*, contribuem para a resolução do PRRCP. Na formulação Fluxo [71], o número de instâncias resolvidas aumentou em duas unidades e, em média, seu tempo de processamento foi reduzido em 3.20%. A formulação Multi-fluxo [93], que não foi capaz de resolver vários casos por problemas relacionados com a falta de memória, foi a formulação mais beneficiada pela aplicação das regras. Esta última formulação obteve um aumento de dez instâncias resolvidas e seu tempo computacional foi reduzido em média, 19.10%. Por fim, no modelo Não-dir, houve um acréscimo de uma nova instância resolvida, e uma redução média de 15.75% no seu tempo de processamento.

4.6 Mineração de dados

Para os algoritmos DM-GRASP e MDM-GRASP foi mantido o valor definido para o parâmetro *maxIt* da heurística estado-da-arte, e o suporte mínimo foi definido como 20%, por apresentar melhor custo benefício entre o tempo despendido e a qualidade de solução, em relação aos outros valores testados (30%, 40% e 50%).

Foram realizados experimentos com 37 instâncias, a partir de dez execuções para cada configuração de suporte, utilizando sementes diferentes para cada execução.

Foram testados valores de suporte apenas de $sup_{min} = 10\%$ até 50% , pois com $sup_{min} = 60\%$ já não há ocorrência de mineração de dados em nenhuma das instâncias testadas. Os resultados deste experimento são apresentados nas Tabelas 8.14, 8.15, 8.16 e 8.17 (DM-GRASP) e Tabelas 8.18, 8.19, 8.20 e 8.21 (MDM-GRASP), localizadas no apêndice desta tese. Nas duas últimas colunas destas tabelas, são apresentados o ganho percentual em relação ao GRASP-GENIUS-RVND, para as tabelas do DM-GRASP, e em relação ao DM-GRASP, para as tabelas do MDM-GRASP.

Após a calibragem do suporte mínimo a ser utilizado, foram realizados experimentos com as 107 instâncias restantes de SILVA [93]. Os resultados deste experimento são apresentados nas Tabelas 4.6 (DM-GRASP) e 4.7 (MDM-GRASP). Para cada estratégia, são apresentados os valores das melhores soluções, das médias de soluções e das médias de tempo despendido. Os melhores resultados estão destacados em negrito.

Tabela 4.6: Comparação entre o GRASP-GENIUS-RVND e o DM-GRASP.

Instância	GRASP-GENIUS-RVND				DM-GRASP			
	Melhor Sol.	Média Sol.	Média T.(s)	Melhor Sol.	Média Sol.	Média T. (s)	Dif_{ms}	Dif_T
att48_R9_T10_W29_25	2291	2291.00	0.82	2291	2291.27	0.77	-0.01	6.70
att48_R9_T29_W10_50	2824	2824.00	4.59	2824	2824.00	3.79	0.00	17.45
att48_R16_T16_W16_75	2650	2650.00	3.71	2650	2650.00	2.98	0.00	19.68
berlin52_R9_T11_W32_25	4861	4861.00	0.86	4861	4861.00	0.78	0.00	8.32
berlin52_R16_T18_W18_50	5186	5186.00	4.57	5186	5186.00	4.13	0.00	9.80
berlin52_R9_T32_W11_75	6916	6916.00	5.99	6916	6916.00	5.87	0.00	2.00
bier127_R24_T77_W26_25	98597	98700.70	79.45	98597	98648.20	65.52	0.05	17.52
bier127_R24_T26_W77_50	74745	74745.00	22.29	74745	74745.00	16.80	0.00	24.63
bier127_R43_T42_W42_75	98311	98390.90	101.08	98311	98325.40	88.57	0.07	12.37
brazil58_R18_T20_W20_25	19417	19417.00	4.02	19417	19417.00	3.14	0.00	22.04
brazil58_R11_T35_W12_50	23312	23312.00	8.12	23312	23312.00	5.90	0.00	27.26
brazil58_R11_T12_W35_75	20466	20466.00	1.75	20466	20466.00	1.43	0.00	18.09
brg180_R36_T36_W108_25	550	550.00	22.64	550	550.00	20.72	0.00	8.52
brg180_R60_T60_W60_50	2360	2360.00	134.29	2360	2360.00	135.35	0.00	-0.79
brg180_R36_T108_W36_75	2210	2211.00	185.08	2210	2215.00	180.94	-0.18	2.23
ch130_R44_T43_W43_25	4383	4383.00	21.78	4383	4383.90	18.96	-0.02	12.95
ch130_R26_T26_W78_50	4139	4139.00	16.65	4139	4139.00	12.46	0.00	25.21
ch130_R26_T78_W26_75	5287	5303.80	87.47	5304	5306.50	72.20	-0.05	17.46
ch150_R50_T50_W50_25	4778	4800.20	58.63	4778	4797.80	42.89	0.05	26.85
ch150_R30_T90_W30_50	5239	5245.50	131.38	5239	5247.80	103.98	-0.04	20.85
ch150_R30_T30_W90_75	4292	4292.00	20.33	4292	4292.00	16.51	0.00	18.77
d198_R39_T40_W119_25	10916	10916.00	53.19	10916	10916.00	38.57	0.00	27.49
d198_R39_T119_W40_50	13598	13602.60	448.89	13598	13599.90	367.46	0.02	18.14
d198_R66_T66_W66_75	12907	12924.40	534.73	12907	12927.90	424.74	-0.03	20.57
eil51_R17_T17_W17_25	273	273.00	2.51	273	273.00	2.32	0.00	7.51
eil51_R9_T11_W31_50	248	248.00	1.03	248	248.00	0.85	0.00	17.37
eil51_R9_T31_W11_75	348	348.50	5.18	348	348.10	4.96	0.11	4.30
eil76_R14_T46_W16_25	387	387.00	12.18	387	387.00	10.26	0.00	15.76
eil76_R24_T26_W26_50	338	338.00	10.54	338	338.00	8.20	0.00	22.21
eil76_R14_T16_W46_75	332	332.00	2.92	332	332.00	2.58	0.00	11.47
eil101_R19_T21_W61_25	356	356.00	7.73	356	356.00	5.99	0.00	22.53
eil101_R19_T61_W21_50	506	506.00	38.8	506	506.00	32.53	0.00	16.15
eil101_R33_T34_W34_75	462	463.40	40.06	462	463.10	37.64	0.06	6.03
gr48_R16_T16_W16_25	3571	3571.00	1.82	3571	3571.00	1.39	0.00	23.95

Tabela 4.6: Comparação entre o GRASP-GENIUS-RVND e o DM-GRASP.

Instância	GRASP-GENIUS-RVND				DM-GRASP				Dif_{ms}	Dif_T
	Melhor Sol.	Média Sol.	Média T.(s)	Melhor Sol.	Média Sol.	Média T. (s)				
gr48_R9_T10_W29_50	3524	3524.00	0.79	3524	3524.00	0.77	0.00	2.00		
gr48_R9_T29_W10_75	4415	4415.00	4.16	4415	4415.00	3.91	0.00	5.96		
gr96_R18_T58_W20_25	42746	42749.60	30.44	42746	42746.00	25.15	0.01	17.40		
gr96_R32_T32_W32_50	35461	35461.00	58.09	35461	35461.00	47.67	0.00	17.95		
gr96_R18_T20_W58_75	30015	30015.00	9.67	30015	30015.00	7.30	0.00	24.50		
gr120_R40_T40_W40_25	4301	4301.00	27.8	4301	4301.00	19.35	0.00	30.40		
gr120_R24_T72_W24_50	5159	5163.70	60.02	5159	5161.80	49.16	0.04	18.10		
gr120_R24_T24_W72_75	4275	4275.00	13.73	4275	4275.00	10.25	0.00	25.38		
gr137_R26_T28_W83_25	39305	39305.00	15.82	39305	39305.00	13.66	0.00	13.62		
gr137_R45_T46_W46_50	46122	46122.00	180.51	46122	46122.00	137.59	0.00	23.78		
gr137_R26_T83_W28_75	58720	58778.00	116.18	58720	58748.20	90.59	0.05	22.02		
hk48_R9_T29_W10_25	9708	9708.60	3.82	9708	9708.00	3.50	0.01	8.24		
hk48_R16_T16_W16_50	7593	7593.00	2.11	7593	7593.00	1.99	0.00	5.82		
hk48_R9_T10_W29_75	7803	7803.00	0.94	7803	7803.00	0.87	0.00	7.95		
kroA100_R20_T20_W60_25	10848	10848.00	5.39	10848	10848.00	4.67	0.00	13.29		
kroA100_R20_T60_W20_50	16544	16544.00	34.1	16544	16544.00	28.29	0.00	17.03		
kroA100_R34_T33_W33_75	14606	14606.00	43.86	14606	14606.00	33.70	0.00	23.15		
kroA150_R50_T50_W50_25	14382	14453.20	115.63	14410	14453.10	104.29	0.00	9.81		
kroA150_R30_T30_W90_50	12846	12964.60	41.69	12882	12915.00	36.51	0.38	12.43		
kroA150_R30_T90_W30_75	21974	21991.00	166.24	21974	21987.00	130.56	0.02	21.47		
kroA200_R40_T120_W40_25	23495	23495.80	303.96	23495	23497.80	246.14	-0.01	19.02		
kroA200_R68_T66_W66_50	17297	17323.90	727.36	17297	17321.70	530.35	0.01	27.09		
kroA200_R40_T40_W120_75	15197	15197.00	93.34	15197	15197.00	80.03	0.00	14.26		
kroB100_R20_T60_W20_25	18406	18406.00	29.93	18406	18406.00	25.11	0.00	16.09		
kroB100_R34_T33_W33_50	14160	14164.00	55.5	14160	14164.00	43.23	0.00	22.10		
kroB100_R34_T33_W33_75	14768	14768.00	52.13	14768	14768.00	37.30	0.00	28.44		
kroB150_R30_T30_W90_25	11467	11477.80	23.54	11467	11477.80	20.07	0.00	14.74		
kroB150_R50_T50_W50_50	15388	15414.80	181.73	15388	15418.20	137.35	-0.02	24.42		
kroB150_R30_T90_W30_75	21489	21541.70	138.07	21489	21557.50	114.85	-0.07	16.82		
kroB200_R40_T120_W40_25	24030	24057.80	320.21	23954	24026.50	257.15	0.13	19.69		
kroB200_R40_T40_W120_50	14103	14126.60	145.33	14125	14134.20	113.80	-0.05	21.69		
kroB200_R68_T66_W66_75	19459	19493.10	614.45	19459	19472.40	469.76	0.11	23.55		
kroC100_R20_T60_W20_25	16785	16785.00	31.5	16785	16785.00	24.50	0.00	22.21		
kroC100_R34_T33_W33_50	13235	13235.00	50.06	13235	13235.00	39.18	0.00	21.73		
kroC100_R20_T20_W60_75	12582	12582.00	11.39	12582	12582.00	7.84	0.00	31.13		
kroE100_R20_T60_W20_25	17199	17206.50	27.83	17199	17199.00	22.99	0.04	17.40		
kroE100_R20_T20_W60_50	10586	10586.00	10.97	10586	10586.00	7.93	0.00	27.70		
kroE100_R34_T33_W33_75	14622	14622.00	47.92	14622	14622.00	36.03	0.00	24.81		
lin105_R21_T21_W63_25	8893	8893.00	7.72	8893	8893.00	5.97	0.00	22.61		
lin105_R21_T63_W21_50	11583	11583.00	102.19	11583	11583.00	76.52	0.00	25.12		
lin105_R35_T35_W35_75	10566	10566.20	69.05	10566	10566.00	54.54	0.00	21.01		
pr76_R14_T16_W46_25	66007	66007.00	2.55	66007	66007.00	2.21	0.00	13.37		
pr76_R24_T26_W26_50	74539	74540.50	15.99	74539	74540.00	14.34	0.00	10.31		
pr76_R14_T46_W16_75	90862	90862.00	16.04	90862	90862.00	13.66	0.00	14.84		
pr107_R20_T65_W22_25	40994	40994.00	33.53	40994	40994.00	24.82	0.00	25.99		
pr107_R20_T22_W65_50	35869	35869.00	4.8	35869	35869.00	3.93	0.00	18.14		
pr107_R35_T36_W36_75	37292	37322.00	36.78	37292	37309.20	26.83	0.03	27.05		
pr124_R42_T41_W41_25	35465	35465.00	61.79	35465	35465.00	51.94	0.00	15.94		
pr124_R24_T75_W25_50	49160	49160.00	115.84	49160	49164.40	87.77	-0.01	24.23		
pr124_R24_T25_W75_75	34359	34359.00	21.25	34359	34359.00	14.65	0.00	31.08		
pr136_R26_T28_W82_25	47620	47620.00	11.09	47620	47620.00	7.77	0.00	29.95		
pr136_R46_T45_W45_50	59445	59512.80	125.37	59445	59477.10	90.14	0.06	28.10		
pr136_R26_T82_W28_75	78624	78629.80	113.31	78624	78653.60	82.56	-0.03	27.14		
pr144_R28_T87_W29_25	46940	46940.00	89.39	46940	46944.80	64.67	-0.01	27.65		
pr144_R28_T29_W87_50	30294	30294.00	18.9	30294	30294.00	14.97	0.00	20.79		

Tabela 4.6: Comparação entre o GRASP-GENIUS-RVND e o DM-GRASP.

Instância	GRASP-GENIUS-RVND				DM-GRASP				Dif_{ms}	Dif_T
	Melhor Sol.	Média Sol.	Média T.(s)	Melhor Sol.	Média Sol.	Média T. (s)				
pr144_R48_T48_W48_75	43713	43726.50	155.46	43713	43743.20	125.64	-0.04	19.19		
pr152_R50_T51_W51_25	50667	50667.20	80.99	50667	50667.30	61.45	0.00	24.13		
pr152_R29_T92_W31_50	61068	61068.00	291.48	61068	61068.00	216.31	0.00	25.79		
pr152_R29_T31_W92_75	49110	49110.00	49.12	49110	49110.00	31.83	0.00	35.19		
rat99_R19_T20_W60_25	792	792.00	6.35	792	792.00	5.41	0.00	14.89		
rat99_R33_T33_W33_50	847	847.00	24.65	847	847.00	20.24	0.00	17.88		
rat99_R19_T60_W20_75	1017	1017.00	38.08	1017	1017.00	30.74	0.00	19.27		
rat195_R65_T65_W65_25	1605	1609.10	344.38	1605	1606.80	249.12	0.14	27.66		
rat195_R39_T39_W117_50	1199	1199.00	79.22	1199	1199.00	54.56	0.00	31.13		
rat195_R39_T117_W39_75	1929	1938.00	378.31	1928	1933.80	286.81	0.22	24.19		
rd100_R34_T33_W33_25	5552	5552.00	16.36	5552	5552.00	11.84	0.00	27.59		
rd100_R20_T60_W20_50	6619	6644.80	32.46	6619	6635.90	26.44	0.13	18.54		
rd100_R20_T20_W60_75	5421	5421.00	6.21	5421	5421.00	5.37	0.00	13.54		
si175_R35_T35_W105_25	4845	4850.20	71.29	4845	4847.60	65.79	0.05	7.71		
si175_R35_T105_W35_50	7011	7039.00	404.75	7011	7032.40	388.96	0.09	3.90		
si175_R59_T58_W58_75	8430	8447.70	379.09	8428	8443.10	375.49	0.05	0.95		
st70_R14_T14_W42_25	419	419.00	2.27	419	419.00	1.75	0.00	22.71		
st70_R14_T42_W14_50	554	554.00	13.92	554	554.00	11.34	0.00	18.52		
st70_R22_T24_W24_75	505	505.00	9.39	505	505.00	8.44	0.00	10.13		
u159_R31_T96_W32_25	35442	35442.00	86.73	35442	35442.00	66.29	0.00	23.57		
u159_R31_T32_W96_50	28613	28613.00	27.23	28613	28613.00	19.05	0.00	30.06		
u159_R53_T53_W53_75	33015	33072.00	132.21	33015	33072.00	106.33	0.00	19.57		
a280_R94_T93_W93_25	1717	1724.60	747.75	1718	1722.50	644.92	0.12	13.75		
a280_R56_T168_W56_50	1996	2001.10	1208.35	1998	2001.10	1024.14	0.00	15.24		
a280_R56_T56_W168_75	1647	1652.00	209.48	1647	1652.00	109.35	0.00	47.80		
gil262_R51_T53_W158_25	1465	1471.30	226.15	1465	1468.20	153.66	0.21	32.05		
gil262_R88_T87_W87_50	1678	1683.10	522.83	1678	1682.10	460.43	0.06	11.94		
gil262_R51_T158_W53_75	1988	1996.90	1383.03	1989	1999.50	1300.50	-0.13	5.97		
gr202_R39_T122_W41_25	32162	32185.40	404.52	32162	32185.20	293.56	0.00	27.43		
gr202_R68_T67_W67_50	28706	28747.00	886.79	28723	28741.70	719.21	0.02	18.90		
gr202_R39_T41_W122_75	25905	25905.00	147.29	25905	25905.00	118.46	0.00	19.58		
gr229_R45_T46_W138_25	61278	61307.30	79.33	61278	61280.50	66.34	0.04	16.37		
gr229_R45_T138_W46_50	96707	97145.60	746.14	96707	97024.60	672.38	0.12	9.88		
gr229_R77_T76_W76_75	90701	90838.20	871.34	90712	90736.30	702.71	0.11	19.35		
lin318_R63_T64_W191_25	16297	16297.00	438.01	16297	16297.00	333.22	0.00	23.93		
lin318_R108_T105_W105_50	23215	23303.00	4425.67	23137	23259.20	3973.73	0.19	10.21		
lin318_R63_T191_W64_75	32879	32920.90	2550.26	32838	32901.90	2126.96	0.06	16.60		
pr226_R76_T75_W75_25	55812	56455.60	474.67	56052	57259.20	311.15	-1.42	34.45		
pr226_R44_T136_W46_50	67323	67773.80	523.85	67323	67794.40	520.75	-0.03	0.59		
pr226_R44_T46_W136_75	50768	50856.00	753.95	50701	50768.80	85.75	0.17	88.63		
pr264_R52_T53_W159_25	27693	27693.00	138.98	27693	27693.00	99.12	0.00	28.68		
pr264_R88_T88_W88_50	34057	34120.90	1597.63	34052	34076.00	1255.50	0.13	21.41		
pr264_R52_T159_W53_75	45157	45398.40	1116.7	45088	45276.40	916.70	0.27	17.91		
pr299_R59_T180_W60_25	34671	34734.60	1518.55	34671	34701.70	1173.45	0.09	22.73		
pr299_R59_T60_W180_50	20186	20195.70	403.78	20184	20215.40	324.81	-0.10	19.56		
pr299_R101_T99_W99_75	29711	29855.80	3219.02	29711	29872.30	2654.89	-0.06	17.52		
rd400_R80_T240_W80_25	10557	10606.90	5892.05	10543	10580.70	4969.05	0.25	15.67		
rd400_R80_T80_W240_50	5747	5747.00	2077.88	5747	5747.00	1399.54	0.00	32.65		
rd400_R136_T132_W132_75	8900	8929.70	8816.49	8892	8918.10	8074.28	0.13	8.42		
ts225_R45_T45_W135_25	87889	88052.10	48.37	87889	88270.70	42.11	-0.25	12.94		
ts225_R45_T135_W45_50	108805	109113.00	229.37	108805	109117.00	225.41	0.00	1.73		
ts225_R75_T75_W75_75	97236	97380.40	463.44	96959	97181.90	392.21	0.20	15.37		
tsp225_R75_T75_W75_25	2676	2676.90	251.12	2676	2676.80	169.60	0.00	32.46		
tsp225_R75_T75_W75_50	2733	2740.70	411.56	2733	2737.00	366.19	0.14	11.02		

Tabela 4.6: Comparação entre o GRASP-GENIUS-RVND e o DM-GRASP.

Instância	GRASP-GENIUS-RVND				DM-GRASP			
	Melhor Sol.	Média Sol.	Média T.(s)	Melhor Sol.	Média Sol.	Média T. (s)	Dif_{ms}	Dif_T
tsp225_R45_T45_W135_75	2447	2447.50	103.88	2447	2447.00	81.79	0.02	21.27

A partir da Tabela 4.6, observa-se que em termos de qualidade da melhor solução, o DM-GRASP vence em 11, empata em 87 e perde em 8 instâncias. Em relação à média de solução, o método híbrido proposto vence em 37, empata em 52 e perde em 18 casos. Como geralmente relatado na literatura, a principal característica da aplicação de mineração de dados é a redução do tempo de processamento. Em termos de tempo médio despendido, o DM-GRASP foi mais rápido em 106 casos e mais lento em apenas um, com uma redução média de 19.28 % por instância.

Tabela 4.7: Comparação entre DM-GRASP e MDM-GRASP.

Instância	DM-GRASP				MDM-GRASP			
	Melhor Sol.	Média Sol.	Média T.	Melhor Sol.	Média Sol.	Média T.	Dif_{ms}	Dif_T
att48_R9_T10_W29_25	2291	2291.27	0.77	2291	2291.50	0.71	-0.01	7.37
att48_R9_T29_W10_50	2824	2824.00	3.79	2824	2824.00	3.67	0.00	3.02
att48_R16_T16_W16_75	2650	2650.00	2.98	2650	2650.00	3.03	0.00	-1.70
berlin52_R9_T11_W32_25	4861	4861.00	0.78	4861	4861.00	0.71	0.00	9.30
berlin52_R16_T18_W18_50	5186	5186.00	4.13	5186	5186.00	4.06	0.00	1.55
berlin52_R9_T32_W11_75	6916	6916.00	5.87	6916	6916.50	5.89	-0.01	-0.22
bier127_R24_T77_W26_25	98597	98648.20	65.52	98453	98600.10	64.16	0.05	2.08
bier127_R24_T26_W77_50	74745	74745.00	16.80	74745	74745.00	15.13	0.00	9.95
bier127_R43_T42_W42_75	98311	98325.40	88.57	98311	98325.40	86.69	0.00	2.13
brazil58_R18_T20_W20_25	19417	19417.00	3.14	19417	19417.00	2.96	0.00	5.57
brazil58_R11_T35_W12_50	23312	23312.00	5.90	23312	23312.00	5.66	0.00	4.17
brazil58_R11_T12_W35_75	20466	20466.00	1.43	20466	20466.00	1.43	0.00	0.36
brg180_R36_T36_W108_25	550	550.00	20.72	550	550.00	21.32	0.00	-2.90
brg180_R60_T60_W60_50	2360	2360.00	135.35	2360	2360.00	137.77	0.00	-1.79
brg180_R36_T108_W36_75	2210	2215.00	180.94	2210	2212.00	181.62	0.14	-0.38
ch130_R44_T43_W43_25	4383	4383.90	18.96	4383	4384.80	18.27	-0.02	3.66
ch130_R26_T26_W78_50	4139	4139.00	12.46	4139	4139.00	11.21	0.00	9.97
ch130_R26_T78_W26_75	5304	5306.50	72.20	5304	5306.50	71.95	0.00	0.35
ch150_R50_T50_W50_25	4778	4797.80	42.89	4778	4800.30	37.26	-0.05	13.12
ch150_R30_T90_W30_50	5239	5247.80	103.98	5239	5247.10	98.91	0.01	4.88
ch150_R30_T30_W90_75	4292	4292.00	16.51	4292	4292.00	15.91	0.00	3.62
d198_R39_T40_W119_25	10916	10916.00	38.57	10916	10916.00	36.91	0.00	4.30
d198_R39_T119_W40_50	13598	13599.90	367.46	13598	13600.00	349.83	0.00	4.80
d198_R66_T66_W66_75	12907	12927.90	424.74	12907	12926.50	389.28	0.01	8.35
eil51_R17_T17_W17_25	273	273.00	2.32	273	273.00	2.30	0.00	0.84
eil51_R9_T11_W31_50	248	248.00	0.85	248	248.00	0.97	0.00	-14.11
eil51_R9_T31_W11_75	348	348.10	4.96	348	348.40	4.98	-0.09	-0.39
eil76_R14_T46_W16_25	387	387.00	10.26	387	387.00	9.65	0.00	5.94
eil76_R24_T26_W26_50	338	338.00	8.20	338	338.00	7.58	0.00	7.53
eil76_R14_T16_W46_75	332	332.00	2.58	332	332.00	2.60	0.00	-0.68
eil101_R19_T21_W61_25	356	356.00	5.99	356	356.00	5.22	0.00	12.88
eil101_R19_T61_W21_50	506	506.00	32.53	506	506.00	29.71	0.00	8.67
eil101_R33_T34_W34_75	462	463.10	37.64	462	463.40	36.18	-0.06	3.89
gr48_R16_T16_W16_25	3571	3571.00	1.39	3571	3571.00	1.29	0.00	6.61
gr48_R9_T10_W29_50	3524	3524.00	0.77	3524	3524.00	0.77	0.00	0.04
gr48_R9_T29_W10_75	4415	4415.00	3.91	4415	4415.00	3.89	0.00	0.64

Tabela 4.7: Comparação entre DM-GRASP e MDM-GRASP.

Instância	DM-GRASP			MDM-GRASP			Dif_{ms}	Dif_T
	Melhor Sol.	Média Sol.	Média T.	Melhor Sol.	Média Sol.	Média T.		
gr96_R18_T58_W20_25	42746	42746.00	25.15	42746	42746.00	23.48	0.00	6.64
gr96_R32_T32_W32_50	35461	35461.00	47.67	35461	35461.00	44.42	0.00	6.81
gr96_R18_T20_W58_75	30015	30015.00	7.30	30015	30015.00	6.77	0.00	7.25
gr120_R40_T40_W40_25	4301	4301.00	19.35	4301	4301.00	15.38	0.00	20.52
gr120_R24_T72_W24_50	5159	5161.80	49.16	5159	5162.30	46.70	-0.01	5.00
gr120_R24_T24_W72_75	4275	4275.00	10.25	4275	4275.00	9.38	0.00	8.50
gr137_R26_T28_W83_25	39305	39305.00	13.66	39305	39305.00	13.44	0.00	1.63
gr137_R45_T46_W46_50	46122	46122.00	137.59	46122	46122.00	133.47	0.00	2.99
gr137_R26_T83_W28_75	58720	58748.20	90.59	58720	58757.60	86.58	-0.02	4.43
hk48_R9_T29_W10_25	9708	9708.00	3.50	9708	9708.60	3.50	-0.01	0.11
hk48_R16_T16_W16_50	7593	7593.00	1.99	7593	7593.00	2.01	0.00	-0.90
hk48_R9_T10_W29_75	7803	7803.00	0.87	7803	7803.00	0.86	0.00	0.79
kroA100_R20_T20_W60_25	10848	10848.00	4.67	10848	10848.00	4.41	0.00	5.64
kroA100_R20_T60_W20_50	16544	16544.00	28.29	16544	16544.00	25.83	0.00	8.70
kroA100_R34_T33_W33_75	14606	14606.00	33.70	14606	14606.00	32.12	0.00	4.71
kroA150_R50_T50_W50_25	14410	14453.10	104.29	14382	14444.20	104.26	0.06	0.02
kroA150_R30_T30_W90_50	12882	12915.00	36.51	12846	12912.60	32.62	0.02	10.65
kroA150_R30_T90_W30_75	21974	21987.00	130.56	21974	21987.50	128.13	0.00	1.86
kroA200_R40_T120_W40_25	23495	23497.80	246.14	23495	23496.20	225.55	0.01	8.36
kroA200_R68_T66_W66_50	17297	17321.70	530.35	17297	17320.20	553.18	0.01	-4.30
kroA200_R40_T40_W120_75	15197	15197.00	80.03	15197	15197.00	76.88	0.00	3.93
kroB100_R20_T60_W20_25	18406	18406.00	25.11	18406	18406.00	24.10	0.00	4.03
kroB100_R34_T33_W33_50	14160	14164.00	43.23	14160	14162.70	42.23	0.01	2.32
kroB100_R34_T33_W33_75	14768	14768.00	37.30	14768	14768.00	33.47	0.00	10.28
kroB150_R30_T30_W90_25	11467	11477.80	20.07	11467	11475.40	19.38	0.02	3.42
kroB150_R50_T50_W50_50	15388	15418.20	137.35	15388	15437.80	133.72	-0.13	2.64
kroB150_R30_T90_W30_75	21489	21557.50	114.85	21489	21556.80	107.64	0.00	6.28
kroB200_R40_T120_W40_25	23954	24026.50	257.15	23954	24037.80	241.35	-0.05	6.15
kroB200_R40_T40_W120_50	14125	14134.20	113.80	14125	14132.80	110.38	0.01	3.01
kroB200_R68_T66_W66_75	19459	19472.40	469.76	19459	19467.20	441.08	0.03	6.11
kroC100_R20_T60_W20_25	16785	16785.00	24.50	16785	16785.00	22.92	0.00	6.46
kroC100_R34_T33_W33_50	13235	13235.00	39.18	13235	13235.00	37.80	0.00	3.53
kroC100_R20_T20_W60_75	12582	12582.00	7.84	12582	12582.00	7.41	0.00	5.54
kroE100_R20_T60_W20_25	17199	17199.00	22.99	17199	17206.50	21.27	-0.04	7.46
kroE100_R20_T20_W60_50	10586	10586.00	7.93	10586	10586.00	7.37	0.00	7.04
kroE100_R34_T33_W33_75	14622	14622.00	36.03	14622	14622.00	32.44	0.00	9.95
lin105_R21_T21_W63_25	8893	8893.00	5.97	8893	8893.00	5.42	0.00	9.25
lin105_R21_T63_W21_50	11583	11583.00	76.52	11583	11583.00	53.66	0.00	29.88
lin105_R35_T35_W35_75	10566	10566.00	54.54	10566	10566.20	52.04	0.00	4.59
pr76_R14_T16_W46_25	66007	66007.00	2.21	66007	66007.00	2.24	0.00	-1.39
pr76_R24_T26_W26_50	74539	74540.00	14.34	74539	74545.00	13.80	-0.01	3.78
pr76_R14_T46_W16_75	90862	90862.00	13.66	90862	90862.00	13.44	0.00	1.60
pr107_R20_T65_W22_25	40994	40994.00	24.82	40994	40994.00	23.71	0.00	4.46
pr107_R20_T22_W65_50	35869	35869.00	3.93	35869	35869.00	4.09	0.00	-3.94
pr107_R35_T36_W36_75	37292	37309.20	26.83	37292	37317.20	24.14	-0.02	10.04
pr124_R42_T41_W41_25	35465	35465.00	51.94	35465	35465.00	51.89	0.00	0.09
pr124_R24_T75_W25_50	49160	49164.40	87.77	49160	49160.00	80.56	0.01	8.22
pr124_R24_T25_W75_75	34359	34359.00	14.65	34359	34359.00	12.24	0.00	16.45
pr136_R26_T28_W82_25	47620	47620.00	7.77	47620	47620.00	6.98	0.00	10.17
pr136_R46_T45_W45_50	59445	59477.10	90.14	59445	59478.00	82.84	0.00	8.10
pr136_R26_T82_W28_75	78624	78653.60	82.56	78624	78624.00	78.24	0.04	5.23
pr144_R28_T87_W29_25	46940	46944.80	64.67	46940	46940.00	55.84	0.01	13.66
pr144_R28_T29_W87_50	30294	30294.00	14.97	30294	30294.00	13.75	0.00	8.20
pr144_R48_T48_W48_75	43713	43743.20	125.64	43713	43713.00	112.97	0.07	10.08
pr152_R50_T51_W51_25	50667	50667.30	61.45	50667	50667.10	55.52	0.00	9.66

Tabela 4.7: Comparação entre DM-GRASP e MDM-GRASP.

