UNIVERSIDADE FEDERAL FLUMINENSE

Luciene Cristina Soares Motta

O Problema de Recobrimento por Rotas: algoritmos e regras de redução

NITERÓI 2010

UNIVERSIDADE FEDERAL FLUMINENSE

Luciene Cristina Soares Motta

O Problema de Recobrimento por Rotas: algoritmos e regras de redução

Tese de Doutorado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Doutor. Área de concentração: Algoritmos e Otimização.

Orientador: Luiz Satoru Ochi

> NITERÓI 2010

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da UFF

M921	Motta, Luciene Cristina Soares. O problema de recobrimento por rotas : algoritmos e regras de redução / Luciene Cristina Soares Motta. – Niterói, RJ : [s.n.], 2010. 175 f.
	Tese (Doutorado em Computação) - Universidade Federal Fluminense, 2010. Orientador: Luiz Satoru Ochi.
	1. Problema de recobrimento por rotas. 2. Heurística. 3. Regras de redução. 4. Algoritmo. 5. Ciência da computação. I. Título.
	CDD 005.136

O Problema de Recobrimento por Rotas: algoritmos e regras de redução

Luciene Cristina Soares Motta

Tese de Doutorado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Doutor.

Aprovada por:

Profe. Luiz Satoru Ochi / IC-UFF (Presidente)

Prof^a. Cláudia Linhares Sales / DC-UFC

Prof^a. Loana Tito Nogueira / IC-UFF

 $\mathrm{Prof}^{a}\overset{\smile}{.}$ Márcia Helena Costa Fampa / COPPE-UFRJ

Prof^a. Simone de Lima Martins / IC-UFF

Niterói, maio de 2010.

Dedico este trabalho ao meu marido, amigo e companheiro de todas as horas. Aos meus pais, irmãos, cunhados e à minha vozinha querida, pois continuam sendo o porto seguro onde sempre posso ancorar o meu barco.

Agradecimentos

Em primeiro lugar agradeço à Deus, por ter me presenteado com a dádiva da vida, por estar sempre ao meu lado me protegendo, guiando meus passos e por fazer de cada amanhecer um momento único para a renovação dos sonhos, esperanças...

Aos meus pais, por todo amor, carinho, dedicação, pelo investimento sempre depositado com confiança na minha formação e, acima de tudo, por terem dado condições de me tornar a pessoa que sou hoje. Amo muito vocês! Obrigado por existirem de forma tão especial na minha vida.

À minha vozinha Jurandy, pelo amor, pela amizade, pelo incentivo e pelas orações incansáveis que, sem dúvida, sempre tiveram um papel fundamental em todas as minhas conquistas.

Ao meu marido, por todo amor, dedicação, carinho, compreensão, mesmo nas horas de mau humor, ausência... Obrigada por compartilhar comigo todos os momentos da elaboração deste trabalho sempre com um sorriso carinhoso, com um abraço aconchegante e por jamais duvidar que sairíamos vitoriosos em mais esta etapa da nossa vida. Que Deus te conserve sempre assim!

Aos meus irmãos Letícia, Lucimar, Leonardo e Rodrigo, amigos e cúmplices de todas as horas. Sem vocês minha vida não teria a mesma alegria. Que bom que vocês existem!

Aos meus padrinhos Inezil, Solange e Rosiene (*in memoriam*), por todo carinho e confiança.

Aos meus cunhados, pela torcida, carinho e amizade.

Ao meu irmão de coração, José Luiz, pela presença constante, pelo apoio, pelas inúmeras madrugadas, pela ajuda nas tabulações dos dados e, acima de tudo, pela amizade sincera e pela mão sempre estendida. À Sônia, Mariana e Bernardo, pelas muitas acolhidas e por todo incentivo.

Ao meu orientador, professor Luiz Satoru Ochi, pelas oportunidades e pela orientação que tornou este trabalho possível. Aos membros da banca examinadora, professoras Cláudia Linhares Sales, Loana Tito Nogueira, Márcia Helena Costa Fampa e Simone de Lima Martins, pelas sugestões que muito contribuíram e enriqueceram este trabalho.

Aos amigos Jacques, Juliana, Diego e Tiago, pelo incentivo nas horas de desânimo e por toda ajuda. Em especial, meu muito obrigado ao Tiago pelas horas de discussões sobre a implementação. Valeu! Aprendi muito!

Aos amigos do IC: Puca, Anand, André Renato, Mário, Luciana Brugiolo, Juliano, Maurício Guedes, Daniel, Idalmis, Alessandro Copetti, Adria, Janine, Luciana Pessôa, Renatha, Aletéia, Haroldo, Cristiano e tantos outros que fizeram parte desta história, direta ou indiretamente.

Aos funcionários do suporte, Carlos e Rafael e às secretárias Angela, Isabela, Maria, Teresa e Viviane pelos auxílios prestados. Ao amigo Hans, meu muito obrigado mais do que especial pelas inúmeras noites que passou comigo discutindo o trabalho via Skype, pelas sugestões que contribuíram de forma única e, acima de tudo, pela amizade e consideração.

Ao professor Valdomiro Neves Lima, por ter acreditado no meu potencial desde o início e por ter plantado a primeira semente da minha história acadêmica.

À Ângela Patrício, pela disponibilidade e boa vontade em ajudar com seu conhecimento estatístico.

Ao Wagner, pela ajuda nas implementações. Nossas conversas sobre C++ e Orientação à Objetos realmente contribuíram para a melhoria das heurísticas implementadas.

À todos aqueles que não foram diretamente citados nestes agradecimentos e que, de uma forma ou de outra, ajudaram a concretizar este trabalho.

"Então me diz qual é a graça De já saber o fim da estrada, Quando se parte rumo ao nada?" Paulinho Moska

Resumo

Este trabalho tem como objetivo apresentar novas alternativas de solução para uma generalização do Problema do Caixeiro Viajante (PCV) ainda pouco explorado na literatura, conhecido como Problema de Recobrimento por Rotas (PRR). O PRR é classificado como NP-Difícil e pode ser definido na estrutura de um grafo não direcionado $G = (V \cup W, E)$, onde V é o conjunto dos vértices que podem ser visitados, W é o conjunto dos vértices que devem ser cobertos e $T \subseteq V$ é o conjunto dos vértices que devem ser visitados. O problema consiste em determinar uma rota de comprimento mínimo sobre um subconjunto de V e contendo todos os vértices de T, de modo que todo vértice de W esteja no máximo a uma distância pré-estabelecida de algum vértice pertencente à rota.

Adicionalmente, este trabalho aborda uma variante do PRR, chamada de Problema de Recobrimento por Rotas Generalizado (PRRG), onde os vértices $w \in W$, ao contrário do PRR, podem fazer parte da solução.

Entre as contribuições apresentadas neste trabalho para o PRR e para o PRRG relacionam-se: novas regras de redução para os grafos associados, uma comparação entre as formulações matemáticas existentes na literatura para ambos os problemas, heurísticas de construção e busca local, versões da meta-heurística GRASP com e sem mecanismos baseados em memória, uma proposta baseada na meta-heurística *Iterated Local Search*, além de alguns resultados teóricos que estabelecem uma relação entre os referidos problemas.

Experimentos computacionais apresentam as soluções exatas e heurísticas obtidas para um conjunto de instâncias do PRR e do PRRG e é verificado o impacto do uso de regras de redução na obtenção destas soluções. Uma análise estatística é realizada para avaliar o desempenho das heurísticas propostas.

Palavras-chave: Problema de Recobrimento por Rotas, Formulações Matemáticas, Regras de Redução, Heurísticas.

Abstract

This work presents new alternative solutions to a generalization of the Traveling Salesman Problem (TSP) few explored in literature, known as the Covering Tour Problem (CTP). The CTP is classified as NP-Hard and can be modeled as an undirected graph $G = (V \cup W, E)$ where V is the set of vertices that can be visited, W is the set of vertices to be covered and $T \subseteq V$ is the set of vertices that must be visited. The CTP consists in determining a minimum length cycle over a subset of V which contains all vertices of T, and every vertex of W must be at a distance not higher than d of any vertex present in the route.

In addition, this work presents a generalization of CTP called Generalized Covering Tour Problem (GCTP), where the vertices $w \in W$ can be part of the solution, in oposition to the CTP.

The proposals presented in this thesis to the CTP and GCTP includes: new reduction rules to the associated graphs, a comparison between the mathematical formulations of the literature for both problems, constructive heuristics, local search algorithms, GRASP versions with and without memory-based mechanisms, a proposal based on Iterated Local Search metaheuristic and some theoretical results that establish a relation between the refferd problems.

Computational experiments show the exact and heuristics solutions obtained for several instances of CTP and GCTP and the impact of the reduction rules on shuch instances is also discussed. An statistical analysis is performed to evaluate the proposed heuristics.

Keywords: The Covering Tour Problem, Mathematical Formulations, Reduction Rules, Heuristics.

Palavras-chave

- 1. Problema de Recobrimento por Rotas
- 2. Formulações Matemáticas
- 3. Regras de Redução
- 4. Meta-heurísticas

Sumário

Li	sta de	Figura	S	xiii
Li	sta de	Tabela	ıs	xv
Li	sta de	Siglas	e Acrônimos	xviii
1	Intro	odução		1
	1.1	Objeti	vos e organização do trabalho	2
2	O Pr	oblema	a de Recobrimento por Rotas e Recobrimento por Rotas Generalizado) 4
	2.1	Definio	;ão dos problemas	4
	2.2	Trabal	hos relacionados na literatura	7
	2.3	Proble	mas correlatos	14
	2.4	Aplica	ções	16
3	Regr	as para	a redução do grafo associado ao PRR e ao PRRG	19
	3.1	Regras	de redução para o PRR	20
		3.1.1	Regras de redução existentes na literatura para o PRR	20
		3.1.2	Nova regra de redução proposta para o PRR	27
		3.1.3	Experimentos computacionais com o conjunto de regras de redução proposto para o PRR	31
	3.2	Regras	de redução para o PRRG	34
		3.2.1	Regras de redução existentes na literatura para o PRRG	34
		3.2.2	Nova regra de redução para o PRRG	37

		3.2.3	Experimentos computacionais com o conjunto de regras de redução proposto para o PRRG	40
4	Forn	nulaçõe	s matemáticas para o PRR e para o PRRG	44
	4.1	Formu	lações matemáticas existentes	44
		4.1.1	Formulação de Gendreau, Laporte e Semet para o PRR	45
		4.1.2	Formulação de Maniezzo, Baldacci, Boschetti e Zamboni para o PRR	46
		4.1.3	Formulação de Motta para o PRR e para o PRRG	48
	4.2	Compa	aração entre as formulações da literatura	50
	4.3	Experi	imentos computacionais para o ajuste dos parâmetros do CPLEX	56
		4.3.1	Parâmetro Nodeselect	57
		4.3.2	Parâmetro Variables elect	58
		4.3.3	Parâmetro Branch	60
		4.3.4	Parâmetro <i>Startalgorithm</i>	61
		4.3.5	Resultados obtidos com a formulação $\mathcal{F}2'$	64
	4.4	Algum	as relações entre o PRR e o PRRG	68
5	Abo	rdagens	s heurísticas propostas para o PRR e para o PRRG	70
	5.1	Heurís	ticas de Construção	71
		5.1.1	Heurística AdicaoSucessiva	71
		5.1.2	Heurística RemocaoSucessiva	74
		5.1.3	Heurística IMP-PRR	78
		5.1.4	Heurística IMD-PRR	81
		5.1.5	Heurística IMB-PRR	86
	5.2	Heurís	ticas de Busca Local	88
		5.2.1	Heurística k-AddDrop	88
		5.2.2	Heurística k-DropAdd	92
		5.2.3	Heurística 20pt-PRR	95

		5.2.4	Heurística	a VNS-PRR	. 97
			5.2.4.1	Heurística VND-PRR	. 99
	5.3	Metod	ologia dos	experimentos	. 102
	5.4	Exper	mentos co	mputacionais com as heurísticas de construção e de busca	
		local			. 105
	5.5	Variaç	ões da met	ta-heurística GRASP	. 114
		5.5.1	GRASP I	Básico	. 116
		5.5.2	GRASP I	Reativo	. 117
			5.5.2.1	GRASP Reativo-EQ	. 118
			5.5.2.2	GRASP Reativo-QA	. 119
	5.6	Iterate	d Local Se	arch (ILS)	. 120
	5.7	Estrat	égia de int	ensificação utilizando a Reconexão por Caminhos	. 122
	5.8	Exper	mentos co	mputacionais com as heurísticas GRASP e ILS	. 125
6	Con	clusões			134
R	eferên	cias			137
$\mathbf{A}_{]}$	pêndio	ce A – 1	nstâncias	para o PRR e para o PRRG	142
	A.1	Grupo	s de instâr	ncias gerados	. 142
	A.2	Nomer	nclatura da	as instâncias e representação das estruturas de dados $\ . \ .$. 145
$\mathbf{A}_{]}$	pêndio	ce B – I	Experimen	tos computacionais com as regras de redução propostas	150
$\mathbf{A}_{]}$	pêndio	ce C – I	Resultados	complementares dos experimentos com o CPLEX	161
$\mathbf{A}_{]}$	pêndio	ce D - 7	Festes de i	nferência estatística	166
	D.1	Anális	e de variâr	ncia (ANOVA)	. 166
	D.2	Teste	t-Student		. 168

Lista de Figuras

2.1	Exemplo de uma instância do PRR com 23 vértices	5
2.2	Exemplo de uma solução viável para o PRR associada ao grafo da Figura 2.1	6
2.3	Diferença entre uma solução viável para o PRR e uma para o PRRG associadas ao grafo da Figura 2.1	7
2.4	Solução obtida para uma instância do CSP pela heurística COVTOUR $\ .\ .\ .$.	8
3.1	Exemplo da utilização da regra de redução R1 em um grafo associado ao ${\rm PRR}$	22
3.2	Exemplo da utilização da regra de redução R3 em um grafo associado ao $\ensuremath{\mathrm{PRR}}$.	23
3.3	Exemplo da aplicação da regra R6 no grafo da Figura 2.1 \ldots	24
3.4	Exemplo da aplicação da regra R5 no grafo da Figura 3.3	24
3.5	Exemplo da aplicação da regra R4 no grafo da Figura 3.4	25
3.6	Grafo da Figura 2.1 reduzido pela aplicação do conjunto de regras propostas para o PRR em (BRITO, 2005)	25
3.7	Exemplo da utilização da regra de redução R2 em um grafo associado ao PRR $\ .$	26
3.8	Exemplos da aplicação da nova regra ${\rm R7}$ em três grafos associados ao ${\rm PRR}$	28
3.9	Exemplo do uso da regra de redução R5 proposta para o PRR quando é aplicada diretamente em um grafo associado ao PRRG	35
3.10	Exemplo da aplicação da regra R8 no grafo da Figura 2.1	36
3.11	Exemplo da aplicação da regra de redução R4 no grafo da Figura 3.10	37
3.12	Grafo associado ao PRRG resultante da aplicação do conjunto de regras de re- dução propostas em (MOTTA, 2001)	37
3.13	Exemplo do uso da regra de redução R9 quando aplicada a um grafo associado ao PRRG	38
4.1	Espaço de soluções do PRR e do PRRG	68

[5.1	Passo a passo da construção de uma solução inicial para CTP a partir da heu- rística <i>AdicaoSucessiva</i>	74
[5.2	Passo a passo da construção de uma solução inicial para CTP a partir da heu- rística <i>RemocaoSucessiva</i>	77
[5.3	Passo a passo da construção de uma solução inicial para CTP a partir da heu- rística <i>IMP-PRR</i>	80
[5.4	Passo a passo da construção de uma solução inicial para CTP a partir da heu- rística <i>IMD-PRR</i>	83
	5.5	Passo a passo da construção de uma solução inicial para CTP a partir da heurística IMD_{MinMax} - PRR	84
[5.6	Passo a passo da construção de uma solução inicial para CTP a partir da heurística IMD_{MaxMin} - PRR	85
	5.7	Passo a passo da construção de uma solução inicial para CTP a partir da heurística <i>IMB-PRR</i>	87
[5.8	Passo a passo de uma iteração da busca local 1- $AddDrop$ realizada em uma solução inicial construída pela heurística IMD_{MinMax} - PRR	90
[5.9	Passo a passo de uma iteração da busca local 2- $AddDrop$ realizada em uma solução inicial construída pela heurística IMD_{MinMax} - PRR	91
[5.10	Passo a passo de uma iteração da busca local 1 - $DropAdd$ realizada em uma solução inicial construída pela heurística IMD_{MinMax} - PRR	94
[5.11	Passo a passo de uma iteração da busca local 2- $DropAdd$ realizada em uma solução inicial construída pela heurística IMD_{MinMax} - PRR	95
[5.12	Passo a passo de uma iteração da busca local $2Opt$ - PRR realizada em uma solução inicial construída pela heurística IMD_{MinMax} - PRR	96
,	5.13	Exemplo da utilização da estratégia de RC proposta para o PRR e para o PRRG	124
4	A.1	Estruturas que representam a cobertura dos vértices de W da instância $ctp23_6-7-10$ do PRR	147
	A.2	Estruturas que representam a cobertura dos vértices de W da instância $gctp23_6-7-10$ do PRRG	148
,	A.3	Representação gráfica de uma rota viável para a instância $ctp23_6-7-10$	149

Lista de Tabelas

3.1	Redução das instâncias do Grupo III (PRR) - Regras {R7,R6,R5,R4}	32
3.2	Redução das instâncias do Grupo IV (PRR) - Regras {R7,R6,R5,R4}	33
3.3	Redução das instâncias do Grupo I (PRRG) - Regras {R9,R8,R4}	42
3.4	Redução das instâncias do Grupo II (PRRG) - Regras {R9,R8,R4} $\ .$	43
4.1	Número de restrições e de variáveis em relação ao tamanho do problema de entrada	51
4.2	Resultados da relaxação linear de $\mathcal{F}2$ e $\mathcal{F}3$ para o PRRG \ldots	53
4.3	Resultados da relaxação linear de $\mathcal{F}2$ e $\mathcal{F}3$ para o PRR $\ldots \ldots \ldots \ldots \ldots$	53
4.4	Comparativo entre os resultados da relaxação linear de $\mathcal{F}2$ e $\mathcal{F}2'$ para as instâncias do PRR e PRRG	54
4.5	Resultados dos experimentos com o parâmetro <i>nodeselect</i>	58
4.6	Resultados dos experimentos com o parâmetro variables elect	59
4.7	Resultados dos experimentos com o parâmetro <i>branch</i>	61
4.8	Resultados dos experimentos com o parâmetro <i>startalgorithm</i>	63
4.9	Custo das soluções ótimas obtidas com a formulação $\mathcal{F}2'$ para o PRRG - GI $$	65
4.10	Custo das soluções ótimas obtidas com a formulação $\mathcal{F}2'$ para o PRR - GIII $~$	66
4.11	Custo das soluções ótimas obtidas com $\mathcal{F}2'$ para instâncias de GII e GIV \ldots	67
5.1	Comparação do desempenho das heurísticas construtivas para o PRRG conside- rando as instâncias originais e reduzidas	108
5.2	Comparação do desempenho das heurísticas construtivas para o PRR conside- rando as instâncias originais e reduzidas	110
5.3	Resultado do teste t-Student com as heurísticas de busca local - Grafos Reduzidos - PRRG	112

5.4	Comparação do desempenho das heurísticas de busca local para as instâncias reduzidas do PRRG	113
5.5	Comparação do desempenho das heurísticas de busca local para as instâncias reduzidas do PRR	114
5.6	Versões do GRASP Básico propostas para o PRR e PRRG	117
5.7	Resultado do teste <i>t-Student</i> com as heurísticas GRASP básicas aplicadas aos grafos reduzidos do PRRG	128
5.8	Comparação do desempenho das heurísticas GRASP básicas aplicadas aos grafos reduzidos do PRRG (GII)	128
5.9	Comparação do desempenho das heurísticas GRASP básicas aplicadas aos grafos reduzidos do PRR (GIV)	129
5.10	Comparação do desempenho das heurísticas GRASP Reativo-EQ e GRASP Reativo- QA aplicadas aos grafos reduzidos do PRRG (GII)	131
5.11	Comparação do desempenho das heurísticas GRASP Reativo-EQ e GRASP Reativo- QA aplicadas aos grafos reduzidos do PRR (GIV)	131
5.12	Comparação do desempenho das heurísticas GRASP Reativo-QA com reconexão por caminhos e ILS-PRR quando aplicadas aos grafos reduzidos do PRRG (GII)	132
5.13	Comparação do desempenho das heurísticas GRASP Reativo-QA com reconexão por caminhos e ILS-PRR quando aplicadas aos grafos reduzidos do PRR (GIV) $$.	133
A.1	Cardinalidade de cada conjunto que compõe as instâncias geradas \ldots .	144
B.1	Redução das instâncias do PRR (Grupo III) - Sequências 1 e 2 das regras de redução	151
B.2	Redução das instâncias do PRR (Grupo III) - Sequências 3 e 4 das regras de redução	152
B.3	Redução das instâncias do PRR (Grupo III) - Sequências 5 e 6 das regras de redução	153
B.4	Redução das instâncias do PRR (Grupo IV) - Sequências 1 e 2 das regras de redução	154
B.5	Redução das instâncias do PRR (Grupo IV) - Sequências 3 e 4 das regras de redução	155

B.6	Redução das instâncias do PRR (Grupo IV) - Sequências 5 e 6 das regras de redução	156
B.7	Redução das instâncias do PRRG (Grupo I) - Sequências 1 e 2 das regras de redução	157
B.8	Redução das instâncias do PRRG (Grupo I) - Sequências 3 e 4 das regras de redução	158
B.9	Redução das instâncias do PRRG (Grupo II) - Sequências 1 e 2 das regras de redução	159
B.10	Redução das instâncias do PRRG (Grupo II) - Sequências 3 e 4 das regras de redução	160
C.1	Ótimos duais obtidos com a formulação $\mathcal{F}2'$ para o PRR - Grupo III $\ldots \ldots$	162
C.2	Ótimos duais obtidos com a formulação $\mathcal{F}2'$ para o PRRG - Grupo I $\ \ldots \ \ldots$	163
С.3	Comparação dos resultados da formulação $\mathcal{F}2'$ obtidos para o PRR - Instâncias originais e reduzidas do Grupo III $\ldots \ldots \ldots$	164
С.4	Comparação dos resultados da formulação $\mathcal{F}2'$ obtidos para o PRRG - Instâncias originais e reduzidas do Grupo I $\ldots \ldots \ldots$	165
E.1	Comparação entre as melhores soluções obtidas pelas oito heurísticas construtivas para o PRR e as soluções ótimas conhecidas	172
E.2	Comparação entre as melhores soluções obtidas pelas oito heurísticas construtivas para o PRRG e as soluções ótimas conhecidas	173
E.3	Comparação entre as melhores soluções obtidas pelo GRASP_Bas1 para o PRR e as soluções ótimas conhecidas	174
E.4	Comparação entre as melhores soluções obtidas pelo GRASP_Bas1 para o PRRG e as soluções ótimas conhecidas	175

Lista de Siglas e Acrônimos

2-Opt	:	$2 ext{-}Optimal$
ACS	:	Ant Colony System
BT	:	Busca Tabu
CF	:	Colônia de Formigas
CSP	:	Covering Salesman Problem
CSPP	:	Covering Shortest-Path Problem
CTACS	:	Covering Tour Ant Colony System
CTP	:	Covering Tour Problem
GACS	:	GENI Ant Colony System
GCSP	:	Geometric Covering Salesman Problem
GTSP	:	Generalized Traveling Salesman Problem
GRASP	:	Greedy Randomized Adaptive Search Procedure
IC	:	Instituto de Computação
ILS	:	Iterated Local Search
LB	:	Lower Bound
LRC	:	Lista Restrita de Candidatos
m-CTP	:	Multi-Vehicle Covering Tour Problem
MCTP	:	Maximal Covering Tour Problem
MRD	:	Método Randômico de Descida
MTP	:	Median Tour Problem
NP	:	Nondeterministic Polynomial Time
PCCTP	:	Prize Collecting Covering Tour Problem
PCTSP	:	Prize Collecting Traveling Salesman Problem
PCV	:	Problema do Caixeiro Viajante
PCVA	:	Problema do Caixeiro Viajante Assimétrico
PLI	:	Programação Linear Inteira
PPLI	:	Problema de Programação Linear Inteira
\mathbf{PR}	:	Path Relinking

PRR	:	Problema de Recobrimento por Rotas
PRR-CP	:	Problema de Recobrimento por Rotas com Coleta de Prêmios
PRRG	:	Problema de Recobrimento por Rotas Generalizado
\mathbf{RC}	:	Reconexão por Caminhos
RVNS	:	Reduced Variable Neighborhood Search
SCACS	:	Set Covering Ant Colony System
SCP	:	Set Covering Problem
SCPP	:	Shortest Covering Path Problem
SS	:	Scatter Search
STSP	:	Selective Traveling Salesman Problem
TPP	:	Traveling Purchaser Problem
TSP	:	Traveling Salesman Problem
UB	:	Upper Bound
UFF	:	Universidade Federal Fluminense
VND	:	Variable Neighborhood Descent
VNRD	:	Variable Neighborhood Random Descent
VNS	:	Variable Neighborhood Search

Capítulo 1

Introdução

Este trabalho aborda duas variantes do clássico Problema do Caixeiro Viajante (PCV) ainda pouco exploradas na literatura, conhecidas como Problema de Recobrimento por Rotas (PRR) e Problema de Recobrimento por Rotas Generalizado (PRRG).

O PRR pode ser definido na estrutura de um grafo não direcionado $G = (V \cup W, E)$, onde V é o conjunto dos vértices que podem ser visitados, W é o conjunto dos vértices que devem ser cobertos e $T \subseteq V$ é o conjunto dos vértices que devem ser visitados. O problema consiste em determinar uma rota de comprimento mínimo sobre um subconjunto de V que contenha T, de modo que todo vértice de W esteja no máximo a uma distância d de algum vértice da rota, sendo $d \ge 0$ a distância de cobertura (CURRENT, 1981).

O PRRG difere do PRR por permitir que vértices do conjunto W possam fazer parte da solução do problema, tanto para cobrir a si próprio, quanto para cobrir outros vértices de W (MOTTA, 2001).

O PRR foi introduzido em (CURRENT, 1981), mas só foi formulado matematicamente em (GENDREAU; LAPORTE; SEMET, 1997), onde também foi proposto um algoritmo exato baseado no método branch-and-cut, uma heurística para fornecer um limite superior inicial para o algoritmo exato e quatro regras para reduzir iterativamente os conjuntos $W \, e \, V$. Além destes, são encontrados na literatura mais quatro trabalhos relacionados ao PRR/PRRG: (MANIEZZO et al., 1999), (MOTTA, 2001), (BRITO, 2005) e (KUBIK, 2007). Maniezzo et al. (1999) propuseram para o PRR três algoritmos baseados na metaheurística Scatter Search, um novo modelo de Programação Linear Inteira (PLI) e uma nova regra para reduzir o conjunto W. Motta (2001) introduziu o PRRG, propôs um conjunto de regras de redução e algoritmos baseados na meta-heurística GRASP (Greedy Randomized Adaptive Search Procedure) para este problema, analisou o conjunto de regras de redução proposto para o PRR em (MANIEZZO et al., 1999) e propôs uma formulação matemática baseada em PLI para ambos os problemas. Brito (2005) desenvolveu para o PRR uma heurística lagrangeana, uma nova regra de redução e algoritmos heurísticos também baseados na meta-heurística GRASP. Kubik (2007) propôs dois algoritmos baseados na meta-heurística Ant Colony System para resolver o PRR. É desconhecida a existência de outra abordagem, exata ou heurística, proposta na literatura para o PRR ou para o PRRG.

Diversas aplicações podem ser encontradas para o PRR e para o PRRG, tais como o roteamento de despachos aéreos, o planejamento de serviços médicos em regiões de difícil acesso, a localização de facilidades (hospitais, escolas, postos policiais), o planejamento do patrulhamento urbano preventivo, coleta de lixo seletivo e o planejamento das rotas de veículos de transporte público de alunos. Entretanto, apesar da evidente importância prática, nota-se que poucos trabalhos relacionados ao PRR/PRRG são encontrados na literatura, fato que motivou a escolha do tema desta tese.

Por serem reduzidos ao Problema do Caixeiro Viajante, quando a distância de cobertura é nula (d = 0) e todo vértice de W coincide com os vértices de V, ambos os problemas em estudo nesta tese são considerados NP-difíceis (CURRENT, 1981), o que sugere maior dificuldade no desenvolvimento de metodologias de solução eficientes para resolver estes problemas, tornando este trabalho de pesquisa ainda mais desafiador.

1.1 Objetivos e organização do trabalho

O presente trabalho tem como objetivo a pesquisa e o desenvolvimento de métodos computacionais eficientes para a solução do Problema de Recobrimento por Rotas e do Problema de Recobrimento por Rotas Generalizado.

Entre os objetivos específicos desta pesquisa, pode-se relacionar a proposta de novas regras de redução para os grafos associados ao PRR e ao PRRG, o estabelecimento de algumas relações teóricas entre estes problemas, um estudo experimental das formulações matemáticas existentes para ambos os problemas, a avaliação do impacto da configuração dos parâmetros do algoritmo *branch-and-cut* utilizado pelo CPLEX¹ na obtenção da solução exata dos problemas em estudo e o desenvolvimento de novas abordagens heurísticas para a solução aproximada do PRR e do PRRG.

¹ O CPLEX é um *software* comercial utilizado para resolver problemas de programação linear. O algoritmo *branch-and-cut* é usado por este *software* para resolver problemas de programação inteira e pode ter seus parâmetros configurados visando a melhoria do seu desempenho.

Este documento está organizado em 5 capítulos, além desta introdução:

Capítulo 2: *O Problema de Recobrimento por Rotas e Recobrimento por Rotas Generalizado*, define formalmente os problemas que são objetos de estudo desta teste, realiza um revisão bibliográfica das abordagens computacionais relacionadas a estes problemas, assim como apresenta uma relação de problemas similares e aplicações relacionadas ao PRR e ao PRRG;

Capítulo 3: Regras de Redução para o PRR e para o PRRG, detalha as regras para a redução dos grafos associados ao PRR e ao PRRG encontradas na literatura, bem como propõe novas regras para ambos. Uma análise experimental da utilização do conjunto de regras proposto na redução de 144 instâncias também é apresentada;

Capítulo 4: Formulações Matemáticas para o PRR e para o PRRG, descreve as formulações matemáticas existentes para o PRR e para o PRRG, apresenta um comparativo entre estas formulações, avalia o impacto da calibração dos parâmetros do algoritmo branchand-cut utilizado pelo CPLEX na solução exata de um conjunto de instâncias destes problemas, além de apresentar resultados teóricos que estabelecem uma relação teórica entre o PRR e o PRRG;

Capítulo 5: Abordagens heurísticas propostas para o PRR e para o PRRG, apresenta as abordagens heurísticas propostas nesta tese para a solução aproximada dos problemas em estudo, destacando-se heurísticas de construção, busca local e versões da meta-heurística GRASP e da meta-heurística *Iterated Local Search*. Testes estatísticos são utilizados para avaliar todas as abordagens heurísticas apresentadas;

Capítulo 6: Conclusões, destaca as conclusões e propostas de trabalhos futuros desta pesquisa.

É importante ressaltar que, como não é conhecida nenhuma biblioteca pública de instâncias para os problemas abordados nesta tese, o Apêndice A detalha a biblioteca de instâncias proposta para o PRR e para o PRRG, que agora encontra-se disponível no *site* http://labic.ic.uff.br/. Os demais apêndices reúnem os resultados computacionais de forma mais detalhada para apreciação dos leitores e fazem uma breve revisão dos testes estatísticos utilizados neste trabalho para avaliar as heurísticas propostas (Apêndice B: *Experimentos computacionais com as regras de redução propostas*; Apêndice C: *Resultados complementares dos experimentos com o CPLEX*; Apêndice D: *Testes de inferência estatística*; Apêndice E: *Resultados complementares dos experimentos com as heurísticas*).

Capítulo 2

O Problema de Recobrimento por Rotas e Recobrimento por Rotas Generalizado

Este capítulo apresenta a definição do Problema de Recobrimento por Rotas e do Problema de Recobrimento por Rotas Generalizado (Seção 2.1), um levantamento dos métodos e abordagens computacionais existentes na literatura para ambos problemas (Seção 2.2), assim como alguns problemas correlatos e aplicações (Seções 2.3 e 2.4).

2.1 Definição dos problemas

O Problema de Recobrimento por Rotas, referido como *Covering Tour Problem* (CTP), é uma generalização do Problema do Caixeiro Viajante definido formalmente em 1981 por John Current (CURRENT, 1981).

O PRR pode ser definido na estrutura de um grafo $G = (V \cup W, E)$, completo e não direcionado, onde $V \cup W = \{v_0, ..., v_k\}$ é o conjunto dos vértices e $E = \{(v_i, v_j) | v_i, v_j \in V \cup W, i < j\}$ é o conjunto das arestas. O conjunto dos vértices é particionado em dois conjuntos disjuntos $(V \cap W = \emptyset)$, sendo V o conjunto dos vértices que **podem ser visitados**¹ e W o conjunto dos vértices que **devem ser cobertos**, mas não fazem parte da solução do problema. Existe ainda um conjunto T, com $T \subseteq V$, chamado conjunto dos vértices obrigatórios, que contém todos os vértices que **precisam ser visitados**. Um vértice $v_i \in W$ é considerado coberto se a distância entre v_i e pelo menos um vértice v_j pertencente à rota² solução do problema, for menor ou igual a uma distância $d \ge 0$

¹ Um vértice $v_i \in V$ é considerado visitado se v_i pertence à rota solução do problema.

² No contexto deste trabalho, uma rota é um caminho onde todos os vértices são distintos, com exceção do primeiro e do último. Portanto, toda rota viável para o PRR é também um ciclo.

previamente estabelecida. Para o PRR abordado neste trabalho, está sendo considerada a métrica euclideana. Desta forma, tanto a distância c_{ij} entre os vértices $v_i \in v_j$ ($\forall v_i, v_j \in E | v_i \neq v_j$) quanto a distância de cobertura *d* são definidas segundo esta métrica. Vale ressaltar que, apesar deste trabalho adotar a métrica euclideana, qualquer outra métrica pode ser utilizada.

O objetivo do PRR consiste em determinar uma rota ou um ciclo de comprimento mínimo sobre um subconjunto de V, que deve conter todos os vértices do subconjunto $T \subseteq V$ e deve cobrir cada vértice do conjunto W. Sem perda de generalidade, o vértice v_0 é considerado o vértice origem ($v_0 \in T$) e a rota deve partir do vértice v_0 e, no final, retornar a ele, não necessariamente passando por todos os vértices de V. O comprimento da rota, também chamado de custo da rota, é calculado somando-se o custo associado a cada aresta que a compõe. A matriz de custos definida sobre o conjunto de arestas E é simétrica e satisfaz a desigualdade triangular. O custo de uma aresta (v_i, v_j), com $i \neq j$, é calculado tomando a distância euclideana entre os vértices $v_i \in v_j$.