Instância	DM-GRASP			MDM-GRASP				
	Melhor Sol.	Média Sol.	Média T.	Melhor Sol.	Média Sol.	Média T.	Dif _{ms}	Dif _T
pr152_R29_T92_W31_50	61068	61068.00	216.31	61068	61068.00	192.14	0.00	11.17
pr152_R29_T31_W92_75	49110	49110.00	31.83	49110	49110.00	28.06	0.00	11.84
rat99_R19_T20_W60_25	792	792.00	5.41	792	792.00	5.22	0.00	3.49
rat99_R33_T33_W33_50	847	847.00	20.24	847	847.00	17.13	0.00	15.35
rat99_R19_T60_W20_75	1017	1017.00	30.74	1017	1017.00	29.68	0.00	3.45
rat195_R65_T65_W65_25	1605	1606.80	249.12	1605	1606.20	217.45	0.04	12.72
rat195_R39_T39_W117_50	1199	1199.00	54.56	1199	1199.00	48.54	0.00	11.03
rat195_R39_T117_W39_75	1928	1933.80	286.81	1928	1930.10	265.36	0.19	7.48
rd100_R34_T33_W33_25	5552	5552.00	11.84	5552	5552.00	10.52	0.00	11.15
rd100_R20_T60_W20_50	6619	6635.90	26.44	6619	6653.30	23.99	-0.26	9.25
rd100_R20_T20_W60_75	5421	5421.00	5.37	5421	5421.00	5.19	0.00	3.37
si175_R35_T35_W105_25	4845	4847.60	65.79	4845	4845.00	60.59	0.05	7.92
si175_R35_T105_W35_50	7011	7032.40	388.96	6987	7025.10	380.22	0.10	2.25
si175_R59_T58_W58_75	8428	8443.10	375.49	8404	8436.90	383.88	0.07	-2.23
st70_R14_T14_W42_25	419	419.00	1.75	419	419.00	1.74	0.00	0.80
st70_R14_T42_W14_50	554	554.00	11.34	554	554.00	11.01	0.00	2.87
st70_R22_T24_W24_75	505	505.00	8.44	505	505.00	8.37	0.00	0.87
u159_R31_T96_W32_25	35442	35442.00	66.29	35442	35442.00	62.34	0.00	5.96
u159_R31_T32_W96_50	28613	28613.00	19.05	28613	28613.00	16.38	0.00	13.99
u159_R53_T53_W53_75	33015	33072.00	106.33	33015	33079.40	104.14	-0.02	2.06
a280_R94_T93_W93_25	1718	1722.50	644.92	1717	1723.70	595.72	-0.07	7.63
a280_R56_T168_W56_50	1998	2001.10	1024.14	2001	2001.40	946.62	-0.01	7.57
a280_R56_T56_W168_75	1647	1652.00	109.35	1647	1652.18	147.49	-0.01	-34.88
gil262_R51_T53_W158_25	1465	1468.20	153.66	1465	1467.50	136.62	0.05	11.09
gil262_R88_T87_W87_50	1678	1682.10	460.43	1678	1680.20	426.51	0.11	7.37
gil262_R51_T158_W53_75	1989	1999.50	1300.50	1987	1996.60	1221.01	0.15	6.11
gr202_R39_T122_W41_25	32162	32185.20	293.56	32162	32181.00	252.00	0.01	14.16
gr202_R68_T67_W67_50	28723	28741.70	719.21	28723	28747.10	655.80	-0.02	8.82
gr202_R39_T41_W122_75	25905	25905.00	118.46	25905	25905.00	102.70	0.00	13.30
gr229_R45_T46_W138_25	61278	61280.50	66.34	61278	61291.10	59.93	-0.02	9.67
gr229_R45_T138_W46_50	96707	97024.60	672.38	96707	96952.40	642.61	0.07	4.43
gr229_R77_T76_W76_75	90712	90736.30	702.71	90701	90765.50	676.10	-0.03	3.79
lin318_R63_T64_W191_25	16297	16297.00	333.22	16297	16297.00	297.43	0.00	10.74
lin318_R108_T105_W105_50	23137	23259.20	3973.73	23137	23245.00	3726.65	0.06	6.22
lin318_R63_T191_W64_75	32838	32901.90	2126.96	32790	32899.40	1945.06	0.01	8.55
pr226_R76_T75_W75_25	56052	57259.20	311.15	56052	57258.30	300.60	0.00	3.39
pr226_R44_T136_W46_50	67323	67794.40	520.75	67286	67702.60	480.44	0.14	7.74
pr226_R44_T46_W136_75	50701	50768.80	85.75	50768	50774.00	89.38	-0.01	-4.24
pr264_R52_T53_W159_25	27693	27693.00	99.12	27693	27693.00	93.38	0.00	5.79
pr264_R88_T88_W88_50	34052	34076.00	1255.50	34000	34054.20	1227.13	0.06	2.26
pr264_R52_T159_W53_75	45088	45276.40	916.70	44957	45232.80	822.49	0.10	10.28
pr299_R59_T180_W60_25	34671	34701.70	1173.45	34671	34709.40	1054.92	-0.02	10.10
pr299_R59_T60_W180_50	20184	20215.40	324.81	20181	20212.00	324.45	0.02	0.11
pr299_R101_T99_W99_75	29711	29872.30	2654.89	29711	29828.60	2424.24	0.15	8.69
rd400_R80_T240_W80_25	10543	10580.70	4969.05	10529	10568.30	4641.00	0.12	6.60
rd400_R80_T80_W240_50	5747	5747.00	1399.54	5747	5747.00	1237.00	0.00	11.61
rd400_R136_T132_W132_75	8892	8918.10	8074.28	8891	8915.45	8295.57	0.03	-2.74
ts225_R45_T45_W135_25	87889	88270.70	42.11	87889	88209.50	38.09	0.07	9.53
ts225_R45_T135_W45_50	108805	109117.00	225.41	108805	109028.00	214.61	0.08	4.79
ts225_R75_T75_W75_75	96959	97181.90	392.21	97106	97327.60	358.43	-0.15	8.61
tsp225_R75_T75_W75_25	2676	2676.80	169.60	2676	2676.70	157.49	0.00	7.14
tsp225_R75_T75_W75_50	2733	2737.00	366.19	2733	2736.50	329.06	0.02	10.14
tsp225_R45_T45_W135_75	2447	2447.00	81.79	2447	2447.00	75.47	0.00	7.73

Uma vez que, na análise anterior, o DM-GRASP foi superior à heurística estado-da-arte, foi decidido comparar o MDM-GRASP apenas com a estratégia DM-GRASP, e utilizando apenas as 107 instâncias não incluídas nos testes de calibração. Na Tabela 4.7 pode ser observado que, em termos de qualidade da melhor solução obtida, o MDM-GRASP foi melhor em 13 casos, empatou em 91 e perdeu em três. No que se refere à solução média, a estratégia com *Multi Data-mining* foi superior em 34 casos, empatou em 50 e perdeu em 23. Em relação ao tempo de execução, o MDM-GRASP conseguiu melhorar ainda mais os resultados do DM-GRASP, sendo mais rápido em 92 instâncias, empatando em uma (com precisão de duas casas decimais) e perdendo em 14. Em média, o MDM-GRASP foi 5.32% mais rápido que o DM-GRASP.

4.6.1 O impacto da utilização de padrões minerados

Nesta subseção, uma execução utilizando a instância kroA150_RT30_T30_W90_50 foi selecionada para demonstrar o impacto da utilização de padrões minerados em soluções iniciais. Para cada estratégia de mineração, duas figuras ilustram os resultados para cada fase da heurística: construtiva e busca local, no qual o eixo das abscissas contém as iterações e o eixo das ordenadas possuem os valores dos custos das soluções.

Para ilustrar as análises da instância kroA150_R30_T30_W90_50, a Figura 4.4 demonstra que, devido ao uso de padrões minerados, valores menores de custo são obtidos. Até a iteração 35 (metade do número máximo de iterações), os custos obtidos por ambas as estratégias DM-GRASP e GRASP-GENIUS-RVND são os mesmos. A partir da iteração 36, logo após a mineração de dados, os custos obtidos pelas duas estratégias se tornam distintos, indicando que os resultados do DM-GRASP são melhores em 92% das 35 iterações restantes. Para a busca local, apresentada na Figura 4.5, a partir do processo de mineração em diante, o DM-GRASP apresenta melhores resultados para 80% das outras 35 iterações, comparado aos valores obtidos pela busca local do GRASP-GENIUS-RVND.

A mesma análise com o DM-GRASP sobre o impacto da utilização de padrões minerados, também foi realizada com o MDM-GRASP. A partir das Figuras 4.6 e 4.7, pode ser observado que após o processo de MD, as fases de construção e busca local apresentam linhas próximas para as duas estratégias de mineração de dados, demonstrando, dessa maneira, comportamentos similares. Para a instância utilizada no experimento, o MDM-GRASP realizou quatro vezes o procedimento de mineração de dados, após as iterações 10, 34, 48, e 62. Diferentemente do DM-GRASP, é possível verificar que o MDM-GRASP, após a primeira execução da mineração, continua a reduzir ligeiramente e gradualmente o custo das soluções obtidas pela construção, já que os padrões são extraídos mais de uma vez.

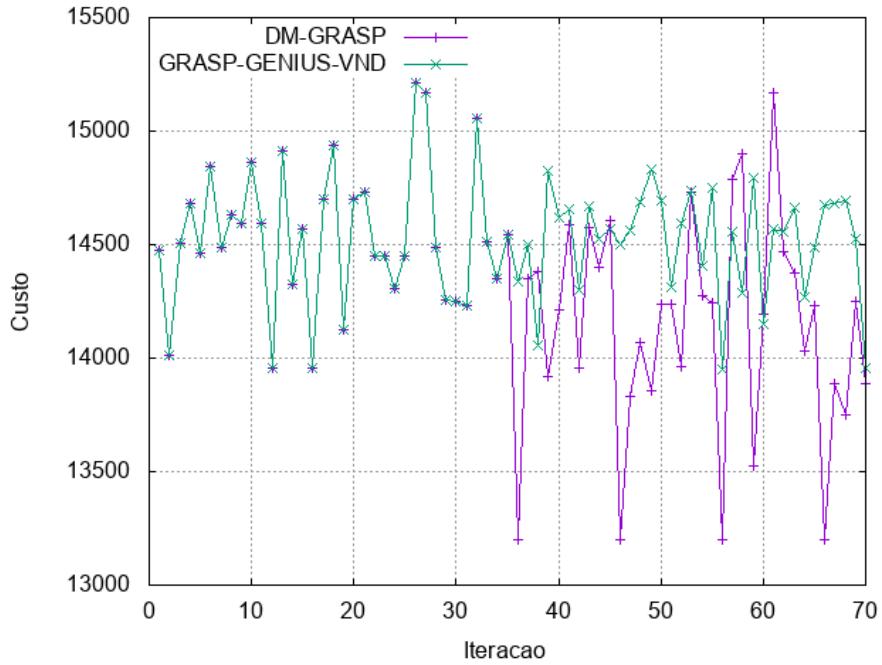


Figura 4.4: Fase de construção

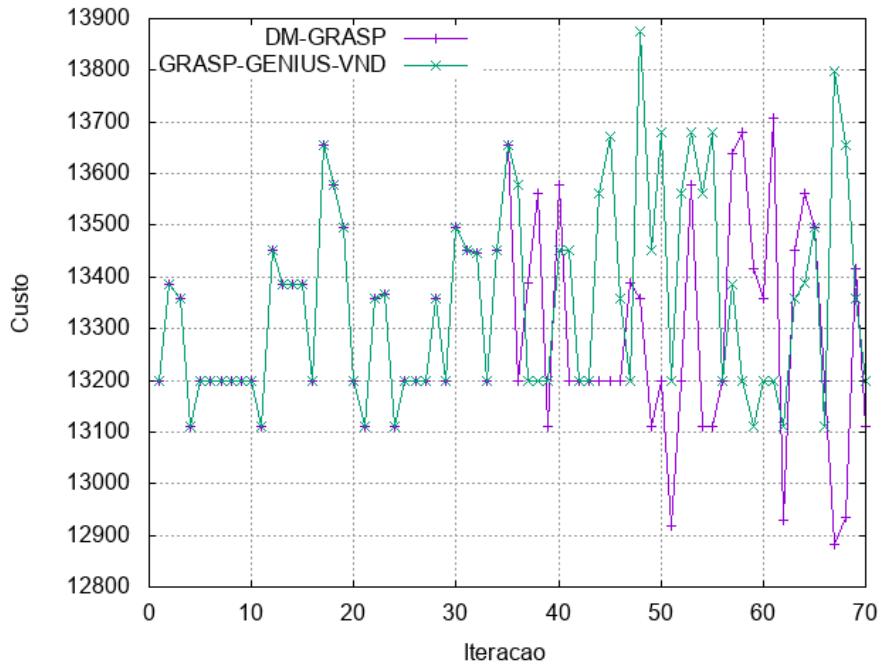


Figura 4.5: Fase de busca local

Observa-se ainda que, as estratégias propostas são capazes de explorar melhor o espaço de busca. Há várias situações em que a heurística estado-da-arte realiza essa exploração de maneira ineficiente. Por exemplo, na Figura 4.7, pode-se verificar que o valor obtido pela busca local do GRASP-GENIUS-RVND não se altera das iterações 5 a 10, 13 a 15, 25 a 27 e 37 a 39. Na execução específica desse experimento, esse comportamento tende a ocorrer menos quando há aplicação de MD.

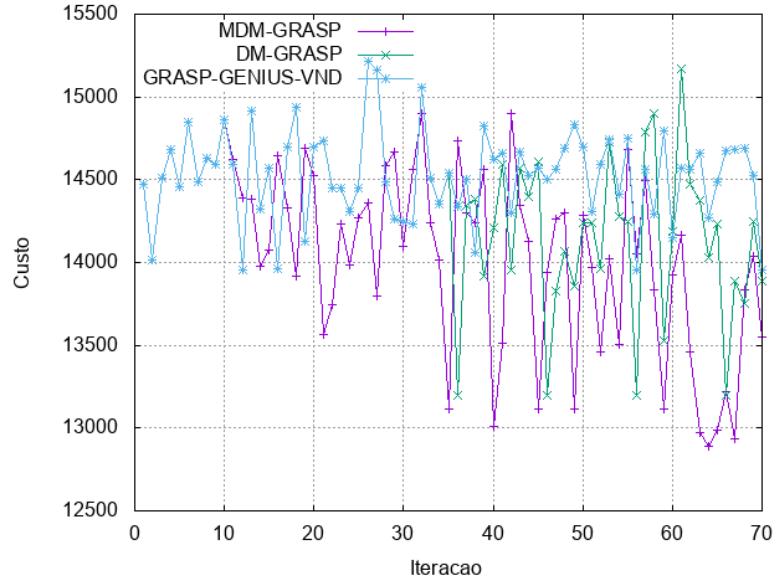


Figura 4.6: Fase de construção

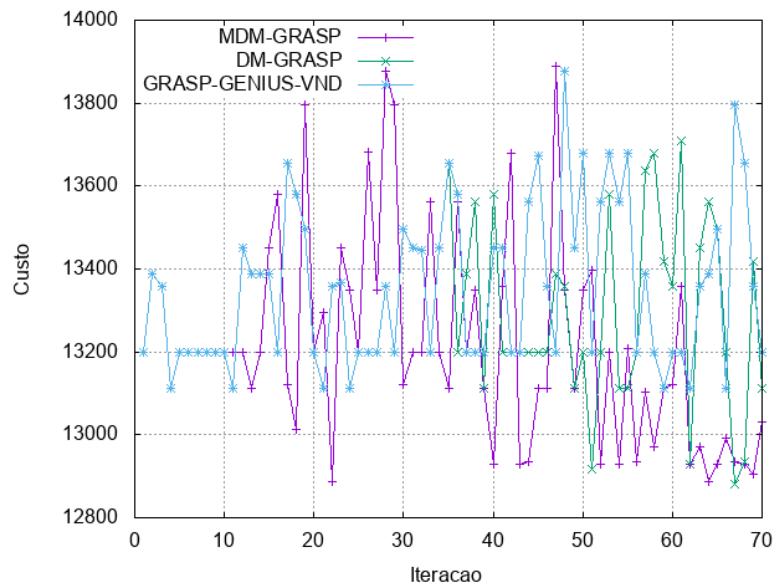


Figura 4.7: Fase de busca local

Nas Tabelas 4.6 e 4.7, foi verificado que a principal contribuição da aplicação de MD na heurística estado-da-arte é na redução do tempo computacional despendido. As Figuras 4.8 e 4.9 ilustram o comportamento das fases de construção e busca local, em termos de tempo computacional utilizado pelo GRASP-GENIUS-RVND, DM-GRASP e MDM-GRASP, por meio da execução de 70 iterações da instância kroA150_VT30_T30_W90_50. As figuras mostram que, para todas as estratégias, a fase de busca local gastou mais tempo que a fase de construção. Uma vez que na estratégia DM-GRASP, o processo de mineração de dados é executado após a iteração 35, pode ser observado na Figura 4.8 que, a partir deste ponto em diante, o

tempo computacional gasto pela fase de construção é reduzido. Originalmente, a construção de uma rota é inicializada por meio da seleção de três vértices aleatórios e, em seguida, vértices são inseridos na rota até que todas as restrições do problema sejam satisfeitas. Com a aplicação da MD, esta inserção de vértices ocorre a partir de um padrão minerado, que geralmente possui mais de três vértices, o que diminui o número de inserções necessárias para satisfazer todas as condições do problema e, consequentemente, reduz o tempo computacional despendido.

A busca local do GRASP-GENIUS-RVND é realizada pela heurística RVND. Essa estratégia é encerrada apenas quando todos os tipos de vizinhanças são explorados, e a cada nova melhoria de solução, o RVND reinicia seu processo a partir da primeira vizinhança da lista. Como as soluções geradas pela construção a partir dos padrões geralmente estão mais próximas de um ótimo local, na Figura 4.9, pode ser observado que a exploração de todos os tipos de vizinhanças é realizada mais rapidamente, pois menos melhorias são realizadas, e o RVND retorna menos vezes para a primeira vizinhança da lista.

Na estratégia MDM-GRASP, o processo de mineração de dados é executado mais de uma vez. Como pode ser observado nas Figuras 4.8 e 4.9, o MDM-GRASP comporta-se de maneira semelhante ao DM-GRASP até o momento em que a primeira mineração é realizada. A partir desta iteração, quanto mais processos de MD são executados, os tempos computacionais das fases de construção e busca local são gradualmente e ligeiramente reduzidos.

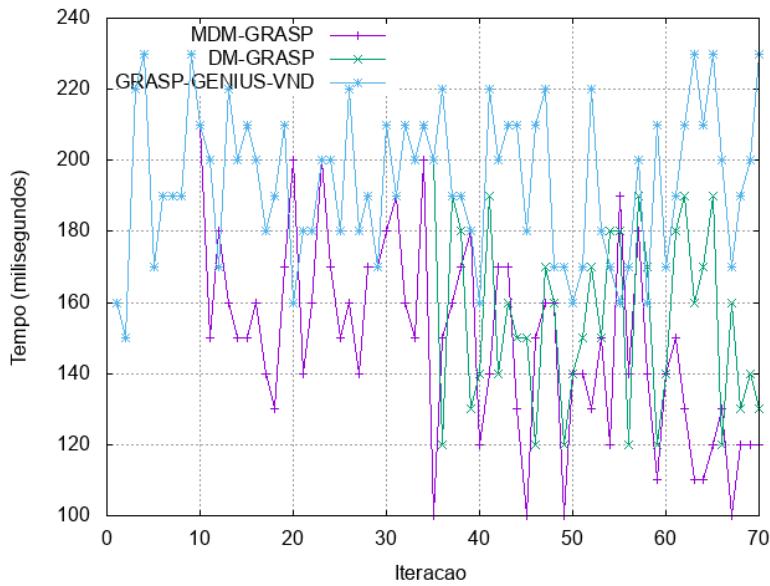


Figura 4.8: Fase de construção

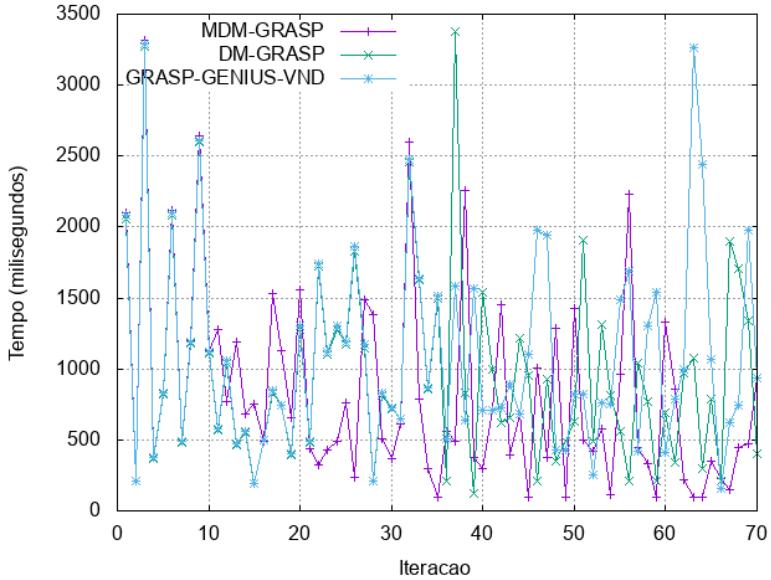


Figura 4.9: Fase de busca local

4.7 Análise das melhores estratégias

Nesta seção, são apresentadas análises de comportamento acerca das melhores estratégias propostas de mineração de dados (MDM-GRASP) e hibridizadas com PLI (MPO_S). Inicialmente, foi realizada uma comparação entre as três estratégias, baseada nos gráficos *Time-to-target plots* (TTTplots) [2], utilizados para analisar o comportamento de algoritmos com componentes aleatórias. Basicamente, esses gráficos mostram a probabilidade acumulada de um algoritmo encontrar um valor de solução melhor ou igual a um valor de solução alvo pré-definido, em um certo tempo de execução.

Os gráficos TTTplots ilustrados nas Figuras 4.10 e 4.11 foram gerados a partir de 100 execuções dos métodos GRASP-GENIUS-RVND, MPO_S e MDM-GRASP a partir da instância kroA150_VT30_T30_W90_50, e utilizando-se sementes que variam de 1 a 100. Para a análise de comportamento em diferentes cenários, foram escolhidos dois valores alvo, um alvo fácil baseado na média atingida nessa instância pela heurística estado-da-arte (12962), e um alvo mais difícil, o valor da melhor solução reportada nessa instância pelo GRASP-GENIUS-RVND (12918).

Nos TTTPlots apresentados, é explícita a superioridade de ambos os métodos propostos em relação ao GRASP-GENIUS-RVND, quando propõem-se alcançar um determinado valor de solução. Para o alvo mais fácil (Figura 4.10), até aproximadamente 30 segundos de execução, o MPO_S e a heurística estado-da-arte possuem probabilidades semelhantes de atingir o alvo. Contudo, esta semelhança rapidamente se desfaz poucos segundos depois. Enquanto que com 40 segundos de execução o MPO_S atinge o alvo com uma chance de 90%, o GRASP-GENIUS-RVND possui

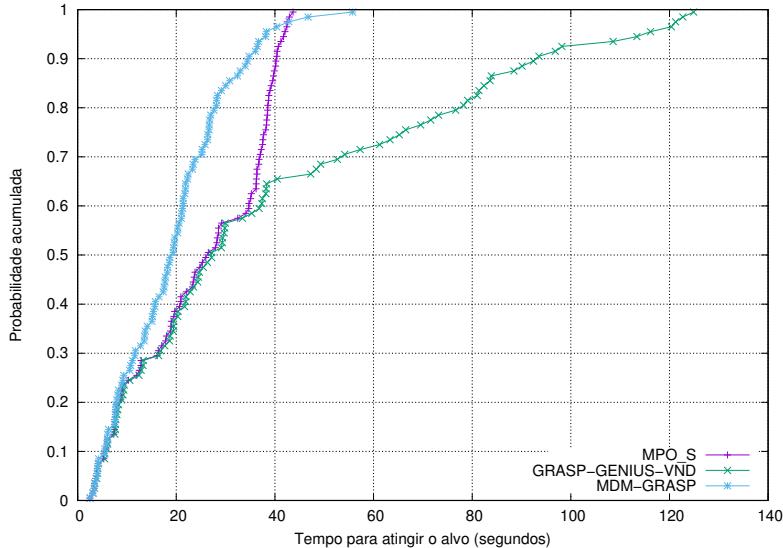


Figura 4.10: TTTplots para a instância kroA150_VT30_T30_W90_50, alvo: 12962.

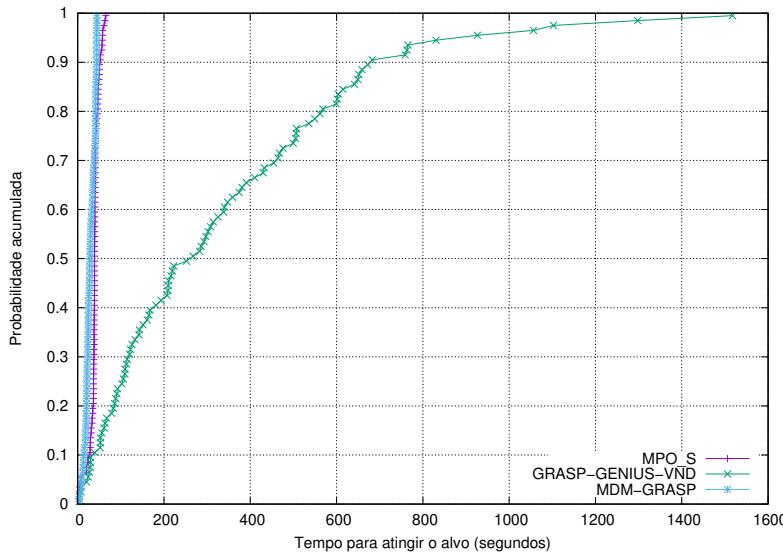


Figura 4.11: TTTplots para a instância kroA150_VT30_T30_W90_50, alvo: 12918.

probabilidade de apenas 66% para aproximadamente atingir o mesmo valor alvo.

Para o alvo mais fácil, o MDM-GRASP possui a maior probabilidade de alcançar o valor alvo em quase todos os seus tempos de execução. Os poucos casos em que isto não ocorre são entre as probabilidades de 99% e 100%, no qual o MPO_S, com estas mesmas chances, atinge o alvo em menor tempo de execução.

Para o alvo mais difícil (Figura 4.11), percebe-se um deslocamento maior das curvas que representam o MPO_S e MDM-GRASP, em relação à curva do GRASP-GENIUS-RVND. Após 561 segundos de execução, a heurística estado-da-arte atinge o alvo com uma probabilidade de 79%, enquanto que com essa mesma chance, o MDM-GRASP e MPO_S necessitam apenas de, respectivamente, 41.71 e 45.77 segundos para atingir o alvo.

4.7.1 Testes estatísticos

Para fortalecer a avaliação de desempenho dos métodos propostos, foram realizados teste de significância estatística (TSE) em todas as instâncias. Por meio da aplicação deste teste, objetiva-se verificar em termos estatísticos, se os resultados gerados pelas heurísticas indicam uma diferença estatística, reforçando a relevância da aplicação de MD e PLI, ou não.

Geralmente, este tipo de teste é utilizado para avaliar algoritmos que possuem componentes aleatórias, e identificar se a diferença entre suas médias foi devido à superioridade de algum destes algoritmos ou apenas devido à aleatoriedade presente nos métodos. Neste trabalho, foi utilizado o teste estatístico não-paramétrico de *Friedman* [92] do pacote R [85], no qual não requer que a amostra analisada siga uma distribuição normal.

Para a realização deste teste, foram considerados os custos obtidos por dez execuções de cada heurística, *p*-valor igual a 0.05 e duas hipóteses:

- Hipótese nula (H_0): Não há diferença entre as médias encontradas pelos dois métodos; e
- Hipótese alternativa (H_1): Há diferença entre as médias encontradas pelos dois métodos.

Neste sentido, H_0 será rejeitada com 95% de certeza se, para cada instância, o teste de *Friedman* [92] retornar um valor menor que o *p*-valor definido. Caso H_0 seja rejeitada, a hipótese alternativa H_1 é considerada.

Nas Tabelas 4.8 e 4.9, para cada par de heurísticas comparadas, é mostrado o número de vitórias, derrotas e empates em termos de melhor solução, média de solução e média de tempo despendido. Para a média de solução, os valores entre parênteses indicam o número de resultados estatisticamente significantes. O MDM-GRASP não foi comparado diretamente com a heurística estado-a-arte, e sim com o DM-GRASP na Tabela 4.7. A Tabela 4.8, apresenta um resumo do que pode ser deduzido a partir das Tabelas 4.6 e 4.7, em relação a uma comparação entre MDM-GRASP e GRASP-GENIUS-RVND.

Tabela 4.8: Resumo dos resultados e teste de significância estatística entre GRASP-GENIUS-RVND e MDM-GRASP.

	Melhor Sol.	Média Sol.	Média T(s)
Vitórias MDM-GRASP	13	41 (10)	105
Empates	90	50	0
Vitórias GRASP-GENIUS-RVND	4	16 (0)	2

Em termos de melhor solução, o MDM-GRASP foi superior em 13, empatou em 90 e perdeu em quatro instâncias. Em relação à média de solução, a heurística com mineração venceu em 41 casos, sendo dez estatisticamente significantes, empatou em 50 e perdeu em 16 instâncias. Para esse teste estatístico, foi observado que as significâncias obtidas concentram-se nas instâncias de maior porte, incluindo-se gr202_R39_T122_W41_25, gr229_R45_T46_W138_25, lin318_R108_T105_W105_50, rd400_R80_T240_W80_25, entre outras. Como já observado em seções anteriores, a aplicação de mineração de dados reduz o tempo computacional da heurística. Com o MDM-GRASP não foi diferente, onde essa heurística foi mais rápida que a heurística estado-da-arte em 105 das 107 instâncias testadas, e com uma redução média de 23.55% por instância.

Tabela 4.9: Resumo dos resultados e do teste de significância estatística entre GRASP-GENIUS-RVND e MPO_S.

	Melhor Sol.	Média Sol.	Média T(s)
Vitórias MPO_S	5	43 (8)	48
Empates	101	58	0
Vitórias GRASP-GENIUS-RVND	1	6 (0)	59

A Tabela 4.9 resume a comparação entre MPO_S e o GRASP-GENIUS-RVND mostrada na Tabela 4.3, com informações adicionais sobre significância estatística. A partir desta tabela, pode ser verificado que o MPO_S é superior em relação à melhor solução e à média de solução. Sobre as médias de solução, o teste de Friedman foi capaz de afirmar, com 95% de certeza, que oito vitórias do MPO_S não são devido à aleatoriedade dos métodos. Por outro lado, nada pode-se afirmar acerca da significância estatística das seis vitórias do GRASP-GENIUS-RVND.

Em relação ao tempo médio de CPU, o MPO_S perde para a heurística estado-da-arte. Para as 107 instâncias testadas nesta tese, este mesmo comportamento foi observado em ambas as hibridizações com PLI. Foi verificado que, em vários casos, o MPO_S atinge a mesma solução encontrada pelo GRASP-GENIUS-RVND em certo momento, porém, devido à necessidade do *solver* em provar a otimalidade dessa solução, o MPO_S continua a ser executado.

Uma evidência para este tipo de comportamento pode ser verificada nos TTTPlots das Figuras 4.10 e 4.11, no qual o MPO_S possui desempenho melhor quando se trata do tempo para atingir um valor alvo específico, pois o MPO_S é encerrado no momento em que este valor é atingido, sem que haja a necessidade de uma prova de otimalidade da solução.

4.8 Considerações finais

Nesta tese foi abordado o PRRCP, um problema derivado do PCV, que é NP-Difícil. Como abordagem exata para a solução do PRRCP, foi proposto: a) uma nova formulação matemática baseada em restrições *cut-sets* para eliminação de ciclos desconexos da origem; e b) desigualdades válidas para o aprimoramento da relaxação linear desta formulação. Para tornar computacionalmente viável a utilização de desigualdades violadas, também foram propostos algoritmos de separação polinomiais para identificá-las.

Seguindo a ideia de procedimentos de pré-processamento encontrados na literatura para o PRRCP, foram propostas novas sete regras de redução e uma investigação foi realizada acerca da melhor ordem de sua aplicação. Como abordagens heurísticas, foram desenvolvidas quatro novos métodos, combinando a heurística estado-da-arte com PLI (MPO_S e LB) e com conceitos de mineração de dados (DM-GRASP e MDM-GRASP).

Os resultados obtidos com a nova formulação foram muito superiores às duas formulações da literatura. Considerando o tempo limite de uma hora, o modelo proposto foi capaz de encontrar uma solução ótima para grande parte das instâncias, e provar a otimalidade para 12 soluções aproximadas da literatura. Um resultado ainda mais satisfatório foi a redução do tempo de CPU despendido pelo *solver*. Considerando o desempenho da melhor formulação da literatura (multi-fluxo), em média o tempo foi reduzido em 43%, havendo redução em 140 das 144 instâncias testadas.

Os resultados das formulações matemáticas foram aprimorados com a aplicação das regras de redução e cortes válidos, este último apenas para o modelo proposto. Com um estudo sobre a melhor ordem de aplicação das regras, foi possível aumentar o número de instâncias resolvidas em todas as três formulações. O modelo que mais se beneficiou com esta aplicação foi a formulação Multi-fluxo [93], que originalmente teve vários problemas relacionados à falta de memória, e que, com a aplicação das regras, este problema foi mitigado em várias instâncias.

Para o modelo proposto, verificou-se que de fato as desigualdades propostas fortalecem a sua relação linear, e isso permitiu que o *solver* encontrasse a solução ótima para mais 14 instâncias. Não foi possível obter uma conclusão sólida, acerca do impacto das desigualdades no tempo de processamento. Para algumas instâncias ocorre um aumento no tempo de processamento e para outras há redução, mas quando esta última acontece, geralmente é bastante significativa.

Em relação às abordagens heurísticas para o PRRCP, também foram obtidos resultados satisfatórios para todos os métodos propostos. Combinando PLI com a heurística estado-da-arte, foi possível melhorar o custo da melhor solução e a

média de solução reportada na literatura. Com a aplicação de mineração de dados no GRASP-GENIUS-RVND, houve uma melhora de desempenho ainda maior: foi possível aprimorar a melhor solução e a média de solução para um número expressivo de instâncias. Além disso, o MDM-GRASP foi capaz de realizar uma redução significativa no tempo de processamento reportado pela heurística estado-da-arte, o que concorda os vários relatos na literatura sobre o impacto da incorporação de MD em heurísticas.

Todas as quatro heurísticas propostas contribuíram para a melhoria dos resultados encontrados na literatura. Para fortalecer as conclusões e entender melhor estes resultados, foram realizados: a) testes de significância para indicar a relevância estatística das diferenças entre os resultados obtidos; e b) análises de comportamento das heurísticas em diferentes cenários. Por fim, foi concluído que a heurística que mais se destacou foi o MDM-GRASP, tanto no aprimoramento da qualidade de solução, quanto na redução do tempo despendido.

Como investigações futuras, pretende-se desenvolver uma nova heurística híbrida, combinando MD e PLI. Nesta nova heurística, provavelmente um conjunto E^0 de maior qualidade seria construído e padrões minerados poderiam ser fixados no modelo de PLI. Como mencionado na introdução, o PRRCP possui características comuns a vários outros problemas e, portanto, um trabalho futuro será adaptar as heurísticas apresentadas nesta tese para a aplicação em outros problemas de otimização.

Capítulo 5

O problema do caixeiro viajante com coleta de prêmios

Poucos problemas de otimização combinatória possuem tanta aplicabilidade como o problema do caixeiro viajante (PCV). Praticamente, qualquer situação que envolva tomadas de decisão que afetem a sequência de tarefas ou operações, possuem algum aspecto do PCV [9]. Às vezes, tais problemas são formulados como instâncias especiais do PCV, ou, mais frequentemente, tornam-se generalizações do PCV. O problema do caixeiro viajante com coleta de prêmios (PCVCP) é uma generalização do PCV e foi inicialmente proposto por BALAS [6].

O PCVCP pode ser descrito como um caixeiro viajante que deve visitar algumas cidades, sendo que cada uma delas possui um prêmio (p_i) e uma penalidade associada (w_i). O prêmio é coletado sempre que uma cidade é visitada; e a penalidade é aplicada sempre que a cidade não é visitada. Além disso, há um custo de viagem entre duas cidades (c_{ij}). O objetivo é minimizar a soma dos custos da viagem e das penalidades pagas, garantindo a coleta de um prêmio mínimo (*PRIZE*). O PCVCP possui várias aplicações práticas como no escalonamento diário de um laminador de aço [6], no planejamento de rotas de helicópteros para as plataformas de petróleo no mar [6], e na criação de rotas para turistas [9].

Como um outro exemplo prático do PCVCP, pode ser citada a entrega de um produto por um distribuidor e seus revendedores [79]. Cada revendedor possui uma determinada demanda de certo produto, e há casos em que as demandas superam a capacidade do distribuidor de atendê-los. Então, para cada revendedor que não recebe o produto, o distribuidor paga uma multa de valor variável, de acordo com o contrato firmado com cada revendedor.

A rota planejada para o veículo do distribuidor tem, por objetivo, a minimização do custo de deslocamentos entre os revendedores que receberão as entregas, e do custo relativo às multas pagas pelas entregas não realizadas. Além disso, a empresa determina que todo veículo utilizado para a distribuição tenha uma taxa percentual

mínima de atendimento, em relação à demanda total dos revendedores (*PRIZE*).

5.1 Revisão da literatura

Nesta seção, é realizada uma revisão da literatura acerca do PCVCP, com apresentação de formulações matemáticas, métodos exatos e heurísticos. Inicialmente são relatados, de maneira cronológica, todos os trabalhos relacionados com esse problema e, nas próximas seções, os algoritmos são apresentados com mais detalhes.

A ideia do PCVCP surgiu originalmente com BALAS e MARTIN [7]. Esses autores desenvolveram um *software* para apoiar a programação da laminação de uma usina de bobinas de aço. Anos depois, essa programação serviria como base para a definição do PCVCP.

Três anos após o trabalho de BALAS e MARTIN [7], FISCHETTI e TOTH [39] propuseram: i) dois limites inferiores baseados em relaxação Lagrangeana e um terceiro baseado na disjunção de dois subproblemas; e ii) utilizam um algoritmo *branch-and-bound* para resolver um conjunto de instâncias de médio porte (até 200 nós). Para tais propostas, os autores introduziram uma formulação inédita para o PCVCP, na qual a eliminação de subciclos é realizada por meio de restrições do tipo *cutset*.

No ano seguinte, BALAS [6] introduziu o PCVCP de forma definitiva na literatura. Em seu trabalho, o autor discutiu propriedades estruturais do problema, e propôs várias desigualdades, relacionadas ao PCV clássico e ao problema da mochila. Ainda nesse trabalho, BALAS [6] apresentou duas formulações matemáticas para o PCVCP, mas nenhuma dessas formulações deixa explícito como é feita a eliminação de subciclos na solução.

As primeiras soluções não exatas para o PCVCP surgiram em BIENSTOCK *et al.* [13], onde foi proposto um algoritmo aproximativo com limite constante. Esse algoritmo é uma combinação da relaxação linear do problema, com o algoritmo de GOODRICH e TAMASSIA [50], amplamente utilizado para resolução do PCV. Como resultado, o algoritmo apresentou nos piores casos uma aproximação de 2,5 vezes a solução ótima do problema.

DELL'AMICO *et al.* [27] fizeram um resumo acerca dos principais resultados da literatura, propuseram um limite inferior para PCVCP assimétrico baseado em relaxação lagrangeana, e realizaram seus testes em instâncias geradas aleatoriamente como em FISCHETTI e TOTH [39].

DELL'AMICO *et al.* [28] desenvolveram uma heurística Lagrangeana para obter um limite superior na forma de uma solução, por meio da relaxação da restrição de prêmio mínimo e aplicando o método de sub-gradientes para resolver o problema dual. As relaxações foram utilizadas em conjunto com a clássica heurística de inserção

mais barata, obtendo um método que, a partir de uma solução relaxada, alcançava uma solução viável. Nesse método, a inserção mais econômica é realizada a cada iteração até que o prêmio mínimo seja coletado. Por fim, propuseram uma heurística de refinamento chamada de Extensão e Colapso (*Extension and Collapse Heuristic*).

AWERBUCH *et al.* [5] elaboraram um algoritmo aproximativo de complexidade ($\log_2 n$) para o PCV com Quota, que com algumas transformações, também contemplava o PCVCP. No PCV com Quota, cada cidade possui um prêmio e o caixeiro deve encontrar a menor rota possível tal que cole uma quota (prêmio mínimo). A diferença desse problema para o PCVCP é que, nesse último a função objetivo não é somente minimizar o custo da rota, mas também as penalidades de cidades não visitadas. Ou seja, o PCV com Quota é um PCVCP com todas as penalidades iguais a zero. O algoritmo proposto por esses autores, foi baseado em um algoritmo aproximativo para o problema *k-Minimum-Spanning-Tree* (k-MST), no qual cada vértice $v_i \in V$ é substituído por cópias p_i (prêmio de v_i) de si mesmos e no mesmo local. E então, o método aproximativo do k-MST é executado nesse grafo, com $k = PRIZE$.

GOMES *et al.* [49] apresentaram uma heurística híbrida que combina GRASP [31] com a heurística de busca VNS [73]. Seus experimentos foram realizados em instâncias próprias criadas a partir da TSPLIB [86]. Até o momento, o autor desta tese desconhece a disponibilização pública de tais instâncias.

Em TORRES e BRITO [95] são abordadas duas variantes do PCV, o PCVCP e o *Orienteering Problem* (OP). Para ambos os problemas, foram propostas novas formulações matemáticas e heurísticas. Em termos de modelagem exata, os autores apresentaram uma nova formulação baseada em BALAS [6] que faz uso de restrições de fluxo. Em termos de heurísticas, foram apresentadas apenas métodos para o OP, que são combinações de GRASP e VNS. De acordo com os autores, as propostas para o PCVCP ainda estariam em desenvolvimento. As estratégias propostas foram avaliadas a partir de um conjunto de instâncias criadas em BALAS [6], e não foram disponibilizadas publicamente.

MELO e MARTINHON [72] também apresentaram uma heurística híbrida que combina a metaheurística GRASP com VNS. Porém, uma versão diferente do GRASP, o GRASP-Progressivo, no qual o valor do parâmetro α é incrementado durante a execução do método e, como consequência, ocorre um aumento progressivo de soluções viáveis iniciais. As mesmas instâncias de GOMES *et al.* [49] foram utilizadas para validar a proposta desse trabalho. Contudo, não foi realizada nenhuma comparação entre a proposta desse trabalho com o método de GOMES *et al.* [49].

No trabalho de CHAVES *et al.* [17] foi apresentado novamente um método híbrido baseado em GRASP, VNS e VND para obter soluções aproximadas do PCVCP. Esse trabalho se diferencia do desenvolvido por GOMES *et al.* [49] pelo

fato de combinar as heurísticas VNS e VND, e se diferencia do trabalho de MELO e MARTINHON [72] por utilizar o GRASP básico com filtro, aplicando a busca local apenas na melhor solução construída, e também por trabalhar com um número mais elevado de vizinhanças para os métodos VNS e VND. Ainda nesse último trabalho, uma adaptação da formulação matemática de BALAS [6] também foi proposta, introduzindo dois conjuntos de restrições de fluxo para evitar sub-rotas na solução. Para a validação dos seus métodos, os autores realizaram experimentos com um novo conjunto de instâncias, disponibilizados em <http://www.decom.ufop.br/prof/marcone/Orientacoes/OrientacoesConcluidas.htm>. Na comparação dos resultados encontrados na modelagem exata e na heurística, verificou-se que os algoritmos heurísticos sempre atingiram a solução ótima para as instâncias em que o ótimo global é conhecido.

CHAVES e LORENA [16] propuseram uma heurística chamada ECS (*Evolutionary Clustering Search*) e outra CS*, que é uma adaptação da primeira. Na segunda heurística, a componente evolucionária foi substituída pelas metaheurísticas GRASP e VNS. Ambas as propostas fizeram uso de conceitos de algoritmos híbridos, combinando metaheurísticas com um processo de clusterização, detectando áreas de busca promissoras. Sempre que uma área era considerada promissora, alguma estratégia agressiva de busca era realizada nessa área. CHAVES e LORENA [16] apresentaram uma nova formulação matemática baseada nos modelos de BALAS [6], e TORRES e BRITO [95]. Para a validação das propostas, CHAVES e LORENA [16] realizaram experimentos com o conjunto de instâncias de CHAVES *et al.* [17]. Os resultados obtidos pelo ECS e CS* mostraram-se competitivos com o modelo exato, para instâncias de até 51 nós, nas quais foram alcançadas as soluções ótimas. Dentre as duas heurísticas, a CS* obteve melhor desempenho em termos de qualidade de solução e tempo de processamento.

CHAVES e LORENA [19] apresentaram uma heurística CS que faz uso do GRASP e VND para a geração de soluções. Os autores conduziram experimentos em um novo conjunto de 108 instâncias, geradas a partir de diferentes valores de *PRIZE*, e diferentes intervalos de penalidades e custos de aresta. Essas instâncias estão disponibilizadas em <http://www.lac.inpe.br/~lorena/antonio/instancias-pctsp.zip>. Os resultados mostraram que o CS é bem competitivo com o modelo exato para instâncias de até 100 nós. Para o restante das instâncias, o CS foi superior tanto em qualidade de solução, quanto em tempo de processamento. Também foi realizada uma comparação entre o CS e o GRASP/VNS (sem CS), e a primeira heurística foi superior para quase todas as instâncias.

GOEMANS [47] apresentou um algoritmo 1,91457-aproximativo para o PCVCP, que foi obtido combinando uma variante randomizada do algoritmo de arredondamento de BIENSTOCK *et al.* [13], com um algoritmo primal-dual de GOEMANS e

WILLIAMSON [46].

BÉRUBÉ *et al.* [12] propuseram um algoritmo *branch-and-cut* para o PCVCP e algumas desigualdades adaptadas do *Orienteering Problem*. Contudo, os autores trataram o PCVCP sem considerar as penalidades dos vértices não visitados. Essa versão do PCVCP é conhecida em vários trabalhos da literatura como o PCV com Quota.

Por fim, o último trabalho que aborda o PCVCP foi realizado por PEDRO *et al.* [80], no qual os autores propuseram uma resolução para o problema por meio de uma busca tabu (BT), que incorpora a heurística GENIUS à sua fase de construção, e uma busca local 2-opt. Para validar essa proposta, os autores fizeram experimentos utilizando as mesmas instâncias de CHAVES e LORENA [16] e compararam seus resultados com a heurística CS. A abordagem CS obteve melhores resultados que a BT para instâncias com até 100 nós, contudo, para as instâncias de maior porte, a BT foi superior.

Nesta tese, são apresentadas com mais detalhes, as melhores heurísticas da literatura nas quais os autores disponibilizaram as instâncias utilizadas em seus experimentos. Na literatura, têm-se cronologicamente os trabalhos de CHAVES *et al.* [17], CHAVES e LORENA [16], CHAVES e LORENA [19], e PEDRO *et al.* [80]. Dentre esses quatro trabalhos, os dois primeiros e o último utilizaram as mesmas instâncias. Em qualidade de solução e tempo de processamento, a heurística CS* de CHAVES e LORENA [16] obteve melhor desempenho em relação às demais. Dessa forma, considerou-se o CS* [16] como sendo a heurística estado-da-arte para o primeiro grupo de instâncias. Já para o segundo grupo de problemas-teste disponibilizado, há apenas experimentos com a heurística CS de CHAVES e LORENA [19]. Portanto, essa foi também considerada heurística estado-da-arte.

5.2 Clustering Search

Uma vez que as melhores heurísticas da literatura são baseadas no método *clustering search* (CS), nessa seção são apresentados mais detalhes dessa abordagem. O CS surgiu como uma proposta genérica que combina metaheurísticas de busca com clusterização, no intuito de detectar áreas promissoras de busca, antes de efetuar uma busca local [75].

Para identificar áreas promissoras, são geradas soluções por uma metaheurística e então essas soluções são clusterizadas em grupos, que posteriormente serão exploradas por heurísticas de busca local [75]. Uma definição mais precisa do CS é apresentada por CHAVES *et al.* [18]: um método híbrido que objetiva combinar metaheurísticas e busca local onde a busca é intensificada apenas em áreas do espaço de busca que mereçam atenção especial.

O CS pode ser separado em quatro partes conceitualmente independentes: a) a metaheurística de busca (SM), b) o componente iterativo de clusterização (IC), c) o módulo analisador (AM), e d) o buscador local (LS). O componente SM é considerado por seus autores como um gerador de soluções diversificadas, que funciona por tempo integral no método CS. O componente IC agrupa soluções similares em *clusters* (grupos), em que cada *cluster* possui um centro que o representa. Uma métrica de distância Δ deve ser pré-definida, que funciona como uma medida de similaridade para o processo de agrupamento.

O componente AM examina cada grupo, em intervalos regulares, indicando um provável *cluster* promissor. Uma densidade de *cluster*, δ , é uma medida que indica o nível de atividade dentro de um *cluster*. Para simplificar, δ_j pode contabilizar o número de soluções geradas por SM e agrupadas no *cluster* c_j . Sempre que δ_j atinge um certo limite δ_{max} , significando que algum modelo de informação torna-se predominantemente gerado por SM, o *cluster* c_j deve ser investigado. Grupos com altos δ_j podem ser mantidos ao longo das iterações, enquanto que *clusters* com baixos δ_j podem ser eliminados ou perturbados, como parte de um mecanismo que permite a criação posterior de outros *clusters*. A eliminação de um grupo não afeta o conjunto de soluções no SM. Apenas o *cluster* é considerado irrelevante para o método. Por fim, o componente LS funciona como um módulo buscador que provê a exploração de uma suposta área promissora, identificada por um *cluster*.

O Algoritmo 12 apresenta um pseudo-código genérico da heurística *Clustering Search* para um problema de minimização. Na linha 1, são criados β *clusters* iniciais, e a quantidade gerada depende da estratégia utilizada. Esse processo inicial não necessariamente precisa ser realizado; nesse caso, *clusters* são gerados ao longo do algoritmo. Das linhas 2 a 14, enquanto um critério de parada não for satisfeito, as seguintes instruções são executadas. Na linha 3, uma solução s_k é gerada pela SM e então é passada para o IC que, por sua vez, tenta agrupá-la em algum *cluster* c_j , de acordo com Δ (linha 4). Dependendo da implementação, caso s_k seja considerada suficientemente nova, um novo *cluster* c_{new} é criado, e s_k torna-se o seu centro.

Na linha 5, s_k é inserida em c_j e então a densidade desse *cluster* é atualizada (linha 6). O passo seguinte consiste numa fase de assimilação de c_j (linha 7), em que c_j é atualizado com a nova informação recebida (s_k). Na literatura, os processos de assimilação mais conhecidos são: i) simples: um movimento determinístico de c_j em direção a s_k ; ii) crossover: similar ao crossover de algoritmos genéticos, soluções são aleatoriamente geradas, a partir da combinação de c_j e s_k ; e iii) caminho: similar ao *path-relinking*, busca por soluções no caminho de c_j a s_k . O processo de assimilação é finalizado quando c_j é substituído pela melhor solução encontrada no processo.

Caso um *cluster* esteja denso o suficiente (linhas 8 a 12), uma busca local é executada nele, e caso uma melhor solução seja encontrada, o seu centro é atualizado.

Em seguida, na linha 13, a melhor solução corrente é atualizada. Por fim, ao atingir o critério de parada do algoritmo, a melhor solução s é retornada como resultado na linha 15.

Algoritmo 12: CS(β, δ_{max})

- 1: Gerar β clusters iniciais por meio de uma heurística geradora de soluções
 - 2: **Enquanto** critério de parada não satisfeito **faça**
 - 3: $s_k \leftarrow$ gerarSolução()
 - 4: encontrar o cluster c_j mais similar a s_k
 - 5: inserir s_k em c_j
 - 6: $\delta_j \leftarrow \delta_j + 1$
 - 7: $c_j \leftarrow$ Assimilação(c_j, s_k)
 - 8: **Se** $\delta_j \geq \delta_{max}$ **então**
 - 9: $\delta_j \leftarrow 0$
 - 10: $s_{bl} \leftarrow$ BuscaLocal(c_j)
 - 11: $c_j \leftarrow \min(s_{bl}, c_j)$
 - 12: **Fim-se**
 - 13: $s \leftarrow \min(s, c_j)$
 - 14: **Fim-enquanto**
 - 15: **Retorne** s
-

5.3 CS aplicada ao PCVCP

Os melhores resultados para as instâncias disponibilizadas, foram obtidos por duas heurísticas baseadas em Clustering Search (CS). Esse método foi inicialmente proposto por CHAVES e LORENA [16], no qual são apresentadas duas heurísticas híbridas, sendo CS* a que obteve melhor desempenho e detentora dos melhores resultados.

5.3.1 A metaheurística GRASP/VNS

Na heurística CS*, tem-se um componente SM composto pela combinação de um GRASP com VNS. A fase de construção do GRASP utiliza a regra de inserção do procedimento *Adding-Nodes* [28] para construir a lista de candidatos (LC). Inicialmente dois nós são adicionados na rota e, então, cada elemento é selecionado de maneira aleatória a partir de uma lista restrita de candidatos (LRC), contendo os melhores candidatos de LC. Esse elemento é adicionado à solução e a lista de candidatos é atualizada. A fase de construção termina quando não existem candidatos com custos positivos e o prêmio mínimo esteja coletado.

A fase de busca local do GRASP utiliza o VNS [73], que explora vizinhanças distantes da atual solução incumbente, e move-se para uma nova vizinhança se e somente se uma melhoria foi feita. Inicialmente um conjunto de estruturas de vizinhanças é definido por meio de movimentos aleatórios. O VNS proposto por CHAVES e LORENA [16] implementa nove estruturas de vizinhanças, por meio dos seguintes movimentos:

- inserir dois nós na rota;
- remover dois nós da rota;
- trocar quatro nós da rota;
- inserir e remover um nó da rota;
- remover três nós da rota;
- remover um nó e trocar quatro nós da rota.

Iniciando a partir da solução corrente, a cada iteração, uma solução vizinha é aleatoriamente selecionada na k -ésima vizinhança da solução incumbente. Essa solução vizinha é então submetida a algum método de busca local. Se a solução obtida é melhor que a incumbente, então a incumbente é atualizada e a busca continua a partir da primeira estrutura de vizinhança. Caso contrário, a busca continua para a próxima vizinhança. Esse procedimento é interrompido quando o tempo sem melhorias ultrapassa 100 segundos [16].

Em seu trabalho, CHAVES e LORENA [16] utilizaram o método Variable Neighborhood Descent (VND) [73] como uma busca local para o VNS, e é composto por três métodos diferentes de melhoria, que combinam dois movimentos, *Add-step* e *Drop-step* [49]. O movimento *Add-step* consiste na adição de um nó que provê o melhor valor da função de adição. O movimento *Drop-step* consiste na remoção de um nó da rota que provê o melhor valor na função de remoção. Em ambos os movimentos, se os valores das funções forem positivos, então a função objetivo será melhor após o movimento. O aspecto principal a ser observado é que todos os movimentos são executados preservando a viabilidade. Assim, os três métodos de melhoria do VND são os seguintes:

- *SeqDrop*: aplicar uma sequência de movimentos *Drop-step* enquanto o valor da função objetivo estiver sendo decrementado;
- *AddDrop*: aplicar um movimento *Add-step* e um movimento *Drop-step*;
- *SeqAdd*: aplicar uma sequência de movimentos *Add-step* enquanto o valor da função objetivo estiver sendo decrementado.

A sequência de movimentos utilizada no VND de CHAVES e LORENA [16] foi a seguinte: 1) *SeqDrop-SeqAdd*: aplicação do *seqDrop* seguida da aplicação do *SeqAdd*; 2) 2-Opt [53]; e 3) *AddDrop*. Sempre que um desses métodos de melhoria obter uma solução melhor, o VND retorna para o primeiro método de melhoria. O critério de parada estabelecido para o VND foi até que não haja mais melhorias na solução, e a última vizinhança ter sido alcançada.

5.3.2 O processo de clusterização

O componente IC é o núcleo do método CS, funcionando como um classificador, mantendo apenas informações relevantes, e dirigindo a intensificação da busca em áreas promissoras do espaço de busca. Primeiramente, um número máximo de clusters (NC) é pre-definido. O i -ésimo *cluster* possui seu próprio centro c_i e um raio r que é idêntico para os outros *clusters*. Soluções geradas pelo GRASP/VNS são passadas para o IC que tenta agrupá-las como informação conhecida, de acordo com uma métrica de distância. A informação ativa o centro de *cluster* mais próximo C_i (centro de *cluster* que minimiza a métrica de distância), causando um tipo de perturbação nele. Nesse trabalho, a métrica de distância foi o número de arestas diferentes entre a solução e o centro de *cluster*. Quanto maior o número de arestas diferentes, maior a dissimilaridade entre eles.

A perturbação significa um processo de assimilação da nova solução, no qual o centro do *cluster* é atualizado pela nova solução gerada. Em seu trabalho, CHAVES e LORENA [16] utilizaram o método *Path Relinking* [45], que gera vários pontos (soluções) contidos no caminho que conecta a solução gerada pelo GRASP/VNS e o centro do *cluster*. Uma vez que cada ponto é avaliado pela função objetivo, o processo de assimilação por si só é um mecanismo de intensificação de solução dentro do *cluster*. O novo centro C_i é a solução melhor avaliada obtida no caminho.

O componente AM é executado sempre que uma solução é atribuída a um *cluster*, verificando se o *cluster* pode ser considerado promissor. Um *cluster* torna-se promissor quando a densidade δ_i atinge um certo limite, dado por:

$$\delta_i \geq PD \times \frac{NS}{|C_i|} \quad (5.1)$$

no qual, NS é o número de soluções geradas no intervalo de análise dos *clusters*, PD é uma constante que representa a densidade de *cluster* desejável além da densidade normal, obtida se NS for igualmente dividida para todos os clusters, e $|C_i|$ é a cardinalidade do *cluster* i . Nesse trabalho, o AM também possui a função de “resfriar” todos os *clusters*, diminuindo a sua densidade, e eventualmente eliminando *clusters* com baixa densidade.

Por fim, o componente LS foi implementado por uma heurística *2-opt* [53]. Essa heurística é baseada no re-sequenciamento da rota que sempre resulta em uma solução de igual ou melhor qualidade, uma vez que os custos das viagens podem ser decrementados, enquanto as penalidades e prêmios coletados são mantidos. Nesse trabalho, o método *2-opt* consiste em realizar duas mudanças na rota, deletando dois arcos e os substituindo por dois outros arcos para formar uma nova rota. O LS continua enquanto houver aprimoramento da rota. O pseudo-código do método CS* é apresentado no Algoritmo 13.

Algoritmo 13: CS*(k_{max}, α)

```
1:  $s^* \leftarrow \emptyset; f(s^*) \leftarrow \infty$ 
2: {Componente SM}
3: Para número de iterações não atingido faca
4:    $s \leftarrow \emptyset$ 
5:   {Fase de construção do GRASP/VNS}
6:   Enquanto (Solução não construída) faca
7:     computar lista de candidatos (LC)
8:      $LRC \leftarrow \text{criarLRC}(LC, \alpha)$ 
9:      $e \leftarrow$  vértice aleatoriamente selecionado de LRC
10:     $s \leftarrow s \cup e$ 
11:   Fim-enquanto
12:   {Fase de busca local do GRASP/VNS}
13:   Enquanto Critério de parada não satisfeito faca
14:      $k \leftarrow 1$ 
15:     Enquanto  $k \leq k_{max}$  faca
16:       Gerar aleatoriamente  $s' \in N^k(s)$ 
17:        $s'' \leftarrow$  Aplicar VND com  $s'$  como ponto inicial
18:       Se  $f(s'') < f(s)$  então
19:          $s \leftarrow s''$ 
20:        $k \leftarrow 1$ 
21:     Senão
22:        $k \leftarrow k + 1$ 
23:     Fim-se
24:   Fim-enquanto
25:   Fim-enquanto
26:   {Componente IC}
27:   Calcular a distância da solução do GRASP/VNS ( $s$ ) para os clusters
28:   Inserir  $s$  no cluster mais similar ( $C_i$ )
29:   Aplicar o processo de assimilação -  $\text{path\_relinking}(s, C_i)$ 
30:   {Componente AM}
31:   Verificar se o cluster pode ser considerado promissor. Se sim, o componente  $LS$  é aplicado nele
32:   {Componente LS}
33:   Aplicar a heurística  $2\text{-opt}$  no cluster promissor
34:    $s^* \leftarrow \min(s^*, C_i)$ 
35: Fim-para
36: Retorne  $s^*$ 
```

A única abordagem heurística com resultados para as instâncias de CHAVES e LORENA [19], é a própria heurística apresentada nesse trabalho (denominada apenas como CS). CHAVES e LORENA [19] realizaram aprimoramentos na heurística CS* apresentada em CHAVES e LORENA [16]. Nessa nova heurística, novas vizinhanças do VND foram propostas, a fase construção do GRASP foi realizada de maneira diferente e um novo limite de densidade dos *clusters* é utilizado. Os novos movimentos

do VND são:

- m1: adiciona um nó à rota;
- m2: remove um nó da rota;
- m3: troca a posição de dois nós;
- m4: adiciona dois nós à rota;
- m5: remove dois nós da rota;
- m6: troca a posição de quatro nós da rota;
- m7: adiciona três nós à rota;
- m8: remove três nós da rota;
- m9: troca a posição de seis nós da rota.

Na fase de construção do GRASP utilizado em CHAVES e LORENA [19], a regra de inserção para adicionar um nó k entre os nós i e j é:

$$g(k) = c_{ij} + \gamma_k - c_{ik} - c_{kj} \quad (5.2)$$

que é composta pelo custo do arco (i, j) , a penalidade do nó k e custos dos arcos (i, k) e (k, j) , respectivamente. No componente AM de CHAVES e LORENA [16] o limite da densidade que indica o quanto promissor é um *cluster* foi substituído por

$$\delta_i \geq PD \times \frac{NS}{|Clus|} \quad (5.3)$$

no qual $|Clus|$ é o número total de *clusters* no momento da aplicação do componente AM. Com exceção das alterações supracitadas, a heurística CS [19] é idêntica ao método de CHAVES e LORENA [16] e, portanto, o restante dessa heurística não é descrito novamente.

5.4 Formulações matemáticas

A maioria das formulações matemáticas de PLI propostas para o PCVCP, são baseadas em estudos anteriores acerca do PCV. A seguir, serão apresentados todos os modelos da literatura utilizados em experimentos com as instâncias disponíveis, além de formulações nas quais pode ser aplicado o procedimento de separação proposto nesta tese.

5.4.1 Direccionadas

Na literatura, para os casos em que os modelos são direcionados, são associadas variáveis binárias x_{ij} para cada arco $(i, j) \in A$, no qual x_{ij} recebe valor um se e somente se o respectivo arco é utilizado na solução; a variável binária y_i é associada a cada vértice $i \in V$, com valor igual a um se e somente se o respectivo vértice é visitado e valor zero caso contrário; e a variável $z_{ij} \geq 0$ corresponde a quantidade de fluxo escoado no arco (i, j) .

Para ambos os casos de modelo (direcionado e não-direcionado) o vértice $i = 1$ representa a origem da rota, e são utilizadas as seguintes contantes: w_i como sendo a penalidade de cada vértice $i \in V$, com $w_1 = \infty$; p_i representando o prêmio associado a cada vértice $i \in V$, com $p_1 = 0$; $prize(V)$ como sendo a soma dos prêmios de todos os vértices em V ; c_{ij} o custo de cada arco $(i, j) \in A$; e $PRIZE$ a quantidade mínima de prêmio que precisa ser coletada. A seguir são apresentadas três formulações da literatura para o PCVCP, respectivamente dos autores TORRES e BRITO [95], CHAVES *et al.* [17] e CHAVES e LORENA [16].

Torres e Brito

$$\text{Minimizar} \quad \sum_{i \in V} \sum_{j \in V \setminus \{i\}} c_{ij} x_{ij} + \sum_{i \in V} \gamma_i (1 - y_i) \quad (5.4)$$

$$s.a.: \sum_{j \in V \setminus \{i\}} x_{ji} = y_i \quad \forall i \in V \quad (5.5)$$

$$\sum_{j \in V \setminus \{i\}} x_{ij} = \sum_{j \in V \setminus \{i\}} x_{ji} \quad \forall i \in V \quad (5.6)$$

$$\sum_{i \in V \setminus \{1\}} p_i y_i \geq PRIZE \quad (5.7)$$

$$\sum_{j \in V \setminus \{1\}} z_{1j} = 0 \quad (5.8)$$

$$\sum_{j \in V \setminus \{i\}} z_{ij} = \sum_{j \in V \setminus \{i\}} z_{ji} + p_i y_i \quad \forall i \in V \setminus \{1\} \quad (5.9)$$

$$\sum_{j \in V \setminus \{1\}} z_{j1} - \sum_{j \in V \setminus \{1\}} p_j y_j = 0 \quad (5.10)$$

$$z_{ij} \geq x_{ij} \quad \forall i \in V \setminus \{1\}, \forall j \in V, i \neq j \quad (5.11)$$

$$prize(V) \geq z_{ij} \quad \forall i \in V \setminus \{1\}, \forall j \in V, i \neq j \quad (5.12)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A, i \neq j \quad (5.13)$$

$$y_i \in \{0, 1\} \quad \forall i \in V \quad (5.14)$$

Segundo TORRES e BRITO [95], a função objetivo (5.4) visa minimizar o somatório dos custos da viagem e as penalidades. As restrições (5.5) garantem que cada vértice i , ou não é visitado (no caso de nenhuma aresta entrar ou sair do vértice i), ou é visitado uma única vez. As restrições (5.6) representam as equações de conservação de fluxo das variáveis x_{ij} . As restrições 5.7 demandam que a soma dos prêmios coletados não deve ser menor que o *PRIZE* estipulado. As restrições (5.8) a (5.10) garantem a conectividade da solução, em que cada vértice da solução pode ser alcançado a partir da origem, ou seja, são as restrições que unidas às restrições (5.5) e (5.6), evitam a formação de rotas desconexas da origem. As variáveis de fluxo z_{ij} são definidas de forma a representar a quantidade de prêmio escoada no arco (i, j) . As restrições (5.11) e (5.12) conectam as variáveis x_{ij} e z_{ij} , de forma que ambas gerem a mesma rota. Por fim, o domínio das variáveis do modelo são determinados pelas restrições (5.13) e (5.14).

Chaves et al.

$$\text{Minimizar} \quad \sum_{i \in V} \sum_{j \in V \setminus \{i\}} c_{ij} x_{ij} + \sum_{i \in V} \gamma_i (1 - y_i) \quad (5.15)$$

$$\text{s.a. : } \sum_{j \in V \setminus \{i\}} x_{ij} = y_i \quad \forall i \in V \quad (5.16)$$

$$\sum_{i \in V \setminus \{j\}} x_{ij} = y_j \quad \forall j \in V \quad (5.17)$$

$$\sum_{i \in V \setminus \{1\}} p_i y_i \geq \text{PRIZE} \quad (5.18)$$

$$\sum_{j \in V \setminus \{i\}} z_{ij} - \sum_{j \in V \setminus \{i\}} z_{ji} = y_i \quad \forall i \in V \setminus \{1\} \quad (5.19)$$

$$z_{ij} \leq (|V| - 1)x_{ij} \quad \forall (i, j) \in A \quad (5.20)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (5.21)$$

$$y_i \in \{0, 1\} \quad \forall i \in V \quad (5.22)$$

$$f_{ij} \geq 0 \quad \forall (i, j) \in A \quad (5.23)$$

A função objetivo da formulação de CHAVES *et al.* [17] (5.15) é idêntica a de TORRES e BRITO [95]. As restrições (5.16) garantem que se o vértice i estiver na rota, o somatório dos arcos que saem dele devem ser igual a um, e zero caso contrário; as restrições (5.17) demandam que se o vértice j estiver na rota, o somatório dos arcos que chegam nele devem ser igual a um, e zero caso contrário. As restrições (5.18) asseguram que o prêmio coletado na rota será maior ou igual ao prêmio mínimo pré-estabelecido. As restrições (5.19) e (5.20) foram propostas em CHAVES *et al.* [17], para que, unidas às restrições (5.16) e (5.17), garantam a eliminação de sub-rotas. As restrições (5.19) demandam que a diferença do fluxo que sai pelo o fluxo que entra no vértice i seja igual a 1, caso o vértice i esteja na solução, e 0 caso contrário. As restrições (5.20) asseguram que o fluxo máximo que pode passar por uma aresta é $(n - 1)$, pois, uma vez que se tem n vértices, o número máximo de arestas possíveis em uma solução é $n - 1$. As restrições (5.21), (5.22) e (5.23) garantem a integralidade e não-negatividade das variáveis do modelo.

Chaves e Lorena

As variáveis dessa formulação diferem das variáveis apresentadas nos modelos anteriores. CHAVES e LORENA [16] introduziram variáveis binárias $x_{ij} = 1$, se o arco (i, j) pertence à solução e 0 caso contrário, $x_{ii} = 1$ se o nó i não é visitado e 0 caso contrário.

$$\text{Minimizar} \quad \sum_{i \in V} \sum_{j \in V} b_{ij} x_{ij} \quad (5.24)$$

$$\text{s.a. : } \sum_{j \in V} x_{ij} = 1 \quad \forall i \in V \quad (5.25)$$

$$\sum_{j \in V} x_{ji} = 1 \quad \forall i \in V \quad (5.26)$$

$$\sum_{i \in V} p_i x_{ii} \leq \text{prize}(V) - \text{PRIZE} \quad (5.27)$$

$$\sum_{i \in V \setminus \{1\}} f_{1i} = 0 \quad (5.28)$$

$$\sum_{j \in V \setminus \{i\}} f_{ij} = \sum_{j \in V \setminus \{i\}} f_{ji} + p_i(1 - x_{ii}) \quad \forall i \in V \setminus \{1\} \quad (5.29)$$

$$\sum_{j \in V \setminus \{1\}} f_{j1} = \sum_{j \in V \setminus \{1\}} p_j(1 - x_{jj}) \quad (5.30)$$

$$f_{ij} \geq x_{ij} \quad \forall i \in V \setminus \{1\}, \forall j \in V, i \neq j \quad (5.31)$$

$$prize(V)x_{ij} \geq f_{ij} \quad \forall i \in V \setminus \{1\}, \forall j \in V, i \neq j \quad (5.32)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V \quad (5.33)$$

$$f_{ij} \geq 0 \quad \forall i, j \in V, i \neq j \quad (5.34)$$

Segundo CHAVES e LORENA [16], na função objetivo do seu modelo, $b_{ij} = c_{ij}$ se $i \neq j$, e $b_{ij} = \gamma_i$ caso contrário. As restrições (5.25) e (5.26) são as bem conhecidas restrições de atribuição. A restrição (5.29) garante que o prêmio requerido será coletado. As restrições (5.30) às (5.32) forçam cada nó visitado a ser conectado com o vértice origem, e unidas às restrições de atribuição asseguram a eliminação de sub-rotas. As variáveis de fluxo f_{ij} são utilizadas para prevenir sub-rotas, atribuindo o prêmio do nó visitado aos arcos que saem desse nó. Por fim, as restrições (5.33) e (5.34) determinam os domínios das variáveis.