A Figura 2.1 ilustra uma instância do PRR com 23 vértices. Os vértices do conjunto Tsão representados por quadrados na cor azul, os vértices do conjunto W são representados por losangos na cor vermelha e os vértices do conjunto $V \setminus T$ por círculos na cor preta. As circunferências de raio d e centro nos vértices de W delimitam a área de cobertura de cada respectivo vértice de W.



Figura 2.1: Exemplo de uma instância do PRR com 23 vértices

Observando a Figura 2.1 é fácil notar que, para cada $j \in W$, qualquer vértice $v_i \in V$, tal que $c_{ij} \leq d$, pode ser utilizado na rota para cobrir j. As arestas do grafo não são ilustradas nesta figura para facilitar a visualização.

O PRR é classificado como um problema NP-difícil (CURRENT, 1981), pois se reduz ao Problema do Caixeiro Viajante quando $d = 0, W = \emptyset$ e V é somente composto por vértices obrigatórios (V = T). Portanto, a tendência é que o uso exclusivo de métodos exatos fique limitado a solução de instâncias pequenas, fato que tem incentivado a utilização de métodos heurísticos, assim como propõe este trabalho no Capítulo 5. A Figura 2.2 mostra uma solução do PRR associada à instância apresentada na Figura 2.1.



Figura 2.2: Exemplo de uma solução viável para o PRR associada ao grafo da Figura 2.1

Com o intuito de abranger um número maior de aplicações reais, Motta (MOTTA, 2001) propôs uma variante do PRR denominada Problema de Recobrimento por Rotas Generalizado . A diferença entre o problema original (PRR) e o problema generalizado (PRRG) reside no fato de que neste último é permitido que os vértices de W façam parte da solução do problema cobrindo a si próprio e a outros vértices deste conjunto, quando for o caso. Portanto, o objetivo do PRRG consiste em determinar uma rota de comprimento mínimo sobre um subconjunto de $V \cup W$ (enquanto o PRR considera somente o conjunto V), que contenha todos os vértices do subconjunto $T \subseteq V$ e cubra todos os vértices do conjunto W.

A Figura 2.3 ilustra uma solução viável para o PRRG (rota na cor verde) sobreposta



à solução do PRR que foi apresentada na Figura 2.2 (rota na cor laranja).

Figura 2.3: Diferença entre uma solução viável para o PRR e uma para o PRRG associadas ao grafo da Figura 2.1

É importante ressaltar que toda solução viável para o PRR é também viável para o PRRG. Porém, uma solução viável para o PRRG não necessariamente é viável para o PRR, assim como pode ser observado nas rotas representadas na Figura 2.3: a rota na cor verde, que é viável para o PRRG, contém o vértice $11 \in W$, responsável por cobrir a si próprio e também o vértice $9 \in W$. Como esta rota contém um vértice de W, não é viável para o PRR. Por outro lado, a rota representada na cor laranja, contém somente vértices de V e conta com o vértice $17 \in V \setminus T$ para cobrir os vértices 11 e $9 \in W$. Por conseguinte, a rota representada em laranja é viável tanto para o PRR quanto para o PRRG.

2.2 Trabalhos relacionados na literatura

A primeira contribuição relacionada ao PRR após sua definição ocorreu em 1989 quando Current e Schilling apresentaram o *Covering Salesman Problem* (CSP) (CUR-RENT; SCHILLING, 1989). O CSP é considerado um problema similar ao PRR e pode ser definido na estrutura de um grafo direcionado G = (V, A), com um custo c_{ij} não negativo associado a cada arco $(i, j) \in A$. O objetivo do CSP consiste em obter uma rota de custo mínimo sobre um subconjunto de V, de modo que todos os vértices fora da rota estejam, no máximo, a uma distância de cobertura d pré-determinada de algum dos vértices da rota $(d \ge 0)$. O referido trabalho formulou o CSP como um problema de programação linear inteira (0 - 1) e propôs uma heurística para resolvê-lo de forma aproximada, chamada COVTOUR, que baseava-se na combinação de procedimentos desenvolvidos originalmente para o *Set Covering Problem* (SCP) e para o PCV. Para demonstrar o modelo e a heurística propostos, foi utilizado como exemplo um grafo definido para o CSP em (CURRENT; REVELLE; COHON, 1988), com 21 vértices, trinta e nove arcos bidirecionados e com uma distância de cobertura d = 90. A Figura 2.4 apresenta a solução obtida pela heurística COVTOUR para o referido grafo. Note que todos os vértices que não se encontram na rota estão cobertos por pelo menos um vértice que fazem parte dela.



Figura 2.4: Solução obtida para uma instância do CSP pela heurística COVTOUR Fonte: (CURRENT; REVELLE; COHON, 1988)

Em 1994, Current, Hasan e Holland propuseram duas abordagens para resolver um outro problema similar ao PRR, chamado originalmente de Shortest Covering Path Problem (SCPP) (CURRENT; HASAN; ROLLAND, 1994). O SCPP pode ser definido em um grafo completo e direcionado G = (V, A), onde V é um conjunto de n vértices e A é um conjunto de m arcos (i, j). O problema consiste em identificar um caminho de custo mínimo, partindo de um vértice origem s até um vértice destino $t \ (s \neq t)$, ambos previamente estabelecidos, de forma que o caminho cubra todos os vértices do grafo. Um vértice é considerado coberto se distanciar, no máximo, de um valor $d \ge 0$ de algum vértice do caminho (d é previamente estabelecido). A primeira abordagem proposta em (CURRENT; HASAN; ROLLAND, 1994) para o SCPP foi uma heurística baseada em uma relaxação lagrangeana do problema. A segunda abordagem consistia em um algoritmo exato baseado no método branch-and-cut, vinculado a um procedimento que utilizava os limites gerados pela heurística lagrangeana (1º algoritmo proposto) para gerar um conjunto de soluções para duas versões distintas do problema: a primeira consistia em minimizar somente o comprimento da rota, enquanto a segunda maximizava o número de vértices cobertos pela rota obtida na versão anterior. Os procedimentos propostos foram testados para 135 instâncias do SCPP.

Três anos mais tarde, Gendreau, Laporte e Semet (GENDREAU; LAPORTE; SEMET, 1997) apresentaram e analisaram a primeira formulação matemática para o PRR, investigaram as propriedades poliédricas de várias restrições, propuseram um algoritmo exato baseado no método *branch-and-cut* e uma heurística para fornecer um limite superior inicial para o algoritmo exato. Esta heurística combina a heurística GENIUS, proposta originalmente para o PCV (GENDREAU; HERTZ; LAPORTE, 1992), com o algoritmo PRI-MAL1, proposto para o SCP (BALAS; HO, 1980). A formulação matemática proposta no referido trabalho será detalhada mais adiante no Capítulo 4.

Um conjunto composto de quatro regras de redução para os grafos associados ao PRR também foi proposto em (GENDREAU; LAPORTE; SEMET, 1997). O Capítulo 3 desta tese aborda em detalhes estas quatro regras.

As propostas apresentadas em (GENDREAU; LAPORTE; SEMET, 1997) foram testadas para 75 instâncias do PRR geradas da seguinte forma: |V| + |W| pares ordenados (x, y)foram gerados aleatoriamente em um retângulo de dimensão $[0, 100] \times [0, 100]$, seguindo uma distribuição uniforme. O conjunto de vértices obrigatórios (T) foi definido considerando os |T| primeiros pontos gerados. Os conjuntos de vértices opcionais $(V \setminus T)$ e vértices a serem cobertos (W) foram definidos considerando os $|V \setminus T|$ e |W| pontos remanescentes, respectivamente. O custo (c_{ij}) entre cada par de vértices do grafo foi definido computando-se a distância euclideana entre os pontos (x_i, y_i) e (x_j, y_j) . A distância de cobertura d foi calculada da seguinte forma:

$$d = \max\left\{\max_{v_k \in V \setminus T} \left(\min_{v_l \in W} \left(c_{ik}\right), \max_{v_l \in W} \left(c_{lk(l)}\right)\right)\right\}$$
(2.1)

onde k(l) é o índice do vértice de $V \setminus T$ que é o segundo mais próximo de $v_j \in W$. Da forma como os autores definiram a distância de cobertura d, foi assegurado que cada vértice de $V \setminus T$ cobriria pelo menos um vértice de W e que cada vértice de W seria coberto por pelo menos dois vértices de $V \setminus T$. A cardinalidade total do conjunto de vértices das 75 instâncias variou de 100 a 600, sendo $|V| = \{50, 75, 100\}, |T| = \{1, 13, 19, 25, 38, 50, 56, 75\}$ $|W| = \{50, 75, 100, 150, 200, 225, 250, 300, 375, 400, 450, 500\}.$

Em 1999, Maniezzo, Baldacci, Boschetti e Zamboni (MANIEZZO et al., 1999) apresentaram algumas abordagens para o PRR, das quais pode-se relacionar: três algoritmos heurísticos baseados na meta-heurística *Scatter Search* (SS), um novo modelo de programação matemática adaptado a partir da formulação conhecida na literatura por *Two-Commodity*, proposta originalmente para o PCV (FINKE; CLAUS; GUNN, 1984), além de um novo conjunto de regras de redução para os grafos associados. Adicionalmente, os autores formularam o PRR como o PCV generalizado, referido na literatura por *Generalized Traveling Salesman Problem* (GTSP) (FISCHETTI; SALAZAR; TOTH, 1995).

A meta-heurística Scatter Search (GLOVER, 1995) opera sobre um conjunto de soluções, chamado Conjunto Referência, que armazena as melhores soluções encontradas durante o processo de busca. Operadores de recombinação são aplicados entre as soluções do Conjunto Referência gerando novas soluções, nas quais é aplicado um processo de busca local. O primeiro algoritmo baseado na meta-heurística Scatter Search proposto em (MANIEZZO et al., 1999) para o PRR, chamado SS-CTP1, inicializa o Conjunto Referência com o procedimento de diversificação proposto em (GLOVER, 1998) e utiliza cortes para obter as desigualdades válidas para inicializar cada laço principal do algoritmo. O segundo algoritmo, SS-CTP2, difere do primeiro por utilizar a técnica star paths (GLOVER, 1995) para gerar as soluções. O terceiro algoritmo, SS-CTPB, baseado no algoritmo SS1 proposto em (GLOVER, 1998), difere dos dois primeiros algoritmos por não utilizar a técnica de cortes.

O modelo de programação matemática proposto para o PRR em (MANIEZZO et al., 1999) descreve-o como um problema de programação linear inteira, associando duas variáveis de fluxo x_{ij} e x_{ji} a cada aresta (i, j) do grafo. Mais detalhes sobre esta formulação serão fornecidos no Capítulo 4.

O conjunto de regras de redução proposto em (MANIEZZO et al., 1999) para os grafos associados ao PRR era composto por três das quatro regras apresentadas em (GENDREAU; LAPORTE; SEMET, 1997), mais uma nova regra. Todas estas regras serão abordadas no Capítulo 3.

O modelo de programação linear inteira, as três heurísticas e o conjunto de regras de redução propostos em (MANIEZZO et al., 1999) foram testados para um conjunto de 75 instâncias geradas como sugerido em (GENDREAU; LAPORTE; SEMET, 1997). Entretanto, os custos de cada aresta $(i, j) \in E$ foram computados como valores inteiros iguais a $\lfloor e_{ij} + 0.5 \rfloor$, onde e_{ij} representava a distância euclidiana entre os vértices $v_i \in v_j$. Em 2001, Motta (MOTTA, 2001) apresentou uma nova formulação matemática para o PRR baseada em variáveis de fluxo e analisou o conjunto de regras de redução proposto em (MANIEZZO et al., 1999). Esta análise mostrou que duas das quatro regras deste conjunto possibilitavam a obtenção de soluções viáveis para o grafo reduzido, porém inviáveis para o grafo original (vide Seção 3.1.1). Adicionalmente, foi apesentado e formulado o Problema de Recobrimento por Rotas Generalizado , para o qual foi proposto um conjunto de regras para reduzir o grafo associado, três heurísticas de construção, três heurísticas de busca local, inclusive uma baseada na meta-heurística VNS (Variable Neighborhood Search) (MLADENOVIĆ, 1995), além de oito versões básicas da meta-heurística GRASP (Greedy Randomized Adaptive Search Procedure) que combinavam as heurísticas de construção e busca local propostas.

O modelo matemático proposto para o PRR e para o PRRG em (MOTTA, 2001) associava uma única variável de fluxo a cada aresta $(i, j) \in E$ para evitar a formação de ciclos desconexos da origem. Mais detalhes acerca destas formulações matemáticas serão vistos no Capítulo 4.

O conjunto de regras proposto em (MOTTA, 2001) para reduzir o grafo associado ao PRRG era composto de duas regras, sendo uma nova regra e uma proposta em (MANIEZZO et al., 1999) para o PRR. Uma abordagem detalhada deste conjunto de regras será vista no Capítulo 3.

As propostas apresentadas em (MOTTA, 2001) para o PRRG foram testadas para 52 instâncias obtidas da seguinte forma: os |V| + |W| pares ordenados (x, y) foram gerados aleatoriamente em um retângulo de dimensão $[0, 560] \times [0, 400]$, seguindo uma distribuição uniforme. Os conjuntos de vértices T, $W \in V \setminus T$ foram definidos considerando |T|primeiros pontos gerados, seguidos dos |W| e dos $|V \setminus T|$ pontos remanescentes, respectivamente. O custo c_{ij} entre cada par de vértices do grafo foi definido computando-se a distância euclideana entre os pontos $(x_i, y_i) \in (x_j, y_j)$. A distância de cobertura d foi calculada de forma a garantir que todo vértice $v_i \in W$ fosse coberto por pelo menos um vértice $v_j \in V \cup W$, já que uma solução viável do PRRG pode conter vértices de W. Experimentos computacionais foram apresentados para mostrar o impacto da utilização do conjunto de regras de redução na solução do PRRG, tanto na utilização de métodos exatos (com grafos de 10 a 25 vértices), quanto na utilização de métodos heurísticos (com grafos de 30 a 300 vértices).

Considerado uma variante do PRR, o Problema de Recobrimento por Rotas com Coleta de Prêmios (PRR-CP) foi proposto por Lyra (LYRA, 2004) e pode ser definido na estrutura de um grafo simétrico $G = (V \cup W, E)$, completo e não direcionado, onde $V \cup W$ e E são respectivamente o conjunto de vértices e arestas, sendo definidos da mesma forma que o PRR. A cada vértice i de V está associado um prêmio não-negativo ω_i . Um prêmio mínimo pré-definido, dado por ω_{min} , deve ser coletado. O PRR-CP consiste em determinar uma rota de comprimento mínimo sobre um subconjunto de V, que contenha todos os vértices obrigatórios (T), que cubra todos os vértices de W e que colete a quantidade mínima de prêmios (ω_{min}) .

As seguintes contribuições foram apresentadas em (LYRA, 2004) para PRR-CP: uma formulação matemática baseada na proposta em (MOTTA, 2001) para o PRR, um conjunto de regras para a redução do grafo associado, duas versões da meta-heurística GRASP obtidas a partir de uma heurística construtiva e duas heurísticas de busca local, além de um módulo de Reconexão por Caminhos (RC) que utilizava as soluções produzidas pelas iterações GRASP como solução base e as soluções do conjunto elite como solução alvo. Neste módulo não eram geradas soluções inviáveis, pois a cada passo no processo da RC, um vértice $v_i \in V \setminus T$ da solução alvo era introduzido na solução base e, somente se não fosse obtida uma solução inviável, um vértice $v_j \in V \setminus T$ da solução guia era removido. Experimentos computacionais foram realizados com diferentes simulações efetuadas, considerando instâncias de 10 a 500 vértices geradas aleatoriamente em um plano cartesiano 100×100 . Os testes com a formulação foram realizados somente com as instâncias de pequenas dimensões ($|V \cup W| = \{10, 15, 20, 30, 40\}$) devido as limitações do ambiente computacional. Para as heurísticas, foram consideradas instâncias de maiores dimensões, sendo $|V \cup W| = \{30, 50, 100, 150, 200, 250, 300, 400, 500\}$.

Em 2005, Brito (BRITO, 2005) utilizou a técnica de relaxação lagrangeana (GEOF-FRION, 1974) e a formulação proposta para o PRR em (MOTTA, 2001) visando obter limites inferiores melhores dos que os obtidos mediante a utilização de uma relaxação linear (BAZARAA; JARVIS; SHERALI, 1990). Também foi apresentada uma heurística lagrangeana com o objetivo de *ajustar* as soluções obtidas a partir da relaxação proposta para produzir soluções viáveis que constituíssem limites superiores para o PRR. Uma nova regra para a redução do grafo também foi introduzida no referido trabalho. Para solucionar o PRR de forma aproximada, foram apresentados seis algoritmos baseados na meta-heurística GRASP, obtidos a partir da combinação dos seguintes procedimentos: adaptação da heurística Vizinho Mais Próximo do PCV como heurística construtiva; duas heurísticas de busca local, sendo uma baseada na troca simultânea de vértices $v_i \in V \setminus T$ da rota com vértices $v_j \in V \setminus T$ fora da rota e outra baseada na meta-heurística VNS; um módulo de RC e um procedimento de filtro para a fase de construção do GRASP. Experimentos computacionais com instâncias geradas aleatoriamente em um plano cartesiano 100×100 , com a distância de cobertura calculada de modo a garantir que todo vértice $v_i \in W$ fosse coberto por pelo menos um vértice $v_j \in V$, possibilitaram a obtenção de soluções viáveis para diversos problemas teste com até 500 vértices.