5.4.2 Não direcionadas

Para as formulações não-direcionadas, seja um subconjunto de vértices S , e $\delta(S)$ um conjunto de arestas com um extremo em S e o outro em $V \setminus S$. Considerando as formulações da literatura, um *framework* pode ser definido como:

$$\text{Minimizar} \quad \sum_{e \in E} c_e x_e + \sum_{i \in V} \gamma_i (1 - y_i) \quad (5.35)$$

$$\text{s.a. : } \sum_{e \in \delta(\{i\})} x_e = 2y_i, \quad \forall i \in V \quad (5.36)$$

$$\sum_{i \in V} p_i y_i \geq PRIZE \quad (5.37)$$

$$\text{Restrições de eliminação de sub-ciclos} \quad (5.38)$$

$$x_e \in \{0, 1\}, \quad \forall e \in E \quad (5.39)$$

$$y_i \in \{0, 1\}, \quad \forall i \in V \quad (5.40)$$

Um das maiores preocupações das formulações não-direcionadas do PCVCP, são as restrições de eliminação de sub-ciclos (5.38). Para os modelos não direcionados, BIENSTOCK *et al.* [13] propuseram as restrições (5.38) como sendo:

$$\sum_{e \in \delta(S)} x_e \geq 2y_i, \quad \forall (S \subset V \setminus \{1\}, i \in S) \quad (5.41)$$

Enquanto que LEIFER e ROSENWEIN [68] propuseram:

$$\sum_{e \in S} x_e \leq |S| - 1, \quad \forall (S \subset V \setminus \{1\}, 3 \leq |S| \leq n - 2) \quad (5.42)$$

Capítulo 6

Contribuições para a resolução do PCVCP

Neste capítulo, são apresentados os métodos propostos para a solução do PCVCP. Em termos de solução exata, a proposta desta tese para o PCVCP é a aplicação inédita do modelo de BIENSTOCK *et al.* [13] para instâncias da literatura, com a adição do seguinte conjunto de restrições:

$$x_e \leq y_v, \quad \forall e = (i, j) \in E, v \in \{i, j\} \quad (6.1)$$

que associa as variáveis de cada aresta na rota aos seus respectivos vértices; se uma aresta está na rota, então seus vértices também devem estar. A esse novo modelo com a adição das restrições supracitadas, denominamos de Bienstock+. Em seu trabalho, BIENSTOCK *et al.* [13] não realizaram experimentos computacionais com seu modelo, e também não apresentaram com detalhes o processo de separação de suas *cutsets* (5.41). Para separar tais restrições, foi aplicado um procedimento análogo ao utilizado no PRRCP para a separação das *cutsets* (3.7), (3.8) e (3.9).

Para o procedimento de separação, considere $G^* = (V^*, E^*)$ como sendo o grafo associado à solução $S = (x^*, y^*)$, obtida por meio de uma relaxação linear do modelo Bienstock⁺; $V^* = \{i \in V : y_i^* > 0\}$ e $E^* = \{e \in E : x_e^* > 0\}$. Considerando x_e^* como sendo o valor da capacidade de cada aresta $e \in E^*$, foi executado um algoritmo de corte-mínimo/fluxo-máximo [21] entre o vértice 1 e todo $i \in V^* \setminus \{1\}$. Caso o resultado do corte seja menor que $2y_i$, então tem-se que a restrição 5.41 é violada e, portanto, adicionada ao modelo para reotimização.

Em relação a soluções aproximadas, é proposta uma heurística híbrida semelhante ao MPO_S desenvolvido para o PRRCP. Essa nova heurística será descrita a seguir.

6.1 Heurística

Para o segundo problema abordado nesta tese, uma nova heurística híbrida foi desenvolvida a partir do MPO_S, com algumas modificações relacionadas à construção da solução e do conjunto E^0 . Esta nova heurística foi chamada apenas de MPO. Assim como o MPO_S, o MPO é um procedimento híbrido que combina GRASP, GENIUS, RVND e PLI, que neste caso faz uso da formulação de BIENSTOCK *et al.* [13]. O pseudocódigo do MPO é apresentado no Algoritmo 14.

Algoritmo 14: MPO($maxIt, seed, model$)

```

1:  $S \leftarrow \emptyset; f^* \leftarrow \infty; E^0 \leftarrow E;$ 
2: Para  $k = 1$  até  $k \leq maxIt$  faz
3:    $S' \leftarrow \text{GENIUS}(seed); \{\text{Fase de construção}\}$ 
4:    $S'' \leftarrow \text{RVND}(S', seed); \{\text{Fase de busca local}\}$ 
5:   Se  $f(S'') \leq f^*$  então
6:      $S \leftarrow S''; f^* \leftarrow f(s);$ 
7:   Fim-se
8:    $k \leftarrow k + 1;$ 
9:    $\text{update}(S'', E^0);$ 
10: Fim-para
11:  $\text{remove}(E^0, model);$ 
12:  $S \leftarrow \text{solve}(model); \{\text{solver matemático resolve o modelo reduzido}\}$ 
13: Retorne  $S;$ 

```

A fase heurística do MPO foi implementada da seguinte maneira. Na construção do GRASP, inicialmente três nós aleatórios são inseridos na rota e, então, enquanto o *PRIZE* não for atendido, os seguintes passos são realizados: i) a lista de candidatos (LC) é atualizada; ii) uma lista restrita de candidatos (LRC) é formada com os melhores elementos de LC; iii) um nó é escolhido aleatoriamente a partir de LRC e inserido na rota por meio do método GENI; e iv) uma tentativa de melhoria da rota é realizada pelo método US. Para a criação da LRC, cada vértice candidato k a ser inserido entre dois vértices i e j , é avaliado segundo a seguinte função gulosa de CHAVES e LORENA [19]:

$$g(k) = \min_{i,j \in S: i \neq j} (c_{ik} + c_{kj} - c_{ij} - w_k) \quad (6.2)$$

onde c_{ij} é o custo da aresta (i, j) , e w_k é a penalidade do vértice k . c_{ik} e c_{kj} são, respectivamente, os custos das arestas (i, k) e (k, j) . A LRC é formada pelos $\alpha \times |LC|$ vértices de menores valores, segundo a função gulosa $g(k)$. Após a construção de uma solução inicial, assim como no MPO_S, uma busca local é realizada a partir de uma heurística RVND.

No MPO_S, o critério de parada do módulo heurístico é um número máximo de iterações, ou quando a “estabilidade” do conjunto E^0 é alcançada. A partir de experimentos com as instâncias do PCVCP, não percebeu-se um ponto em que o E^0 se tornava estável; para várias instâncias todas as iterações eram executadas. Portanto, para o MPO foi decidido remover o critério de “estabilidade” adotado no MPO_S.

O restante do método MPO é exatamente o mesmo do MPO_S implementado para o PRRCP: após a construção do conjunto E^0 , as respectivas variáveis de suas arestas são removidas do modelo, e então o modelo é resolvido.

Capítulo 7

Resultados para o PCVCP

De modo a validar as propostas desta tese, foram conduzidos experimentos com dois grupos de instâncias da literatura, e um novo conjunto de problemas proposto. Nos experimentos foram utilizados o mesmo ambiente de execução e a mesma linguagem de programação do PRRCP (ver Capítulo 4). Os valores dos parâmetros comuns entre MPO_S e MPO, e do *solver* matemático também foram mantidos. Como não foi possível a utilização de todos os códigos-fonte originais dos métodos comparados, as formulações da literatura foram implementadas pelo autor desta tese, e para as heurísticas, foram utilizados os resultados reportados na literatura para conversões de tempos despendidos.

7.1 Instâncias de Chaves *et al.*

O primeiro grupo de instâncias foi proposto por CHAVES *et al.* [17] e consiste em problemas com $|V| \in \{10, 20, 30, 50, 100, 250, 500\}$, gerados aleatoriamente nos seguintes intervalos. Custo de viagem entre os vértices, $c_{ij} \in [50, 1000]$; prêmio associado a cada vértice, $y_i \in [1, 750]$; e o prêmio mínimo a ser coletado (*PRIZE*), representando 75% da soma dos prêmios de todos os vértices. Essas instâncias estão disponíveis em <http://www.decom.ufop.br/prof/marcone/Orientacoes/OrientacoesConcluidas.htm>.

7.1.1 Formulações matemáticas

A Tabela 7.1 apresenta uma comparação entre os resultados obtidos por três formulações da literatura: TORRES e BRITO [95], CHAVES *et al.* [17], CHAVES e LORENA [16], e a formulação modificada de BIENSTOCK *et al.* [13] (Bienstock⁺). A primeira coluna refere-se ao nome das instâncias testadas e, para cada modelo, são apresentados o valor da solução obtida e o tempo despendido. Os melhores resultados estão destacados em negrito e o símbolo “-” indica que nenhuma solução

foi encontrada.

Tabela 7.1: Resultados das formulações matemáticas para as instâncias de CHAVES *et al.* [17].

Inst.	Bienstock ⁺		Chaves 2004 [17]		Torres 2003[95]		Chaves 2005[16]	
	Sol	T (s)	Sol	T (s)	Sol	T (s)	Sol	T (s)
v10	1765	0.00	1765	0.06	1765	0.03	1765	0.03
v20	2302	0.01	2302	0.74	2302	0.70	2302	1.22
v30a	3582	0.00	3582	4.13	3582	2.66	3582	4.43
v30b	2515	0.00	2515	3.29	2515	3.67	2515	6.41
v30c	3236	0.03	3236	12.34	3236	12.63	3236	15.47
v50a	4328	0.10	4328	328.07	4328	314.74	4328	422.65
v50b	3872	0.12	3872	371.30	3872	375.30	3872	688.77
v100a	6762	0.27	7001	3600.00	7633	3600.00	7454	3600.00
v100b	6760	0.17	7223	3600.00	7290	3600.00	7668	3600.00
v250a	14083	12.67	15586	3600.00	-	3600.00	-	3600.00
v250b	13632*	8.35	14974	3600.00	14935	3600.00	14715	3600.00
v500a	25848*	102.06	-	3600.00	-	3600.00	-	3600.00
v500b	26389*	130.17	-	3600.00	-	3600.00	-	3600.00

*Optimalidade inédita

A partir da Tabela 7.1, pode ser observado que a formulação Bienstock+ é superior às demais da literatura, tanto em qualidade de solução como em tempo de processamento. A nova formulação foi capaz de resolver todos os 13 problemas-teste propostos por CHAVES *et al.* [17], provando optimalidade inédita em três casos (v250b, v500a e v500b). Já as formulações da literatura foram capazes de resolver apenas instâncias com até 50 vértices, considerando o tempo limite de uma hora. Dentre os modelos da literatura, a formulação de CHAVES e LORENA [16] necessitou de um tempo de CPU maior que as outras duas, mas, em geral, os três modelos direcionados apresentaram desempenho semelhante.

Alguns resultados apresentados na Tabela 7.1 diferem dos reportados na literatura, porque os autores utilizaram um ambiente de execução ou *solver* diferentes. Porém, o único resultado reportado na literatura, que supera algum resultado apresentado nesta tese, é para o modelo de CHAVES e LORENA [16]. Na instância v100a, foi atingida uma solução sub-ótima de valor 6879 em um tempo de 153059.09 segundos, 42 vezes maior que o limite de uma hora, utilizado em nossos experimentos. CHAVES e LORENA [16] não relataram se o *solver* foi interrompido devido à problemas de memória ou tempo limite de execução.

7.1.2 Heurísticas

Nos experimentos com as heurísticas, cada algoritmo proposto foi executado dez vezes, com dez sementes distintas para cada uma das instâncias. Como o código-fonte original do CS* não foi disponibilizado e, os experimentos reportados em CHAVES e LORENA [16] foram conduzidos em um processador diferente, foi adotada a estratégia de PINTO *et al.* [81], que considera uma taxa de escala aproximada para comparar a velocidade dos dois processadores. De acordo com PASSMARK [78], considerando que os algoritmos foram testados em um ambiente com uma única *thread*, essa taxa é de $463/2184 \approx 0.214$.

A Tabela 7.2 apresenta uma comparação entre a heurística estado-da-arte (para as instâncias de CHAVES *et al.* [17]), e a heurística proposta neste trabalho. CHAVES e LORENA [16] não apresentaram resultados para algumas instâncias, tais como: v100b, v250b e v500b. Experimentos com o CS* utilizando essas três instâncias são relatados em PEDRO *et al.* [80], porém esses últimos autores não apresentaram o tempo utilizado pelas heurísticas testadas. Portanto, na Tabela 7.2, o símbolo “?” indica que o tempo da respectiva instância não foi reportado.

A primeira coluna indica o nome da instância testada e, para cada método, são apresentados a melhor solução, a média de solução das dez execuções, o tempo de CPU utilizado (em segundos), e o *gap* percentual do valor da melhor solução em relação ao valor ótimo. Além disto, os tempos de CPU para o CS* são os reportados em CHAVES e LORENA [16] multiplicados pelo fator de escala 0.214.

Tabela 7.2: Resultados das heurísticas para as instâncias de CHAVES *et al.* [17] (Tempos de execução originais do CS* multiplicados por 0.214).

Instância	Ótimo	CS*					MPO				
		Melhor Sol.	Média Sol.	Média T (s)	Gap*	Melhor Sol.	Média Sol.	Média T (s)	Gap*		
v10	1765	1765	1765.0	0.01	0.0	1765	1765.0	0.23	0.0		
v20	2302	2302	2302.0	0.21	0.0	2302	2302.0	0.29	0.0		
v30a	3582	3582	3591.0	0.22	0.0	3582	3582.0	1.03	0.0		
v30b	2515	2515	2515.0	0.24	0.0	2515	2515.0	1.05	0.0		
v30c	3236	3236	3241.0	0.25	0.0	3236	3236.0	1.10	0.0		
v50a	4328	4328	4346.0	8.22	0.0	4328	4328.0	21.58	0.0		
v50b	3872	3872	3881.0	7.98	0.0	3872	3873.6	22.87	0.0		
v100a	6762	6832	6906.0	153.92	1.0	6762	6762.0	158.63	0.0		
v100b	6760	6782	6844.0	?	0.3	6760	6767.8	146.58	0.0		
v250a	14083	15162	15284.0	248.69	7.7	14083	14083.0	171.52	0.0		
v250b	13632	14078	14176.0	?	3.3	13632	13632.4	160.75	0.0		
v500a	25848	28213	28462.0	440.49	9.1	26269	26269.0	245.42	1.6		
v500b	26389	28133	28334.0	?	6.6	26640	26640.0	249.76	1.0		

A partir da Tabela 7.2, pode ser constatado, que em termos de melhor solução

obtida, o MPO superou o CS* em seis casos, atingindo a solução ótima em quatro instâncias a mais que a heurística da literatura. Em média de solução, o desempenho do MPO foi ainda superior, vencendo em dez casos. Em relação ao tempo de execução, o CS* venceu para instâncias com até 100 nós, enquanto o MPO foi mais rápido para as demais. Em geral, pode ser observado que a heurística proposta foi capaz de melhorar as soluções obtidas com o CS*, principalmente para as maiores instâncias.

Análise da função de crescimento

Nesta subseção, é apresentada uma análise de crescimento em relação ao tempo médio de CPU demandado pelas heurísticas. Segundo BARR *et al.* [11], na ausência de um ambiente ideal, as comparações de tempo entre heurísticas executadas em diferentes plataformas são, no mínimo, imprecisas.

Embora uma função de complexidade possa dar uma ideia de como os tempos de solução podem crescer com o tamanho do problema, uma estimativa desse crescimento pode ser bastante útil para avaliar como uma heurística será executada em grandes instâncias de um problema [11]. Na Figura 7.1, são ilustradas as funções de crescimento das respectivas heurísticas MPO e CS*.

Para auxiliar na visualização da função de crescimento, foi considerado que os valores de $|V|$ e tempo médio de CPU são polinomialmente relacionados, e então uma regressão polinomial foi aplicada, utilizando o pacote R [85]. Na Figura 7.1, para cada série (CS* e MPO), foi traçada uma linha que corresponde à curva obtida por uma regressão polinomial de grau 2. Ao lado do rótulo de cada heurística é apresentada a equação do polinômio $a + b(x) + cx^2$ (x correspondendo a $|V|$), os coeficientes do polinômio obtidos a , b e c , e o coeficiente de determinação R^2 . O coeficiente R^2 é uma medida estatística que assume valores entre 0 e 1, e indica quanto próximos os dados estão da linha de regressão ajustada [15].

Quanto maior o valor de R^2 , mais explicativo é o modelo de regressão, e melhor ele prevê resultados futuros. Por exemplo, o R^2 da regressão das médias do MPO é 0.94, e isto significa que 94.2% da variável dependente ($|V|$) consegue ser explicada pelo modelo. Outras regressões como a linear, exponencial e logarítmica foram testadas, mas a polinomial foi a que melhor descreveu os dados. Foram testados outros graus para a regressão polinomial, mas uma vez que com o grau dois já se tem um valor razoável de R^2 , decidiu-se não utilizar graus maiores de modo a evitar *overfitting* do modelo.

Para a geração da Figura 7.1, foi considerado o tempo médio de cada subconjunto de tamanho de instâncias, desconsiderando para ambas as heurísticas, os casos em que o tempo não foi reportado na literatura. No caso do CS, para o cálculo da média, foram utilizados os tempos convertidos. Nesta figura, pode ser observado um comportamento semelhante do crescimento de tempo de ambas as heurísticas até

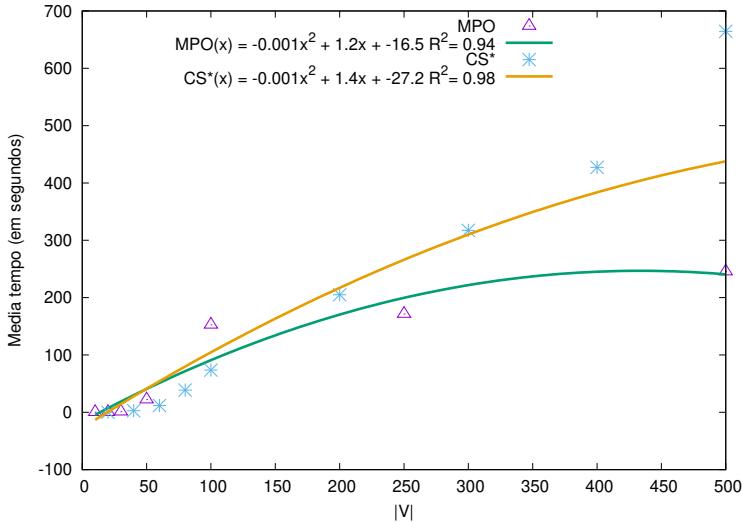


Figura 7.1: Função de crescimento do tempo das heurísticas CS* e MPO.

instâncias com 50 vértices. A partir de casos com 100 nós, a função de crescimento da heurística CS* cresce mais rápida que a função do MPO.

Para o CS*, a diferença de tempo de CPU gasto entre instâncias com 50 e 100 nós é de 145.94 segundos, entre 250 e 100 nós é de 94.77 segundos, e entre 250 e 500 nós é de 191.80 segundos. Já para o MPO, a crescimento da função é mais suave. A diferença entre 100 e 50 nós é de 130.38 segundos, entre 250 e 100 nós é de 18.91 segundos, e entre 250 e 500 nós é de 73.90 segundos.

Além do MPO, foram realizados experimentos com duas MIP heurísticas, uma baseada em *relax-and-fix* e outra em *guided diving*. Contudo, uma vez que essas heurísticas não apresentaram resultados promissores para os demais conjuntos de instância, decidiu-se omiti-las desta tese. Os resultados desses experimentos foram reportados em CLIMACO *et al.* [22].

7.2 Instâncias de Chaves e Lorena

O segundo grupo de problemas-teste foi proposto por CHAVES e LORENA [19]. São instâncias com $|V| \in \{20, 40, 60, 80, 100, 200, 300, 400, 500\}$, custos de viagens $c_{ij} \in [1, M]$ com $M \in \{1000, 10000\}$, prêmios $p_i \in [1, 100]$, e penalidades $\gamma \in [1, N]$ com $N \in \{100, 1000, 10000\}$. O valor do prêmio mínimo foi definido com uma fração do prêmio total, a partir da expressão $\sigma (\sum_{i \in V} p_i)$, com $\sigma \in \{0.2, 0.5, 0.8\}$. Essas instâncias estão disponíveis em <http://www.lac.inpe.br/~lorena/instancias.html>.

7.2.1 Formulações matemáticas

Nas Tabelas 7.3, 7.4, 7.5 e 7.6, as primeiras colunas indicam os valores de σ e o tamanho da instância. Em seguida, para cada formulação é apresentada a solução encontrada e o tempo despendido (em segundos). O tempo limite para a execução dos modelos foi definido como 3600 segundos, os melhores resultados estão destacados em negrito, e os valores de solução com prova de otimalidade inédita na literatura são marcados com o símbolo “*”.

Tabela 7.3: Resultados das formulações para instâncias de CHAVES e LORENA [19], $c_{ij} \in [1, 1000]$ e $\gamma_i \in [1, 100]$.

σ	$ V $	Bienstock+		CHAVES et al.[17]		CHAVES e LORENA [16]		TORRES e BRITO [95]	
		Sol	T (s)	Sol	T (s)	Sol	T (s)	Sol	T (s)
	20	903	0.0	903	0.4	903	0.4	903	0.3
	40	996	0.0	996	17.0	996	16.7	996	18.7
	60	1314	0.1	1314	1247.7	1314	1786.5	1314	1900.5
	80	1384	0.3	1508	3600.0	1541	3600.0	1478	3600.0
0.2	100	1502*	1.6	1876	3600.0	2053	3600.0	2284	3600.0
	200	1625*	8.1	2933	3600.0	2786	3600.0	-	3600.0
	300	1665*	17.5	-	3600.0	-	3600.0	-	3600.0
	400	1854*	56.4	-	3600.0	-	3600.0	-	3600.0
	500	1861*	51.3	-	3600.0	-	3600.0	-	3600.0
	20	1123	0.1	1123	1.0	1123	0.8	1123	0.7
	40	996	0.0	996	16.4	996	12.0	996	19.0
	60	1314	0.1	1314	1337.9	1340	3600.0	1314	1459.0
	80	1384	0.4	1466	3600.0	1573	3600.0	1485	3600.0
0.5	100	1502*	1.8	2031	3600.0	1963	3600.0	2249	3600.0
	200	1625*	7.5	2933	3600.0	2786	3600.0	-	3600.0
	300	1665*	17.6	-	3600.0	-	3600.0	-	3600.0
	400	1854*	56.7	-	3600.0	-	3600.0	-	3600.0
	500	1861*	51.5	-	3600.0	-	3600.0	-	3600.0
	20	1354	0.0	1354	0.5	1354	2.1	1354	1.2
	40	1129	0.1	1129	44.8	1129	157.5	1129	88.0
	60	1319	0.2	1319	862.1	1336	3600.0	1319	2987.5
	80	1384	0.4	1488	3600.0	1530	3600.0	1497	3600.0
0.8	100	1502*	1.5	1918	3600.0	1957	3600.0	2012	3600.0
	200	1625*	7.4	2933	3600.0	2492	3600.0	2814	3600.0
	300	1665*	19.5	-	3600.0	-	3600.0	-	3600.0
	400	1854*	65.4	-	3600.0	-	3600.0	-	3600.0
	500	1861*	51.2	-	3600.0	-	3600.0	-	3600.0

*Optimalidade inédita

A partir das Tabelas 7.3, 7.4, 7.5 e 7.6, pode-se perceber que a formulação Bienstock+ foi superior a todas as outras da literatura, em tempo de CPU e qualidade de solução. A formulação Bienstock+ foi capaz de resolver todas as 108 instâncias de CHAVES e LORENA [19] em menos de 3600 segundos cada, provando otimalidade inédita para 54 casos. Nos experimentos realizados nesta tese, os outros três modelos não conseguem atingir o ótimo em pelo menos 50% dos casos.

Para as instâncias de CHAVES e LORENA [19], as três formulações baseadas em fluxo apresentaram desempenhos semelhantes entre si, resolvendo instâncias com até 80 vértices. Desses três modelos, CHAVES e LORENA [16] é o que possui pior desempenho em termos de tempo de CPU e qualidade de solução, não resolvendo instâncias de 60 nós que foram resolvidas pelas demais formulações.

No mesmo trabalho em que CHAVES e LORENA [19] disponibilizaram as suas

Tabela 7.4: Resultados das formulações para as instâncias de CHAVES e LORENA [19], $c_{ij} \in [1, 10000]$ e $\gamma_i \in [1, 1000]$.

σ	$ V $	Bienstock+		CHAVES et al.[17]		CHAVES e LORENA [16]		TORRES e BRITO [95]	
		Sol	T (s)	Sol	T (s)	Sol	T (s)	Sol	T (s)
		20	11677	0.01	11677	0.4	11677	0.3	11677
0.2	40	10776	0.05	10776	65.9	10776	54.6	10776	45.1
	60	14236	0.23	14236	1133.2	14236	1631.1	14236	985.0
	80	14484	0.37	15202	3600.0	15828	3600.0	15871	3600.0
	100	13442*	2.19	20775	3600.0	17838	3600.0	18968	3600.0
	200	13364*	5.58	20435	3600.0	23334	3600.0	-	3600.0
	300	17271*	15.05	-	3600.0	-	3600.0	-	3600.0
	400	15065*	75.59	-	3600.0	-	3600.0	-	3600.0
	500	17860*	124.04	-	3600.0	-	3600.0	-	3600.0
	20	12900	0.04	12900	1.4	12900	1.1	12900	1.8
	40	10776	0.05	10776	42.5	10776	74.1	10776	50.6
0.5	60	14236	0.23	14236	669.5	14236	3287.6	14236	2257.9
	80	14484	0.38	15360	3600.0	16144	3600.0	15213	3600.0
	100	13438*	1.17	16730	3600.0	20495	3600.0	18049	3600.0
	200	13364*	5.58	20435	3600.0	23334	3600.0	-	3600.0
	300	17271*	15.07	-	3600.0	-	3600.0	-	3600.0
	400	15065*	75.28	-	3600.0	-	3600.0	-	3600.0
	500	17860*	123.97	-	3600.0	-	3600.0	-	3600.0
	20	16559	0.04	16559	1.3	16559	1.9	16559	0.8
	40	10776	0.05	10776	30.0	10776	56.6	10776	78.0
	60	14864	0.31	14864	3600.0	15183	3600.0	14864	3219.9
0.8	80	14484	0.42	15286	3600.0	16514	3600.0	15671	3600.0
	100	13438*	1.17	16599	3600.0	15309	3600.0	23907	3600.0
	200	13364*	5.60	20435	3600.0	24098	3600.0	-	3600.0
	300	17271*	15.13	-	3600.0	-	3600.0	-	3600.0
	400	15065*	75.66	-	3600.0	-	3600.0	-	3600.0
	500	17860*	135.70	-	3600.0	-	3600.0	-	3600.0

*Optimalidade inédita

Tabela 7.5: Resultado das formulações para as instâncias de CHAVES e LORENA [19], $c_{ij} \in [1, 10000]$ e $\gamma_i \in [1, 100]$.

σ	$ V $	Bienstock+		CHAVES et al.[17]		CHAVES e LORENA [16]		TORRES e BRITO [95]	
		Sol	T (s)	Sol	T (s)	Sol	T (s)	Sol	T (s)
		20	3011	0.0	3011	0.3	3011	0.6	3011
0.2	40	3506	0.1	3506	5.8	3506	3.3	3506	3.6
	60	4251	0.3	4251	80.1	4251	140.8	4251	105.0
	80	4903	0.6	4903	1851.9	4903	699.1	4903	749.5
	100	5635	0.8	5852	3600.0	5635	3600.0	5635	3600.0
	200	7945*	4.8	11161	3600.0	17459	3600.0	-	3600.0
	300	10815*	10.3	-	3600.0	-	3600.0	-	3600.0
	400	11259*	20.3	-	3600.0	-	3600.0	-	3600.0
	500	12361*	48.3	-	3600.0	-	3600.0	-	3600.0
	20	4313	0.1	4313	2.0	4313	2.8	4313	2.2
	40	4694	0.2	4694	27.9	4694	31.1	4694	20.0
0.5	60	6120	0.6	6326	3600.0	6368	3600.0	6120	2945.5
	80	6319	1.4	7182	3600.0	7357	3600.0	7539	3600.0
	100	6869	3.2	8410	3600.0	8786	3600.0	9040	3600.0
	200	8080*	11.6	12338	3600.0	12930	3600.0	-	3600.0
	300	10815*	15.1	-	3600.0	-	3600.0	-	3600.0
	400	11259*	55.5	-	3600.0	-	3600.0	-	3600.0
	500	12361*	48.4	-	3600.0	-	3600.0	-	3600.0
	20	7797	0.1	7797	1.7	7797	1.4	7797	1.3
	40	9070	0.1	9070	280.0	9070	104.9	9070	149.3
	60	9459	0.8	9769	3600.0	9616	3600.0	10219	3600.0
0.8	80	9669	0.8	11098	3600.0	13847	3600.0	12266	3600.0
	100	10002	3.8	14122	3600.0	15087	3600.0	14056	3600.0
	200	10332*	91.7	15247	3600.0	16283	3600.0	-	3600.0
	300	11998*	54.1	-	3600.0	-	3600.0	-	3600.0
	400	11810*	152.2	-	3600.0	-	3600.0	-	3600.0
	500	12446*	244.6	-	3600.0	-	3600.0	-	3600.0

*Optimalidade inédita

Tabela 7.6: Resultado das formulações para as instâncias de CHAVES e LORENA [19], $c_{ij} \in [1, 1000]$ e $\gamma_i \in [1, 10000]$.

σ	$ V $	Bienstock+		CHAVES <i>et al.</i> [17]		CHAVES e LORENA [16]		TORRES e BRITO [95]	
		Sol	T (s)	Sol	T (s)	Sol	T (s)	Sol	T (s)
0.2	20	1192	0.0	1192	0.8	1192	1.3	1192	1.1
	40	1449	0.2	1449	22.8	1449	126.6	1449	103.4
	60	1666	0.8	1666	1767.1	1666	1104.0	1666	1645.3
	80	1794	0.8	2001	3600.0	2290	3600.0	2101	3600.0
	100	1601	1.3	1792	3600.0	2495	3600.0	2172	3600.0
	200	1675*	2.5	2650	3600.0	2511	3600.0	2574	3600.0
	300	1789*	5.2	-	3600.0	-	3600.0	-	3600.0
	400	1799*	58.1	-	3600.0	-	3600.0	-	3600.0
	500	1874*	155.7	-	3600.0	-	3600.0	-	3600.0
	20	1192	0.0	1192	0.8	1192	1.3	1192	1.1
0.5	40	1449	0.2	1449	22.8	1449	127.4	1449	103.1
	60	1666	0.8	1666	1766.6	1666	1103.9	1666	1645.8
	80	1794	0.8	2001	3600.0	2290	3600.0	2101	3600.0
	100	1601	1.3	1792	3600.0	2495	3600.0	2172	3600.0
	200	1675*	2.5	2650	3600.0	2511	3600.0	2574	3600.0
	300	1789*	5.2	-	3600.0	-	3600.0	-	3600.0
	400	1799*	58.2	-	3600.0	-	3600.0	-	3600.0
	500	1874*	155.7	-	3600.0	-	3600.0	-	3600.0
	20	1192	0.0	1192	0.7	1192	1.3	1192	1.2
	40	1449	0.2	1449	22.7	1449	127.6	1449	103.2
0.8	60	1666	0.8	1666	1767.6	1666	990.7	1666	1644.9
	80	1794	0.8	2001	3600.0	2290	3600.0	1893	3600.0
	100	1601	1.3	1792	3600.0	2008	3600.0	2172	3600.0
	200	1675*	2.5	2650	3600.0	2511	3600.0	2963	3600.0
	300	1789*	5.2	-	3600.0	-	3600.0	-	3600.0
	400	1799*	58.2	-	3600.0	-	3600.0	-	3600.0
	500	1874*	156.3	-	3600.0	-	3600.0	-	3600.0
	20	1192	0.0	1192	0.7	1192	1.3	1192	1.2
	40	1449	0.2	1449	22.7	1449	127.6	1449	103.2
	60	1666	0.8	1666	1767.6	1666	990.7	1666	1644.9

*Optimalidade inédita

instâncias, esses autores realizaram experimentos com a formulação CHAVES e LORENA [16] utilizando um limite de tempo igual a 100000 segundos. A Tabela 7.7 apresenta os resultados para 21 casos, em que CHAVES e LORENA [19] reportaram soluções de melhor qualidade, que os resultados apresentados nesta tese. Apesar de um tempo limite cerca de 27 vezes maior que o estipulado nesta tese, é reportado que a formulação de CHAVES e LORENA [16] não atingiu o ótimo em instâncias com mais de 100 vértices.

7.2.2 Heurísticas

As Tabelas 7.8, 7.9, 7.10 e 7.11 apresentam resultados do MPO obtidos por 10 execuções para cada instância, utilizando 10 sementes diferentes. Como o código-fonte da heurística estado-da-arte CS¹ não foi disponibilizado, nesta tese são apresentados os mesmos resultados reportados em CHAVES e LORENA [19], com exceção do tempo despendido, onde foi usada a ideia de PINTO *et al.* [81], com uma taxa de conversão igual 0.3. Para cada instância, é apresentada a melhor solução obtida, a média de solução, o tempo despendido e uma coluna “Gap*(%)” indicando a diferença percentual da melhor solução em relação ao ótimo. Os melhores resultados estão destacados em negrito.

¹Heurística diferente do CS* de CHAVES e LORENA [16]

Tabela 7.7: Resultados para o modelo de CHAVES e LORENA [16], em que a literatura apresenta melhores soluções que nesta tese.

$c_{ij} \in [1, 1000]$ e $\gamma_i \in [1, 100]$				$c_{ij} \in [1, 10000]$ e $\gamma_i \in [1, 1000]$			
σ	$ V $	Sol	T (s)	σ	$ V $	Sol	T (s)
0.2	80	1384	26692.9	0.5	80	14484	68464.4
	60	1314	468.5		80	32121.2	
	80	1384			60	14464	18508.2
0.8	60	1319	474.8	0.8	80	14484	70205.2
	80	1384	27498.3		60		

$c_{ij} \in [1, 10000]$ e $\gamma_i \in [1, 100]$				$c_{ij} \in [1, 1000]$ e $\gamma_i \in [1, 1000]$			
σ	$ V $	Sol	T (s)	σ	$ V $	Sol	T (s)
0.2	100	5635	10824.2	0.5	80	1794	41885.0
	60	6120	700.4		100	1601	22767.7
	80	6319	72518.9		100	6869	69562.8
0.8	60	9459	10855.0	0.5	80	1794	32442.2
	80	9699	98073.2		100	1601	22730.5
				0.8	80	1794	27699.3
					100	1601	22722.6

Tabela 7.8: Resultados das heurísticas para instâncias de CHAVES e LORENA [19], $c_{ij} \in [1, 1000]$ e $\gamma_i \in [1, 100]$ (Tempos de execução originais do CS multiplicados por 0,3).

σ	$ V $	MPO				CS			
		Melhor Sol	Média Sol	Média T (s)	Gap* (%)	Melhor Sol	Média Sol	Média T (s)	Gap* (%)
0.2	20	903	903.0	0.1	0.0	903	903.0	0.009	0.0
	40	996	1040.8	0.4	0.0	996	996.0	0.684	0.0
	60	1323	1389.6	1.0	0.7	1314	1314.0	4.119	0.0
	80	1392	1460.2	2.0	0.6	1386	1392.8	24.909	0.1
	100	1561	1612.2	4.4	3.9	1508	1526.4	58.83	0.4
	200	1739	1786.0	25.8	7.0	1816	1834.4	150.882	11.8
	300	1904	1947.2	81.5	14.4	2281	2313.0	320.733	37.0
	400	2074	2194.8	196.4	11.9	2504	2554.2	363.855	35.0
	500	2185	2269.6	466.4	17.4	3233	3281.1	406.686	73.7
	60	1132	1132.0	0.5	0.8	1123	1123.0	0.09	0.0
0.5	40	996	996.0	2.8	0.0	996	996.0	0.702	0.0
	60	1314	1314.0	7.1	0.0	1314	1314.0	3.426	0.0
	80	1384	1384.0	14.6	0.0	1388	1396.6	21.822	0.3
	100	1513	1513.8	30.5	0.7	1513	1534.6	54.3	0.7
	200	1625	1625.8	259.3	0.0	1816	1844.2	171.738	11.8
	300	1665	1665.0	969.6	0.0	2171	2250.5	364.089	30.4
	400	1855	1855.0	1055.0	0.1	2489	2579.7	447.147	34.3
	500	1861	1861.6	1866.3	0.0	3159	3200.7	535.374	69.7
	60	1354	1354.0	0.2	0.0	1354	1354.0	0.045	0.0
	80	1129	1129.0	1.0	0.0	1129	1137.0	0.885	0.0
0.8	60	1336	1342.0	2.7	1.3	1319	1344.2	5.238	0.0
	80	1384	1384.0	5.6	0.0	1396	1400.2	19.158	0.9
	100	1512	1513.6	10.7	0.7	1519	1537.6	55.962	1.1
	200	1625	1625.9	83.1	0.0	1768	1797.2	241.503	8.8
	300	1665	1665.0	305.5	0.0	2148	2213.0	458.487	29.0
	400	1854	1854.8	782.6	0.0	2455	2494.3	500.67	32.4
	500	1861	1861.0	1758.8	0.0	3214	3324.2	512.508	72.7

A Tabela 7.8 fornece resultados para todos os valores de σ e $|V|$, nos quais $c_{ij} \in [1, 1000]$ e $\gamma_i \in [1, 10000]$. Percebe-se que em termos de qualidade da melhor solução obtida, o MPO é superior em 55.55% das 27 instâncias, atingindo o ótimo em 16; enquanto que o CS vence em 18.51% dos casos, atingindo o ótimo em nove. Em relação à média de solução, o MPO aumenta a sua superioridade, vencendo em

66.66% das instâncias, enquanto o CS mantém o número de vitórias. Em relação ao tempo despendido, o desempenho de ambas as heurísticas é semelhante: MPO e CS venceram em, respectivamente, 14 e 13 instâncias.

Tabela 7.9: Resultados das heurísticas para instâncias de CHAVES e LORENA [19], $c_{ij} \in [1, 10000]$ e $\gamma_i \in [1, 1000]$ (Tempos de execução originais do CS multiplicados por 0.3).

σ	$ V $	MPO				CS			
		Melhor Sol	Média Sol	Média T (s)	Gap* (%)	Melhor Sol	Média Sol	Média T (s)	Gap* (%)
0.2	20	11677	11945.2	0.1	0.0	11677	11677.0	0.01	0.0
	40	11095	11245.2	0.4	0.0	10776	10776.0	0.95	0.0
	60	14236	15039.4	1.1	0.7	14243	14314.1	2.133	0.0
	80	15038	15192.0	2.3	0.6	14609	14760.9	34.45	0.9
	100	14234	14690.2	3.9	3.9	13620	14015.0	31.25	1.3
	200	15246	15621.6	22.3	7.0	15303	15628.2	158.55	14.5
	300	19518	20005.4	79.0	14.4	21869	22158.0	198.69	26.6
	400	18031	18485.8	197.5	11.9	24390	25099.6	406.32	61.9
	500	20972	21633.0	479.4	17.4	31090	31558.7	492.94	74.1
	20	13061	13916.6	0.5	0.8	12900	12900.0	0.05	0.0
0.5	40	10845	10845.0	2.6	0.0	10776	10776.0	1.67	0.0
	60	14236	14236.0	7.7	0.0	14349	14421.9	4.11	0.8
	80	14505	14521.8	15.0	0.0	14512	14830.0	33.57	0.2
	100	13438	13441.2	29.7	0.7	13900	14089.1	35.50	3.4
	200	13364	13364.2	216.9	0.0	15190	15440.4	199.25	13.17
	300	17271	17271.0	1016.7	0.0	22731	23211.7	209.02	31.6
	400	15065	15075.8	2951.5	0.1	23898	24525.3	526.62	58.6
	500	17860	17863.2	6554.3	0.0	30275	30842.0	651.93	69.5
	20	16775	16940.0	0.3	0.0	16559	16559.0	0.02	0.0
	40	10845	10845.0	1.3	0.0	10776	10776.0	1.398	0.0
0.8	60	14864	14867.8	3.6	1.3	14864	15017.1	9.822	0.0
	80	14505	14549.8	7.1	0.0	14740	14793.9	27.831	1.8
	100	13438	13440.0	12.4	0.7	13704	13971.9	28.593	2.0
	200	13364	13364.0	84.4	0.1	15200	15376.2	215.082	13.7
	300	17271	17271.0	345.0	0.0	22168	22467.7	449.031	28.4
0.9	400	15065	15065.2	723.9	0.1	22790	23688.3	585.91	51.3
	500	17860	17860.0	1614.9	0.0	30385	30707.0	701.05	70.1

A Tabela 7.9 apresenta resultados para as instâncias com $c_{ij} \in [1, 10000]$ e $\gamma_i \in [1, 1000]$. Para essas instâncias, em relação à qualidade da melhor solução obtida, a heurística proposta MPO vence o CS em 66.66% dos casos, atingindo o ótimo em 14; enquanto o CS vence 25.92% das instâncias, alcançando o ótimo em sete. Em termos de média de solução, o CS melhora levemente o seu desempenho vencendo em 33.33% do casos, mas o MPO permanece superior, atingindo melhor média em 66.66% das instâncias. Em relação ao tempo de CPU despendido, o MPO vence em 16, e o CS em 11 casos.

Os resultados para o grupo de instâncias com $c_{ij} \in [1, 10000]$ e $\gamma_i \in [1, 100]$ são expostos na Tabela 7.10. O MPO possui melhor desempenho em termos de qualidade de solução, superando o CS em 70.37% das instâncias. Em relação à média de solução, o MPO vence 74.07% das instâncias enquanto o CS vence em 22.22% dos casos. Em relação ao tempo de CPU despendido, o MPO vence em 14 e o CS em 13 casos. Além disso, o MPO é melhor quando o quesito é atingir soluções ótimas. A heurística proposta atinge a solução ótima em seis casos a mais que o CS.

A Tabela 7.11 reporta resultados para instâncias com $c_{ij} \in [1, 1000]$ e $\gamma_i \in$

Tabela 7.10: Resultados das heurísticas para instâncias de CHAVES e LORENA [19], $c_{ij} \in [1, 10000]$ e $\gamma_i \in [1, 100]$ (Tempos de execução originais do CS multiplicados por 0.3).

σ	$ V $	MPO				CS			
		Melhor Sol	Média Sol	Média T (s)	Gap* (%)	Melhor Sol	Média Sol	Média T (s)	Gap* (%)
0.2	20	3099	3440.2	0.1	2.9	3011	3011.0	0.003	0.0
	40	3570	3641.6	0.4	1.8	3506	3531.6	0.201	0.0
	60	4420	4448.0	1.0	4.0	4277	4287.2	4.458	0.6
	80	4903	5159.0	2.2	0.0	4903	5044.3	6.783	0.0
	100	5683	5732.8	4.0	0.9	5702	5802.0	17.913	1.2
	200	8036	8188.8	21.7	1.1	9035	9129.0	91.023	13.7
	300	11299	11355.4	68.0	4.5	14592	14875.2	95.742	34.9
	400	12095	12234.6	159.7	7.4	16651	16850.3	192.051	47.9
	500	13588	13788.0	326.6	9.9	20612	21305.7	251.004	66.8
	20	4313	4313.0	0.4	0.0	4313	4313.0	0.051	0.0
0.5	40	4694	5177.0	2.5	0.0	4694	4694.0	1.287	0.0
	60	6120	6311.6	7.2	0.0	6232	6361.7	4.215	1.8
	80	6394	6478.8	14.8	1.2	6528	6628.1	19.107	3.3
	100	6940	6978.0	27.4	1.0	7710	7833.7	35.856	12.2
	200	8080	8126.2	185.7	0.0	10293	10578.0	131.64	27.4
	300	10823	10830.0	734.3	0.1	15312	15698.3	164.895	41.6
	400	11259	11268.8	1746.2	0.0	17263	17535.7	252.51	53.3
	500	12371	12371.4	3961.8	0.1	20896	21623.7	316.824	69.0
	20	8047	8127.6	0.3	3.2	7797	7797.0	0.033	0.0
	40	9070	9070.0	1.2	0.0	9070	9171.6	2.265	0.0
0.8	60	9459	9619.1	3.6	0.0	9664	9810.5	9.831	2.2
	80	9669	9721.4	6.8	0.0	9991	10048.0	29.106	3.3
	100	10002	10014.1	11.4	0.0	10641	10724.1	47.388	6.4
	200	10332	10342.5	81.3	0.0	12650	13024.0	185.052	22.4
	300	11998	12000.7	263.8	0.0	18253	18740.8	330.09	52.1
	400	11810	11812.4	634.8	0.0	18501	18955.7	472.368	56.7
	500	12449	12451.5	1565.2	0.0	23234	23590.3	576.276	86.7

[1, 10000]. Neste caso, em termos de melhor solução, o MPO vence a heurística estado-da-arte em 51.85% das instâncias e perde para 28.62%. Em relação à solução média, o MPO vence em 29.62% e perde em 59.25% das instâncias. Da mesma maneira que na Tabela 7.10, a heurística proposta atinge um número maior de soluções ótimas, três a mais que o CS. Em relação ao tempo despendido, o MPO e o CS vencem em, respectivamente, 9 e 18 instâncias.

A partir das tabelas apresentadas nesta subseção, foi realizada uma análise acerca do *gap* das heurísticas, em relação à solução ótima do problema. As Figuras 7.2(a), 7.2(b), 7.2(c) e 7.2(d) ilustram, separadamente para problemas com até 100 e com mais de 100 vértices, as médias de *gap* do CS e MPO. Cada figura representa um dos quatro subgrupos de instâncias, no qual o eixo das abscissas indica o valor de σ e o eixo das ordenadas apresenta os valores das médias.

Nas Figuras 7.2(a) e 7.2(b), para instâncias com menos de 100 vértices, pode ser observado que os *gaps* médios do CS são levemente melhores que os do MPO. Em contrapartida, nas instâncias de maior porte (mais de 100 vértices), é clara a superioridade do MPO.

Nas Figuras 7.2(c) e 7.2(d), mantém-se claro o desempenho superior do MPO para problemas com mais de 100 nós. Já para o subgrupo $c_{ij} \in [1, 10000]$ e $\gamma_i \in [1, 100]$, o MPO possui GAPs levemente inferiores que o CS, e o oposto ocorre no subgrupo

Tabela 7.11: Resultados das heurísticas para instâncias de CHAVES e LORENA [19], $c_{ij} \in [1, 1000]$ e $\gamma_i \in [1, 10000]$ (Tempos de execução originais do CS multiplicados por 0.3).

σ	$ V $	MPO				CS*			
		Melhor Sol.	Média Sol.	Média T (s)	Gap* (%)	Melhor Sol.	Média Sol.	Média T (s)	Gap* (%)
0.2	20	1192	1192.0	0.3	0.0	1192	1192.0	0.07	0.0
	40	1449	1449.0	1.4	0.0	1449	1449.0	1.95	0.0
	60	1929	2468.3	0.7	15.8	1666	1670.2	10.48	0.0
	80	2373	2945.0	0.7	32.3	1807	1814.6	43.48	0.7
	100	2023	2824.6	5.8	26.4	1628	1647.0	84.32	1.7
	200	2447	2979.5	88.4	46.1	1898	1946.8	273.35	13.3
	300	1987	2176.5	967.6	11.1	2246	2334.0	276.28	25.5
	400	2880	2935.5	697.4	60.1	2880	2933.7	411.06	60.1
	500	13926	20094.4	173.6	80.6	3385	3428.5	736.60	80.6
	20	1192	1192.0	0.3	0.0	1192	1192.0	0.07	0.0
0.5	40	1449	1450.6	1.4	0.0	1449	1449.0	1.81	0.0
	60	1690	1789.6	1.8	1.4	1666	1670.8	10.89	0.0
	80	1794	1794.4	4.3	0.0	1813	1821.6	39.47	1.1
	100	1660	1744.9	6.8	3.7	1655	1665.3	66.15	3.4
	200	2303	2504.5	73.0	37.5	1968	2012.0	158.93	17.5
	300	1987	2176.7	972.3	11.1	2300	2382.3	345.73	28.6
	400	3361	3886.9	853.4	86.8	2842	2935.3	404.59	58.0
	500	2011	2048.9	1285.3	7.3	3274	3332.6	616.34	74.7
	20	1192	1192.0	0.3	0.0	1192	1192.0	0.07	0.0
	40	1449	1449.0	1.4	0.0	1449	1449.0	4.22	0.0
0.8	60	1666	1666.0	3.8	0.0	1666	1669.6	13.92	0.0
	80	1794	1794.0	8.7	0.0	1801	1815.2	33.12	0.4
	100	1601	1601.7	13.9	0.0	1626	1658.8	70.76	1.6
	200	1675	1678.9	236.6	0.0	1978	1999.8	182.95	18.1
	300	2035	2221.5	958.8	13.8	2319	2371.7	330.29	29.6
	400	1810	1841.9	2401.7	0.6	2837	2849.0	466.36	57.7
	500	1874	1877.7	7179.5	0.0	3305	3333.2	640.35	76.4

$c_{ij} \in [1, 1000]$ e $\gamma_i \in [1, 10000]$.

Como uma forma de melhor avaliar o tempo despendido pelas duas heurísticas MPO e CS*, realizou-se a mesma análise feita para as instâncias de CHAVES e LORENA [16] (ver Figuras 7.3(a), 7.3(b), 7.3(c) e 7.3(d)). Nessas figuras, o eixo das abcissas indica o tamanho da instância ($|V|$), enquanto o eixo das ordenadas representa o respectivo tempo médio de $|V|$, obtido por meio das médias de tempo em cada valor de σ .

Diferentemente dos gráficos gerados para as instâncias de CHAVES e LORENA [16], os gráficos das Figuras 7.3(a), 7.3(b), 7.3(c) e 7.3(d) não apresentam a superioridade clara de alguma das heurísticas, em termos de crescimento do tempo. Porém, pode-se observar que, para os quatro subgrupos de instâncias, a heurística CS possui um crescimento mais acelerado que o MPO até instâncias de 200 vértices. Já para instâncias com 300 e 500 vértices, a função do MPO cresce mais rapidamente quando comparada à função de crescimento do CS.

7.3 Novas instâncias

Nas seções anteriores, mostrou-se que o modelo Bienstock+ é capaz de resolver todas as instâncias disponíveis na literatura, em um tempo de CPU inferior a 3600

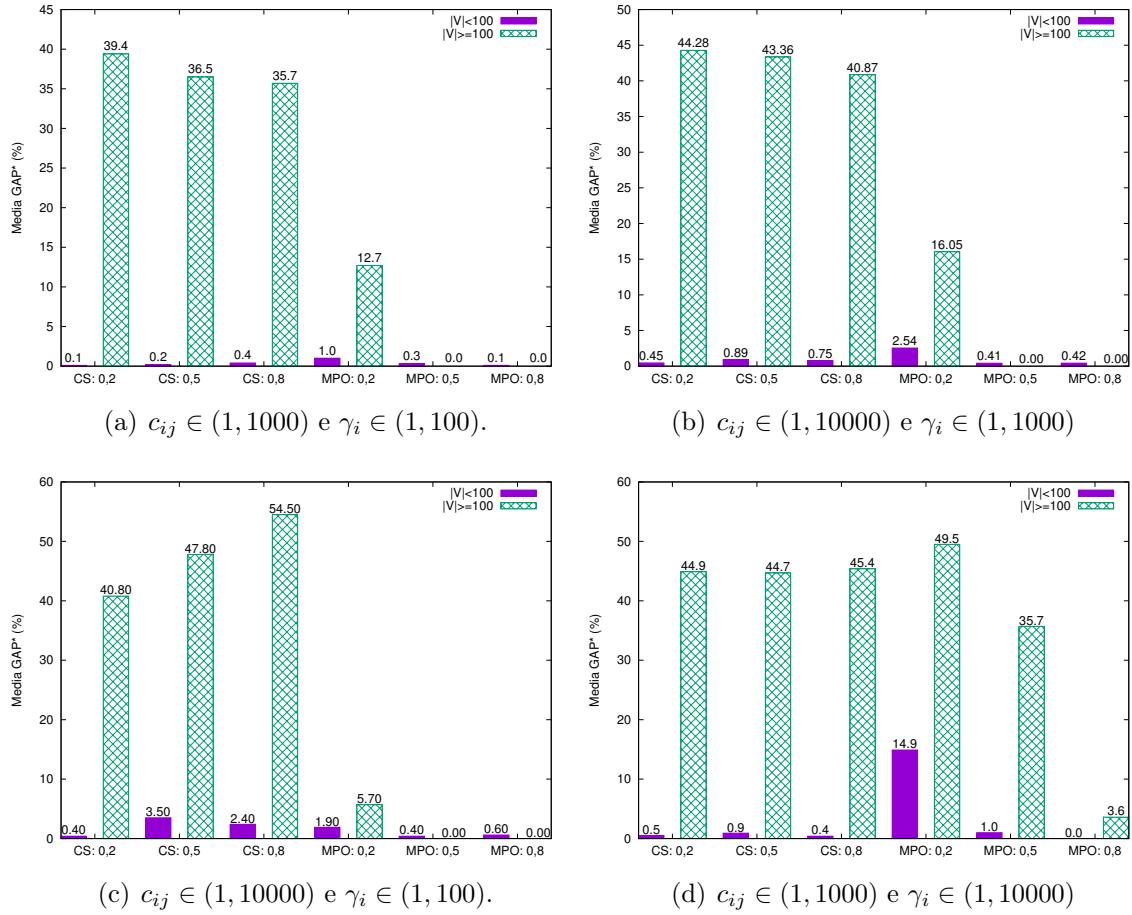


Figura 7.2: gap médio das heurísticas CS e MPO, em relação à solução ótima.

segundos. Portanto, nesta tese é proposto um novo grupo de instâncias, com um número maior de vértices e arestas.

Este novo grupo contém 15 instâncias, adaptadas a partir de problemas-teste do PCV disponíveis no repositório TSPLIB [86]. Assim como em CHAVES e LORENA [16], para ambos, o prêmio p_i e a penalidade γ_i de cada vértice i , foram atribuídos valores aleatórios no intervalo $[1, 100]$; enquanto o *PRIZE* a ser coletado corresponde a 75% do prêmio total da instância. Estas instâncias estão disponíveis em <http://www.deinf.ufma.br/~glaubos/papers/instances/pctsp/new/>.

Mantendo o tempo limite de 3600 segundos, a formulação Bienstock+ não foi capaz de obter nenhuma solução para nenhuma das novas instâncias, devido a problemas relacionados à falta de memória do computador utilizado. Portanto, para tais instâncias foram realizados experimentos com a heurística MPO proposta nesta tese, que manteve os valores de parâmetros dos experimentos anteriores. Os resultados deste experimento são apresentados na Tabela 7.12.

A partir da Tabela 7.12 percebe-se que, de fato, para as estratégias desenvolvidas nesta tese, as novas instâncias são bem mais difíceis que as da literatura. Há casos em que o problema possui 2392 vértices (pr2392), requerendo um tempo médio de

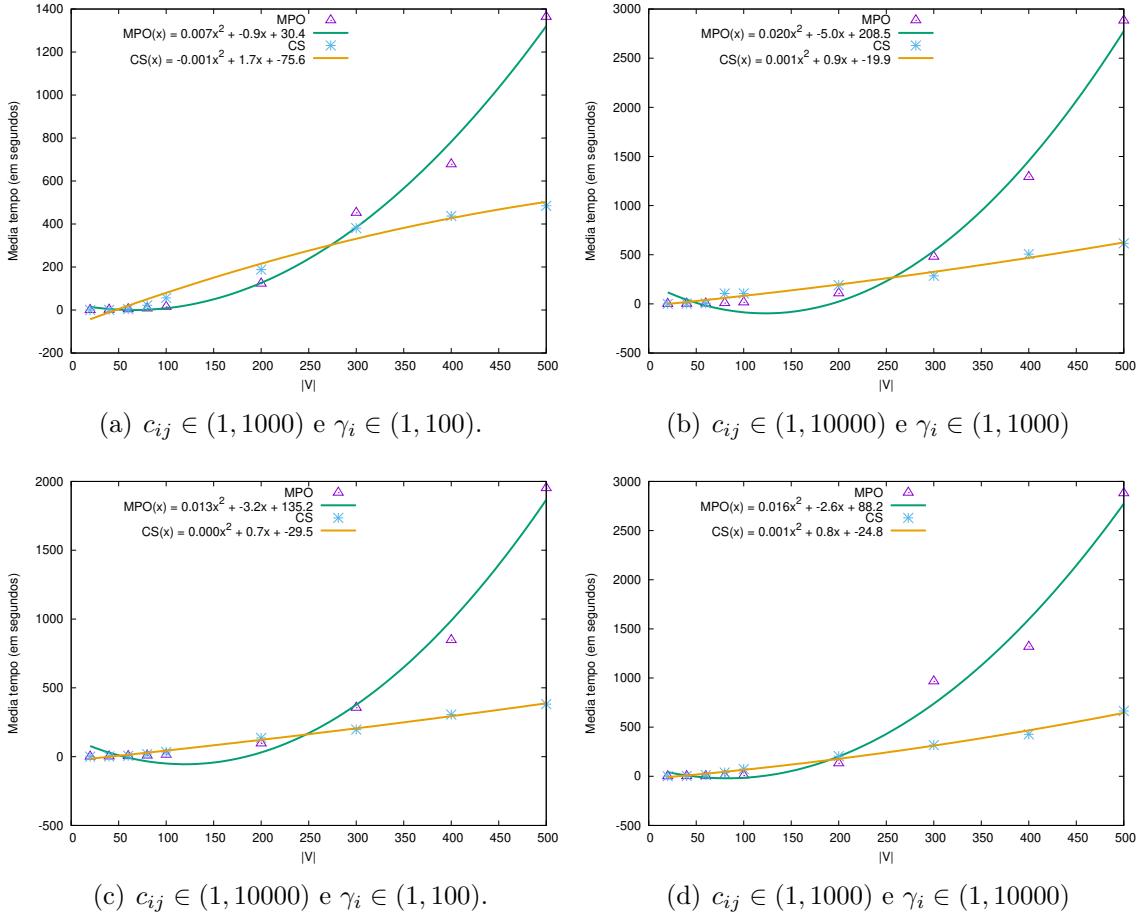


Figura 7.3: Funções de crescimento das heurísticas CS e MPO, para as instâncias de CHAVES e LORENA [19].

142900 segundos para ser resolvido pelo MPO; e também há problemas mais fáceis, como o p654, que mesmo sendo resolvido pelo MPO em 1952.2 segundos, não pode ser resolvido pela formulação Bienstock+.

7.4 Considerações finais

Além do PRRCP, nesta tese foi abordado o PCVCP. Como método exato para a resolução do PCVCP, foram implementadas e testadas quatro formulações já conhecidas da literatura. A formulação de BIENSTOCK *et al.* [13] possui um número exponencial de restrições *cutsets*. Portanto, foi proposto um procedimento de complexidade polinomial para a sua separação.

Em relação à abordagem não exata, utilizou-se uma heurística semelhante ao MPO_S apresentado no PRRCP. Experimentos preliminares com instâncias do PCVCP, mostraram que é mais vantajoso permitir que a heurística seja executada pelo número máximo de iterações, ao contrário do que ocorre com PRRCP, onde a heurística é interrompida assim que E^0 se torna estável.

Tabela 7.12: Resultados do MPO para a novas instâncias do PCVCP.

Instância	MPO		
	Melhor Sol	Média Sol	Média T (s)
d1291	63808	64911.0	20388.2
d657	50856	51226.0	2258.7
f1577	38542	38934.7	60984.8
gr666	243835	250758.0	2282.4
nrw1379	65517	66007.2	26864.3
p654	36525	37047.5	1952.2
pcb1173	65191	65972.4	15168.5
pr1002	237275	237727.0	8230.4
pr2392	371962	376414.0	142900.0
rat783	15902	16245.5	5168.0
rl1889	323509	325744.0	50727.7
u1060	205934	207250.0	5747.0
u1817	74493	75532.0	53103.8
vm1084	225980	227398.0	9594.4
vm1748	317133	319585.0	40351.9

Após a definição das estratégias para a solução do PCVCP, experimentos foram conduzidos em três conjuntos de instâncias. Dois desses contém problemas-teste da literatura e disponíveis para *download*, e o terceiro possui instâncias propostas nesta tese.

Para o primeiro grupo de instâncias [16], a formulação Bienstock+ foi superior às demais da literatura, resolvendo todos os casos em um menor tempo de processamento. As formulações da literatura não foram capazes de resolver instâncias com mais de 50 nós, utilizando um tempo limite de 3600 segundos.

Ainda no primeiro grupo de instâncias, comparou-se o desempenho da heurística proposta com a melhor heurística da literatura (CS*). Em geral, o MPO foi superior ao CS* nos quesitos melhor solução e média de solução, principalmente para as instâncias de maior porte. Para uma comparação justa dos tempos de CPU despendidos, foi realizada uma conversão dos tempos do CS* considerando os processadores utilizados para executar as heurísticas. Após a conversão dos tempos, verificou-se que o CS* é mais rápido para a metade das instâncias de menor porte, enquanto o MPO é mais rápido para as instâncias maiores.

Nesta comparação também foi apresentado o *gap* das melhores soluções das heurísticas, em relação ao ótimo, e foi observado que o MPO e o CS* atingem o ótimo para, respectivamente, 84.61% e 53.84% dos casos. Por fim, foi realizado um experimento para verificar o crescimento do tempo de CPU da heurística, em relação ao tamanho da instância ($|V|$). Com este experimento, foi constatado que a função de crescimento do MPO é suave, em relação à do CS*, que cresce mais rapidamente.

Para o segundo grupo de instâncias [19], inicialmente foram conduzidos os mesmos experimentos com as quatro formulações, e a formulação Bienstock+ resolveu todas as 27 instâncias de cada um dos quatro subgrupos. À medida que o modelo Bienstock+ resolveu todos os casos em até 300 segundos, as outras formulações da literatura não resolveram instâncias com mais de 60 vértices. Considerando os quatro subgrupos e o desempenho de todos os modelos, o subgrupo mais difícil foi o $c_{ij} \in [1, 1000]$ e $\gamma_i \in [1, 100]$ com 43 casos, em que foi encontrada uma solução ótima, ao passo que mais fácil foi o $c_{ij} \in [1, 10000]$ e $\gamma_i \in [1, 1000]$, com 58 instâncias resolvidas.

Experimentos com o MPO também foram realizados para as instâncias de CHAVES e LORENA [19], e comparados com os melhores resultados da literatura (heurística CS [19]). Os resultados desta comparação mostraram que o MPO é superior para os três primeiros subgrupos de instâncias, à medida que o CS vence na maioria das instâncias do quarto subgrupo.

Além da superioridade do MPO no número de vitórias em termos de qualidade de solução, a quantidade de soluções ótimas alcançadas pelo CS sempre é inferior, à quantidade atingida pelo MPO. Outra observação é que, o MPO encontra o ótimo em instâncias de diversos tamanhos, enquanto que o CS só atinge o ótimo para problemas com até 80 vértices.

Como estas duas heurísticas foram executadas em ambientes distintos, e para termos uma comparação mais justa em termos de tempo despendido, a mesma conversão feita para o primeiro grupo de instâncias, foi realizada para os tempos de processamento do CS. Assim, como no grupo anterior de instâncias, também foi realizada uma análise acerca do crescimento do tempo de CPU utilizado, em relação ao tamanho das instâncias. Esta análise mostrou que a função de tempo do MPO cresce inicialmente mais lenta que a do CS mas, a partir de 300 vértices, cresce mais rapidamente.

Após os experimentos com as instâncias da literatura, foi constatado que o modelo Bienstock+ não encontrou dificuldades em resolvê-las. Portanto, nesta tese foram propostos 15 novos problemas-teste com um número de vértices maiores que os da literatura, de modo a verificar melhor o desempenho desta formulação. Após realizar experimentos com a formulação Bienstock+, percebeu-se que, para nenhuma instância proposta, o modelo conseguiu atingir uma solução viável devido a problemas de memória do computador. Desta forma, foi aplicada a heurística MPO que não teve problemas em encontrar soluções sub-ótimas, para nenhuma das dez execuções de cada instância.

Como futuras investigações para o PCVCP, pretende-se realizar um estudo mais aprofundado sobre a relação do conjunto E^0 com as características do problema; verificar a viabilidade da utilização de mineração de dados para o PCVCP; e aplicar a mesma heurística MPO para outras variantes do PCV, já que vários desses problemas

diferem entre si apenas na função objetivo ou em poucas restrições.

Capítulo 8

Apêndice A

Tabela 8.1: Resultados do local branching com $\delta = 3$.

Instâncias	LB com $\delta = 3$				
	Melhor Sol.	Média Sol.	Média T (s)	Dif _{ms}	Dif _t
att48_R9_T10_W29_25	2291	2291.00	0.25	0.00	69.98
berlin52_R9_T11_W32_25	4861	4861.00	0.33	0.00	61.17
bier127_R24_T77_W26_25	98597	98684.80	78.17	0.02	-2.64
brazil58_R18_T20_W20_25	19417	19417.00	4.20	0.00	-8.53
brg180_R36_T36_W108_25	550	550.00	21.12	0.00	1.82
ch130_R44_T43_W43_25	4383	4383.00	15.85	0.00	26.78
ch150_R50_T50_W50_25	4778	4797.60	49.28	0.05	14.35
d198_R39_T40_W119_25	10916	10916.00	60.01	0.00	-18.39
eil51_R17_T17_W17_25	273	273.00	1.07	0.00	57.72
eil76_R14_T46_W16_25	387	387.00	6.30	0.00	47.98
eil101_R19_T21_W61_25	356	356.00	4.31	0.00	43.05
gr48_R16_T16_W16_25	3571	3571.00	0.47	0.00	74.79
gr96_R18_T58_W20_25	42746	42746.00	30.51	0.01	0.01
gr120_R40_T40_W40_25	4301	4301.00	14.74	0.00	46.55
gr137_R26_T28_W83_25	39305	39305.00	25.21	0.00	-61.00
hk48_R9_T29_W10_25	9708	9708.00	2.30	0.01	38.92
kroA100_R20_T20_W60_25	10848	10848.00	5.14	0.00	4.60
kroA150_R50_T50_W50_25	14382	14429.20	113.93	0.17	2.08
kroB100_R20_T60_W20_25	18406	18406.00	24.12	0.00	17.95
kroB150_R30_T30_W90_25	11467	11476.60	37.95	0.01	-70.21
kroC100_R20_T60_W20_25	16785	16785.00	26.05	0.00	16.48
kroE100_R20_T60_W20_25	17199	17204.00	23.26	0.01	17.50
lin105_R21_T21_W63_25	8893	8893.00	7.91	0.00	-5.01
pr76_R14_T16_W46_25	66007	66007.00	1.45	0.00	43.43
pr107_R20_T65_W22_25	40994	40994.00	33.51	0.00	-0.20
pr124_R42_T41_W41_25	35465	35465.00	60.67	0.00	0.78
pr136_R26_T28_W82_25	47620	47620.00	8.29	0.00	26.02
pr144_R28_T87_W29_25	46940	46940.00	88.26	0.00	-1.71
pr152_R50_T51_W51_25	50667	50667.00	87.21	0.00	-11.65
rat99_R19_T20_W60_25	792	792.00	2.22	0.00	65.03
rd100_R34_T33_W33_25	5552	5552.00	16.32	0.00	1.64
si175_R35_T35_W105_25	4845	4850.20	69.75	0.00	0.92
st70_R14_T14_W42_25	419	419.00	1.80	0.00	21.29
u159_R31_T96_W32_25	35442	35442.00	86.10	0.00	-2.15
gr229_R45_T46_W138_25	61278	61304.20	110.28	0.01	-38.38
pr264_R52_T53_W159_25	27693	27693.00	156.18	0.00	-11.35
ts225_R45_T45_W135_25	87889	88290.40	33.80	-0.27	27.53

Nas Tabelas 8.4, 8.5, 8.6, 8.7, 8.8 e 8.9, para cada sequência, as três primeiras colunas apresentam o tamanho resultante dos conjuntos de vértices R , T e W após a aplicação das regras. Na quarta coluna, é apresentado o número total de reduções executadas em cada instância, isto é, o número de vértices que foram removidos do

Tabela 8.2: Resultados do local branching com $\delta = 4$.