Kubik (KUBIK, 2007) dividiu o PRR em duas componentes, sendo uma equivalente ao PCV e outra ao SCP e combinou métodos de solução desenvolvidos para estes dois problemas com a finalidade de encontrar soluções de boa qualidade para o PRR. Uma abordagem semelhante já havia sido utilizada em (GENDREAU; LAPORTE; SEMET, 1997) e (CURRENT; SCHILLING, 1989), porém neste último para resolver de forma aproximada o CSP. A heurística proposta por Kubik, denominado CTACS (*Covering Tour Ant Colony System*), baseia-se na meta-heurística Colônia de Formigas (CF) (DORIGO, 1992) e é composta por dois outros algoritmos: o GACS (GENI Ant Colony System) e o SCACS (Set *Covering Ant Colony System*), utilizados para resolver o PCV e o SCP respectivamente. O algoritmo GACS combina a metodologia da CF com a fase GENI da heurística GENIUS para resolver o PCV e o algoritmo SCACS utiliza a combinação de duas abordagens também baseada na CF para resolver o SCP, sendo estas apresentadas em (LESSING; DUMITRESCU; STÜTZLE, 2004) e (MARCHIORI; STEENBEEK, 2000). O SCP é resolvido primeiro e os vértices da melhor solução obtida é utilizada para formular a componente do PCV, que é resolvida separadamente em seguida.

O algoritmo proposto em (KUBIK, 2007) foi testado para 45 instâncias aleatórias contendo 1000 vértices, sendo $|T| = \{3, 5, 10\}, |V| = \{103, 105, 110, 153, 155, 160, 203, 205, 210\}$ e |W| = 1000 - |V|, mais 45 instâncias de 2000 vértices, com $|T| = \{3, 5, 10\}, |V| = \{63, 65, 70, 83, 85, 90, 103, 105, 110\}$ e |W| = 2000 - |V|. A distância de cobertura $d \ge 0$ foi estabelecida como sendo o valor mínimo para que cada instância se tornasse viável para o PRR, ou seja, d foi estabelecida de modo que todo vértice em W fosse coberto por pelo menos um vértice em V. Resultados computacionais mostram que a heurística CTACS apresentou resultados melhores que a heurística proposta em (GENDREAU; LAPORTE; SE-MET, 1997) para aproximadamente 82% das instâncias testadas. Entretanto, a heurística proposta por Kubik é dependente de um número muito grande de parâmetros a serem calibrados, o que dificulta sua utilização e a torna muita específica para determinadas instâncias.

Em 2009, Silva (SILVA, 2009) propôs três novas regras de redução para o PRR-CP, uma nova formulação matemática com o número de variáveis e restrições da ordem de $O(n^3)$, um algoritmo evolutivo e cinco versões de um algoritmo baseado na meta-heurística Iterated Local Search (ILS). Todas as versões dos algoritmos baseados no ILS tinham o mecanismo de busca local baseado no Método Randômico de Descida (MRD). Para gerar soluções iniciais para esses algoritmos, foram propostos cinco procedimentos heurísticos, sendo três deles adaptações de heurísticas clássicas do PCV: GENIUS (GENDREAU; HERTZ; LAPORTE, 1992), Inserção Mais Barata (ROSENKRANTZ; STEARNS; LEWIS, 1977), Vizinho Mais Próximo (BELLMORE; NEMHAUSER, 1968). As propostas apresentadas em (SILVA, 2009) foram testadas para dois grupos de instâncias do PCV disponíveis no repositório TSPLIB (REINELT, 1991), que foram adaptadas para o PRR-CP, sendo o primeiro grupo envolvendo instâncias de até 200 vértices e o segundo de 201 a 400 vértices.

Não é de nosso conhecimento a existência de outro algoritmo, aproximado ou exato, proposto para o PRR ou para o PRRG na literatura afim.

2.3 Problemas correlatos

A presente seção aborda alguns problemas considerados similares ao PRR e ao PRRG, entre os quais estão os referidos na literatura por Traveling Purchaser Problem, Prize Collecting Traveling Salesman Problem, Selective Traveling Salesman Problem, Median Tour Problem, Maximal Covering Tour Problem, Geometric Covering Salesman Problem, Multi-vehicle Covering Tour Problem e Bi-objective Covering Tour Problem.

O Traveling Purchaser Problem (TPP) (RAMESH, 1981) é uma generalização do Problema do Caixeiro Viajante, onde um comprador deve planejar uma rota sobre os vértices de um grafo G de modo que, visitando os mercados distribuídos pelas cidades (vértices do grafo), possa adquirir de forma mais barata possível (minimizando os custos de compra e de percurso), um conjunto de itens que são necessários ao comprador e oferecidos com diferentes custos e quantidades nestas cidades. Considerando um domicílio v_0 , um conjunto de mercados $M = \{v_1, v_2, ..., v_m\}$ e um conjunto de produtos $K = \{f_1, f_2, ..., f_n\}$, o TPP pode ser representado em um grafo G = (V, E) simples e não direcionado, onde $V = \{v_0\} \cup M$ é o conjunto de vértices e $E = \{(v_i, v_j) : v_i, v_j \in V, i < j\}$ é o conjunto de arestas. Cada produto f_k está associado a uma demanda d_k e está disponível em um subconjunto de mercados $M_k \subseteq M$, sendo b_{ki} o preço do produto f_k no mercado v_i e c_{ij} o custo da viagem entre $v_i \in v_j$. O TPP consiste em determinar uma rota sobre um subconjunto de vértices de G, iniciando e terminando em v_0 , de modo que a partir desta rota seja possível adquirir todos os produtos f_k necessários ao atendimento das demandas d_k , minimizando-se o custo de aquisição dos produtos somado ao custo do deslocamento
no grafo.

O Prize Collecting Traveling Salesman Problem (PCTSP) (BALAS, 1989) pode ser visto como um problema similar ao PRR, pois sua solução consiste na obtenção de uma rota sobre um subconjunto dos vértices do grafo, de forma que esta rota colete um prêmio mínimo ω pré-estabelecido. Cada vértice v_i possui um prêmio $\omega_i \geq 0$ e uma penalidade $p_i \geq 0$ associados. Um caixeiro coleta um prêmio ω_k a cada cidade v_k visitada, paga uma penalidade p_j a cada cidade v_j que deixa de visitar e gasta c_{ij} no deslocamento da cidade v_i até a cidade v_j . Assim, o PCTSP deseja minimizar o comprimento da rota adicionado às penalidades pagas pelas cidades não visitadas, de modo que o total de prêmios coletados por esta rota atenda ao prêmio mínimo ω . O Selective Traveling Salesman Problem (STSP) (LAPORTE; MARTELLO, 1990) pode ser visto como uma variação do PCTSP, cujo objetivo é construir uma rota sobre um subconjunto dos vértices do grafo, de modo que o total de prêmios coletados seja maximizado, desde que o comprimento desta rota não ultrapasse um valor pré-determinado $q \geq 0$.

O Median Tour Problem (MTP) e o Maximal Covering Tour Problem (MCTP) foram introduzidos em 1994 (CURRENT; SCHILLING, 1994) e podem ser definidos em um grafo G = (V, E), completo e não direcionado. O MTP tem como objetivo construir uma rota de comprimento mínimo que contenha p dos n vértices do grafo ($p \in \aleph^*$), minimizando o custo total da distância de acesso à rota por parte dos vértices que não fazem parte dela. O MCTP é um caso particular do MTP e consiste em construir uma rota de comprimento mínimo que contenha p dos n vértices do grafo ($p \in \aleph^*$), assim como o MTP, porém visa adicionalmente minimizar a demanda total dos vértices não cobertos pela rota. Um vértice é considerado coberto pela rota quando a distância deste a qualquer um dos vértices da rota for menor ou igual a uma distância máxima de acesso previamente estabelecida. Tanto o MTP quanto o MCTP são classificados pelos autores como problemas da classe NP-difícil.

O Geometric Covering Salesman Problem (GCSP) (ARKIN; HASSIN, 1994) é uma variação do PRR em que os vértices do grafo não são um conjunto discreto, mas uma região de um plano.

O Multi-vehicle Covering Tour Problem (m-CTP) (HACHICHA; HODGSON; LAPORTE, 2000) pode ser definido na estrutura de um grafo $G = (V \cup W, E)$, completo e não direcionado, onde o conjunto de vértices V, assim como no PRR, é particionado em vértices que podem ser visitados $(V \setminus T)$ e em vértices que devem ser visitados $(T \subseteq V)$. O conjunto W contém os vértices que devem ser cobertos por uma distância $d \ge 0$ previamente estabelecida. O objetivo do m-CTP é determinar um conjunto de $m \in \aleph^*$ rotas de comprimento mínimo, de modo que as rotas satisfaçam as seguintes condições:

- 1. Existem no máximo m rotas de comprimento mínimo, todas partindo e retornando ao depósito s ($s \in T$ é previamente estabelecido);
- 2. Cada vértice de $V \setminus T$ pertence, no máximo, a uma rota e cada vértice de $T \subseteq V$ pertence a exatamente uma única rota;
- 3. Cada vértice de W deve estar coberto por pelo menos uma rota, isto é, deve estar a uma distância máxima $d \ge 0$ de pelo menos um vértice pertencente a uma das mrotas (é assumido que o depósito s não cobre nenhum $i \in W$);
- 4. O número de vértices de cada rota (excluindo o depósito) não deve exceder a um valor fixo $p \in \aleph^*$ previamente estabelecido;
- 5. O comprimento de cada rota não deve exceder um valor fixo $q \in \Re^+$ previamente estabelecido.

O m-CTP se reduz ao PRR quando o número máximo de rotas for igual a um (m = 1), quando o valor q que determina o comprimento máximo de cada rota for grande o suficiente para que, no pior caso, a rota solução do PRR contenha todos os vértices de V (no caso do PRRG, q deve ser grande o suficiente para conter todos os vértices de $V \cup W$).

O Bi-objective Covering Tour Problem (JOZEFOWIEZ; SEMET; TALBI, 2007) é um problema multi-objetivo definido na estrutura de um grafo simétrico $G = (V \cup W, E)$, completo e não direcionado, que consiste em determinar uma rota sobre um subconjunto de V, que contenha todos os vértices de $T \subseteq V$ e que, ao mesmo tempo, minimize o comprimento da rota (objetivo 1) e a cobertura desta (objetivo 2). A cobertura de uma solução é definida como sendo a maior distância entre um vértice $v_i \in W$ e o vértice $v_j \in V$ pertencente à rota mais próxima de v_i .

2.4 Aplicações

Assim como mencionado no Capítulo 1, muitas aplicações reais podem ser associadas ao PRR/PRRG, entre as quais esta seção destaca o roteamento de despachos aéreos, o planejamento de serviços médicos em regiões de difícil acesso, a localização de postos de coletas dos correios, a localização de facilidades, o planejamento do patrulhamento urbano preventivo e o planejamento das rotas de veículos de transporte público de alunos.

No problema de roteamento de despachos aéreos, consideram-se os vértices do caminho como sendo as cidades que devem ser servidas por aeronaves. Os vértices cobertos por este caminho, mas não necessariamente contidos nele, devem ser cobertos por um meio de transporte secundário, geralmente terrestre. Pode-se ainda considerar uma distância máxima de cobertura, que pode estar associada a uma distância ou um tempo máximo com a conexão ar-terra. Neste caso particular, a aplicação refere-se ao MCTP.

O problema de localização e roteamento de caixas de correios (LABBE; LAPORTE, 1986) considera um subconjunto de locais candidatos, onde todos os usuários devem estar localizados a uma distância razoável de alguma das caixas. O custo da rota através de todas as caixas deve ser minimizado para reduzir o gasto com despacho das correspondências. O problema de localização de facilidades é similar ao problema de localização e roteamento de caixas de correios, pois também considera locais candidatos para determinar quais destes são os melhores locais para instalação das facilidades, de forma que todos os clientes sejam atendidos a um custo mínimo. Neste caso, as facilidades podem ser vistas como postos de pronto-socorro, hospitais, escolas, postos de policiamento, entre outros. O custo pode estar relacionado à distância das casas dos usuários até a facilidade mais próxima a ser considerada.

No roteamento de equipes para a entrega de medicamentos em países subdesenvolvidos (OPPONG; HODGSON, 1994), os medicamentos somente podem ser entregues a um subconjunto de vilarejos. Todos os usuários dos demais vilarejos devem ser capazes de atingir pelo menos um vilarejo da rota, visando suprir a necessidade de serviços e medicamentos. Pode-se ainda associar este problema ao MTP e ao MCTP: quando o tempo médio de acesso é a restrição principal, o MTP pode ser considerado para determinar quais vilarejos devem-se visitar. Quando existir uma distância máxima a ser percorrida pelos usuários, pode-se considerar o MCTP.

O problema de planejamento do patrulhamento urbano preventivo (OLIVEIRA, 2008), cujo objetivo é a construção de rotas a serem percorridas pelos veículos da força policial, também pode ser facilmente associado ao PRR/PRRG. O grafo $G = (V \cup W, E)$ representa a região geográfica a ser patrulhada. Os pontos de visita obrigatória são associados aos vértices $v_i \in T$ que devem pertencer às rotas e os pontos de cobertura obrigatória são associados aos vértices $v_j \in W$ que devem estar suficientemente próximos das rotas. O objetivo final é construir rotas eficientes (com o menor custo possível), que comecem e terminem na base do serviço. A elaboração destas rotas tem como finalidade fornecer uma boa visibilidade para o patrulhamento, de modo a proporcionar uma sensação de segurança para a população, permitir o atendimento rápido em caso de ocorrências, evitar a prática de atos ilegais e fazer vigilância de estabelecimentos públicos como hospitais, escolas etc. O planejamento considera o número de veículos disponíveis para realizar uma distribuição equânime de trabalho.

No problema do planejamento das rotas de veículos de transporte público de alunos (LIMA; SILVA, 2004), os vértices obrigatórios ($v_i \in T$) podem ser vistos como as grandes escolas, onde o transporte público escolar deve passar. O mesmo vale para as escolas infantis, onde devem-se entregar/pegar os alunos na porta da escola. As áreas determinadas pelos vértices a serem cobertos ($v_j \in W$) podem ser vistas como as regiões de grande compactação de alunos, onde o transporte escolar deve passar por pelo menos um ponto para atender diversas pequenas escolas. Como os horários das atividades escolares nas unidades são variados, considera-se que um número pré-determinado de veículos circularão pela rota definida, em intervalos de tempo de forma a atender a demanda de alunos. Por este motivo, pode-se desconsiderar a capacidade do veículo.

Outros problemas associados ao PRR/PRRG são o problema de roteamento dos veículos de coleta de resíduos sólidos urbanos ((BRASILEIRO; LACERDA, 2008) e (DETOFENO, 2009)) e o problema de definição de rotas para enfermeiros *home-care* (COLIN, 2007).

Capítulo 3

Regras para a redução do grafo associado ao PRR e ao PRRG

Um dos maiores gargalos identificados na solução de muitos problemas de otimização combinatória está associado ao esforço computacional necessário para explorar um espaço de busca de forma eficiente, uma vez que para cada solução deste espaço, é preciso verificar um conjunto de restrições que determinam a viabilidade do problema que está sendo resolvido.

Para muitos problemas modelados como problemas de otimização, a exploração completa do espaço de busca é computacionalmente inviável, dada a grande quantidade de informações que precisam ser analisadas. Nos últimos anos, além do desenvolvimento de algoritmos heurísticos e exatos cada vez mais eficientes, muito tem-se investido na determinação de regras para reduzir o espaço de busca. Estas são comumente chamadas de regras de redução e atuam na redução do grafo de entrada de modo a não eliminar nenhuma solução ótima. Neste contexto, este capítulo descreve as regras de redução existentes na literatura para o PRR e para o PRRG (Seções 3.1.1 e 3.2.1) e apresenta como contribuições: novas regras para a redução dos grafos associados a estes problemas (Seções 3.1.2 e 3.2.2), um novo conjunto de regras de redução para ambos os problemas e uma demostração teórica que prova a corretude de cada uma das regras que compõem estes conjuntos (Seções 3.1.3 e 3.2.3).

Uma análise experimental da utilização dos novos conjuntos de regras de redução é realizada nas Seções 3.1.3 e 3.2.3. Nestes experimentos foram consideradas 72 instâncias para o PRR e 72 instâncias para o PRRG, variando de 15 a 1000 vértices no total (vide Apêndice A para mais informações sobre as instâncias utilizadas neste trabalho).

3.1 Regras de redução para o PRR

Esta seção descreve as seis regras de redução existentes na literatura para o PRR (Seção 3.1.1) e apresenta como contribuições uma nova regra para a redução do conjunto W, um novo conjunto de regras para a redução dos grafos associados ao PRR e a demonstração teórica da corretude de todas as regras que compõem este conjunto (Seção 3.1.2). Experimentos computacionais mostram o impacto da utilização do novo conjunto de regras na redução dos soluções viáveis de 72 instâncias do PRR (Seção 3.1.3).

3.1.1 Regras de redução existentes na literatura para o PRR

Pode-se se definir formalmente uma regra de redução \mathcal{R} para um grafo G = (V, E)associado ao problema \mathcal{P} , como sendo toda regra que pode ser aplicada sobre o conjunto de vértices V ou sobre conjunto de arestas E, de modo que as alterações promovidas nestes conjuntos não eliminem nenhuma solução ótima de \mathcal{P} em G e que todas as soluções viáveis para o grafo reduzido pela regra \mathcal{R} também sejam viáveis para o grafo original.

O primeiro conjunto de regras de redução formulado para os grafos associados ao PRR foi proposto em (GENDREAU; LAPORTE; SEMET, 1997) para reduzir iterativamente os conjuntos W e V. Uma matriz $\Delta = [\delta_{ij}]$ de dimensão $|W| \times |V \setminus T|$ foi definida para possibilitar a aplicação destas regras, sendo $\delta_{ij} = 1 \Leftrightarrow v_i \in W$ for coberto por $v_j \in V \setminus T$.