Instâncias	LB com $\delta = 4$				
	Melhor Sol.	Média Sol.	Média T (s)	Dif _{ms}	Dif _t
att48_R9_T10_W29_25	2291	2291.00	0.22	0.00	73.82
berlin52_R9_T11_W32_25	4861	4861.00	0.31	0.00	63.74
bier127_R24_T77_W26_25	98531	98634.20	79.87	0.07	-4.88
brazil58_R18_T20_W20_25	19417	19417.00	4.15	0.00	-7.24
brg180_R36_T36_W108_25	550	550.00	21.19	0.00	1.51
ch130_R44_T43_W43_25	4383	4383.00	16.00	0.00	26.08
ch150_R50_T50_W50_25	4778	4797.60	49.58	0.05	13.82
d198_R39_T40_W119_25	10916	10916.00	60.52	0.00	-19.39
eil51_R17_T17_W17_25	273	273.00	1.07	0.00	57.57
eil76_R14_T46_W16_25	387	387.00	6.27	0.00	48.18
eil101_R19_T21_W61_25	356	356.00	4.30	0.00	43.10
gr48_R16_T16_W16_25	3571	3571.00	0.45	0.00	76.07
gr96_R18_T58_W20_25	42746	42746.00	31.16	0.01	-2.15
gr120_R40_T40_W40_25	4301	4301.00	14.81	0.00	46.28
gr137_R26_T28_W83_25	39305	39305.00	17.16	0.00	-9.63
hk48_R9_T29_W10_25	9708	9708.00	2.31	0.01	38.49
kroA100_R20_T20_W60_25	10848	10848.00	5.16	0.00	4.23
kroA150_R50_T50_W50_25	14382	14411.20	119.79	0.29	-2.95
kroB100_R20_T60_W20_25	18406	18406.00	24.22	0.00	17.63
kroB150_R30_T30_W90_25	11467	11476.60	30.69	0.01	-37.64
kroC100_R20_T60_W20_25	16785	16785.00	26.56	0.00	14.84
kroE100_R20_T60_W20_25	17199	17204.00	23.63	0.01	16.20
lin105_R21_T21_W63_25	8893	8893.00	8.50	0.00	-12.92
pr76_R14_T16_W46_25	66007	66007.00	1.45	0.00	43.27
pr107_R20_T65_W22_25	40994	40994.00	34.20	0.00	-2.25
pr124_R42_T41_W41_25	35465	35465.00	61.43	0.00	-0.45
pr136_R26_T28_W82_25	47620	47620.00	8.43	0.00	24.80
pr144_R28_T87_W29_25	46940	46940.00	90.31	0.00	-4.09
pr152_R50_T51_W51_25	50667	50667.00	96.21	0.00	-23.18
rat99_R19_T20_W60_25	792	792.00	2.26	0.00	64.40
rd100_R34_T33_W33_25	5552	5552.00	16.25	0.00	2.07
si175_R35_T35_W105_25	4845	4845.00	69.29	0.11	1.57
st70_R14_T14_W42_25	419	419.00	1.79	0.00	21.64
u159_R31_T96_W32_25	35442	35442.00	86.81	0.00	-2.99
gr229_R45_T46_W138_25	61278	61304.20	111.98	0.01	-40.51
pr264_R52_T53_W159_25	27693	27693.00	156.20	0.00	-11.37
ts225_R45_T45_W135_25	87889	88290.40	34.09	-0.27	26.91

problema ou transformados em obrigatórios. Por fim, a quinta coluna apresenta o tempo despendido pela aplicação da sequência. O tamanho original desses conjuntos pode ser verificado no nome da instância.

Tabela 8.4: Resultados da aplicação das sequências A e B em grafos completos.

Instância	A					B				
	R	T	W	# Red.	T (s)	R	T	W	# Red.	T (s)
att48_R9_T10_W29_25	5	14	12	25	0.02	5	14	12	25	0.03
berlin52_R9_T11_W32_25	7	13	10	26	0.04	7	13	10	26	0.02
bier127_R24_T77_W26_25	24	77	0	26	69.93	24	77	0	26	70.24
brazil58_R18_T20_W20_25	18	20	9	11	0.3	18	20	9	11	0.28
brg180_R36_T36_W108_25	36	36	0	108	17.89	36	36	0	108	17.89
ch130_R44_T43_W43_25	7	61	24	74	30.57	26	61	24	55	36.57
ch150_R50_T50_W50_25	20	67	28	69	69.46	33	67	28	56	82.58
d198_R39_T40_W119_25	39	40	0	119	31.3	39	40	0	119	31.37
eil51_R17_T17_W17_25	17	17	14	3	0.17	17	17	14	3	0.15
eil76_R14_T46_W16_25	14	46	6	10	2.93	14	46	6	10	2.93
eil101_R19_T21_W61_25	18	22	16	47	0.6	18	22	16	47	0.64
gr48_R16_T16_W16_25	14	18	10	10	0.13	14	18	10	10	0.11
gr96_R18_T58_W20_25	18	58	0	20	14.67	18	58	0	20	14.64
gr120_R40_T40_W40_25	13	52	23	56	18.78	28	52	23	41	22.34

Tabela 8.4: Resultados da aplicação das sequências A e B em grafos completos.

Instância	A						B					
	R	T	W	# Red.	T (s)	R	T	W	# Red.	T (s)		
gr137_R26_T28_W83_25	26	28	0	83	3.52	26	28	0	83	3.49		
hk48_R9_T29_W10_25	9	29	3	7	0.23	9	29	3	7	0.22		
kroA100_R20_T20_W60_25	20	20	1	59	0.6	20	20	1	59	0.57		
kroA150_R50_T50_W50_25	50	50	0	50	74.04	50	50	0	50	73.98		
kroB100_R20_T60_W20_25	20	60	0	20	18.36	20	60	0	20	18.36		
kroB150_R30_T30_W90_25	30	30	6	84	6.14	30	30	6	84	6.14		
kroC100_R20_T60_W20_25	20	60	0	20	16.04	20	60	0	20	15.97		
kroE100_R20_T60_W20_25	20	60	0	20	18.58	20	60	0	20	18.53		
lin105_R21_T21_W63_25	21	21	0	63	0.8	21	21	0	63	0.78		
pr76_R14_T16_W46_25	14	16	0	46	0.15	14	16	0	46	0.12		
pr107_R20_T65_W22_25	9	71	13	26	24.89	14	71	13	21	21		
pr124_R42_T41_W41_25	42	41	0	41	27.13	42	41	0	41	26.36		
pr136_R26_T28_W82_25	26	28	11	71	3.38	26	28	11	71	3.38		
pr144_R28_T87_W29_25	28	87	0	29	162.56	28	87	0	29	162.31		
pr152_R50_T51_W51_25	50	51	0	51	82	50	51	0	51	81.91		
rat99_R19_T20_W60_25	6	33	21	65	0.55	6	33	21	65	0.46		
rd100_R34_T33_W33_25	32	35	18	19	6.43	32	35	18	19	6.83		
si175_R35_T35_W105_25	31	39	43	70	15.33	31	39	43	70	17.22		
st70_R14_T14_W42_25	12	16	12	34	0.08	12	16	12	34	0.09		
u159_R31_T96_W32_25	28	99	17	21	288.37	28	99	17	21	276.24		
gr229_R45_T46_W138_25	45	46	0	138	74.13	45	46	0	138	74.15		
pr264_R52_T53_W159_25	52	53	0	159	198.5	52	53	0	159	198.74		
ts225_R45_T45_W135_25	18	70	39	148	73.41	20	70	39	146	79.56		

Tabela 8.5: Resultados da aplicação das sequências C e D em grafos completos.

Instância	C						D					
	R	T	W	# Red.	T (s)	R	T	W	# Red.	T (s)		
att48_R9_T10_W29_25	5	14	4	33	0.00	5	14	4	33	0.00		
berlin52_R9_T11_W32_25	7	13	2	34	0.00	7	13	2	34	0.00		
bier127_R24_T77_W26_25	24	77	0	26	0.00	24	77	0	26	0.00		
brazil58_R18_T20_W20_25	18	20	9	11	0.00	18	20	9	11	0.00		
brg180_R36_T36_W108_25	36	36	0	108	0.00	36	36	0	108	0.00		
ch130_R44_T43_W43_25	5	61	2	98	0.00	5	61	2	98	0.00		
ch150_R50_T50_W50_25	16	67	7	94	0.00	16	67	7	94	0.00		
d198_R39_T40_W119_25	39	40	0	119	0.00	39	40	0	119	0.00		
eil51_R17_T17_W17_25	17	17	14	3	0.00	17	17	14	3	0.00		
eil76_R14_T46_W16_25	14	46	6	10	0.00	14	46	6	10	0.00		
eil101_R19_T21_W61_25	18	22	12	51	0.00	18	22	12	51	0.00		
gr48_R16_T16_W16_25	14	18	6	14	0.00	14	18	6	14	0.00		
gr96_R18_T58_W20_25	18	58	0	20	0.00	18	58	0	20	0.00		
gr120_R40_T40_W40_25	11	52	8	73	0.00	11	52	8	73	0.00		
gr137_R26_T28_W83_25	26	28	0	83	0.00	26	28	0	83	0.00		
hk48_R9_T29_W10_25	9	29	3	7	0.00	9	29	3	7	0.00		
kroA100_R20_T20_W60_25	20	20	1	59	0.00	20	20	1	59	0.00		
kroA150_R50_T50_W50_25	50	50	0	50	0.00	50	50	0	50	0.00		
kroB100_R20_T60_W20_25	20	60	0	20	0.00	20	60	0	20	0.00		
kroB150_R30_T30_W90_25	30	30	6	84	0.00	30	30	6	84	0.00		
kroC100_R20_T60_W20_25	20	60	0	20	0.00	20	60	0	20	0.00		
kroE100_R20_T60_W20_25	20	60	0	20	0.00	20	60	0	20	0.00		
lin105_R21_T21_W63_25	21	21	0	63	0.00	21	21	0	63	0.00		
pr76_R14_T16_W46_25	14	16	0	46	0.00	14	16	0	46	0.00		
pr107_R20_T65_W22_25	3	71	1	44	0.00	3	71	1	44	0.00		
pr124_R42_T41_W41_25	42	41	0	41	0.00	42	41	0	41	0.00		

Tabela 8.5: Resultados da aplicação das sequências C e D em grafos completos.

Instância	C						D					
	R	T	W	# Red.	T (s)	R	T	W	# Red.	T (s)		
pr136_R26_T28_W82_25	26	28	11	71	0.00	26	28	11	71	0.00		
pr144_R28_T87_W29_25	28	87	0	29	0.00	28	87	0	29	0.00		
pr152_R50_T51_W51_25	50	51	0	51	0.00	50	51	0	51	0.00		
rat99_R19_T20_W60_25	4	33	2	86	0.00	4	33	2	86	0.00		
rd100_R34_T33_W33_25	32	35	15	22	0.00	32	35	15	22	0.00		
si175_R35_T35_W105_25	31	39	26	87	0.00	31	39	26	87	0.00		
st70_R14_T14_W42_25	12	16	6	40	0.00	12	16	6	40	0.00		
u159_R31_T96_W32_25	28	99	13	25	0.00	28	99	13	25	0.00		
gr229_R45_T46_W138_25	45	46	0	138	0.02	45	46	0	138	0.02		
pr264_R52_T53_W159_25	52	53	0	159	0.00	52	53	0	159	0.00		
ts225_R45_T45_W135_25	15	70	8	182	0.00	15	70	8	182	0.00		

Tabela 8.6: Resultados da aplicação das sequências E e F em grafos completos.

Instância	E						F					
	R	T	W	# Red.	T (s)	R	T	W	# Red.	T (s)		
att48_R9_T10_W29_25	5	14	4	33	0.02	4	14	12	26	0.01		
berlin52_R9_T11_W32_25	7	13	2	34	0.04	7	13	10	26	0.02		
bier127_R24_T77_W26_25	24	77	0	26	66.69	24	77	0	26	67.12		
brazil58_R18_T20_W20_25	18	20	9	11	0.00	18	20	9	11	0.28		
brg180_R36_T36_W108_25	36	36	0	108	16.84	36	36	0	108	17.12		
ch130_R44_T43_W43_25	5	61	2	98	38.78	7	61	24	74	31.68		
ch150_R50_T50_W50_25	16	67	7	94	88.29	20	67	28	69	73.35		
d198_R39_T40_W119_25	39	40	0	119	28.63	39	40	0	119	28.91		
eil51_R17_T17_W17_25	17	17	14	3	0.00	17	17	14	3	0.14		
eil76_R14_T46_W16_25	14	46	6	10	0.00	14	46	6	10	2.94		
eil101_R19_T21_W61_25	18	22	12	51	0.74	18	22	16	47	0.60		
gr48_R16_T16_W16_25	14	18	6	14	0.13	14	18	10	10	0.12		
gr96_R18_T58_W20_25	18	58	0	20	19.54	18	58	0	20	13.58		
gr120_R40_T40_W40_25	11	52	8	73	22.28	13	52	23	56	18.36		
gr137_R26_T28_W83_25	26	28	0	83	0.00	26	28	0	83	3.39		
hk48_R9_T29_W10_25	9	29	3	7	0.00	9	29	3	7	0.24		
kroA100_R20_T20_W60_25	20	20	1	59	0.00	20	20	1	59	0.59		
kroA150_R50_T50_W50_25	50	50	0	50	74.28	50	50	0	50	74.87		
kroB100_R20_T60_W20_25	20	60	0	20	17.92	20	60	0	20	17.93		
kroB150_R30_T30_W90_25	30	30	6	84	0.00	30	30	6	84	6.40		
kroC100_R20_T60_W20_25	20	60	0	20	0.00	20	60	0	20	16.38		
kroE100_R20_T60_W20_25	20	60	0	20	17.66	20	60	0	20	17.71		
lin105_R21_T21_W63_25	21	21	0	63	0.00	21	21	0	63	0.83		
pr76_R14_T16_W46_25	14	16	0	46	0.00	14	16	0	46	0.14		
pr107_R20_T65_W22_25	3	71	1	44	22.09	9	71	13	26	26.02		
pr124_R42_T41_W41_25	42	41	0	41	23.49	42	41	0	41	23.86		
pr136_R26_T28_W82_25	26	28	11	71	0.00	26	28	11	71	3.53		
pr144_R28_T87_W29_25	28	87	0	29	152.37	28	87	0	29	152.93		
pr152_R50_T51_W51_25	50	51	0	51	189.03	50	51	0	51	73.89		
rat99_R19_T20_W60_25	4	33	2	86	0.51	6	33	21	65	0.52		
rd100_R34_T33_W33_25	32	35	15	22	6.88	32	35	18	19	6.53		
si175_R35_T35_W105_25	31	39	26	87	21.21	31	39	43	70	16.47		
st70_R14_T14_W42_25	12	16	6	40	0.11	12	16	12	34	0.08		
u159_R31_T96_W32_25	28	99	13	25	262.72	28	99	17	21	272.00		
gr229_R45_T46_W138_25	45	46	0	138	69.32	45	46	0	138	69.69		
pr264_R52_T53_W159_25	52	53	0	159	0.00	52	53	0	159	191.07		
ts225_R45_T45_W135_25	15	70	8	182	96.37	18	70	39	148	82.17		

Tabela 8.3: Resultados do local branching com $\delta = 5$

Instâncias	LB com $\delta = 5$					
	Melhor Sol.	Média Sol.	Média T (s)	Dif _{ms}	Dif _t	
att48_R9_T10_W29_25	2291	2291.00	0.22	0.00	73.82	
berlin52_R9_T11_W32_25	4861	4861.00	0.31	0.00	63.74	
bier127_R24_T77_W26_25	98453	98636.60	79.96	0.07	-4.88	
brazil58_R18_T20_W20_25	19417	19417.00	4.21	0.00	-7.24	
brg180_R36_T36_W108_25	550	550.00	21.38	0.00	1.51	
ch130_R44_T43_W43_25	4383	4383.00	16.10	0.00	26.08	
ch150_R50_T50_W50_25	4778	4795.80	49.89	0.05	13.82	
d198_R39_T40_W119_25	10916	10916.00	60.81	0.00	-19.39	
eil51_R17_T17_W17_25	273	273.00	1.06	0.00	57.57	
eil76_R14_T46_W16_25	387	387.00	6.28	0.00	48.18	
eil101_R19_T21_W61_25	356	356.00	4.37	0.00	43.10	
gr48_R16_T16_W16_25	3571	3571.00	0.45	0.00	76.07	
gr96_R18_T58_W20_25	42746	42746.00	32.69	0.01	-2.15	
gr120_R40_T40_W40_25	4301	4301.00	14.84	0.00	46.28	
gr137_R26_T28_W83_25	39305	39305.00	19.32	0.00	-9.63	
hk48_R9_T29_W10_25	9708	9708.00	2.32	0.01	38.49	
kroA100_R20_T20_W60_25	10848	10848.00	5.20	0.00	4.23	
kroA150_R50_T50_W50_25	14382	14404.40	136.00	0.29	-2.95	
kroB100_R20_T60_W20_25	18406	18406.00	24.55	0.00	17.63	
kroB150_R30_T30_W90_25	11467	11476.60	28.47	0.01	-37.64	
kroC100_R20_T60_W20_25	16785	16785.00	26.69	0.00	14.84	
kroE100_R20_T60_W20_25	17199	17204.00	23.77	0.01	16.20	
lin105_R21_T21_W63_25	8893	8893.00	9.92	0.00	-12.92	
pr76_R14_T16_W46_25	66007	66007.00	1.48	0.00	43.27	
pr107_R20_T65_W22_25	40994	40994.00	35.00	0.00	-2.25	
pr124_R42_T41_W41_25	35465	35465.00	61.95	0.00	-0.45	
pr136_R26_T28_W82_25	47620	47620.00	8.50	0.00	24.80	
pr144_R28_T87_W29_25	46940	46940.00	93.63	0.00	-4.09	
pr152_R50_T51_W51_25	50667	50667.00	97.42	0.00	-23.18	
rat99_R19_T20_W60_25	792	792.00	2.22	0.00	64.40	
rd100_R34_T33_W33_25	5552	5552.00	16.35	0.00	2.07	
si175_R35_T35_W105_25	4845	4845.00	69.14	0.11	1.57	
st70_R14_T14_W42_25	419	419.00	1.78	0.00	21.64	
u159_R31_T96_W32_25	35442	35442.00	86.90	0.00	-2.99	
gr229_R45_T46_W138_25	61278	61304.20	112.20	0.01	-40.51	
pr264_R52_T53_W159_25	27693	27693.00	157.20	0.00	-11.37	
ts225_R45_T45_W135_25	87889	88290.40	34.04	-0.27	26.91	

Tabela 8.7: Resultados da aplicação das sequências A e B em grafos esparsos.

Instância	A						B					
	R	T	W	# Red.	T (s)	R	T	W	# Red.	T (s)		
att48_R9_T10_W29_25	2	17	12	31	0.01	2	17	12	31	0.01		
berlin52_R9_T11_W32_25	3	16	10	29	0.01	3	16	10	29	0.01		
bier127_R24_T77_W26_25	24	77	0	101	12.26	24	77	0	101	12.45		
brazil58_R18_T20_W20_25	18	20	9	47	0.38	18	20	9	47	0.38		
brg180_R36_T36_W108_25	36	36	0	72	4.57	36	36	0	72	4.54		
ch130_R44_T43_W43_25	5	64	24	93	3.46	5	64	24	93	2.87		
ch150_R50_T50_W50_25	21	69	28	118	15.09	21	69	28	118	14.92		
d198_R39_T40_W119_25	38	40	0	78	4.05	38	40	0	78	4.04		
eil51_R17_T17_W17_25	14	19	14	47	0.12	14	19	14	47	0.12		
eil76_R14_T46_W16_25	9	48	6	63	0.47	9	48	6	63	0.47		
eil101_R19_T21_W61_25	11	28	16	55	0.44	11	28	16	55	0.42		
gr48_R16_T16_W16_25	1	25	9	35	0.02	3	23	9	35	0.01		
gr96_R18_T58_W20_25	18	58	0	76	1.59	18	58	0	76	1.71		
gr120_R40_T40_W40_25	12	57	23	92	2.76	12	57	23	92	2.7		
gr137_R26_T28_W83_25	26	28	0	54	0.69	26	28	0	54	0.69		
hk48_R9_T29_W10_25	8	29	3	40	0.06	8	29	3	40	0.06		
kroA100_R20_T20_W60_25	20	20	1	41	0.15	20	20	1	41	0.15		
kroA150_R50_T50_W50_25	50	50	0	100	13.06	50	50	0	100	12.83		
kroB100_R20_T60_W20_25	19	60	0	79	3.25	19	60	0	79	3.25		
kroB150_R30_T30_W90_25	30	30	6	66	3.57	30	30	6	66	3.54		

Tabela 8.7: Resultados da aplicação das sequências A e B em grafos esparsos.

Instância	A						B					
	R	T	W	# Red.	T (s)	R	T	W	# Red.	T (s)		
kroC100_R20_T60_W20_25	20	60	0	80	2.83	20	60	0	80	2.87		
kroE100_R20_T60_W20_25	18	60	0	78	2.56	18	60	0	78	2.56		
lin105_R21_T21_W63_25	21	21	0	42	0.17	21	21	0	42	0.17		
pr76_R14_T16_W46_25	13	16	0	29	0.03	13	16	0	29	0.03		
pr107_R20_T65_W22_25	4	77	13	94	2.95	4	77	13	94	2.88		
pr124_R42_T41_W41_25	42	41	0	83	3.66	42	41	0	83	3.66		
pr136_R26_T28_W82_25	23	29	10	62	1.04	23	29	10	62	0.97		
pr144_R28_T87_W29_25	28	87	0	115	10.95	28	87	0	115	11.66		
pr152_R50_T51_W51_25	50	51	0	101	8.72	50	51	0	101	9.15		
rat99_R19_T20_W60_25	1	40	21	62	0.25	0	39	21	60	0.24		
rd100_R34_T33_W33_25	21	36	18	75	1.60	21	36	18	75	1.56		
si175_R35_T35_W105_25	27	43	43	113	24.94	27	43	43	113	24.89		
st70_R14_T14_W42_25	11	17	12	40	0.09	11	17	12	40	0.08		
u159_R31_T96_W32_25	8	106	16	130	16.12	8	106	16	130	15.48		
gr229_R45_T46_W138_25	45	46	0	91	8.61	45	46	0	91	9.48		
pr264_R52_T53_W159_25	51	53	0	104	26.34	51	53	0	104	26.36		
ts225_R45_T45_W135_25	14	76	39	129	18.63	14	76	39	129	17.51		

Tabela 8.8: Resultados da aplicação das sequências C e D em grafos esparsos.

Instância	C						D					
	R	T	W	# Red.	T (s)	R	T	W	# Red.	T (s)		
att48_R9_T10_W29_25	5	14	4	33	0.00	5	14	4	33	0.00		
berlin52_R9_T11_W32_25	6	13	2	35	0.01	6	13	2	35	0.01		
bier127_R24_T77_W26_25	24	77	0	26	0.00	24	77	0	26	0.00		
brazil58_R18_T20_W20_25	18	20	9	11	0.00	18	20	9	11	0.00		
brg180_R36_T36_W108_25	36	36	0	108	0.00	36	36	0	108	0.00		
ch130_R44_T43_W43_25	5	61	2	98	2.30	5	61	2	98	2.29		
ch150_R50_T50_W50_25	16	67	7	94	8.95	16	67	7	94	8.93		
d198_R39_T40_W119_25	38	40	0	120	13.69	38	40	0	120	13.34		
eil51_R17_T17_W17_25	16	17	14	4	0.04	16	17	14	4	0.04		
eil76_R14_T46_W16_25	11	46	6	13	0.45	11	46	6	13	0.45		
eil101_R19_T21_W61_25	17	22	12	52	0.20	17	22	12	52	0.19		
gr48_R16_T16_W16_25	6	20	2	28	0.01	6	20	2	28	0.01		
gr96_R18_T58_W20_25	18	58	0	20	0.00	18	58	0	20	0.00		
gr120_R40_T40_W40_25	11	52	8	73	1.77	11	52	8	73	1.78		
gr137_R26_T28_W83_25	26	28	0	83	0.00	26	28	0	83	0.00		
hk48_R9_T29_W10_25	8	29	3	8	0.06	8	29	3	8	0.06		
kroA100_R20_T20_W60_25	20	20	1	59	0.00	20	20	1	59	0.00		
kroA150_R50_T50_W50_25	50	50	0	50	0.00	50	50	0	50	0.00		
kroB100_R20_T60_W20_25	19	60	0	21	4.97	19	60	0	21	4.99		
kroB150_R30_T30_W90_25	30	30	6	84	0.00	30	30	6	84	0.00		
kroC100_R20_T60_W20_25	20	60	0	20	0.00	20	60	0	20	0.00		
kroE100_R20_T60_W20_25	18	60	0	22	3.72	18	60	0	22	3.69		
lin105_R21_T21_W63_25	21	21	0	63	0.00	21	21	0	63	0.00		
pr76_R14_T16_W46_25	13	16	0	47	0.07	13	16	0	47	0.07		
pr107_R20_T65_W22_25	3	71	1	44	2.70	3	71	1	44	2.72		
pr124_R42_T41_W41_25	42	41	0	41	0.00	42	41	0	41	0.00		
pr136_R26_T28_W82_25	24	28	10	74	0.98	24	28	10	74	0.96		
pr144_R28_T87_W29_25	28	87	0	29	0.00	28	87	0	29	0.00		
pr152_R50_T51_W51_25	50	51	0	51	0.00	50	51	0	51	0.00		
rat99_R19_T20_W60_25	4	33	2	86	0.00	4	33	2	86	0.00		
rd100_R34_T33_W33_25	22	35	15	32	0.96	22	35	15	32	0.97		
si175_R35_T35_W105_25	31	39	26	87	0.00	31	39	26	87	0.00		
st70_R14_T14_W42_25	12	16	6	40	0.00	12	16	6	40	0.00		

Tabela 8.8: Resultados da aplicação das sequências C e D em grafos esparsos.

Instância	C						D					
	R	T	W	# Red.	T (s)	R	T	W	# Red.	T (s)		
u159_R31_T96_W32_25	15	99	12		39	13.58	15	99	12		39	13.46
gr229_R45_T46_W138_25	45	46	0		138	0.00	45	46	0		138	0.00
ts225_R45_T45_W135_25	45	45	135		225	31.35	15	70	8		182	20.23

Tabela 8.9: Resultados da aplicação das sequências E e F em grafos esparsos.

Instância	E						F					
	R	T	W	# Red.	T (s)	R	T	W	# Red.	T (s)		
att48_R9_T10_W29_25	5	14	4		33	0.01	5	14	12		25	0.00
berlin52_R9_T11_W32_25	6	13	2		35	0.01	6	13	10		27	0.01
bier127_R24_T77_W26_25	24	77	0		26	12.17	24	77	0		26	12.28
brazil58_R18_T20_W20_25	18	20	9		11	0.00	18	20	9		11	0.09
brg180_R36_T36_W108_25	36	36	0		108	4.65	36	36	0		108	4.57
ch130_R44_T43_W43_25	5	61	2		98	1.69	7	61	24		74	2.30
ch150_R50_T50_W50_25	16	67	7		94	9.77	20	67	28		69	8.81
d198_R39_T40_W119_25	38	40	0		120	4.25	38	40	0		120	4.22
eil51_R17_T17_W17_25	16	17	14		4	0.04	16	17	14		4	0.04
eil76_R14_T46_W16_25	11	46	6		13	0.46	11	46	6		13	0.44
eil101_R19_T21_W61_25	17	22	12		52	0.22	17	22	16		48	0.17
gr48_R16_T16_W16_25	6	20	2		28	0.01	6	20	9		21	0.01
gr96_R18_T58_W20_25	18	58	0		20	2.37	18	58	0		20	1.57
gr120_R40_T40_W40_25	11	52	8		73	2.01	13	52	23		56	1.73
gr137_R26_T28_W83_25	26	28	0		83	0.00	26	28	0		83	0.75
hk48_R9_T29_W10_25	8	29	3		8	0.06	8	29	3		8	0.06
kroA100_R20_T20_W60_25	20	20	1		59	0.00	20	20	1		59	0.16
kroA150_R50_T50_W50_25	50	50	0		50	13.05	50	50	0		50	13.38
kroB100_R20_T60_W20_25	19	60	0		21	3.34	19	60	0		21	3.39
kroB150_R30_T30_W90_25	30	30	6		84	0.00	30	30	6		84	1.64
kroC100_R20_T60_W20_25	20	60	0		20	0.00	20	60	0		20	2.98
kroE100_R20_T60_W20_25	18	60	0		22	2.61	18	60	0		22	2.65
lin105_R21_T21_W63_25	21	21	0		63	0.00	21	21	0		63	0.20
pr76_R14_T16_W46_25	13	16	0		47	0.07	13	16	0		47	0.03
pr107_R20_T65_W22_25	3	71	1		44	2.07	9	71	13		26	2.73
pr124_R42_T41_W41_25	42	41	0		41	3.67	42	41	0		41	4.08
pr136_R26_T28_W82_25	24	28	10		74	0.98	24	28	10		74	0.66
pr144_R28_T87_W29_25	28	87	0		29	11.32	28	87	0		29	11.35
pr152_R50_T51_W51_25	50	51	0		51	21.69	50	51	0		51	9.29
rat99_R19_T20_W60_25	4	33	2		86	0.12	6	33	21		65	0.14
rd100_R34_T33_W33_25	22	35	15		32	1.00	22	35	18		29	0.96
si175_R35_T35_W105_25	31	39	26		87	4.41	31	39	43		70	3.38
st70_R14_T14_W42_25	12	16	6		40	0.05	12	16	12		34	0.03
u159_R31_T96_W32_25	15	99	12		39	12.04	15	99	16		35	13.42
gr229_R45_T46_W138_25	45	46	0		138	8.56	45	46	0		138	9.07
ts225_R45_T45_W135_25	15	70	8		182	9.67	18	70	39		148	7.65

Na Tabela 8.13 são apresentados os resultados das formulações matemáticas, executadas após a aplicação das regras de redução. A primeira coluna apresenta o nome da instância. Em seguida, para cada formulação, é apresentada a solução obtida e o ganho percentual de tempo de CPU, em relação à execução sem aplicação das regras. Valores em negrito indicam que se obteve melhoria na solução, e o símbolo + indica que o ótimo foi atingido.

Tabela 8.10: Resultados da aplicação das sequências AA e BB antes e após a remoção de E^0 .

Instância	AA						BB					
	R	T	W	# Red.	T (s)	R	T	W	# Red.	T (s)		
att48_VT9_T10_W29_25	7	7	17	31	0.03	4	14	12	26	0.01		
berlin52_VT9_T11_W32_25	7	5	22	34	0.02	7	13	10	26	0.02		
bier127_VT24_T77_W26_25	0	0	26	26	23.53	24	77	0	26	67.12		
brazil58_VT18_T20_W20_25	0	0	11	11	0.97	18	20	9	11	0.28		
brg180_VT36_T36_W108_25	0	0	108	108	7.98	36	36	0	108	17.12		
ch130_VT44_T43_W43_25	38	19	29	86	6.39	7	61	24	74	31.68		
ch150_VT50_T50_W50_25	32	19	23	74	27.97	20	67	28	69	73.35		
d198_VT39_T40_W119_25	1	0	119	120	9.33	39	40	0	119	28.91		
eil51_VT17_T17_W17_25	3	2	3	8	0.31	17	17	14	3	0.14		
eil76_VT14_T46_W16_25	5	2	10	17	1.21	14	46	6	10	2.94		
eil101_VT19_T21_W61_25	9	7	45	61	0.95	18	22	16	47	0.60		
gr48_VT16_T16_W16_25	14	7	7	28	0.04	14	18	10	10	0.12		
gr96_VT18_T58_W20_25	0	0	20	20	3.63	18	58	0	20	13.58		
gr120_VT40_T40_W40_25	31	17	21	69	5.43	13	52	23	56	18.36		
gr137_VT26_T28_W83_25	1	0	83	84	1.61	26	28	0	83	3.39		
hk48_VT9_T29_W10_25	1	0	7	8	0.15	9	29	3	7	0.24		
kroA100_VT20_T20_W60_25	0	0	59	59	0.46	20	20	1	59	0.59		
kroA150_VT50_T50_W50_25	0	0	50	50	27.18	50	50	0	50	74.87		
kroB100_VT20_T60_W20_25	1	0	20	21	6.78	20	60	0	20	17.93		
kroB150_VT30_T30_W90_25	0	0	84	84	7.43	30	30	6	84	6.40		
kroC100_VT20_T60_W20_25	0	0	20	20	6.86	20	60	0	20	16.38		
kroE100_VT20_T60_W20_25	2	0	20	22	6.09	20	60	0	20	17.71		
lin105_VT21_T21_W63_25	0	0	63	63	0.43	21	21	0	63	0.83		
pr76_VT14_T16_W46_25	2	0	46	48	0.07	14	16	0	46	0.14		
pr107_VT20_T65_W22_25	15	11	11	37	6.89	9	71	13	26	26.02		
pr124_VT42_T41_W41_25	0	0	41	41	8.41	42	41	0	41	23.86		
pr136_VT26_T28_W82_25	3	1	72	76	2.69	26	28	11	71	3.53		
pr144_VT28_T87_W29_25	0	0	29	29	24.5	28	87	0	29	152.93		
pr152_VT50_T51_W51_25	0	0	51	51	23.2	50	51	0	51	73.89		
rat99_VT19_T20_W60_25	16	16	43	75	0.67	6	33	21	65	0.52		
rd100_VT34_T33_W33_25	13	3	15	31	3.59	32	35	18	19	6.53		
si175_VT35_T35_W105_25	8	8	62	78	59.7	31	39	43	70	16.47		
st70_VT14_T14_W42_25	2	2	34	38	0.15	12	16	12	34	0.08		
u159_VT31_T96_W32_25	25	10	16	51	31.85	28	99	17	21	272.00		
gr229_VT45_T46_W138_25	0	0	138	138	20.91	45	46	0	138	69.69		
pr264_VT52_T53_W159_25	2	0	159	161	49.76	52	53	0	159	191.07		
ts225_VT45_T45_W135_25	32	31	102	165	40.51	18	70	39	148	82.17		

Tabela 8.13: Resultados das formulações matemáticas após a aplicação das regras de redução.

Instância	Fluxo			Multi-fluxo			Não-dir		
	Sol.	Dif _t	Sol.	Dif _t	Sol.	Dif _t	Sol.	Dif _t	Sol.
att48_R9_T10_W29_25	2291	65.9	2291	46.2	2291	27.0			
att48_R9_T29_W10_50	3169	0.0	2824	49.7	2824	45.6			
att48_R16_T16_W16_75	2655	0.0	2650	54.3	2650	73.0			
berlin52_R9_T11_W32_25	4861	45.5	4861	46.8	4861	24.9			
berlin52_R16_T18_W18_50	5529	0.0	5186	44.9	5186	85.2			
berlin52_R9_T32_W11_75	8022	0.0	6916	55.1	6916	55.2			
bier127_R24_T77_W26_25	-	0.0	-	0.0	98453	10.2			
bier127_R24_T26_W77_50	121081	0.0	74745	56.1	74745	77.7			
bier127_R43_T42_W42_75	314056	0.0	-	0.0	98311	13.7			
brazil58_R18_T20_W20_25	20351	0.0	19417	46.9	19417	12.6			
brazil58_R11_T35_W12_50	27586	0.0	23312	56.4	23312	48.2			

Tabela 8.13: Resultados das formulações matemáticas após a aplicação das regras de redução.