As quatro regras que compõem o conjunto proposto em (GENDREAU; LAPORTE; SE-MET, 1997) são:

- (**R1**) Se $\delta_{ij} = 1, \forall v_j \in V \setminus T$, então remova $v_i \in W$;
- (R2) Se existirem v_i, v_j, ..., v_l ∈ W e diferentes entre si, tal que δ_{ik} = δ_{jk} = ... = δ_{lk}, ∀v_k ∈ V\T, então considere apenas um dos vértices entre v_i, v_j, ..., v_l e remova os demais;
- (R3) Se existirem $v_i \neq v_j$, com $v_i, v_j \in W$, tal que $\delta_{ik} \leq \delta_{jk}$, $\forall v_k \in V \setminus T$, então remova $v_i \in W$;
- (R4) Se $\delta_{ij} = 0, \forall v_i \in W$, então remova $v_j \in V \setminus T$.

O segundo conjunto de regras de redução formulado para os grafos associados ao PRR foi proposto em (MANIEZZO et al., 1999)¹ e é composto pelas regras R1, R3, R4 propostas

¹ Este trabalho foi publicado anos mais tarde como capítulo de livro em (MANIEZZO et al., 2005).

em (GENDREAU; LAPORTE; SEMET, 1997) e uma nova regra, que pode ser enunciada da seguinte forma:

(R5) Se existir $v_j \in T$, tal que $\delta_{ij} = 1$, então remova $v_i \in W$;

O conjunto de regras proposto em (MANIEZZO et al., 1999) é aplicado iterativamente ao grafo associado ao PRR na seguinte sequência: {R1, R3, R5, R4}.

Uma análise da utilização de cada uma das quatro regras de redução que compõem o conjunto proposto em (MANIEZZO et al., 1999) foi realizada em (MOTTA, 2001). Nesta análise, foi constatado que a aplicação da regra descrita em R1 pode reduzir o grafo de modo que uma solução viável para o grafo reduzido seja inviável para o grafo original. Para exemplificar este fato, observe o grafo composto por 10 vértices apresentado na Figura 3.1A. Note que todos os vértices de $V \setminus T$ (6, 7, 8 e 9) cobrem todos os vértices de W (3, 4 e 5). Logo, de acordo com a regra R1, pode-se remover os vértices 3, 4 e $5 \in W$ (Figura 3.1B), resultando no grafo apresentado na Figura 3.1C. Como pode ser constatado na rota representada na cor verde (Figura 3.1D), a redução do grafo a partir da regra R1 permitiu que uma solução viável para o grafo reduzido, seja inviável para o grafo original, o que invalida a utilização desta regra tanto para a redução dos grafos associados ao PRR, quanto dos grafos associados ao PRRG, uma vez que toda solução viável do PRR é também viável para o PRRG. A rota representada na cor laranja ilustra uma solução viável do PRR tanto para o grafo original, quanto para o grafo reduzido, pois contém todos os vértices obrigatórios e cobre todos os vértices de W (o vértice $8 \in V \setminus T$ cobre os vértices 3, 4 e $5 \in W$).

A regra R3, também analisada em (MOTTA, 2001), remove do grafo original todo vértice $v_i \in W$, se existir um vértice $v_j \in W$ ($v_i \neq v_j$), tal que todo $v_k \in V \setminus T$ que cobre v_i também cobre v_j , mas nem todo $v_l \in V \setminus T$ que cobre v_j , necessariamente cobre v_i . Para exemplificar a utilização desta regra em um grafo associado ao PRR, observe a Figura 3.2.

Assim como identificado na análise da regra R1, a aplicação da regra R3 tanto em grafos associados ao PRR quanto ao PRRG, pode produzir grafos reduzidos que contenham soluções viáveis para ele e inviáveis para os grafos originais, fato que invalida o emprego da regra R3 na redução do espaço de soluções dos referidos problemas. Exemplificando tal situação, observe as vizinhanças dos vértices 2 e 3 \in W na Figura 3.2A. Note que todo vértice $v_k \in V \setminus T$ que cobre o vértice 2 (vértices 6 e 7), também cobre o vértice 3. Entretanto, nem todo vértice $v_t \in V \setminus T$ que cobre o vértice 3, também cobre o



Figura 3.1: Exemplo da utilização da regra de redução R1 em um grafo associado ao PRR

vértice 2 (vértices 4 e 9). Portanto, a redução deste grafo a partir da regra R3 resulta no grafo apresentado na Figura 3.2C, que tem como uma solução viável a rota representada em verde na Figura 3.2D. Observe que esta rota ilustra uma solução viável para o grafo reduzido que é inviável para grafo original, pois não contém nenhum vértice $v_k \in V \setminus T$ que cubra o vértice $2 \in W$. A rota representada em laranja na Figura 3.2D é viável para ambos os grafos, pois contém todos os vértices obrigatórios e cobre todos os vértices de W (o vértice $7 \in V \setminus T$ cobre os vértices $2 \in 3 \in W$, ao contrário do vértice $9 \in V \setminus T$ que cobre apenas o vértice $3 \in W$).

Em 2005, a partir da análise realizada por Motta (MOTTA, 2001), Brito (BRITO, 2005) apresentou uma nova regra, aqui chamada de R6, que associada às regras R4 e R5, formara o seguinte conjunto de regras de redução para o grafo associado ao PRR:



Figura 3.2: Exemplo da utilização da regra de redução R3 em um grafo associado ao PRR

- (**R4**) Se $\delta_{ij} = 0, \forall v_i \in W$, então remova $v_j \in V \setminus T$;
- (R5) Se existir $v_j \in T$, tal que $\delta_{ij} = 1$, então remova $v_i \in W$;
- (R6) Se para algum $v_i \in W$, $v_j \in V \setminus T$ é o único vértice tal que $\delta_{ij} = 1$, então transforme $v_j \in V \setminus T$ em um vértice do conjunto T.

A seguinte sequência foi sugerida em (BRITO, 2005) para a aplicação das três regras supra citadas: {R6, R5, R4}. Para exemplificar sua aplicação em um grafo associado ao PRR, considere o grafo ilustrado na Figura 2.1 (Capítulo 2). Observe que o vértice $17 \in V \setminus T$ é o único que cobre o vértice $9 \in W$, isto é, $\delta_{ij} = 1$, com $v_i = 9$ e $v_j = 17$. De acordo com a regra R6, o vértice 17 deve ser transformado em um vértice obrigatório (T), assim como apresenta a Figura 3.3.



Figura 3.3: Exemplo da aplicação da regra R6 no grafo da Figura 2.1

Aplicando-se a regra R5 no grafo resultante da regra R6 (Figura 3.3), obtém-se a remoção de cinco vértices (7, 8, 9, 10 e 11), assim como ilustra a Figura 3.4. Observe que a remoção dos vértices 9 e 11 só ocorreu porque o vértice $17 \in V \setminus T$ foi transformado em um vértice obrigatório (T) pela aplicação da regra R6.



Figura 3.4: Exemplo da aplicação da regra R5 no grafo da Figura 3.3

Com a aplicação da regra R4 no grafo resultante da aplicação consecutiva das regras R6 e R5 (Figura 3.4), obtém-se a remoção de mais seis vértices (13, 16, 18, 20, 21 e 23), assim como é mostrado na Figura 3.5.



Figura 3.5: Exemplo da aplicação da regra R4 no grafo da Figura 3.4

A Figura 3.6 apresenta o grafo resultante da aplicação iterativa do conjunto de regras de redução propostas em (BRITO, 2005), assim como uma solução viável do PRR para este grafo reduzido. Note que a rota ilustrada também é uma solução viável para o grafo original, fato que poderia não ocorrer se fossem utilizadas as regras R1 e R3, assim como mostrado no início desta seção.



Figura 3.6: Grafo da Figura 2.1 reduzido pela aplicação do conjunto de regras propostas para o PRR em (BRITO, 2005)

Pode-se observar no grafo da Figura 3.6 que após a aplicação do conjunto de regras proposto em (BRITO, 2005), obteve-se um percentual de redução de aproximadamente

44% sobre total de vértices do grafo original apresentado na Figura 2.1.

A regra R2 proposta em (GENDREAU; LAPORTE; SEMET, 1997) para o PRR não foi analisada em (MOTTA, 2001), assim como as regras R1, R3, R4 e R5. Desta forma, este trabalho apresenta uma análise da referida regra quando aplicada aos grafos associados ao PRR. Para tanto, considere o grafo ilustrado na Figura 3.7A (mesmo grafo utilizado na Figura 3.1 para exemplificar a utilização regra R1).



Figura 3.7: Exemplo da utilização da regra de redução R2 em um grafo associado ao PRR

De acordo com a regra R2, se existirem $v_i, v_j, ..., v_l \in W$ e diferentes entre si, tal que $\delta_{ik} = \delta_{jk} = ... = \delta_{lk}, \forall v_k \in V \setminus T$, então pode-se considerar apenas um dos vértices entre $v_i, v_j, ..., v_l$ e remover os demais. Assim, observando o grafo da Figura 3.7A, pode-se remover, por exemplo, os vértices 3 e 4, desde que o vértice 5 seja mantido (Figura 3.7B). O grafo resultante destas remoções pode ser visto na Figura 3.7C.

Analisando a Figura 3.7D, nota-se que a redução do grafo a partir da regra R2 permitiu que uma solução viável para o grafo reduzido, seja inviável para o grafo original (rota representada na cor verde), o que invalida a utilização desta regra para os grafos associados ao PRR. Entretanto, como toda solução viável do PRR é também viável para o PRRG, a utilização da regra R2 também fica invalidada para os grafos associados ao PRRG. Observe que a rota representada na cor laranja ilustra uma solução viável do PRR tanto para o grafo original, quanto para o grafo reduzido, pois contém todos os vértices obrigatórios e cobre todos os vértices de W (o vértice $8 \in V \setminus T$ cobre os vértices 3, 4 e $5 \in W$, como pode ser constatado na Figura 3.7A).

3.1.2 Nova regra de redução proposta para o PRR

Como descrito na Seção 3.1.1, em (MOTTA, 2001) foi provado através de contraexemplos que, das quatro regras que compõem o conjunto proposto para o PRR em (MANIEZZO et al., 1999) para reduzir iterativamente os conjuntos $W \in V$, duas (R1 e R3) podem resultar em grafos reduzidos que contenham soluções viáveis para estes e inviáveis para os grafos originais, invalidando o emprego das referidas regras na redução do espaço de soluções deste problema.

Partindo da análise da regra R1, é proposto neste trabalho uma nova regra R7 que reduz iterativamente o conjunto W da seguinte forma:

(R7) Se existirem $v_i, v_j \in W$, com $v_i \neq v_j$, tal que $\delta_{ik} \geq \delta_{jk}$, $\forall v_k \in V$, então remova $v_i \in W$.

Em outras palavras, a regra R7 remove do grafo o vértice $v_i \in W$, se existir $v_j \in W$, com $v_i \neq v_j$ e se a vizinhança de v_j estiver inteiramente contida na vizinhança de v_i . Para ilustrar a aplicação desta regra, considere os três grafos apresentados na Figura 3.8.

Observando o grafo 1 da Figura 3.8, pode-se perceber que a vizinhança do vértice 2 está inteiramente contida na vizinhança do vértice 3, fato que pela regra R7 implica na remoção do vértice 3. Note que toda solução viável para o grafo reduzido deve conter o vértice 6 e/ou o vértice 8, além dos vértices obrigatórios. Sendo assim, toda rota viável para o grafo resultante da regra R7 também será viável para o grafo original, pois tanto o vértice 6 quanto o vértice 8 cobrem o vértice 3 que foi removido pela referida regra.

No grafo 2, observa-se facilmente que a vizinhança do vértice 2 está inteiramente contida tanto na vizinhança do vértice 3 quanto na do vértice 4 e portanto, pela regra R7, pode-se remover os vértices 3 e 4. Analogamente às considerações feitas para o grafo 1 reduzido, toda rota viável para o grafo 2 reduzido deve conter todos os vértices obrigatórios e o vértice 10, pois este é o único vértice que cobre o vértice 4, tornando todas estas soluções também viáveis para o grafo original, já que o vértice 10 cobre os dois vértices removidos por R7.



Figura 3.8: Exemplos da aplicação da nova regra R7 em três grafos associados ao PRR

Analisando o grafo 3 da Figura 3.8, nota-se que as vizinhanças dos vértices 3 e 5

estão inteiramente contidas na vizinhança do vértice 4. Logo, pela regra R7, remove-se o vértice 4. Considerando o grafo 3 reduzido, observa-se que qualquer rota viável para este grafo deve conter todos os vértices obrigatórios, pelo menos um vértice da vizinhança do vértice 3 e pelo menos um da vizinhança do vértice 5, o que consequentemente garantirá a cobertura do vértice 4 que foi removido.

No presente trabalho é proposto adicionar a nova regra R7 às regras R4, R5 e R6, formando o seguinte conjunto de regras para reduzir os grafos associados ao PRR:

- (**R4**) Se $\delta_{ij} = 0, \forall v_i \in W$, então remova $v_j \in V \setminus T$;
- (R5) Se existir $v_j \in T$, tal que $\delta_{ij} = 1$, então remova $v_i \in W$;
- (R6) Se para algum $v_i \in W$, $v_j \in V \setminus T$ for o único vértice tal que $\delta_{ij} = 1$, então transforme $v_j \in V \setminus T$ em um vértice do conjunto T.
- (R7) Se existirem $v_i, v_j \in W$, com $v_i \neq v_j$, tal que $\delta_{ik} \geq \delta_{jk}$, $\forall v_k \in V$, então remova $v_i \in W$.

Sendo $G = (V \cup W, E)$ um grafo associado ao PRR, é correto afirmar que G pode ser reduzido a um grafo $G' = (V' \cup W', E')$, sendo $V' \subseteq V$, $T' \subseteq T$, $W' \subseteq W$ e $E' \subseteq E$, se toda solução viável em G' também for viável em G e se a solução ótima, denotada por r_{opt} , estiver em G'. Neste caso, pode-se dizer que G' é uma redução de G.

Denotando-se por S_G o conjunto de todas as soluções viáveis associadas a um grafo G, pode-se afirmar que G se reduz a G' se as seguintes condições forem verificadas:

Condição 1: Toda solução viável em G' também for viável em G; Condição 2: $r_{opt} \in S_{G'}$.

Vale ressaltar que, por definição, os vértices de W não fazem parte de nenhuma solução viável do PRR. Assim sendo, para este problema, tanto S_G quanto $S_{G'}$ não possuem soluções que contenham vértices de W.

A seguir, é apresentado como uma das contribuições deste trabalho, a demostração teórica de que as condições 1 e 2 definidas anteriormente são verificadas quando o conjunto de regras proposto nesta seção é aplicado a qualquer grafo G associado ao *PRR*. Para facilitar as demostrações, considere $G\beta'$ como sendo uma redução de G obtida a partir da aplicação da regra R β . Por exemplo, se a regra R4 for aplicada ao grafo G, então G4' é uma redução de G obtida a partir da aplicação da regra R4.

Afirmação 1: G4' é uma redução de G (PRR).

Por hipótese, o conjunto de vértices de G4' é formado por todos os vértices de T e pelos vértices $v_j \in V \setminus T$, onde $\delta_{ij} = 1$ para pelo menos um $v_i \in W$, isto é, T' = T, W' = We $V' \setminus T \subseteq V \setminus T$. Se $s' \in S_{G4'}$, então s' contém todos os vértices de T e cobre todos os vértices de W, portanto $s' \in S_G$, provando a condição 1.

Por definição, se r_{opt} é uma solução ótima do PRR em G, então r_{opt} contém todos os vértices de T e cobre todos os vértices de W, além de não existir nenhuma outra solução $s \in S_G$ tal que $custo(s) < custo(r_{opt})$. Suponha por absurdo que $r_{opt} \notin S_{G4'}$. Como por hipótese T' = T e W' = W, então existe pelo menos um $v_j \in V \setminus V'$, tal que $v_j \in r_{opt}$. Como todos os vértices em $V \setminus V'$ não cobrem nenhum $v_k \in W$, então conclui-se que $v_j \notin r_{opt}$, que é um absurdo, pois r_{opt} é ótima.

Afirmação 2: G5' é uma redução de G.

De fato a condição 1 se verifica, pois o vértice $v_i \in W$ terá sua cobertura assegurada em qualquer $s' \in S_{G5'}$, já que todo vértice de $v_k \in T$ deve estar em qualquer rota viável para o problema, seja em G ou em G5', uma vez que nenhum vértice de W faz parte de uma solução do PRR. Além disso, como nenhum vértice de V é removido por esta regra, toda solução viável em G será também viável em G5', em particular a solução ótima, o que verifica a condição 2.

Afirmação 3: G6' é uma redução de G.

Por hipótese, $T' = T \cup \{v_k \mid v_k \in V \setminus T \text{ é o único vértice onde } \delta_{ik} = 1, v_i \in W\}$. Assim sendo, qualquer $s' \in S_{G6'}$, por definição, contém todos os vértices de T' e cobre todos os vértices de W, já que W' = W. Por outro lado, toda solução viável em G, deve conter todos os vértices de T e cobrir todos os vértices de W. Como v_k é o único vértice que cobre $v_i \in W$, então v_k está em toda solução de S_G e, portanto $s' \in S_G$.

Seja r_{opt} uma solução ótima do PRR em G, então $r_{opt} \in S_G$ e possui custo mínimo. Como não existe solução viável em G que não contenha $v_k \in V \setminus T$, pois v_k é o único vértice onde $\delta_{ik} = 1, v_i \in W$, então r_{opt} contém v_k . Assim sendo, como $v_k \in T'$, nenhum vértice foi removido de V ou W, então $r_{opt} \in S_{G6'}$.

Afirmação 4: G7' é uma redução de G.

Seja $s' \in S_{G7'}$ e $v_l \in V$, tal que $v_l \in s'$. Se $\delta_{jl} = 1$ e $\delta_{jk} = 1$ para algum k, δ_{ik} será igual a 1, estará coberto por s' e, portanto, será desnecessário em qualquer solução. Como por hipótese $\delta_{ik} \geq \delta_{jk}, \forall v_k \in V$, pode-se afirmar que $s' \in S_G$, já que $\delta_{il} = 1$. Além disso, como nenhum vértice de V é removido por esta regra, toda solução viável em G será também viável em G' e em particular, $r_{opt} \in S_{G7'}$.