Instância	Fluxo		Multi-fluxo		Não-dir	
	Sol.	Dif_t	Sol.	Dif_t	Sol.	Dif_t
brazil58_R11_T12_W35_75	20466	0.0	20466	44.9	20466	33.2
brg180_R36_T36_W108_25	-	0.0	-	0.0	550	73.1
brg180_R60_T60_W60_50	-	0.0	-	0.0	2350	95.2
brg180_R36_T108_W36_75	-	0.0	-	0.0	2190	90.7
ch130_R44_T43_W43_25	-	0.0	-	0.0	4383	-16.8
ch130_R26_T26_W78_50	11000	0.0	4139	57.2	4139	39.7
ch130_R26_T78_W26_75	-	0.0	-	0.0	5287	-2.7
ch150_R50_T50_W50_25	-	0.0	-	0.0	4778	37.5
ch150_R30_T90_W30_50	-	0.0	-	0.0	5239	10.3
ch150_R30_T30_W90_75	6695	0.0	4292	54.9	4292	63.5
d198_R39_T40_W119_25	15099	0.0	-	0.0	11214	0.0
d198_R39_T119_W40_50	-	0.0	-	0.0	14260	0.0
d198_R66_T66_W66_75	-	0.0	-	0.0	13653	0.0
eil51_R17_T17_W17_25	273	34.8	273	53.0	273	34.8
eil51_R9_T11_W31_50	248	27.8	248	46.6	248	-28.6
eil51_R9_T31_W11_75	435	0.0	348	46.7	348	46.6
eil76_R14_T46_W16_25	727	0.0	387	52.2	387	73.6
eil76_R24_T26_W26_50	417	0.0	338	55.3	338	85.3
eil76_R14_T16_W46_75	332	71.6	332	52.7	332	86.5
eil101_R19_T21_W61_25	409	0.0	349	85.2	356	42.6
eil101_R19_T61_W21_50	-	0.0	-	0.0	506	15.2
eil101_R33_T34_W34_75	1094	0.0	462	39.2	462	60.8
gr48_R16_T16_W16_25	3571	8.3	3542	51.9	3571	67.7
gr48_R9_T10_W29_50	3811	55.8	3524	46.3	3524	26.8
gr48_R9_T29_W10_75	4886	0.0	4395	-187.5	4415	50.3
gr96_R18_T58_W20_25	71100	0.0	42746	54.6	42746	11.2
gr96_R32_T32_W32_50	73369	0.0	35461	61.4	35461	1.1
gr96_R18_T20_W58_75	32411	0.0	30015	46.9	30015	68.3
gr120_R40_T40_W40_25	15081	0.0	4229	55.2	4301	74.7
gr120_R24_T72_W24_50	-	0.0	-	0.0	5159	55.6
gr120_R24_T24_W72_75	7277	0.0	4275	56.1	4275	66.5
gr137_R26_T28_W83_25	66989	0.0	39305⁺	48.9	39305	-17.22
gr137_R45_T46_W46_50	-	0.0	-	0.0	50905	0.0
gr137_R26_T83_W28_75	-	0.0	-	0.0	59093	0.0
hk48_R9_T29_W10_25	10368	0.0	9708	52.9	9708	4.5

Tabela 8.13: Resultados das formulações matemáticas após a aplicação das regras de redução.

Instância	Fluxo		Multi-fluxo		Não-dir	
	Sol.	Dif_t	Sol.	Dif_t	Sol.	Dif_t
hk48_R16_T16_W16_50	7837	-6.5	7593	52.4	7593	55.7
hk48_R9_T10_W29_75	7803	88.9	7803	38.3	7803	55.0
kroA100_R20_T20_W60_25	17757	0.0	10848	51.0	10848	53.5
kroA100_R20_T60_W20_50	-	0.0	16544 ⁺	32.4	16544	-1.5
kroA100_R34_T33_W33_75	51428	0.0	18107	0.0	14641	0.0
kroA150_R50_T50_W50_25	-	0.0	-	0.0	16158	0.0
kroA150_R30_T30_W90_50	18713	0.0	13329	0.0	13239	0.0
kroA150_R30_T90_W30_75	-	0.0	-	-0.1	21974 ⁺	0.0
kroA200_R40_T120_W40_25	-	0.0	-	-0.2	25550	0.0
kroA200_R68_T66_W66_50	-	0.0	-	-0.1	20143	0.0
kroA200_R40_T40_W120_75	71979	0.0	-	0.0	15197	82.4
kroB100_R20_T60_W20_25	40728	0.0	-	0.0	18544	0.0
kroB100_R34_T33_W33_50	24403	0.0	20617	0.0	14257	0.0
kroB100_R34_T33_W33_75	24403	0.0	19195	0.0	14768	-4.4
kroB150_R30_T30_W90_25	14769	0.0	14279	0.0	12034	0.0
kroB150_R50_T50_W50_50	-	0.0	-	-0.1	17228	0.0
kroB150_R30_T90_W30_75	-	0.0	-	-0.1	21489	-15.4
kroB200_R40_T120_W40_25	-	0.0	-	-0.2	28081	0.0
kroB200_R40_T40_W120_50	31941	0.0	-	0.0	14393	0.0
kroB200_R68_T66_W66_75	-	0.0	-	-0.1	20778	0.0
kroC100_R20_T60_W20_25	97982	0.0	16785	59.6	16785	5.1
kroC100_R34_T33_W33_50	44592	0.0	13235	58.4	13235	0.0
kroC100_R20_T20_W60_75	15847	0.0	12582	53.1	12582	56.4
kroE100_R20_T60_W20_25	77100	0.0	17199	57.8	17199	85.8
kroE100_R20_T20_W60_50	11939	0.0	10586	50.7	10586	-32.5
kroE100_R34_T33_W33_75	25646	0.0	15104	0.0	14622	3.1
lin105_R21_T21_W63_25	14194	0.0	8893	51.7	8893	84.7
lin105_R21_T63_W21_50	53464	0.0	11583 ⁺	48.2	11583	-46.0
lin105_R35_T35_W35_75	-	0.0	10566 ⁺	32.7	10742	0.0
pr76_R14_T16_W46_25	72173	0.0	66007	53.3	66007	-35.9
pr76_R24_T26_W26_50	128227	0.0	74539	55.9	75594	0.0
pr76_R14_T46_W16_75	146621	0.0	90862	58.2	90862	72.7
pr107_R20_T65_W22_25	-	0.0	40994 ⁺	52.2	40994	57.0
pr107_R20_T22_W65_50	45930	0.0	35869	50.0	35869	84.5
pr107_R35_T36_W36_75	54404	0.0	37292 ⁺	30.9	37292	97.8

Tabela 8.13: Resultados das formulações matemáticas após a aplicação das regras de redução.

Instância	Fluxo		Multi-fluxo		Não-dir	
	Sol.	Dif_t	Sol.	Dif_t	Sol.	Dif_t
pr124_R42_T41_W41_25	-	0.0	-	0.0	41123	0.0
pr124_R24_T75_W25_50	-	0.0	-	0.0	49184	0.0
pr124_R24_T25_W75_75	58515	0.0	36719	0.0	34359	29.4
pr136_R26_T28_W82_25	65025	0.0	47620	56.2	47620	33.3
pr136_R46_T45_W45_50	99500	0.0	-	0.0	63947	0.0
pr136_R26_T82_W28_75	-	0.0	-	-0.1	78624	0.0
pr144_R28_T87_W29_25	-	0.0	-	0.0	47261	0.0
pr144_R28_T29_W87_50	45456	0.0	30294	54.6	32302	0.0
pr144_R48_T48_W48_75	-	0.0	-	0.0	47039	0.0
pr152_R50_T51_W51_25	-	0.0	-	0.0	62092	0.0
pr152_R29_T92_W31_50	-	0.0	-	0.0	78156	0.0
pr152_R29_T31_W92_75	72564	0.0	49110⁺	13.5	49110	19.9
rat99_R19_T20_W60_25	792⁺	49.7	792	52.8	792	-45.9
rat99_R33_T33_W33_50	1727	0.0	863	-0.1	847	-86.5
rat99_R19_T60_W20_75	1420	0.0	1017⁺	31.8	1017	-78.6
rat195_R65_T65_W65_25	-	0.0	-	0.0	1605	-1.0
rat195_R39_T39_W117_50	2099	0.0	1239	0.0	1199	-61.2
rat195_R39_T117_W39_75	-	0.0	-	0.0	1928	0.3
rd100_R34_T33_W33_25	13598	0.0	5595	0.0	5579	0.0
rd100_R20_T60_W20_50	-	0.0	6794	0.0	6619	-3.0
rd100_R20_T20_W60_75	5992	0.0	5421	54.1	5421	-9.6
si175_R35_T35_W105_25	10441	0.0	4845	56.3	4845	-46.2
si175_R35_T105_W35_50	-	0.0	-	0.0	6871	-10.7
si175_R59_T58_W58_75	-	0.0	-	0.0	8402	98.4
st70_R14_T14_W42_25	419⁺	19.0	419	52.2	419	46.2
st70_R14_T42_W14_50	858	0.0	554	54.9	554	-106.5
st70_R22_T24_W24_75	715	0.0	505⁺	48.4	505	21.7
u159_R31_T96_W32_25	-	0.0	-	0.0	35573	0.0
u159_R31_T32_W96_50	50525	0.0	28613	59.2	28613	-52.7
u159_R53_T53_W53_75	-	0.0	-	0.0	33015	73.5
a280_R94_T93_W93_25	-	0.0	-	0.0	2052	0.0
a280_R56_T168_W56_50	-	0.0	-	0.0	1996	0.0
a280_R56_T56_W168_75	-	0.0	-	0.0	1652	0.0
gil262_R51_T53_W158_25	-	0.0	-	0.0	1501	0.0
gil262_R88_T87_W87_50	-	0.0	-	0.0	1846	0.0

Tabela 8.13: Resultados das formulações matemáticas após a aplicação das regras de redução.

Instância	Fluxo		Multi-fluxo		Não-dir	
	Sol.	Dif_t	Sol.	Dif_t	Sol.	Dif_t
gil262_R51_T158_W53_75	-	0.0	-	0.0	2588	0.0
gr202_R39_T122_W41_25	-	0.0	-	0.0	32215	0.0
gr202_R68_T67_W67_50	-	0.0	-	0.0	31524	0.0
gr202_R39_T41_W122_75	97200	0.0	-	0.0	26202	0.0
gr229_R45_T46_W138_25	409000	0.0	-	0.0	65650	0.0
gr229_R45_T138_W46_50	-	0.0	-	0.0	118544	0.0
gr229_R77_T76_W76_75	-	0.0	-	0.0	104398	0.0
lin318_R63_T64_W191_25	-	0.0	-	0.0	36906	0.0
lin318_R108_T105_W105_50	-	0.0	-	0.0	62036	0.0
lin318_R63_T191_W64_75	-	0.0	-	0.0	51335	0.0
pr226_R76_T75_W75_25	-	0.0	-	0.0	77825	0.0
pr226_R44_T136_W46_50	-	0.0	-	0.0	-	0.0
pr226_R44_T46_W136_75	-	0.0	-	0.0	52519	0.0
pr264_R52_T53_W159_25	-	0.0	-	0.0	33687	0.0
pr264_R88_T88_W88_50	-	0.0	-	0.0	38553	0.0
pr264_R52_T159_W53_75	-	0.0	-	0.0	63158	0.0
pr299_R59_T180_W60_25	-	0.0	-	0.0	40486	0.0
pr299_R59_T60_W180_50	-	0.0	-	0.0	22948	0.0
pr299_R101_T99_W99_75	-	0.0	-	0.0	38250	0.0
rd400_R80_T240_W80_25	-	0.0	-	0.0	-	0.0
rd400_R80_T80_W240_50	-	0.0	-	0.0	12402	0.0
rd400_R136_T132_W132_75	-	0.0	-	0.0	12800	0.0
ts225_R45_T45_W135_25	-	0.0	-	0.0	87889	6.9
ts225_R45_T135_W45_50	-	0.0	-	0.0	116968	0.0
ts225_R75_T75_W75_75	-	0.0	-	0.0	100697	0.0
tsp225_R75_T75_W75_25	-	0.0	-	0.0	2929	0.0
tsp225_R75_T75_W75_50	-	0.0	-	0.0	3277	0.0
tsp225_R45_T45_W135_75	8630	0.0	-	0.0	2447	6.8

Instância	CC						DD					
	R	T	W	# Red.	T (s)	R	T	W	# Red.	T (s)		
att48_VT9_T10_W29_25	6	6	27	39	0.02	6	6	27	39	0.02		
berlin52_VT9_T11_W32_25	7	5	32	44	0.02	7	5	32	44	0.02		
bier127_VT24_T77_W26_25	0	0	26	26	23.73	0	0	26	26	23.52		
brazil58_VT18_T20_W20_25	0	0	11	11	0.97	0	0	11	11	0.97		
brg180_VT36_T36_W108_25	0	0	108	108	7.97	0	0	108	108	7.96		
ch130_VT44_T43_W43_25	40	21	42	103	8.46	40	21	42	103	8.47		
ch150_VT50_T50_W50_25	32	19	44	95	27.8	32	19	44	95	27.86		
d198_VT39_T40_W119_25	1	0	119	120	9.33	1	0	119	120	9.32		
eil51_VT17_T17_W17_25	3	2	7	12	0.31	3	2	7	12	0.31		
eil76_VT14_T46_W16_25	5	2	14	21	1.21	5	2	14	21	1.21		
eil101_VT19_T21_W61_25	9	7	58	74	0.94	9	7	58	74	0.94		
gr48_VT16_T16_W16_25	16	9	16	41	0.05	16	9	16	41	0.05		
gr96_VT18_T58_W20_25	0	0	20	20	3.63	0	0	20	20	3.63		
gr120_VT40_T40_W40_25	31	17	38	86	5.47	31	17	38	86	5.48		
gr137_VT26_T28_W83_25	1	0	83	84	1.61	1	0	83	84	1.61		
hk48_VT9_T29_W10_25	1	0	7	8	0.15	1	0	7	8	0.15		
kroA100_VT20_T20_W60_25	0	0	59	59	0.47	0	0	59	59	0.46		
kroA150_VT50_T50_W50_25	0	0	50	50	27.39	0	0	50	50	27.18		
kroB100_VT20_T60_W20_25	1	0	20	21	6.78	1	0	20	21	6.78		
kroB150_VT30_T30_W90_25	0	0	84	84	7.43	0	0	84	84	7.43		
kroC100_VT20_T60_W20_25	0	0	20	20	6.86	0	0	20	20	6.86		
kroE100_VT20_T60_W20_25	2	0	20	22	6.09	2	0	20	22	6.09		
lin105_VT21_T21_W63_25	0	0	63	63	0.42	0	0	63	63	0.43		
pr76_VT14_T16_W46_25	2	0	46	48	0.07	2	0	46	48	0.07		
pr107_VT20_T65_W22_25	16	12	21	49	7.43	16	12	21	49	7.44		
pr124_VT42_T41_W41_25	0	0	41	41	8.47	0	0	41	41	8.42		
pr136_VT26_T28_W82_25	3	1	74	78	2.99	3	1	74	78	2.99		
pr144_VT28_T87_W29_25	0	0	29	29	24.38	0	0	29	29	24.4		
pr152_VT50_T51_W51_25	0	0	51	51	23.19	0	0	51	51	23.2		
rat99_VT19_T20_W60_25	19	19	60	98	0.73	19	19	60	98	0.73		
rd100_VT34_T33_W33_25	13	3	19	35	3.58	13	3	19	35	3.58		
si175_VT35_T35_W105_25	8	8	90	106	59.3	8	8	90	106	59.37		
st70_VT14_T14_W42_25	3	3	38	44	0.23	3	3	38	44	0.24		
u159_VT31_T96_W32_25	31	16	32	79	33	31	16	32	79	33.03		
gr229_VT45_T46_W138_25	0	0	138	138	20.84	0	0	138	138	20.81		
pr264_VT52_T53_W159_25	2	0	159	161	49.8	2	0	159	161	49.99		
ts225_VT45_T45_W135_25	32	31	129	192	44.24	32	31	129	192	44.17		

Tabela 8.11: Resultados da aplicação das sequências CC e DD antes e após a remoção de E^0 .

Instância	EE						FF					
	R	T	W	# Red.	T (s)	R	T	W	# Red.	T (s)		
att48_VT9_T10_W29_25	6	6	27	39	0.02	7	7	17	31	0.03		
berlin52_VT9_T11_W32_25	7	5	32	44	0.02	7	5	22	34	0.02		
bier127_VT24_T77_W26_25	0	0	26	26	23.47	0	0	26	26	23.49		
brazil58_VT18_T20_W20_25	0	0	11	11	0.97	0	0	11	11	0.97		
brg180_VT36_T36_W108_25	0	0	108	108	7.99	0	0	108	108	7.97		
ch130_VT44_T43_W43_25	40	21	42	103	6.8	38	19	29	86	6.36		
ch150_VT50_T50_W50_25	32	19	44	95	27.52	32	19	23	74	27.94		
d198_VT39_T40_W119_25	1	0	119	120	9.33	1	0	119	120	9.32		
eil51_VT17_T17_W17_25	3	2	7	12	0.31	3	2	3	8	0.31		
eil76_VT14_T46_W16_25	5	2	14	21	1.21	5	2	10	17	1.2		
eil101_VT19_T21_W61_25	9	7	58	74	0.94	9	7	45	61	0.94		
gr48_VT16_T16_W16_25	16	9	16	41	0.05	16	9	7	32	0.05		
gr96_VT18_T58_W20_25	0	0	20	20	3.63	0	0	20	20	3.63		
gr120_VT40_T40_W40_25	31	17	38	86	5.47	31	17	21	69	5.41		
gr137_VT26_T28_W83_25	1	0	83	84	1.62	1	0	83	84	1.61		
hk48_VT9_T29_W10_25	1	0	7	8	0.15	1	0	7	8	0.15		
kroA100_VT20_T20_W60_25	0	0	59	59	0.47	0	0	59	59	0.47		
kroA150_VT50_T50_W50_25	0	0	50	50	27.22	0	0	50	50	27.17		
kroB100_VT20_T60_W20_25	1	0	20	21	6.78	1	0	20	21	6.78		
kroB150_VT30_T30_W90_25	0	0	84	84	7.43	0	0	84	84	7.43		
kroC100_VT20_T60_W20_25	0	0	20	20	6.9	0	0	20	20	6.86		
kroE100_VT20_T60_W20_25	2	0	20	22	6.11	2	0	20	22	6.08		
lin105_VT21_T21_W63_25	0	0	63	63	0.42	0	0	63	63	0.43		
pr76_VT14_T16_W46_25	2	0	46	48	0.07	2	0	46	48	0.07		
pr107_VT20_T65_W22_25	16	12	21	49	7.23	15	11	11	37	6.88		
pr124_VT42_T41_W41_25	0	0	41	41	8.41	0	0	41	41	8.42		
pr136_VT26_T28_W82_25	3	1	74	78	2.99	3	1	72	76	2.99		
pr144_VT28_T87_W29_25	0	0	29	29	24.38	0	0	29	29	24.39		
pr152_VT50_T51_W51_25	0	0	51	51	23.19	0	0	51	51	23.2		
rat99_VT19_T20_W60_25	19	19	60	98	0.73	16	16	43	75	0.66		
rd100_VT34_T33_W33_25	13	3	19	35	3.59	13	3	15	31	3.58		
si175_VT35_T35_W105_25	8	8	90	106	59.19	8	8	62	78	59.85		
st70_VT14_T14_W42_25	3	3	38	44	0.23	2	2	34	38	0.15		
u159_VT31_T96_W32_25	31	16	32	79	32.92	31	16	16	63	32.95		
gr229_VT45_T46_W138_25	0	0	138	138	20.8	0	0	138	138	20.75		
pr264_VT52_T53_W159_25	2	0	159	161	49.85	2	0	159	161	50.73		
ts225_VT45_T45_W135_25	32	31	129	192	42.2	32	31	102	165	39.42		

Tabela 8.12: Resultados da aplicação das sequências EE e FF antes e após a remoção de E^0 .

Tabela 8.14: Resultados do DM-GRASP com $sup_{min} = 20\%$.

Instância	DM-GRASP ($sup_{min} = 20\%$)				
	Melhor Sol.	Média Sol.	Média T (s)	Dif _{ms}	Dif _t
att48_R9_T10_W29_25	2291	2291.27	0.77	-0.01	6.70
berlin52_R9_T11_W32_25	4861	4861.00	0.78	0.00	8.32
bier127_R24_T77_W26_25	98597	98648.20	65.52	0.05	17.52
brazil58_R18_T20_W20_25	19417	19417.00	3.14	0.00	22.04
brg180_R36_T36_W108_25	550	550.00	20.72	0.00	8.52
ch130_R44_T43_W43_25	4383	4383.90	18.96	-0.02	12.95
ch150_R50_T50_W50_25	4778	4797.80	42.89	0.05	26.85
d198_R39_T40_W119_25	10916	10916.00	38.57	0.00	27.49
eil51_R17_T17_W17_25	273	273.00	2.32	0.00	7.51
eil76_R14_T46_W16_25	387	387.00	10.26	0.00	15.76
eil101_R19_T21_W61_25	356	356.00	5.99	0.00	22.53
gr48_R16_T16_W16_25	3571	3571.00	1.39	0.00	23.95
gr96_R18_T58_W20_25	42746	42746.00	25.15	0.01	17.40
gr120_R40_T40_W40_25	4301	4301.00	19.35	0.00	30.40
gr137_R26_T28_W83_25	39305	39305.00	13.66	0.00	13.62
hk48_R9_T29_W10_25	9708	9708.00	3.50	0.01	8.24
kroA100_R20_T20_W60_25	10848	10848.00	4.67	0.00	13.29
kroA150_R50_T50_W50_25	14410	14453.10	104.29	0.00	9.81
kroB100_R20_T60_W20_25	18406	18406.00	25.11	0.00	16.09
kroB150_R30_T30_W90_25	11467	11477.80	20.07	0.00	14.74
kroC100_R20_T60_W20_25	16785	16785.00	24.50	0.00	22.21
kroE100_R20_T60_W20_25	17199	17199.00	22.99	0.04	17.40
lin105_R21_T21_W63_25	8893	8893.00	5.97	0.00	22.61
pr76_R14_T16_W46_25	66007	66007.00	2.21	0.00	13.37
pr107_R20_T65_W22_25	40994	40994.00	24.82	0.00	25.99
pr124_R42_T41_W41_25	35465	35465.00	51.94	0.00	15.94
pr136_R26_T28_W82_25	47620	47620.00	7.77	0.00	29.95
pr144_R28_T87_W29_25	46940	46944.80	64.67	-0.01	27.65
pr152_R50_T51_W51_25	50667	50667.30	61.45	0.00	24.13
rat99_R19_T20_W60_25	792	792.00	5.41	0.00	14.89
rd100_R34_T33_W33_25	5552	5552.00	11.84	0.00	27.59
si175_R35_T35_W105_25	4845	4847.60	65.79	0.05	7.71
st70_R14_T14_W42_25	419	419.00	1.75	0.00	22.71
u159_R31_T96_W32_25	35442	35442.00	66.29	0.00	23.57
gr229_R45_T46_W138_25	61278	61280.50	66.34	0.04	16.37
pr264_R52_T53_W159_25	27693	27693.00	99.12	0.00	28.68
ts225_R45_T45_W135_25	87889	88270.70	42.11	-0.25	12.94

Tabela 8.15: Resultados do DM-GRASP com $sup_{min} = 30\%$.

Instância	DM-GRASP ($sup_{min} = 30\%$)				
	Melhor Sol.	Média Sol.	Média T (s)	Dif _{ms}	Dif _t
att48_VT9_T10_W29_25	2291	2291.00	0.80	0.00	2.65
berlin52_VT9_T11_W32_25	4861	4861.00	0.81	0.00	4.87
bier127_VT24_T77_W26_25	98453	98612.30	69.32	0.09	12.74
brazil58_VT18_T20_W20_25	19417	19417.00	3.44	0.00	14.53
brg180_VT36_T36_W108_25	550	550.00	21.18	0.00	6.45
ch130_VT44_T43_W43_25	4383	4385.70	19.93	-0.06	8.50
ch150_VT50_T50_W50_25	4778	4800.30	45.00	0.00	23.24
d198_VT39_T40_W119_25	10916	10916.00	41.35	0.00	22.27
eil51_VT17_T17_W17_25	273	273.00	2.56	0.00	-1.78
eil76_VT14_T46_W16_25	387	387.00	10.87	0.00	10.75
eil101_VT19_T21_W61_25	356	356.00	6.29	0.00	18.64
gr48_VT16_T16_W16_25	3571	3571.00	1.39	0.00	23.71
gr96_VT18_T58_W20_25	42746	42746.00	27.21	0.01	10.64
gr120_VT40_T40_W40_25	4301	4301.00	19.65	0.00	29.30
gr137_VT26_T28_W83_25	39305	39305.00	14.24	0.00	9.95
hk48_VT9_T29_W10_25	9708	9708.50	3.61	0.00	5.46
kroA100_VT20_T20_W60_25	10848	10848.00	5.04	0.00	6.53
kroA150_VT50_T50_W50_25	14410	14472.30	113.41	-0.13	1.92
kroB100_VT20_T60_W20_25	18406	18406.00	26.39	0.00	11.83
kroB150_VT30_T30_W90_25	11479	11479.00	20.88	-0.01	11.29
kroC100_VT20_T60_W20_25	16785	16785.00	25.13	0.00	20.20
kroE100_VT20_T60_W20_25	17199	17201.50	23.28	0.03	16.34
lin105_VT21_T21_W63_25	8893	8893.00	6.85	0.00	11.21
pr76_VT14_T16_W46_25	66007	66007.00	2.23	0.00	12.47
pr107_VT20_T65_W22_25	40994	40994.00	24.02	0.00	28.35
pr124_VT42_T41_W41_25	35465	35465.00	54.57	0.00	11.68
pr136_VT26_T28_W82_25	47620	47620.00	8.03	0.00	27.55
pr144_VT28_T87_W29_25	46940	46940.00	73.46	0.00	17.82
pr152_VT50_T51_W51_25	50667	50667.30	62.97	0.00	22.26
rat99_VT19_T20_W60_25	792	792.00	5.60	0.00	11.88
rd100_VT34_T33_W33_25	5552	5552.00	13.37	0.00	18.26
si175_VT35_T35_W105_25	4845	4850.20	71.97	0.00	-0.95
st70_VT14_T14_W42_25	419	419.00	1.93	0.00	14.85
u159_VT31_T96_W32_25	35442	35442.00	69.07	0.00	20.36
gr229_VT45_T46_W138_25	61278	61304.20	67.98	0.01	14.30
pr264_VT52_T53_W159_25	27693	27693.00	104.01	0.00	25.16
ts225_VT45_T45_W135_25	87889	88112.20	44.16	-0.07	8.70

Tabela 8.16: Resultados do DM-GRASP com $sup_{min} = 40\%$.

Instâncias	DM-GRASP ($sup_{min} = 40\%$)				
	Melhor Sol.	Média Sol.	Média T (s)	Dif _{ms}	Dif _t
att48_VT9_T10_W29_25	2291	2291.10	0.83	0.00	-0.87
berlin52_VT9_T11_W32_25	4861	4861.00	0.91	0.00	-6.46
bier127_VT24_T77_W26_25	98453	98496.20	71.79	0.21	9.64
brazil58_VT18_T20_W20_25	19417	19417.00	3.45	0.00	14.31
brg180_VT36_T36_W108_25	550	550.00	21.24	0.00	6.19
ch130_VT44_T43_W43_25	4383	4385.80	20.56	-0.06	5.61
ch150_VT50_T50_W50_25	4778	4799.70	48.14	0.01	17.89
d198_VT39_T40_W119_25	10916	10916.00	44.64	0.00	16.08
eil51_VT17_T17_W17_25	273	273.00	2.55	0.00	-1.57
eil76_VT14_T46_W16_25	387	387.00	11.33	0.00	7.02
eil101_VT19_T21_W61_25	356	356.00	6.58	0.00	14.90
gr48_VT16_T16_W16_25	3571	3571.00	1.47	0.00	19.32
gr96_VT18_T58_W20_25	42746	42746.00	28.77	0.01	5.49
gr120_VT40_T40_W40_25	4301	4301.00	19.36	0.00	30.35
gr137_VT26_T28_W83_25	39305	39305.00	14.59	0.00	7.76
hk48_VT9_T29_W10_25	9708	9708.50	3.71	0.00	2.86
kroA100_VT20_T20_W60_25	10800	10848.00	5.30	0.00	1.76
kroA150_VT50_T50_W50_25	14382	14445.00	116.05	0.06	-0.36
kroB100_VT20_T60_W20_25	18406	18406.00	26.73	0.00	10.67
kroB150_VT30_T30_W90_25	11479	11479.00	22.05	-0.01	6.30
kroC100_VT20_T60_W20_25	16785	16785.00	27.77	0.00	11.84
kroE100_VT20_T60_W20_25	17199	17201.50	25.44	0.03	8.57
lin105_VT21_T21_W63_25	8893	8893.00	7.13	0.00	7.62
pr76_VT14_T16_W46_25	66007	66007.00	2.29	0.00	10.25
pr107_VT20_T65_W22_25	40994	40994.00	24.43	0.00	27.12
pr124_VT42_T41_W41_25	35500	35465.00	55.18	0.00	10.69
pr136_VT26_T28_W82_25	47620	47620.00	8.50	0.00	23.35
pr144_VT28_T87_W29_25	46940	46942.40	78.24	-0.01	12.47
pr152_VT50_T51_W51_25	50667	50667.00	63.77	0.00	21.26
rat99_VT19_T20_W60_25	792	792.00	5.70	0.00	10.32
rd100_VT34_T33_W33_25	5552	5552.00	13.94	0.00	14.79
si175_VT35_T35_W105_25	4845	4847.60	71.34	0.05	-0.07
st70_VT14_T14_W42_25	419	419.00	2.00	0.00	11.53
u159_VT31_T96_W32_25	35442	35442.00	74.26	0.00	14.37
gr229_VT45_T46_W138_25	56052	57109.50	391.76	-1.16	17.47
pr264_VT52_T53_W159_25	27693	27693.00	109.05	0.00	21.53
ts225_VT45_T45_W135_25	87889	88184.60	46.57	-0.15	3.72

Tabela 8.17: Resultados do DM-GRASP com $sup_{min} = 50\%$.

Instâncias	DM-GRASP ($sup_{min} = 50\%$)				
	Melhor Sol.	Média Sol.	Média T (s)	Dif _{ms}	Dif _t
att48_VT9_T10_W29_25	2291	2291.00	0.85	0.00	-3.59
berlin52_VT9_T11_W32_25	4861	4861.00	0.90	0.00	-5.33
bier127_VT24_T77_W26_25	98453	98602.90	75.63	0.10	4.81
brazil58_VT18_T20_W20_25	19417	19417.00	3.48	0.00	13.40
brg180_VT36_T36_W108_25	550	550.00	21.40	0.00	5.51
ch130_VT44_T43_W43_25	4383	4387.50	21.20	-0.10	2.65
ch150_VT50_T50_W50_25	4778	4800.30	53.27	0.00	9.15
d198_VT39_T40_W119_25	10916	10916.00	46.98	0.00	11.67
eil51_VT17_T17_W17_25	273	273.00	2.55	0.00	-1.64
eil76_VT14_T46_W16_25	387	387.00	11.56	0.00	5.10
eil101_VT19_T21_W61_25	356	356.00	6.51	0.00	15.82
gr48_VT16_T16_W16_25	3571	3571.00	1.52	0.00	16.84
gr96_VT18_T58_W20_25	42746	42749.60	29.90	0.00	1.78
gr120_VT40_T40_W40_25	4301	4301.00	20.04	0.00	27.93
gr137_VT26_T28_W83_25	39305	39305.00	14.78	0.00	6.53
hk48_VT9_T29_W10_25	9708	9708.60	3.79	0.00	0.71
kroA100_VT20_T20_W60_25	10800	10848.00	5.31	0.00	1.49
kroA150_VT50_T50_W50_25	14410	14465.20	116.66	-0.08	-0.89
kroB100_VT20_T60_W20_25	18406	18406.00	27.15	0.00	9.28
kroB150_VT30_T30_W90_25	11467	11479.80	21.79	-0.02	7.41
kroC100_VT20_T60_W20_25	16785	16785.00	27.89	0.00	11.46
kroE100_VT20_T60_W20_25	17199	17204.00	26.54	0.01	4.63
lin105_VT21_T21_W63_25	8893	8893.00	7.09	0.00	8.15
pr76_VT14_T16_W46_25	66007	66007.00	2.31	0.00	9.25
pr107_VT20_T65_W22_25	40994	40994.00	27.29	0.00	18.62
pr124_VT42_T41_W41_25	35500	35465.00	57.17	0.00	7.47
pr136_VT26_T28_W82_25	47620	47620.00	9.01	0.00	18.73
pr144_VT28_T87_W29_25	46940	46940.00	80.61	0.00	9.82
pr152_VT50_T51_W51_25	50667	50667.20	62.52	0.00	22.81
rat99_VT19_T20_W60_25	792	792.00	5.66	0.00	10.88
rd100_VT34_T33_W33_25	5552	5552.00	14.19	0.00	13.23
si175_VT35_T35_W105_25	4845	4847.60	71.12	0.05	0.24
st70_VT14_T14_W42_25	419	419.00	2.06	0.00	8.91
u159_VT31_T96_W32_25	35442	35442.00	74.93	0.00	13.61
gr229_VT45_T46_W138_25	55932	57094.50	420.28	-1.13	11.46
pr264_VT52_T53_W159_25	27693	27693.00	122.33	0.00	11.98
ts225_VT45_T45_W135_25	87889	88273.90	47.61	-0.25	1.56

Tabela 8.18: Resultados do MDM-GRASP com $sup_{min} = 20\%$.

Instância	MDM-GRASP ($sup_{min} = 20\%$)				
	Melhor Sol.	Média Sol.	Média T (s)	Dif _{ms} (DM)	Dif _t (DM)
att48_VT9_T10_W29_25	2291	2291.50	0.71	-0.01	7.54
berlin52_VT9_T11_W32_25	4861	4861.00	0.71	0.00	8.74
bier127_VT24_T77_W26_25	98453	98600.10	64.16	0.05	2.07
brazil58_VT18_T20_W20_25	19417	19417.00	2.96	0.00	5.70
brg180_VT36_T36_W108_25	550	550.00	21.32	0.00	-2.88
ch130_VT44_T43_W43_25	4383	4384.80	18.27	-0.02	3.66
ch150_VT50_T50_W50_25	4778	4800.30	37.26	-0.05	13.13
d198_VT39_T40_W119_25	10916	10916.00	36.91	0.00	4.30
eil51_VT17_T17_W17_25	273	273.00	2.30	0.00	0.70
eil76_VT14_T46_W16_25	387	387.00	9.65	0.00	5.91
eil101_VT19_T21_W61_25	356	356.00	5.22	0.00	12.90
gr48_VT16_T16_W16_25	3571	3571.00	1.29	0.00	6.90
gr96_VT18_T58_W20_25	42746	42746.00	23.48	0.00	6.66
gr120_VT40_T40_W40_25	4301	4301.00	15.38	0.00	20.53
gr137_VT26_T28_W83_25	39305	39305.00	13.44	0.00	1.61
hk48_VT9_T29_W10_25	9708	9708.60	3.50	-0.01	0.06
kroA100_VT20_T20_W60_25	10848	10848.00	4.41	0.00	5.57
kroA150_VT50_T50_W50_25	14382	14444.20	104.26	0.06	0.02
kroB100_VT20_T60_W20_25	18406	18406.00	24.10	0.00	4.03
kroB150_VT30_T30_W90_25	11467	11475.40	19.38	0.02	3.43
kroC100_VT20_T60_W20_25	16785	16785.00	22.92	0.00	6.45
kroE100_VT20_T60_W20_25	17199	17206.50	21.27	-0.04	7.47
lin105_VT21_T21_W63_25	8893	8893.00	5.42	0.00	9.17
pr76_VT14_T16_W46_25	66007	66007.00	2.24	0.00	-1.19
pr107_VT20_T65_W22_25	40994	40994.00	23.71	0.00	4.48
pr124_VT42_T41_W41_25	35465	35465.00	51.89	0.00	0.10
pr136_VT26_T28_W82_25	47620	47620.00	6.98	0.00	10.21
pr144_VT28_T87_W29_25	46940	46940.00	55.84	0.01	13.66
pr152_VT50_T51_W51_25	50667	50667.10	55.52	0.00	9.66
rat99_VT19_T20_W60_25	792	792.00	5.22	0.00	3.53
rd100_VT34_T33_W33_25	5552	5552.00	10.52	0.00	11.13
si175_VT35_T35_W105_25	4845	4845.00	60.59	0.05	7.91
st70_VT14_T14_W42_25	419	419.00	1.74	0.00	0.71
u159_VT31_T96_W32_25	35442	35442.00	62.34	0.00	5.97
gr229_VT45_T46_W138_25	56052	57258.30	300.60	0.00	3.39
pr264_VT52_T53_W159_25	27693	27693.00	93.38	0.00	5.79
ts225_VT45_T45_W135_25	87889	88209.50	38.09	0.07	9.53

Tabela 8.19: Resultados do MDM-GRASP com $sup_{min} = 30\%$.

Instância	MDM-GRASP ($sup_{min} = 30\%$)				
	Melhor Sol.	Média Sol.	Média T (s)	Dif _{ms} (DM)	Dif _t (DM)
att48_VT9_T10_W29_25	2291	2291.00	0.80	0.01	-3.90
berlin52_VT9_T11_W32_25	4861	4861.00	0.81	0.00	-3.85
bier127_VT24_T77_W26_25	98453	98612.30	69.32	0.04	-5.80
brazil58_VT18_T20_W20_25	19417	19417.00	3.44	0.00	-9.55
brg180_VT36_T36_W108_25	550	550.00	21.18	0.00	-2.22
ch130_VT44_T43_W43_25	4383	4385.70	19.93	-0.04	-5.12
ch150_VT50_T50_W50_25	4778	4800.30	45.00	-0.05	-4.92
d198_VT39_T40_W119_25	10916	10916.00	41.35	0.00	-7.21
eil51_VT17_T17_W17_25	273	273.00	2.56	0.00	-10.34
eil76_VT14_T46_W16_25	387	387.00	10.87	0.00	-5.95
eil101_VT19_T21_W61_25	356	356.00	6.29	0.00	-5.01
gr48_VT16_T16_W16_25	3571	3571.00	1.39	0.00	0.00
gr96_VT18_T58_W20_25	42746	42746.00	27.21	0.00	-8.19
gr120_VT40_T40_W40_25	4301	4301.00	19.65	0.00	-1.55
gr137_VT26_T28_W83_25	39305	39305.00	14.24	0.00	-4.25
hk48_VT9_T29_W10_25	9708	9708.50	3.61	-0.01	-3.14
kroA100_VT20_T20_W60_25	10848	10848.00	5.04	0.00	-7.92
kroA150_VT50_T50_W50_25	14410	14472.30	113.41	-0.13	-8.74
kroB100_VT20_T60_W20_25	18406	18406.00	26.39	0.00	-5.10
kroB150_VT30_T30_W90_25	11479	11479.00	20.88	-0.01	-4.04
kroC100_VT20_T60_W20_25	16785	16785.00	25.13	0.00	-2.57
kroE100_VT20_T60_W20_25	17199	17201.50	23.28	-0.01	-1.26
lin105_VT21_T21_W63_25	8893	8893.00	6.85	0.00	-14.74
pr76_VT14_T16_W46_25	66007	66007.00	2.23	0.00	-0.90
pr107_VT20_T65_W22_25	40994	40994.00	24.02	0.00	3.22
pr124_VT42_T41_W41_25	35465	35465.00	54.57	0.00	-5.06
pr136_VT26_T28_W82_25	47620	47620.00	8.03	0.00	-3.35
pr144_VT28_T87_W29_25	46940	46940.00	73.46	0.01	-13.59
pr152_VT50_T51_W51_25	50667	50667.30	62.97	0.00	-2.47
rat99_VT19_T20_W60_25	792	792.00	5.60	0.00	-3.51
rd100_VT34_T33_W33_25	5552	5552.00	13.37	0.00	-12.92
si175_VT35_T35_W105_25	4845	4850.20	71.97	-0.05	-9.39
st70_VT14_T14_W42_25	419	419.00	1.93	0.00	-10.29
u159_VT31_T96_W32_25	35442	35442.00	69.07	0.00	-4.19
gr229_VT45_T46_W138_25	55943	57095.50	355.65	0.29	-14.30
pr264_VT52_T53_W159_25	27693	27693.00	104.01	0.00	-4.93
ts225_VT45_T45_W135_25	87889	88112.20	44.16	0.18	-4.87

Tabela 8.20: Resultados do MDM-GRASP com $sup_{min} = 40\%$.

Instância	MDM-GRASP ($sup_{min} = 40\%$)				
	Melhor Sol.	Média Sol.	Média T (s)	Dif _{ms} (DM)	Dif _t (DM)
att48_VT9_T10_W29_25	2291	2291.10	0.83	0.01	-7.92
berlin52_VT9_T11_W32_25	4861	4861.00	0.91	0.00	-16.84
bier127_VT24_T77_W26_25	98453	98496.20	71.79	0.15	-9.57
brazil58_VT18_T20_W20_25	19417	19417.00	3.45	0.00	-9.76
brg180_VT36_T36_W108_25	550	550.00	21.24	0.00	-2.52
ch130_VT44_T43_W43_25	4383	4385.80	20.56	-0.04	-8.44
ch150_VT50_T50_W50_25	4778	4799.70	48.14	-0.04	-12.24
d198_VT39_T40_W119_25	10916	10916.00	44.64	0.00	-15.73
eil51_VT17_T17_W17_25	273	273.00	2.55	0.00	-9.98
eil76_VT14_T46_W16_25	387	387.00	11.33	0.00	-10.41
eil101_VT19_T21_W61_25	356	356.00	6.58	0.00	-9.82
gr48_VT16_T16_W16_25	3571	3571.00	1.47	0.00	-5.76
gr96_VT18_T58_W20_25	42746	42746.00	28.77	0.00	-14.41
gr120_VT40_T40_W40_25	4301	4301.00	19.36	0.00	-0.06
gr137_VT26_T28_W83_25	39305	39305.00	14.59	0.00	-6.79
hk48_VT9_T29_W10_25	9708	9708.50	3.71	-0.01	-5.91
kroA100_VT20_T20_W60_25	10848	10848.00	5.30	0.00	-13.39
kroA150_VT50_T50_W50_25	14382	14445.00	116.05	0.06	-11.28
kroB100_VT20_T60_W20_25	18406	18406.00	26.73	0.00	-6.47
kroB150_VT30_T30_W90_25	11479	11479.00	22.05	-0.01	-9.89
kroC100_VT20_T60_W20_25	16785	16785.00	27.77	0.00	-13.34
kroE100_VT20_T60_W20_25	17199	17201.50	25.44	-0.01	-10.66
lin105_VT21_T21_W63_25	8893	8893.00	7.13	0.00	-19.46
pr76_VT14_T16_W46_25	66007	66007.00	2.29	0.00	-3.39
pr107_VT20_T65_W22_25	40994	40994.00	24.43	0.00	1.55
pr124_VT42_T41_W41_25	35465	35465.00	55.18	0.00	-6.23
pr136_VT26_T28_W82_25	47620	47620.00	8.50	0.00	-9.36
pr144_VT28_T87_W29_25	46940	46942.40	78.24	0.01	-20.99
pr152_VT50_T51_W51_25	50667	50667.00	63.77	0.00	-3.78
rat99_VT19_T20_W60_25	792	792.00	5.70	0.00	-5.33
rd100_VT34_T33_W33_25	5552	5552.00	13.94	0.00	-17.72
si175_VT35_T35_W105_25	4845	4847.60	71.34	0.00	-8.44
st70_VT14_T14_W42_25	419	419.00	2.00	0.00	-14.56
u159_VT31_T96_W32_25	35442	35442.00	74.26	0.00	-12.03
gr229_VT45_T46_W138_25	56052	57109.50	391.76	0.26	-25.91
pr264_VT52_T53_W159_25	27693	27693.00	109.05	0.00	-10.02
ts225_VT45_T45_W135_25	87889	88184.60	46.57	0.10	-10.59

Tabela 8.21: Resultados do MDM-GRASP com $sup_{min} = 50\%$.

Instâncias	MDM-GRASP ($sup_{min} = 50\%$)				
	Melhor Sol.	Média Sol.	Média T (s)	Dif _{ms} (DM)	Dif _t (DM)
att48_VT9_T10_W29_25	2291	2291.00	0.85	0.01	-10.82
berlin52_VT9_T11_W32_25	4861	4861.00	0.90	0.00	-15.60
bier127_VT24_T77_W26_25	98453	98602.90	75.63	0.05	-15.42
brazil58_VT18_T20_W20_25	19417	19417.00	3.48	0.00	-10.92
brg180_VT36_T36_W108_25	550	550.00	21.40	0.00	-3.26
ch130_VT44_T43_W43_25	4383	4387.50	21.20	-0.08	-11.84
ch150_VT50_T50_W50_25	4778	4800.30	53.27	-0.05	-24.19
d198_VT39_T40_W119_25	10916	10916.00	46.98	0.00	-21.82
eil51_VT17_T17_W17_25	273	273.00	2.55	0.00	-10.05
eil76_VT14_T46_W16_25	387	387.00	11.56	0.00	-12.70
eil101_VT19_T21_W61_25	356	356.00	6.51	0.00	-8.64
gr48_VT16_T16_W16_25	3571	3571.00	1.52	0.00	-9.01
gr96_VT18_T58_W20_25	42746	42749.60	29.90	-0.01	-18.90
gr120_VT40_T40_W40_25	4301	4301.00	20.04	0.00	-3.54
gr137_VT26_T28_W83_25	39305	39305.00	14.78	0.00	-8.23
hk48_VT9_T29_W10_25	9708	9708.60	3.79	-0.01	-8.25
kroA100_VT20_T20_W60_25	10848	10848.00	5.31	0.00	-13.70
kroA150_VT50_T50_W50_25	14410	14465.20	116.66	-0.08	-11.86
kroB100_VT20_T60_W20_25	18406	18406.00	27.15	0.00	-8.12
kroB150_VT30_T30_W90_25	11467	11479.80	21.79	-0.02	-8.58
kroC100_VT20_T60_W20_25	16785	16785.00	27.89	0.00	-13.82
kroE100_VT20_T60_W20_25	17199	17204.00	26.54	-0.03	-15.44
lin105_VT21_T21_W63_25	8893	8893.00	7.09	0.00	-18.77
pr76_VT14_T16_W46_25	66007	66007.00	2.31	0.00	-4.55
pr107_VT20_T65_W22_25	40994	40994.00	27.29	0.00	-9.93
pr124_VT42_T41_W41_25	35465	35465.00	57.17	0.00	-10.07
pr136_VT26_T28_W82_25	47620	47620.00	9.01	0.00	-15.96
pr144_VT28_T87_W29_25	46940	46940.00	80.61	0.01	-24.64
pr152_VT50_T51_W51_25	50667	50667.20	62.52	0.00	-1.75
rat99_VT19_T20_W60_25	792	792.00	5.66	0.00	-4.67
rd100_VT34_T33_W33_25	5552	5552.00	14.19	0.00	-19.87
si175_VT35_T35_W105_25	4845	4847.60	71.12	0.00	-8.11
st70_VT14_T14_W42_25	419	419.00	2.06	0.00	-17.95
u159_VT31_T96_W32_25	35442	35442.00	74.93	0.00	-13.03
gr229_VT45_T46_W138_25	55932	57094.50	420.28	0.29	-35.07
pr264_VT52_T53_W159_25	27693	27693.00	122.33	0.00	-23.42
ts225_VT45_T45_W135_25	87889	88273.90	47.61	0.00	-13.07

Referências Bibliográficas

- [1] AGRAWAL, R., SRIKANT, R., “Fast algorithms for mining association rules”. In: *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94)*. (San Francisco, CA, USA, 1994), Morgan Kaufmann Publishers Inc., p. 487–499.
- [2] AIEX, R. M., RESENDE, M. G. C., RIBEIRO, C. C., 2006, “TTTplots: a perl program to create time-to-target plots”, *Optimization Letters*, v. 1, pp. 355–366.
- [3] ALMEDER, C., 2010, “A hybrid optimization approach for multi-level capacitated lot-sizing problems”, *European Journal of Operational Research*, v. 200, pp. 599–606.
- [4] ALOISE, D., RIBEIRO, C. C., 2011, “Adaptive memory in multistart heuristics for multicommodity network design”, *Journal of Heuristics*, v. 17, pp. 153–179.
- [5] AWERBUCH, B., AZAR, Y., BLUM, A., VEMPALA, S., 1998, “New approximation guarantees for minimum-weight k-trees and prize-collecting salesmen”, *SIAM Journal on computing*, v. 28, pp. 254–262.
- [6] BALAS, E., 1989, “The prize collecting traveling salesman problem”, *Networks*, v. 19, pp. 621–636.
- [7] BALAS, E., MARTIN, G., 1985, “ROLL-A-ROUND: Software package for scheduling the rounds of a rolling mill”, *Copyright Balas and Martin Associates*, v. 104.
- [8] BALAS, E., 1975, “Facets of the knapsack polytope”, *Mathematical programming*, v. 8, pp. 146–164.
- [9] BALAS, E., 2007, “The Prize Collecting Traveling Salesman Problem and its Applications”. In: Gutin, G., Punnen, A. P. (Eds.), *The Traveling Salesman Problem and Its Variations*, pp. 663–695, Boston, MA, Springer US. ISBN: 978-0-306-48213-7. doi: 10.1007/0-306-48213-4_14. Disponível em: <https://doi.org/10.1007/0-306-48213-4_14>.
- [10] BARBALHO, H., ROSSETI, I., MARTINS, S. L., PLASTINO, A., 2013, “A hybrid data mining GRASP with path-relinking for the 2-path network design problem”, *Computers & Operations Research*, v. 40, pp. 3159–3173.

- [11] BARR, R. S., GOLDEN, B. L., KELLY, J. P., RESENDE, M. G. C., STEWART, W. R., 1995, “Designing and reporting on computational experiments with heuristic methods”, *Journal of Heuristics*, v. 1, pp. 9–32. ISSN: 1572-9397. doi: 10.1007/BF02430363. Disponível em: <<https://doi.org/10.1007/BF02430363>>.
- [12] BÉRUBÉ, J., GENDREAU, M., POTVIN, J., 2009, “A branch-and-cut algorithm for the undirected prize collecting traveling salesman problem”, *Networks: An International Journal*, v. 54, pp. 56–67.
- [13] BIENSTOCK, D., GOEMANS, M. X., SIMCHI-LEVI, D., WILLIAMSON, D., 1993, “A note on the prize collecting traveling salesman problem”, *Mathematical programming*, v. 59, pp. 413–420.
- [14] BLUM, C., ROLI, A., SAMPELS, M., 2008, *Hybrid metaheuristics: an emerging approach to optimization*, v. 114. Berlim, Germany. Springer.
- [15] CARPENTER, R., 1960, “Principles and procedures of statistics, with special reference to the biological sciences”, *The Eugenics Review*, v. 52, pp. 172.
- [16] CHAVES, A. A., LORENA, L. A. N., 2005, “Hybrid algorithms with detection of promising areas for the prize collecting travelling salesman problem”. In: *Fifth International Conference on Hybrid Intelligent Systems (HIS'05)*, pp. 49–54. IEEE Computer Society. Rio de Janeiro, Brazil.
- [17] CHAVES, A. A., BIAJOLI, F. L., MINE, O. M., SOUZA, M. J. F., 2004, “Modelagens Exata e Heurística para Resolução do Problema do Caixeiro Viajante com Coleta de Prêmios”. In: *XXIV Encontro Nacional de Engenharia de Produção*, pp. 3151–3158. Florianópolis-SC, Brasil.
- [18] CHAVES, A. A., LORENA, L. A. N., MIRALLES, C., 2009, “Hybrid metaheuristic for the assembly line worker assignment and balancing problem”. In: *International Workshop on Hybrid Metaheuristics*, pp. 1–14. Springer. Plymouth, United Kingdom.
- [19] CHAVES, A. A., LORENA, L. A. N., 2008, “Hybrid metaheuristic for the prize collecting travelling salesman problem”. In: *European Conference on Evolutionary Computation in Combinatorial Optimization*, pp. 123–134. Springer. Leipzig, Germany.
- [20] CHVÁTAL, V., 1973, “Edmonds polytopes and weakly Hamiltonian graphs”, *Mathematical programming*, v. 5, pp. 29–40.
- [21] CIUPALA, L., CIUREA, E., 2008, “About preflow algorithms for the minimum flow problem”, *WSEAS Transactions on Computer Research*, v. 3, pp. 35–42.
- [22] CLIMACO, G., SIMONETTI, L., ROSSETI, I., “A branch-and-cut and MIP-based heuristics for the Prize-Collecting Travelling Salesman Problem”, *RAIRO-Operations Research*.

- [23] CLÍMACO, G., ROSSETI, I., SIMONETTI, L., GUERINE, M., 2018, “Combining integer linear programming with a state-of-the-art heuristic for the 2-path network design problem”, *International Transactions in Operational Research*, v. 26, pp. 615–641.
- [24] CROWDER, H., JOHNSON, E. L., PADBERG, M., 1983, “Solving large-scale zero-one linear programming problems”, *Operations Research*, v. 31, pp. 803–834.
- [25] CURRENT, J. R., SCHILLING, D. A., 1989, “The covering salesman problem”, *Transportation science*, v. 23, pp. 208–213.
- [26] DA SILVA, R., 2014, *Metaheurística aplicada ao Problema de Recobrimento de Rotas com Coleta de Prêmios*. Dissertação de Mestrado, Programa de Pós-Graduação em Computação, Instituto de Computação, Universidade Federal Fluminense, Niterói, RJ.
- [27] DELL’AMICO, M., MAFFIOLI, F., VÄRBRAND, P., 1995, “On prize-collecting tours and the asymmetric travelling salesman problem”, *International Transactions in Operational Research*, v. 2, pp. 297 – 308. ISSN: 0969-6016. doi: [https://doi.org/10.1016/0969-6016\(95\)00010-5](https://doi.org/10.1016/0969-6016(95)00010-5). Disponível em: <<http://www.sciencedirect.com/science/article/pii/0969601695000105>>.
- [28] DELL’AMICO, M., MAFFIOLI, F., SCIOMACHEN, A., 1998, “A lagrangian heuristic for the prize collectingtravelling salesman problem”, *Annals of Operations Research*, v. 81, pp. 289–306.
- [29] EDMONDS, J., 1965, “Maximum matching and a polyhedron with 0, 1-vertices”, *Journal of Research of the National Bureau of Standards B*, v. 69, pp. 55–56.
- [30] EIBEN, A. E., RUDOLPH, G., 1999, “Theory of evolutionary algorithms: a bird’s eye view”, *Theoretical Computer Science*, v. 229, pp. 3–9.
- [31] FEO, T. A., RESENDE, M. G. C., 1995, “Greedy randomized adaptive search procedures”, *Journal of global optimization*, v. 6, pp. 109–133.
- [32] FERREIRA, C. E., MARTIN, A., WEISMANTEL, R., 1996, “Solving multiple knapsack problems by cutting planes”, *SIAM Journal on Optimization*, v. 6, pp. 858–877.
- [33] FESTA, P., RESENDE, M. G. C., 2009, “An annotated bibliography of GRASP–Part I: Algorithms”, *International Transactions in Operational Research*, v. 16, pp. 1–24.
- [34] FESTA, P., RESENDE, M. G. C., 2009, “An annotated bibliography of GRASP–Part II: Applications”, *International Transactions in Operational Research*, v. 16, pp. 131–172.
- [35] FISCHETTI, M., TOTH, P., 1997, “A polyhedral approach to the asymmetric traveling salesman problem”, *Management Science*, v. 43, pp. 1520–1536.

- [36] FISCHETTI, M., GONZALEZ, J. J. S., TOTH, P., 1998, “Solving the orienteering problem through branch-and-cut”, *INFORMS Journal on Computing*, v. 10, pp. 133–148.
- [37] FISCHETTI, M., GLOVER, F., LODI, A., 2005, “The feasibility pump”, *Mathematical Programming*, v. 104, pp. 91–104.
- [38] FISCHETTI, M., LODI, A., 2003, “Local branching”, *Mathematical programming*, v. 98, pp. 23–47.
- [39] FISCHETTI, M., TOTH, P., 1988, “An additive approach for the optimal solution of the prize collecting traveling salesman problem”, *Vehicle routing: Methods and studies*, v. 231, pp. 319–343.
- [40] FLEURENT, C., GLOVER, F., 1999, “Improved constructive multistart strategies for the quadratic assignment problem using adaptive memory”, *INFORMS Journal on Computing*, v. 11, pp. 198–204.
- [41] FÜRNKRANZ, J., 2001, “Round robin rule learning”. In: *Proceedings of the 18th International Conference on Machine Learning (ICML-01)*: 146–153. Citeseer. Williamstown, MA, USA.
- [42] GENDREAU, M., HERTZ, A., LAPORTE, G., 1992, “New insertion and postoptimization procedures for the traveling salesman problem”, *Operations Research*, v. 40, pp. 1086–1094.
- [43] GENDREAU, M., LAPORTE, G., SEMET, F., 1997, “The covering tour problem”, *Operations Research*, v. 45, pp. 568–576.
- [44] GENDREAU, M., HERTZ, A., LAPORTE, G., 1992, “New insertion and postoptimization procedures for the traveling salesman problem”, *Operations Research*, , n. 6, pp. 1086–1094.
- [45] GLOVER, F., LAGUNA, M., MARTÍ, R., 2000, “Fundamentals of scatter search and path relinking”, *Control and Cybernetics*, v. 29, pp. 653–684.
- [46] GOEMANS, M. X., WILLIAMSON, D. P., 1995, “A general approximation technique for constrained forest problems”, *SIAM Journal on Computing*, v. 24, pp. 296–317.
- [47] GOEMANS, M. X., 2009, “Combining Approximation Algorithms for the Prize-Collecting TSP”, *ArXiv*, v. abs/0910.0553.
- [48] GOETHALS, B., ZAKI, M. J., 2004, “Advances in Frequent Itemset Mining Implementations: report on FIMI’03”, *SIGKDD Exploration Newsletter*, v. 6, pp. 109–117. ISSN: 1931-0145. doi: 10.1145/1007730.1007744. Disponível em: <<http://doi.acm.org/10.1145/1007730.1007744>>.

- [49] GOMES, L., DINIZ, V., MARTINHON, C. A., 2000, “An Hybrid GRASP+ VND Metaheuristic for the Prize-Collecting Traveling Salesman Problem”. In: *XXXII Simpósio Brasileiro de Pesquisa Operacional*, pp. 1657–1665. Viçosa, Brazil.
- [50] GOODRICH, M. T., TAMASSIA, R., 2015, “The Christofides Approximation Algorithm”, *Algorithm Design and Applications*, Wiley, pp. 513–514.
- [51] GOWER, J. C., ROSS, G. J., 1969, “Minimum spanning trees and single linkage cluster analysis”, *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, v. 18, pp. 54–64.
- [52] GRAHNE, G., ZHU, J., 2003, “Efficiently using prefix-trees in mining frequent itemsets.” In: *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, v. 90, pp. 236–245. Melbourne, The United States of America.
- [53] GRÖTSCHEL, M., PADBERG, M. W., 1979, “On the symmetric travelling salesman problem I: inequalities”, *Mathematical Programming*, v. 16, pp. 265–280.
- [54] GRÖTSCHEL, M., PADBERG, M. W., 1979, “On the symmetric travelling salesman problem II: Lifting theorems and facets”, *Mathematical Programming*, v. 16, pp. 281–302.
- [55] GU, Z., NEMHAUSER, G. L., SAVELSBERGH, M. W., 1999, “Lifted flow cover inequalities for mixed 0-1 integer programs”, *Mathematical Programming*, v. 85, pp. 439–467.
- [56] GUERINE, M., ROSSETI, I., PLASTINO, A., 2020, “A hybrid data mining heuristic to solve the point-feature cartographic label placement problem”, *International Transactions in Operational Research*, v. 27, pp. 1189–1209.
- [57] GUERINE, M., ROSSETI, I., PLASTINO, A., 2014, “Extending the Hybridization of Metaheuristics with Data Mining to a Broader Domain”. In: *Proceedings of the 16th International Conference on Enterprise Systems*, pp. 395–406, Lisboa, Portugal.
- [58] HAN, J., PEI, J. AND KAMBER, M., 2011, *Data mining: concepts and techniques*. Elsevier.
- [59] HAN, J., PEI, J., YIN, Y., 2000, “Mining frequent patterns without candidate generation”, *ACM sigmod record*, v. 29, pp. 1–12.
- [60] HANSEN, P., MLADENOVIĆ, N., 1997, “Variable neighborhood search for the p-median”, *Location Science*, v. 5, pp. 207–226.
- [61] HANSEN, P., MLADENOVIĆ, N., 2001, “Variable neighborhood search: Principles and applications”, *European journal of operational research*, v. 130, pp. 449–467.

- [62] HANSEN, P., MLADENOVIĆ, N., 2006, “First vs. best improvement: An empirical study”, *Discrete Applied Mathematics*, v. 154, pp. 802 – 817. ISSN: 0166-218X. doi: <https://doi.org/10.1016/j.dam.2005.05.020>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0166218X05003070>>. IV ALIO/EURO Workshop on Applied Combinatorial Optimization.
- [63] HELD, M., KARP, R. M., 1970, “The traveling-salesman problem and minimum spanning trees”, *Operations Research*, v. 18, pp. 1138–1162.
- [64] HIFI, M., NEGRE, S., WU, L., 2014, “Hybrid greedy heuristics based on linear programming for the three-dimensional single bin-size bin packing problem”, *International Transactions in Operational Research*, v. 21, pp. 59–79.
- [65] HWANG, F. K., RICHARDS, D. S., 1992, “Steiner tree problems”, *Networks*, v. 22, pp. 55–89.
- [66] JEPSEN, M. K. AND PETERSEN, B., SPOORENDONK, S., PISINGER, D., 2014, “A branch-and-cut algorithm for the capacitated profitable tour problem”, *Discrete Optimization*, v. 14, pp. 78–96.
- [67] JÜNGER, M., REINELT, G., RINALDI, G., 1995, “The traveling salesman problem”, *Handbooks in operations research and management science*, v. 7, pp. 225–330. Elsevier.
- [68] LEIFER, A. C., ROSENWEIN, M. B., 1994, “Strong linear programming relaxations for the orienteering problem”, *European Journal of Operational Research*, v. 73, pp. 517–523.
- [69] LIN, S., KERNIGHAN, B. W., 1973, “An effective heuristic algorithm for the traveling-salesman problem”, *Operations research*, v. 21, pp. 498–516.
- [70] LOURENÇO, H. R., MARTIN, O. C., STÜTZLE, T., 2003, “Iterated Local Search”. In: Glover, F., Kochenberger, G. A. (Eds.), *Handbook of Metaheuristics*, pp. 320–353, Boston, MA, Springer US. ISBN: 978-0-306-48056-0. doi: 10.1007/0-306-48056-5_11. Disponível em: <https://doi.org/10.1007/0-306-48056-5_11>.
- [71] LYRA, A. R. D., 2004, *O problema de recobrimento de rotas com coleta de prêmios: regras de redução, formulação matemática e heurísticas*. Dissertação de Mestrado, Programa de Pós-Graduação em Computação, Instituto de Computação, Universidade Federal Fluminense, Niterói, RJ.
- [72] MELO, V. A., MARTINHON, C. A., 2000, “Metaheurísticas híbridas para o problema do caixeiro viajante com coleta de prêmios”. In: *Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, v. 36, pp. 1295–1306. Viçosa, Brazil.
- [73] MLADENOVIĆ, N., HANSEN, P., 1997, “Variable Neighborhood Search”, *Computers & Operations Research*, v. 24, pp. 1097–1100. ISSN: 0305-0548. doi:

10.1016/S0305-0548(97)00031-2. Disponível em: <[http://dx.doi.org/10.1016/S0305-0548\(97\)00031-2](http://dx.doi.org/10.1016/S0305-0548(97)00031-2)>.

- [74] NADDEF, D., 2007, “Polyhedral Theory and Branch-and-Cut Algorithms for the Symmetric TSP”. In: Gutin, G., Punnen, A. P. (Eds.), *The Traveling Salesman Problem and Its Variations*, pp. 29–116, Boston, MA, Springer US. ISBN: 978-0-306-48213-7. doi: 10.1007/0-306-48213-4_2. Disponível em: <https://doi.org/10.1007/0-306-48213-4_2>.
- [75] OLIVEIRA, A. C. M. D., CHAVES, A. A., LORENA, L. A. N., 2013, “Clustering search”, *Pesquisa operacional*, v. 33, pp. 105–121.
- [76] OPTIMIZATION, G., 2018. “Gurobi optimizer reference manual”. Disponível em: <<http://www.gurobi.com>>. Acessado em: 10 de Novembro de 2018.
- [77] PADBERG, M. W., RAO, M. R., 1982, “Odd minimum cut-sets and b-matchings”, *Mathematics of Operations Research*, v. 7, pp. 67–80.
- [78] PASSMARK, 2019. “CPU Benchmarks.” Disponível em: <<https://www.cpubenchmark.net/compare.php>>. Acessado em: 25 de Setembro de 2019.
- [79] PEDRO, O. R., 2013, *Uma abordagem de Busca Tabu para o Problema do Caixeiro Viajante com Coleta de Prêmios*. Dissertação de Mestrado, Universidade Federal de Minas Gerais, Belo Horizonte, MG.
- [80] PEDRO, O., SALDANHA, R., CAMARGO, R., 2013, “A Tabu Search Approach for the Prize Collecting Traveling Salesman Problem”, *Electronic Notes in Discrete Mathematics*, v. 41, pp. 261 – 268. ISSN: 1571-0653. doi: <https://doi.org/10.1016/j.endm.2013.05.101>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1571065313001042>>.
- [81] PINTO, B. Q., RIBEIRO, C. C., ROSSETI, I., PLASTINO, A., 2018, “A biased random-key genetic algorithm for the maximum quasi-clique problem”, *European Journal of Operational Research*, v. 271, pp. 849 – 865. ISSN: 0377-2217. doi: <https://doi.org/10.1016/j.ejor.2018.05.071>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221718304909>>.
- [82] PLASTINO, A., FONSECA, E., FUCHSHUBER, R., et al., 2009, “A Hybrid Data Mining Metaheuristic for the p-median Problem.” In: *Proceedings of the SIAM International Conference on Data Mining*, pp. 189–198, Sparks, NV, EUA.
- [83] PLASTINO, A., FUCHSHUBER, R., MARTINS, S. L., FREITAS, A. A., SALHI, S., 2011, “A hybrid data mining metaheuristic for the p-median problem”, *Statistical Analysis and Data Mining: The ASA Data Science Journal*, v. 4, pp. 313–335.
- [84] PLASTINO, A., BARBALHO, H., SANTOS, L. F. M., FUCHSHUBER, R., MARTINS, S., 2014, “Adaptive and multi-mining versions of the DM-GRASP hybrid metaheuristic”, *Journal of Heuristics*, v. 20, pp. 39–74.

- [85] R CORE TEAM, 2017, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. Disponível em: <<https://www.R-project.org/>>.
- [86] REINELT, G., 1991, “TSPLIB—A traveling salesman problem library”, *ORSA journal on computing*, v. 3, pp. 376–384.
- [87] REINELT, G., 1994, *The traveling salesman: computational solutions for TSP applications*. Berlim, Germany. Springer-Verlag.
- [88] RESENDE, M. G., RIBEIRO, C. C., 2005, “GRASP with Path-Relinking: Recent Advances and Applications”. In: Ibaraki, T., Nonobe, K., Yagiura, M. (Eds.), *Metaheuristics: Progress as Real Problem Solvers*, pp. 29–63, Boston, MA, Springer US. ISBN: 978-0-387-25383-1. doi: 10.1007/0-387-25383-1_2. Disponível em: <https://doi.org/10.1007/0-387-25383-1_2>.
- [89] RIBEIRO, M. H., PLASTINO, A., MARTINS, S. L., 2006, “Hybridization of GRASP metaheuristic with data mining techniques”, *Journal of Mathematical Modelling and Algorithms*, v. 5, pp. 23–41.
- [90] ROCHAT, Y., TAILLARD, E. D., 1995, “Probabilistic diversification and intensification in local search for vehicle routing”, *Journal of heuristics*, v. 1, pp. 147–167.
- [91] SANTOS, L. F., RIBEIRO, M. H., PLASTINO, A., MARTINS, S. L., 2005, “A hybrid GRASP with data mining for the maximum diversity problem”. In: *International Workshop on Hybrid Metaheuristics*, pp. 116–127. Springer. Barcelona, Spain.
- [92] SIEGEL, S., 1956, *Nonparametric statistics for the behavioral sciences*. New York, United States of America. McGraw-hill.
- [93] SILVA, M. S. A., 2009, *Problema de Recobrimento de Rotas com Coleta de Prêmios*. Dissertação de Mestrado, Programa de Pós-Graduação em Computação, Instituto de Computação, Universidade Federal Fluminense, Niterói, RJ.
- [94] TALBI, E.-G., 2002, “A taxonomy of hybrid metaheuristics”, *Journal of heuristics*, v. 8, pp. 541–564.
- [95] TORRES, R. D., BRITO, J. A. M., 2003, “Problemas de coleta de prêmios seletiva”. In: *Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, v. 35, pp. 1359–1371. Natal, Brazil.
- [96] VANSTEENWEGEN, P., SOUFFRIAU, W., VAN OUDHEUSDEN, D., 2011, “The orienteering problem: A survey”, *European Journal of Operational Research*, v. 209, pp. 1–10.
- [97] WOLSEY, L. A., NEMHAUSER, G. L., 2014, *Integer and combinatorial optimization*. John Wiley & Sons.