Tendo sido provado nesta seção que os grafos G' associados ao PRR obtidos a partir da aplicação do conjunto de regras proposto {R6, R5, R7, R4} são reduções de G, a Seção 3.1.3 apresenta o resultado da utilização destas regras quando aplicadas aos grafos associados às instâncias do PRR geradas neste trabalho (vide Apêndice A).

3.1.3 Experimentos computacionais com o conjunto de regras de redução proposto para o PRR

Esta seção apresenta o resultado dos testes realizados para avaliar o impacto do conjunto de regras de redução proposto na Seção 3.1.2 quando aplicado às instâncias do PRR geradas neste trabalho (vide Apêndice A).

A partir de uma análise detalhada do impacto causado pela aplicação prévia de uma regra de redução sobre a outra, considerando as regras que compõem o conjunto proposto para o PRR na Seção 3.1.2, constatou-se que a ordem na qual estas regras são aplicadas pode influenciar no percentual de redução do grafo resultante (vide Apêndice B). Logo, visando aumentar o percentual de redução total do grafo, sugere-se a adoção de uma das seguintes sequências para a aplicação das quatro regras do conjunto proposto:

Sequência 1: $R7 \rightarrow R6 \rightarrow R5 \rightarrow R4$ Sequência 2: $R6 \rightarrow R7 \rightarrow R5 \rightarrow R4$

Sequência 3: $R6 \rightarrow R5 \rightarrow R7 \rightarrow R4$

A Tabela 3.1 apresenta o percentual de redução dos grafos associados às instâncias do Grupo III (instâncias de 15 a 60 vértices), considerando a sequência 1 descrita anteriormente para a aplicação das regras propostas nesta seção. Na 1^a coluna encontra-se relacionado o nome de cada uma das instâncias submetidas à redução. Na 2^a e 3^a colunas tem-se, respectivamente, o percentual de redução total de cada instância e o percentual de redução do conjunto W.

	Regras {R7,R6,R5,R4}		
Instâncias do	l das Reduções		
Grupo IV	V + W	W	
ctp15 3-3-9	73,33%	100,00%	
ctp15 3-9-3	53,33%	88,89%	
ctp15 4-4-7	60,00%	100,00%	
ctp15 4-6-5	53.33%	100.00%	
ctp15 7-7-1	53,33%	100,00%	
ctp15 9-3-3	33,33%	100,00%	
ctp20 10-10-0	50,00%	100,00%	
ctp20 12-4-4	40,00%	100,00%	
ctp20 4-12-4	75,00%	100,00%	
ctp20 4-4-12	65,00%	100,00%	
ctp20 6-6-8	65,00%	100,00%	
ctp20 6-8-6	65,00%	100,00%	
ctp25 12-12-1	$52,\!00\%$	100,00%	
	40,00%	100,00%	
	76,00%	100,00%	
	60,00%	80,00%	
	68,00%	100,00%	
	64,00%	100,00%	
	50,00%	100,00%	
 ctp3018-6-6	$36,\!67\%$	100,00%	
ctp30 6-18-6	$73,\!33\%$	100,00%	
ctp30 6-6-18	$76,\!67\%$	100,00%	
ctp30_9-12-9	$66,\!67\%$	100,00%	
ctp30_9-9-12	$66,\!67\%$	$100,\!00\%$	
ctp35_10-14-11	60,00%	100,00%	
ctp35_11-11-13	$57,\!14\%$	$100,\!00\%$	
ctp35_17-17-1	$51,\!43\%$	100,00%	
$ctp35_21-7-7$	$31,\!43\%$	$100,\!00\%$	
ctp35_7-21-7	$71,\!43\%$	100,00%	
$\operatorname{ctp35}_{7-7-21}$	60,00%	85,71%	
$ctp40_{12-16-12}$	$60,\!00\%$	$93,\!75\%$	
$ctp40_{13-13-14}$	50,00%	$92,\!31\%$	
$\operatorname{ctp40}_20\text{-}20\text{-}0$	$50,\!00\%$	$100,\!00\%$	
$\operatorname{ctp40}{\underline{}24\text{-}8\text{-}8}$	$37,\!50\%$	$100,\!00\%$	
$\operatorname{ctp40}_{8-24-8}$	$77,\!50\%$	$100,\!00\%$	
$\operatorname{ctp40}_{8-8-24}$	$60,\!00\%$	$87,\!50\%$	
${ m ctp50_10-10-30}$	$^{66,00\%}$	90,00%	
$ctp50_{10-30-10}$	$80,\!00\%$	$100,\!00\%$	
$ctp50_{15-20-15}$	66,00%	100,00%	
$ctp50_{16-16-18}$	$^{64,00\%}$	$100,\!00\%$	
${ m ctp50}_{25-25-0}$	50,00%	$100,\!00\%$	
ctp50_30-10-10	$40,\!00\%$	$100,\!00\%$	
$ctp60_{12-12-36}$	$^{63,33\%}$	$91,\!67\%$	
$ctp60_{12-36-12}$	80,00%	100,00%	
$ctp60_{18-24-18}$	68,33%	100,00%	
$ctp60_{19-19-22}$	$66,\!67\%$	$100,\!00\%$	
$ctp60_{30-30-0}$	50,00%	$100,\!00\%$	
36-12-12	40,00%	100,00%	
Médias	58,70%	98,12%	

Tabela 3.1: Redução das instâncias do Grupo III (PRR) - Regras $\{R7, R6, R5, R4\}$

Como pode ser observado na 3^a coluna da Tabela 3.1, aproximadamente 83% dos grafos reduziram 100% do conjunto W, o que certamente reduzirá o custo computacional da exploração do espaço de busca destas instâncias, já que não será mais necessário verificar a restrição de cobertura dos vértices do referido conjunto. O percentual de redução total dos grafos ficou em torno de 58,7% (2^a coluna), o que comprova empiricamente o impacto causado pela utilização do conjunto de regras proposto para o PRR na redução das instâncias do Grupo III.

A Tabela 3.2 apresenta o percentual de redução dos grafos associados às instâncias do Grupo IV (instâncias de 100 a 700 vértices), considerando a sequência 1 para a aplicação das regras. As colunas da referida tabela estão organizadas conforme descrito anteriormente na Tabela 3.1.

Instâncias do Grupo IV	Regras $\{R7, R6, R5, R4\}$	
	Percentual das Reduções	
	V + W	W
ctp100_20-60-20	$80,\!00\%$	$100,\!00\%$
ctp100_33-33-34	57,00%	96,97%
ctp100_50-50-0	50,00%	$100,\!00\%$
$ctp100_60\text{-}20\text{-}20$	40,00%	$100,\!00\%$
ctp200_100-100-0	50,00%	$100,\!00\%$
$ctp200_{120-40-40}$	40,00%	$100,\!00\%$
ctp200_40-120-40	$73,\!50\%$	97,50%
$\operatorname{ctp200_66-66-68}$	$56,\!50\%$	93,94%
ctp300_150-150-0	$50,\!00\%$	$100,\!00\%$
$ctp300_{180-60-60}$	$39,\!67\%$	$100,\!00\%$
ctp300_60-180-60	$77,\!00\%$	98,89%
$ctp300_{99-99-102}$	$^{62,67\%}$	96,97%
ctp500_100-300-100	$72,\!20\%$	97,33%
$ctp500_{165-165-170}$	$60,\!60\%$	95,15%
$ctp500_{250-250-0}$	50,00%	$100,\!00\%$
$ctp500_{300-100-100}$	$39,\!80\%$	$100,\!00\%$
ctp700_140-420-140	77,71%	99,29%
${\rm ctp700}_231\text{-}231\text{-}238$	$64,\!29\%$	97,84%
$ctp700_{350-350-0}$	50,00%	$100,\!00\%$
ctp700_420-140-140	$39,\!43\%$	99,29%
ctp1000_200-600-200	$74,\!90\%$	97,50%
$ctp1000_{330-330-340}$	$^{62,40\%}$	96,36%
$ctp1000_{500-500-0}$	50,00%	$100,\!00\%$
$ctp1000_{600-200-200}$	39,40%	99,50%
Médias	$56,\!54\%$	98,61%

Tabela 3.2: Redução das instâncias do Grupo IV (PRR) - Regras {R7,R6,R5,R4}

Observa-se na Tabela 3.2 que 12 instâncias do Grupo IV obtiveram redução total acima de 50%. Em relação à redução do conjunto W, 46% das instâncias eliminaram

todos os vértices deste conjunto, o que confirma empiricamente o impacto causado pela utilização do conjunto de regras proposto para o PRR na redução das instâncias com até 1000 vértices.

Experimentos computacionais realizados com seis sequências distintas para a aplicação das quatro regras (incluindo as três sequências descritas nesta seção), confirmaram empiricamente que as três sequências aqui sugeridas aumentam a redução dos grafos associados às 72 instâncias relacionadas nos experimentos. Os resultados completos das reduções obtidas a partir destas seis sequências podem ser encontrados no Apêndice B.

3.2 Regras de redução para o PRRG

Esta seção apresenta as duas regras de redução existentes na literatura para o PRRG (Seção 3.2.1) e relaciona, como contribuições para este problema, uma nova regra para a redução do conjunto W, um novo conjunto de regras para a redução dos grafos associados ao PRRG e uma demonstração teórica da corretude de todas as regras que compõem o referido conjunto (Seção 3.2.2). Experimentos computacionais mostram o impacto do emprego do novo conjunto de regras na redução de 72 instâncias do PRRG (Seção 3.2.3).

3.2.1 Regras de redução existentes na literatura para o PRRG

O único conjunto de regras para a redução do grafo associado ao PRRG foi proposto em (MOTTA, 2001).

De acordo com (MOTTA, 2001), se $G = (V \cup W, E)$ é um grafo completo e não direcionado associado à uma instância do PRRG, onde $V \cup W = \{0, ..., v_k\}$ é o conjunto dos vértices e $E = \{(v_i, v_j) \mid v_i, v_j \in V \cup W, v_i \neq v_j\}$ é o conjunto das arestas, então os conjuntos $V \setminus T$ e W podem ser reduzidos ou transformados aplicando-se uma única vez as seguintes regras:

- (**R4**) Se $\forall v_j \in W, \ \delta_{ji} = 0$, então remova $v_i \in V \setminus T$;
- (R8) Se existir $v_k \in T$ e $v_i \in W$, tal que $\delta_{ik} = 1$, então transforme v_i em um vértice optativo pertencente ao conjunto $V \setminus T$.

A regra R8 foi proposta em (MOTTA, 2001) após ter sido realizada uma análise do comportamento das regras de redução descritas em (MANIEZZO et al., 1999) para a redução do grafo associado ao PRRG. Uma das observações importantes identificadas na referida análise está relacionada à regra R5 que, se empregada diretamente na redução do grafo associado ao PRRG, pode eliminar do grafo reduzido a solução ótima do grafo original, fato que descarta a possibilidade de aproveitamento desta regra para o PRRG. A Figura 3.9 mostra o que pode acontecer quando a regra de redução R5 é aplicada diretamente à um grafo associado ao PRRG.



Figura 3.9: Exemplo do uso da regra de redução R5 proposta para o PRR quando é aplicada diretamente em um grafo associado ao PRRG

Como pode ser observado na Figura 3.9B, a regra R5 remove os vértices 3 e $4 \in W$, pois ambos possuem a cobertura assegurada por qualquer rota viável associada ao grafo em questão, uma vez que o vértice $1 \in T$ pertence tanto à vizinhança do vértice 3 quanto do vértice 4. Note também que ao remover o vértice 4, a solução ótima associada ao grafo original (rota em laranja representada na Figura 3.9D) é eliminada do grafo reduzido, sendo o custo da solução ótima associada ao grafo reduzido (rota em verde representada na Figura 3.9D) maior do que o custo da solução ótima do grafo original, quando estes custos deveriam ser iguais. Logo, pode-se afirmar que, no caso dos grafos associados ao PRRG, faz-se necessário verificar não somente a sua condição de cobertura de um vértice $v_i \in W$ em relação aos vértices $v_k \in T$, como também em relação aos demais vértices $v_j \in W$ da vizinhança de v_i ($v_i \neq v_j$).

Para resolver o problema ocasionado pela remoção prematura dos vértices $v_i \in W$, com $\delta_{ik} = 1$ e $v_k \in T$, assim como mostrado na sequência da Figura 3.9, as regras propostas em (MOTTA, 2001) realizam a remoção dos vértices de W em dois estágios: R8 relaciona todos os vértices $v_i \in W$ que possuem a cobertura assegurada por algum vértice de $v_k \in T$, enquanto R4 remove todos os vértices (inclusive os relacionados por R8) que não são úteis para cobrir algum outro vértice $v_j \in W | v_j \neq v_i$. A Figura 3.10 apresenta o resultado da aplicação da regra R8 no grafo da Figura 2.1.



Figura 3.10: Exemplo da aplicação da regra R8 no grafo da Figura 2.1

Observe que a regra R8 apenas transformou os vértices 7, 8 e $10 \in W$ em vértices de $V \setminus T$. Quando a regra R4 for aplicada ao grafo da Figura 3.10, serão removidos todos os vértices de $V \setminus T$ que não pertencerem à vizinhança de pelo menos um vértice de W, inclusive os transformados pela regra R8.

A Figura 3.11 ilustra a aplicação da regra R4 no grafo da Figura 3.10, destacando a remoção dos vértices 7, 10, 13, 16, 18, 20 e 23.



Figura 3.11: Exemplo da aplicação da regra de redução R4 no grafo da Figura 3.10

O grafo resultante da aplicação do conjunto de regras proposto para o PRRG em (MOTTA, 2001) é apresentado na Figura 3.12. O percentual de redução atingido neste caso é de 32%, quando observado o grafo original apresentado na Figura 2.1.



Figura 3.12: Grafo associado ao PRRG resultante da aplicação do conjunto de regras de redução propostas em (MOTTA, 2001)

3.2.2 Nova regra de redução para o PRRG

A partir da análise do emprego da regra R1 nos grafos associados ao PRRG, é proposto neste trabalho uma nova regra R9 que transforma iterativamente os vértices do conjunto W da seguinte forma:

(R9) Se existirem $v_i, v_j \in W$, com $v_i \neq v_j$, tal que $\delta_{ik} \geq \delta_{jk}$, $\forall v_k \in V$, então transforme v_i em um vértice optativo pertencente ao conjunto $V \setminus T$;

Em outras palavras, a regra R9 transforma o vértice $v_i \in W$ em um vértice opcional pertencente ao conjunto $V \setminus T$, se existir $v_j \in W$, com $v_i \neq v_j$ e se a vizinhança de v_j estiver inteiramente contida na vizinhança de v_i .

Para ilustrar a aplicação da regra R9, considere o grafo associado ao *PRRG* apresentado na Figura 3.13.



Figura 3.13: Exemplo do uso da regra de redução R9 quando aplicada a um grafo associado ao PRRG

Observando os vértices 3 e 5 da Figura 3.13A, nota-se que suas vizinhanças estão inteiramente contidas na vizinhança do vértice 4, fato que pela regra R9 permite transformar o vértice 4 em um vértice optativo pertencente ao conjunto $V \setminus T$ (Figura 3.13B). Analisando o grafo transformado pela regra R9 (Figura 3.13C), constata-se que qualquer rota viável para este grafo deve conter todos os vértices obrigatórios e pelo menos um vértice da vizinhança do vértice 3 e do vértice 5, o que garante a cobertura do vértice 4 que foi transformado em vértice optativo.

Tendo sido enunciada a nova regra R9, é proposto a seguir um novo conjunto de regras para reduzir os grafos associados ao *PRRG*:

- (**R4**) Se $\delta_{ji} = 0, \forall v_j \in W$, então remova $v_i \in V \setminus T$;
- (**R8**) Se existir $v_k \in T$ e $v_i \in W$, tal que $\delta_{ik} = 1$, então transforme v_i em um vértice optativo pertencente ao conjunto $V \setminus T$;
- (**R9**) Se existirem $v_i, v_j \in W$, com $v_i \neq v_j$, tal que $\delta_{ik} \geq \delta_{jk}$, $\forall v_k \in V$, então transforme v_i em um vértice optativo pertencente ao conjunto $V \setminus T$;

Analogamente ao PRR, para provar a validade do conjunto de regras de redução proposto nesta seção para o PRRG, deve-se verificar as mesmas duas condições definidas na Seção 3.1.2:

Condição 1: Toda solução viável em G' também seja viável em G; Condição 2: $r_{opt} \in S_{G'}$.

Lembre-se que no caso do PRRG, por definição, os vértices de W podem fazer parte das soluções viáveis do problema. Por isso, tanto S_G quanto $S_{G'}$ podem possuir soluções que contenham vértices de W. No que segue, prova-se que as condições 1 e 2 são verificadas quando o conjunto de regras proposto nesta seção é aplicado a qualquer grafo Gassociado ao PRRG, assim como os grafos obtidos a partir da aplicação das regras R9, R8, e R4 são reduções de G.

Afirmação 5: G4' é uma redução de G (PRRG).

A prova desta afirmação é similar aquela descrita para a afirmação 4 na Seção 3.1.2. A diferença que vale ser ressaltada relaciona-se ao domínio do conjunto de vértices de G'que, no caso desta afirmação 7, é formado por todos os vértices de $T \cup W$ e por todos os vértices $v_j \in V \setminus T$, onde $\delta_{jk} = 1$ para pelo menos um $v_i \in W$. Afirmação 6: G8' é uma redução de G.

De fato a condição 1 se verifica, pois o vértice $v_i \in W$ terá sua cobertura assegurada em qualquer $s' \in G$, já que todo vértice de $v_k \in T$ deve estar em qualquer rota viável para o problema, seja em G ou em G8'.

Quanto a condição 2, como por hipótese $v_i \in W$ passou a ser um vértice de $V \setminus T$, nenhum vértice foi eliminado do grafo por esta regra e, por conseguinte, todo $v_j \in G$ tal que $v_j \in r_{opt}$ também está em G8'. Logo, $r_{opt} \in G8'$.

Afirmação 7: G9' é uma redução de G.

Seja $s' \in S_{G9'}$ e $v_l \in V$, tal que $v_l \in s'$ e $\delta_{jl} = 1$. Como por hipótese $\delta_{ik} \geq \delta_{jk}$, $\forall v_k \in V$, pode-se afirmar que $s' \in S_G$, já que $\delta_{il} = 1$.

Sabe-se que a regra R9 não remove vértices do grafo e, portanto, tem-se $V' \cup W' = V \cup W$ e E' = E. Como transformar um vértice $v_j \in W$ em um vértice $v_j \in V \setminus T$ não impede que v_j pertença a r_{opt} , conclui-se que $S_{G9'} = S_G$, provando a condição 2.

A próxima seção apresenta o resultado da aplicação do conjunto de regras de redução proposto nesta seção para o PRRG, tendo sido provado que os grafos associados ao PRRG obtidos a partir da aplicação das regras R9, R8 e R4 são reduções de G.

3.2.3 Experimentos computacionais com o conjunto de regras de redução proposto para o PRRG

Esta seção apresenta o resultado dos testes realizados para avaliar o impacto da utilização do conjunto de regras de redução proposto na Seção 3.2.2, quando considerados os grafos associados aos dois grupos de instâncias do PRRG gerados neste trabalho (vide Apêndice A).

Assim como mencionado na Seção 3.1.3, a ordem na qual as regras de redução são aplicadas pode influenciar no percentual de redução do grafo resultante. Portanto, visando potencializar as reduções, sugere-se neste trabalho a adoção de uma das seguintes sequências para a aplicação das três regras que compõem o conjunto proposto: Sequência 1: $R9 \rightarrow R8 \rightarrow R4$ Sequência 2: $R8 \rightarrow R9 \rightarrow R4$

O Apêndice B apresenta o resultado completo dos experimentos com quatro sequências testadas para a aplicação das três regras que compõem o conjunto proposto nesta seção.

As Tabelas 3.3 e 3.4 mostram, respectivamente, o percentual de redução dos grafos associados as instâncias do Grupo I (15 a 60 vértices) e do Grupo II (100 a 1000 vértices), após a utilização das regras de redução aplicadas na ordem sugerida na sequência 1 recém descrita. Na 1^{*a*} coluna destas tabelas encontra-se relacionado o nome de cada uma das instâncias submetidas à redução, enquanto na 2^{*a*} coluna tem-se o percentual de redução total de cada instância e na 3^{*a*} o percentual de redução do conjunto W.

Observando a 2^a coluna da Tabela 3.3, nota-se que cerca de 27% das instâncias reduziram mais de 50% do total de vértices. As duas instâncias que obtiveram a menor redução total foram $gctp15_3-9-3$ e $gctp40_12-16-12$, com 20% e 22,5% de redução total respectivamente. As duas instâncias que atingiram o maior percentual de redução total são $gctp35_11-11-13$, com 68,57% e a $gctp50_10-10-30$, com 76%.

Ao analisar o percentual de redução do conjunto W (3^{*a*} coluna - Tabela 3.3), observase que somente cerca de 8% das instâncias conseguiram reduzir totalmente este conjunto. Em média, a redução do conjunto W ficou em torno de 74%, mas 10,4% das instâncias reduziram menos de 50% o referido conjunto.

Para as instâncias de maiores dimensões (100 a 1000 vértices), assim como pode ser constatado na Tabela 3.4, 20,8% das instâncias atingiram uma redução total acima de 50%. A menor redução total foi de 30% para a instância $gctp100_{-}60-20-20$ e a maior foi de 62,50% para a instância $gctp1000_{-}330-330-340$.

Diferente do ocorrido para as instâncias de pequenas dimensões (Tabelas 3.3), nenhuma instância do Grupo II atingiu 100% de redução do conjunto W (3^a coluna - Tabela 3.4). Entretanto, 50% das instâncias do referido grupo obtiveram redução acima de 90%. A menor redução do conjunto W foi de 71,67% para a instância $gctp100_20-60-20$ e a maior foi de 99,60% para a instância $gctp1000_500-500-0$.

	Regras {R9.R8.R4}		
Instâncias reduzida Grupo I	Percentual das Beduções		
	V + W	W	
gctp15 3-3-9	46,67%	33,33%	
gctp15_3-9-3	20.00%	44,44%	
gctp15 4-4-7	40.00%	75.00%	
gctp15_4-6-5	26.67%	33.33%	
gctp15_7-7-1	40.00%	85.71%	
gctn15_9-3-3	26.67%	66 67%	
gctp20_10-10-0	35.00%	90.00%	
$gctp20_10_{10} 10^{-10}$	40.00%	100.00%	
$gctp20_12 + 4$ $gctp20_4.12.4$	25.00%	66 67%	
$gctp20_4-12-4$	60.00%	75.00%	
$gctp20_4-4-12$	40.00%	50.00%	
$gctp20_0-6-8$	40,0070 60.00%	87 50%	
getp20_0-0-0	52.00%	100.00%	
getp25_12-12-1	52,0070 40,0007		
gctp25_15-5-5	40,007	100,00%	
gctp25_5-15-5	44,00%	80,00%	
gctp25_5-5-15	48,00%	60,00%	
gctp25_7-10-8	44,00%	70,00%	
gctp25_8-8-9	60,00%	87,50%	
gctp30_15-15-0	30,00%	80,00%	
gctp30_18-6-6	33,33%	83,33%	
gctp30_6-18-6	43,33%	72,22%	
gctp30_6-6-18	53,33%	50,00%	
gctp30_9-12-9	63,33%	91,67%	
gctp30_9-9-12	40,00%	66,67%	
gctp35_10-14-11	25,71%	50,00%	
gctp35_11-11-13	68,57%	$100,\!00\%$	
gctp35_17-17-1	$28,\!57\%$	76,47%	
gctp35_21-7-7	$34,\!29\%$	85,71%	
gctp35_7-21-7	42,86%	80,95%	
gctp35_7-7-21	$60,\!00\%$	71,43%	
$gctp40_{12-16-12}$	$22,\!50\%$	56,25%	
gctp40_13-13-14	47,50%	76,92%	
$g \operatorname{ctp} 40 20-20-0$	$30,\!00\%$	90,00%	
$g \operatorname{ctp40}_{24-8-8}$	$35,\!00\%$	87,50%	
$g \operatorname{ctp} 40 8-24-8$	$57,\!50\%$	83,33%	
$g \operatorname{ctp40}_{8-8-24}$	$55,\!00\%$	37,50%	
$gctp50_{10-10-30}$	$60,\!00\%$	70,00%	
$\operatorname{gctp50_10-30-10}$	$76,\!00\%$	96,67%	
$\mathrm{gctp50_15}\text{-}20\text{-}15$	$44,\!00\%$	75,00%	
$gctp50_{16-16-18}$	$46,\!00\%$	75,00%	
$\operatorname{gctp50}_25\text{-}25\text{-}0$	$34,\!00\%$	92,00%	
$\mathrm{gct}\mathrm{p50}_30\text{-}10\text{-}10$	$28,\!00\%$	70,00%	
gctp60_12-12-36	$35,\!00\%$	33,33%	
gctp60_12-36-12	$35,\!00\%$	69,44%	
gctp60_18-24-18	$51,\!67\%$	83,33%	
gctp60_19-19-22	$45,\!00\%$	68,42%	
gctp60_30-30-0	$33,\!33\%$	83,33%	
gctp60_36-12-12	$36,\!67\%$	91,67%	
Médias	42,57%	$\mathbf{74,03\%}$	

Tabela 3.3: Redução das instâncias do Grupo I (PRRG) - Regras $\{\rm R9, R8, R4\}$

Instâncias reduzida Grupo II	Regras {R9,R8,R4}		
	Percentual das Reduções		
	V + W	W	
gctp100_20-60-20	$31,\!00\%$	$71,\!67\%$	
gctp100_33-33-34	$45,\!00\%$	75,76%	
$gctp100_50-50-0$	$33,\!00\%$	$^{84,00\%}$	
$gctp100_{60-20-20}$	$30,\!00\%$	75,00%	
gctp200_100-100-0	$47,\!00\%$	97,00%	
$gctp200_{120-40-40}$	$^{34,50\%}$	87,50%	
$gctp200_{40-120-40}$	41,50%	80,83%	
$gctp200_{66-66-68}$	$56,\!50\%$	89, 39%	
gctp300_150-150-0	$41,\!67\%$	95,33%	
gctp300_180-60-60	$35,\!67\%$	91,67%	
$gctp300_{60-180-60}$	$38,\!67\%$	83,33%	
$gctp300_{99-99-102}$	$54,\!33\%$	90,91%	
$gctp500_{100-300-100}$	$37,\!80\%$	73,00%	
$gctp500_{165-165-170}$	$50,\!20\%$	84,24%	
$gctp500_{250-250-0}$	$40,\!20\%$	93,20%	
$gctp500_{300-100-100}$	$37,\!20\%$	94,00%	
$gctp700_{140-420-140}$	$37,\!43\%$	79,29%	
$gctp700_231-231-238$	$^{61,29\%}$	96,97%	
$gctp700_{350-350-0}$	$47,\!57\%$	98,86%	
$gctp700_420-140-140$	$38,\!14\%$	97,14%	
$gctp1000_200-600-200$	$48,\!30\%$	85,83%	
$gctp1000_{330-330-340}$	$62,\!50\%$	96,97%	
$gctp1000_500-500-0$	49,40%	99,60%	
$gctp1000_600-200-200$	$37,\!60\%$	95,00%	
Médias	43,19%	88,19%	

Tabela 3.4: Redução das instâncias do Grupo II (PRRG) - Regras {R9,R8,R4}

Como poder ser constatado nas Tabelas 3.3 e 3.4, os experimentos computacionais realizados com o conjunto de regras proposto {R4, R8, R9} para a redução dos grafos associados ao PRRG, confirmaram empiricamente uma redução bastante significativa nas 72 instâncias envolvidas nos experimentos.

Capítulo 4

Formulações matemáticas para o PRR e para o PRRG

Este capítulo apresenta as formulações matemáticas existentes na literatura para o PRR e para o PRRG (Seção 4.1) e relaciona como contribuições da tese neste contexto: um comparativo entre estas formulações (Seção 4.2); uma análise da calibração dos parâmetros do algoritmo *branch-and-cut* utilizado pelo CPLEX na solução exata de um conjunto de instâncias destes problemas (Seção 4.3) e alguns resultados teóricos que estabelecem uma relação entre o PRR e o PRRG (Seção 4.4). Nos resultados computacionais apresentados na Seção 4.3.5, são destacadas as soluções ótimas obtidas para cada um dos problemas em estudo.

4.1 Formulações matemáticas existentes

O PRR foi primeiramente formulado como um Problema de Programação Linear Inteira (PPLI) por Gendreau, Laporte e Semet (1997). Dois anos mais tarde, Maniezzo *et al.* (MANIEZZO et al., 1999) propuseram um novo modelo de programação linear inteira para o PRR baseado na proposta de Finke, Claus e Gunn (FINKE; CLAUS; GUNN, 1984) para o PCV. Em 2001, Motta (MOTTA, 2001) propôs uma formulação matemática para o PRR e para o PRRG que utiliza variáveis de fluxo para estabelecer as restrições de conectividade e eliminação de ciclos da rota. As seções a seguir apresentam a descrição destas três formulações.

4.1.1 Formulação de Gendreau, Laporte e Semet para o PRR

Segundo Gendreau, Laporte e Semet (1997), o PRR pode ser formulado como um PPLI a partir da definição do seguinte conjunto de variáveis:

- y_k é uma variável binária definida ∀k ∈ V, onde y_k = 1 se, e somente se, o vértice k ∈ V estiver na rota e y_k = 0, caso contrário. Esta variável é necessariamente igual a 1 para todo k ∈ T;
- x_{ij} é uma variável binária definida para $i, j \in V$, com i < j, onde $x_{ij} = 1$ se, e somente se, a aresta (i, j) pertencer à rota e $x_{ij} = 0$, caso contrário;
- Δ = (δ_{lk}) é uma matriz binária, onde δ_{lk} = 1 se, e somente se, l ∈ W for coberto por k ∈ V e δ_{lk} = 0, caso contrário;
- H_l = {k ∈ V | δ_{lk} = 1}, isto é, H_l é o conjunto dos vértices k ∈ V que cobrem o vértice l ∈ W;
- S é qualquer subconjunto próprio de V, tal que $V \setminus S \neq \emptyset$;

Considerando $G = (V \cup W, E)$ um grafo completo e não direcionado, $c_{ij} \ge 0$ o custo da aresta $(i, j) \in E | i < j, |V| = n$ e tendo definido o conjunto de variáveis descritas anteriormente, Gendreau, Laporte e Semet apresentaram o PRR como um PPLI da seguinte forma:

$$(\mathcal{F}1) \qquad minimizar \sum_{i,j \in V | i < j} c_{ij} x_{ij} \tag{4.1}$$

sujeito à:

$$\sum_{k \in H_l} y_k \ge 1, \qquad \forall l \in W \tag{4.2}$$

$$\sum_{i < k} x_{ik} + \sum_{j > k} x_{kj} = 2y_k, \qquad \forall k \in V$$
(4.3)

$$\sum_{(i,j)\,|\,i\in S;\,j\in V\setminus S} x_{ij} \ge 2y_t, \qquad \forall t\in S, \ S\subset V$$

$$(4.4)$$

$$y_k = 1, \qquad \forall k \in T \tag{4.5}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall (i, j) \in E, \ i < j$$
(4.6)

$$y_k \in \{0, 1\}, \qquad \forall k \in V \setminus T \tag{4.7}$$

A função objetivo de $\mathcal{F}1$ dada por (4.1) deseja minimizar o custo total da rota sujeito às restrições (4.2)-(4.7). O conjunto de restrições geradas por (4.2) garante que todos os vértices de W estarão cobertos pela rota, enquanto as restrições (4.5) asseguram a presença de todos os vértices obrigatórios na rota. As restrições (4.3) determinam o grau dos vértices pertencentes à rota (grau 2). As restrições (4.4) são responsáveis pela conectividade da rota, onde $S \subsetneq V, 2 \leq |S| \leq n-2$ e S contém um vértice t da rota. Estas restrições forçam a presença de pelo menos duas arestas entre quaisquer subconjuntos Se $V \setminus S$. Finalmente, as restrições (4.6) e (4.7) garantem a integralidade das variáveis do problema.

Vale ressaltar que, por definição, a rota degenerada $\{0,0\}$, onde 0 é o depósito, é considerada inviável.

4.1.2 Formulação de Maniezzo, Baldacci, Boschetti e Zamboni para o PRR

A formulação proposta em (MANIEZZO et al., 1999) descreve o PRR como um PPLI através de uma adaptação da formulação proposta originalmente para o PCV (FINKE; CLAUS; GUNN, 1984), conhecida na literatura por *Two-Commodity*. O princípio básico desta formulação é associar duas variáveis de fluxo z_{ij} e z_{ji} a cada aresta (i, j) do grafo.

Para apresentar a referida formulação, algumas considerações prévias fazem-se necessárias:

- O vértice s pode ser considerado o ponto de partida da rota (origem) se $s \in T$;
- z_{ij} é uma variável inteira positiva que define dois circuitos para qualquer rota que representar uma solução viável para o PRR. Se o caixeiro viaja de i para j, então a variável z_{ij} representa o número de vértices que podem ser visitados e z_{ji} representa o número de vértices que podem ser visitados e z_{ji} representa o número de vértices que já foram visitados. A variável z_{ij} é definida para todo i, j ∈ V, i ≠ j;
- x_{ij} é uma variável binária definida ∀(i, j) ∈ E tal que i < j, onde x_{ij} = 1 se, e somente se, a aresta (i, j) está na solução e x_{ij} = 0, caso contrário;

- y_k é uma variável binária definida ∀k ∈ V, onde y_k = 1 se, e somente se, o vértice k ∈ V estiver na rota e y_k = 0, caso contrário;
- H_l = {i ∈ V | c_{il} ≤ d, l ∈ W}, isto é, sendo d ≥ 0 a distância de cobertura, H_l é o conjunto dos vértices i ∈ V que cobrem o vértice l ∈ W.

O resultado da formulação proposta por Maniezzo *et al.* (1999), sendo $G = (V \cup W, E)$ um grafo completo e não direcionado (|V| = n), pode ser visto da seguinte forma:

(F2)
$$minimizar \sum_{i,j \in V | i < j} c_{ij} x_{ij}$$
 (4.8)

sujeito à:

$$\sum_{j \in V, j \neq i} z_{ij} - \sum_{j \in V, j \neq i} z_{ji} = -2y_i, \qquad \forall i \in V - \{s\}$$
(4.9)

$$\sum_{j \in V \ j \neq i} (z_{ij} + z_{ji}) = 2y_i(|V| - 1), \qquad \forall i \in V$$
(4.10)

$$\sum_{j \in H_l} y_j \ge 1, \qquad \forall l \in W \tag{4.11}$$

$$z_{ij} + z_{ji} = (n-1)x_{ij}, \qquad \forall (i,j) \in E, \ i < j$$
(4.12)

$$y_j = 1, \qquad \forall j \in T \tag{4.13}$$

$$z_{ij} \ge 0, \qquad \forall (i,j) \in E, i \ne j \ e \ z_{ij} \in Z^+$$

$$(4.14)$$

$$y_j \in \{0, 1\}, \qquad \forall j \in V \setminus T$$

$$(4.15)$$

$$x_{ij} \in \{0, 1\}, \qquad \forall (i, j) \in E, \ i < j$$
(4.16)

Em $\mathcal{F}2$, a função objetivo a ser minimizada é dada por (4.8). O conjunto de restrições geradas a partir de (4.9) e (4.14) definem um fluxo viável para as variáveis z_{ij} . As restrições (4.10) e (4.12) asseguram que todo vértice *i* pertencente à rota possui necessariamente grau 2. As restrições (4.11) garantem a cobertura de todos os vértices $h \in W$, enquanto a presença de todos os vértices $i \in T$ na solução é assegurada pelas restrições (4.13). Finalmente, as restrições (4.15) e (4.16) definem as condições de integralidade do problema.

4.1.3 Formulação de Motta para o PRR e para o PRRG

A proposta apresentada por Motta (2001) introduz variáveis de fluxo nas restrições do PRR, assim como a formulação proposta por Maniezzo *et al.* (1999). Entretanto, cada aresta da rota possui uma única variável de fluxo associada, já que a referida proposta estende para o PRR/PRRG a ideia da modelagem conhecida na literatura por *Single-Commodity* (GAVISH; GRAVES, 1978).

A variável de fluxo usada no modelo é definida por z_{ij} e associa a cada aresta $(i, j) \in E$, um inteiro não negativo que representa o fluxo escoado na aresta (i, j). Também são definidas uma matriz de custo $C = (c_{ij})$, duas matrizes binárias $X = (x_{ij})$ e $\Delta = (\delta_{lk})$, além de um vetor de pertinência de vértices na rota $Y = (y_k)$, onde:

- $C = (c_{ij}) \in \Re^+$. Cada aresta $(i, j), i \neq j$ é associada a um valor c_{ij} ;
- y_k é uma variável binária definida ∀k ∈ V, tal que y_k = 1 se, e somente se, k
 pertencer a rota e y_k = 0, caso contrário. Portanto, para todo k ∈ T, então y_k é
 necessariamente igual a 1;
- x_{ij} é uma variável binária definida ∀i, j ∈ V, i < j, onde x_{ij} = 1 se, e somente se, a aresta (i, j) pertencer a rota e x_{ij} = 0, caso contrário;
- Δ = (δ_{lk}) é uma matriz binária, onde δ_{lk} = 1 se, e somente se, l ∈ W puder ser coberto por k ∈ V (i.e., c_{lk} ≤ d). Vale ressaltar que, ∑_{j∈V} δ_{ij} ≥ 1, ∀i ∈ W, caso contrário, o PRR não possui solução viável;
- $H_l = \{k \in V \mid \delta_{lk} = 1, l \in W\}$, isto é, H_l é o conjunto dos vértices $k \in V$ que cobrem o vértice $l \in W$.

O conjunto de restrições que definem o PPLI para o PRR proposto em (MOTTA, 2001), sendo $G = (V \cup W, E)$ um grafo completo e não direcionado (|V| = n), é dado por:

(F3)
$$minimizar \sum_{i,j \in V \mid i < j} c_{ij} x_{ij}$$
 (4.17)

sujeito à:

$$\sum_{k \in H_l} y_k \ge 1, \qquad \forall l \in W \tag{4.18}$$

$$\sum_{i < k} x_{ik} + \sum_{k < j} x_{kj} = 2y_k, \qquad \forall k \in V$$
(4.19)

$$\sum_{j \in V} z_{kj} = \sum_{i \in V} z_{ik} + y_k, \qquad \forall k \in V - \{s\}$$

$$(4.20)$$

$$\sum_{j \in V} z_{sj} = 1 \tag{4.21}$$

$$\sum_{j \in V} z_{js} = \sum_{j \in V} y_j \tag{4.22}$$

$$x_{ij} \le z_{ij}, \qquad \forall (i,j) \in E, i < j$$

$$(4.23)$$

$$x_{ij} \ge \left(\frac{z_{ij}}{|V| + |W| + 1}\right), \qquad \forall i, j \in V \mid i < j$$

$$(4.24)$$

$$y_k = 1, \qquad \forall k \in T \tag{4.25}$$

$$z_{ij} \ge 0, \qquad \forall (i,j) \in E, i < j \ e \ z_{ij} \in Z^+$$

$$(4.26)$$

$$y_k \in \{0, 1\}, \qquad \forall k \in V \setminus T \tag{4.27}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall (i, j) \in E, i < j$$
(4.28)

A equação (4.17) de $\mathcal{F}3$ define a função objetivo a ser minimizada. As restrições (4.18) asseguram a cobertura de todos os vértices de W por pelo menos um vértice $k \in V$ pertencente à rota. As restrições (4.19) asseguram que qualquer vértice pertencente à rota tenha grau 2, enquanto as restrições (4.20), (4.21) e (4.22) proíbem a formação de subrotas desconexas da origem s. Sem perda de generalidade, é assumido que s = 0 é o vértice origem ($s \in T$). As restrições (4.23) e (4.24) determinam que a rota gerada a partir das variáveis de fluxo (z_{ij}) coincida com a rota gerada a partir das variáveis de decisão (x_{ij}). Finalmente, as restrições (4.25) garantem que todos os vértices de T estarão na rota e as restrições (4.26), (4.27) e (4.28) definem as condições de integralidade do problema.

No que diz respeito ao PRRG, este foi formulado como um PPLI pela primeira vez em (MOTTA, 2001), no mesmo trabalho que definiu o problema. Esta formulação pode ser facilmente obtida a partir de $\mathcal{F}3$ estendendo o domínio da função objetivo (4.17) e das restrições (4.19), (4.20), (4.21), (4.22), (4.23) e (4.24) de forma que sejam considerados os vértices $i \in V \cup W$, onde estes se restringem somente ao conjunto V. As variáveis y_k , x_{ij} e z_{ij} também precisam ter seus domínios alterados, passando a ser definidas sobre o conjunto $V \cup W$. É importante salientar que as alterações descritas são bastante simples de serem implementadas a partir de $\mathcal{F}3$. Porém, o número de restrições e de variáveis associadas ao modelo do PRRG aumenta proporcionalmente à cardinalidade do conjunto W, quando comparado ao número de restrições e de variáveis associadas ao modelo do PRR para um mesmo grafo de entrada. Este aumento implica diretamente na necessidade de mais memória para armazenar as restrições e um maior esforço computacional para obter uma solução para o PRRG quando comparado ao PRR em condições semelhantes, assim como comprova experimentalmente os resultados apresentados nas Seções 4.2 e 4.3.

Uma outra forma de estender a formulação $\mathcal{F}3$ para o PRRG é promover alterações no grafo de entrada, considerando para cada $i \in W$ um outro vértice $j \in V$ coincidente. Neste caso, não é necessário alterar o domínio de nenhuma das equações que definem $\mathcal{F}3$ e nem tão pouco alterar o domínio das variáveis y_k , x_{ij} e z_{ij} .

4.2 Comparação entre as formulações da literatura

As seções anteriores apresentaram três formulações para modelar o PRR como um PPLI, podendo qualquer uma destas formulações ser adaptada para o PRRG de forma análoga ao que foi descrito na Seção 4.1.3 para a adaptação da formulação $\mathcal{F}3$. Entretanto, não se tem conhecimento de nenhum tipo de comparação presente na literatura entre estas formulações. Neste sentido, esta seção apresenta como uma das contribuições desta tese uma visão comparativa que relaciona o número de restrições e o número de variáveis em função do tamanho do problema, além de um comparativo experimental entre as formulações $\mathcal{F}2$ e $\mathcal{F}3$ para o PRR e para o PRRG¹.

A Tabela 4.1 apresenta os valores calculados a partir das equações que compõem as formulações $\mathcal{F}1$, $\mathcal{F}2$ e $\mathcal{F}3$, considerando o número de restrições, de variáveis binárias e de variáveis inteiras, sabendo que a cardinalidade dos conjuntos T, W e V é dada respectivamente por m, $r \in n$.

¹ A adaptação de $\mathcal{F}2$ para o PRRG é realizada pela primeira vez neste trabalho. A forma escolhida para fazer esta adaptação nos experimentos apresentados no decorrer deste capítulo foi promover alterações no grafo de entrada, assim como foi descrito no final da seção anterior
Formulação	$\operatorname{Restri}_{ ilde{o}}$	Variáveis Binárias	Variáveis Inteiras
$\mathcal{F}1$	$2^n + n + m + r$	$(n^2 + n - 2m)/2$	0
$\mathcal{F}2$	$(n^2 + 2 \times (m + r - 1) + 3n)/2$	$(n^2 + n - 2m)/2$	$2 \times (n^2 - n)$
$\mathcal{F}3$	$n^2 + n + m + r + 1$	$(n^2 + n - 2m)/2$	$(n^2 - n)/2$

Tabela 4.1: Número de restrições e de variáveis em relação ao tamanho do problema de entrada

Os resultados organizados na Tabela 4.1 consideram as formulações $\mathcal{F}1$, $\mathcal{F}2$ e $\mathcal{F}3$ com um grafo de entrada associado ao PRR. Uma tabela equivalente pode ser feita para estas formulações estendidas para o PRRG. Neste caso, vale lembrar que o número de restrições e variáveis associadas aos modelos aumentam proporcionalmente à cardinalidade do conjunto W.

Observando os dados apresentados na Tabela 4.1, verificou-se que:

- A formulação *F*1 possui o maior número de restrições quando comparada às demais, pois somente a equação (4.4) é responsável por gerar aproximadamente 2ⁿ restrições, uma vez que para cada circuito Hamiltoneano possível tem-se uma restrição definida por (4.4);
- O fato da formulação \$\mathcal{F}\$1 produzir um número de restrições consideravelmente superior às outras duas formulações (\$O(2^n)\$ restrições para \$\mathcal{F}\$1 e \$O(n^2)\$ restrições para \$\mathcal{F}\$2 e \$\mathcal{F}\$3\$), não implica necessariamente que tal formulação seja ineficiente. Orman e Williams (ORMAN; WILLIAMS, 2004) apresentaram um comparativo entre oito diferentes formulações para o Problema do Caixeiro Viajante Assimétrico (PCVA) e verificaram através de um exemplo numérico que a formulação que utiliza as restrições tradicionais do PCV (assim como \$\mathcal{F}\$1) para evitar a formação de subciclos desconexos da origem apresentou, de modo geral, os melhores resultados;
- O número de variáveis binárias nas três formulações é igual, sendo da ordem de O(n²);
- A formulação *F*1 não utiliza variáveis inteiras, enquanto as formulações *F*2 e *F*3 apresentam O(n²) variáveis inteiras associadas às variáveis de fluxo empregadas em ambas as formulações.

O número exponencial de restrições da formulação $\mathcal{F}1$ restringe sua utilização na prática, devido às limitações de memória RAM do ambiente computacional utilizado

neste trabalho. Portanto, somente as formulações $\mathcal{F}2$ e $\mathcal{F}3$ foram selecionadas para os experimentos computacionais comparativos apresentados no decorrer desta seção.

Com a finalidade de comparar o desempenho médio das formulações $\mathcal{F}2$ e $\mathcal{F}3$ na obtenção dos limites duais para o PRR e para o PRRG, experimentos computacionais foram realizados com um subconjunto das instâncias descritas no Apêndice A. Estas instâncias foram selecionadas pelo total de vértices que, nesta seção, variam de 15 a 300 vértices no total. Foram utilizadas a relaxação linear das referidas formulações produzidas pelo próprio CPLEX.

Todos os experimentos computacionais relacionados neste capítulo utilizaram o *software* ILOG CPLEX na versão 11.2 e foram executados em um computador com processador Intel® CoreTM 2 Quad Q9550, com 2.83 GHz de frequência, 12 MB de memória cache L2 e 8 GB de memória RAM. Como sistema operacional foi utilizada a distribuição GNU/Linux Ubuntu, com kernel na versão 2.6.28-15-generic.

Para comparar as formulações $\mathcal{F}2 \in \mathcal{F}3$ foram realizados experimentos computacionais com a relaxação linear destas formulações. As seguintes métricas foram utilizadas para organizar os resultados obtidos nestes experimentos:

- *MelhorDual*: Para cada instância, *MelhorDual* equivale ao valor do melhor limite dual encontrado pela relaxação linear das formulações consideradas no experimento;
- *Dif% MelhorDual*: Para cada instância, *Dif% MelhorDual* é diferença percentual entre o valor do ótimo dual da formulação considerada e *MelhorDual*;
- MDif% MelhorDual: Para cada formulação *F*, MDif% MelhorDual é a média de todos os valores Dif% MelhorDual encontrados por *F* para um determinado grupo de instâncias;
- Tempo: Para cada formulação *F*, Tempo equivale a média do tempo total consumido por *F* para cada uma das instâncias até para alcançar o ótimo dual;
- MedIter: Para cada formulação *F*, MedIter é a média do total de iterações que *F* realizou para cada uma das instâncias até para alcançar o ótimo dual;

A Tabela 4.2 apresenta os resultados da relaxação linear de $\mathcal{F}2$ e $\mathcal{F}3$ para 60 instâncias do PRRG, das quais 48 são instâncias de menores dimensões com até 60 vértices (GI) e 12 são instâncias com até 300 vértices (GII).

	G	II	G	HI
	$\mathcal{F}2$	$\mathcal{F}3$	$\mathcal{F}2$	$\mathcal{F}3$
MDif% MelhorDual	0,00%	19,70%	0,00%	$19,\!50\%$
Tempo (seg)	$0,\!16$	0,23	$66,\!77$	$870,\!67$
MedIter	$1.756,\!51$	$1.722,\!11$	$69.618,\!83$	$244.915,\!17$

Tabela 4.2: Resultados da relaxação linear de $\mathcal{F}2$ e $\mathcal{F}3$ para o PRRG

Observando os resultados apresentados na Tabela 4.2, pode-se verificar que $\mathcal{F}2$ obteve o melhor limite dual para todas as 60 instâncias do PRRG (GI e GII). A qualidade dos limites duais obtidos por $\mathcal{F}3$ para as mesmas 60 instâncias foi aproximadamente 19% pior do que os obtidos por $\mathcal{F}2$. Quando são consideradas as instâncias com até 60 vértices (GI), a média do tempo total consumido por $\mathcal{F}3$ para obter os ótimos duais foi 43% maior do que a média do tempo gasto por $\mathcal{F}2$. Para as instâncias de 100 a 300 vértices (GII), a média do tempo total gasto por $\mathcal{F}3$ foi aproximadamente 13 vezes maior do que a média do tempo consumido por $\mathcal{F}2$, fato que é justificado pela média do total de iterações que $\mathcal{F}3$ realizou para as instâncias de GII.

Os resultados obtidos pela relaxação linear de $\mathcal{F}2$ e $\mathcal{F}3$ para 60 instâncias do PRR com até 300 vértices (GIII e GIV) são apresentados na Tabela 4.3.

	G	III	G	IV
	$\mathcal{F}2$	$\mathcal{F}3$	$\mathcal{F}2$	$\mathcal{F}3$
MDif% MelhorDual	0,00%	$18,\!51\%$	0,00%	$16,\!45\%$
Tempo (seg)	$0,\!04$	0,03	$7,\!24$	$146,\!42$
MedIter	712,38	$399,\!62$	$23.849{,}50$	$104.355,\!25$

Tabela 4.3: Resultados da relaxação linear de $\mathcal{F}2$ e $\mathcal{F}3$ para o PRR

Assim como observado no experimento realizado para o PRRG, $\mathcal{F}2$ obteve o melhor limite dual para as 60 instâncias do PRR consideradas neste experimento (GIII e GIV). A média do tempo total consumido por $\mathcal{F}2$ e $\mathcal{F}3$ para obter os ótimos duais para as instâncias com até 60 vértices (GIII) foi praticamente igual, apesar de $\mathcal{F}2$ ter realizado em média 43,9% mais iterações. Quando são consideradas as instâncias de maiores dimensões (100 a 300 vértices em GIV), nota-se que $\mathcal{F}3$ consome aproximadamente 20 vezes mais tempo em média e realiza 4,4 vezes mais iterações para alcançar o ótimo dual. Para melhorar a qualidade dos limites inferiores fornecidos por $\mathcal{F}2$, Maniezzo *et al.* (1999) sugeriram a inclusão de desigualdades válidas derivadas da observação de que em qualquer solução inteira viável para $\mathcal{F}2$, se $x_{ij} = 1$, com $(i, j) \in E$, i < j, $i \neq s$ e $j \neq s$, então $z_{ij} \geq 1$ e $z_{ji} \geq 1$. Portanto, as desigualdades a seguir satisfazem qualquer solução inteira viável para $\mathcal{F}2$, mas não necessariamente são verificadas para a sua relaxação linear:

$$z_{ij} - x_{ij} \ge 0, \qquad \forall (i,j) \in E, \ i < j, \ i \neq s, \ j \neq s \tag{4.29}$$

$$z_{ji} - x_{ij} \ge 0, \qquad \forall (i,j) \in E, \ i < j, \ i \neq s, \ j \neq s \tag{4.30}$$

Apesar de (4.29) e (4.30) terem sido sugeridas em (MANIEZZO et al., 1999), não é conhecido na literatura nenhum resultado teórico ou experimental para validar a qualidade das referidas desigualdades. Assim sendo, a Tabela 4.4 apresenta os resultados que avaliam empiricamente o benefício de acrescentar as restrições (4.29) e (4.30) em $\mathcal{F}2$. Foram consideradas as mesmas 120 instâncias utilizadas nos experimentos apresentados nas Tabelas 4.3 e 4.2. No contexto deste trabalho, $\mathcal{F}2'$ representará a formulação $\mathcal{F}2$ acrescida das restrições (4.29) e (4.30).

Tabela 4.4: Comparativo entre os resultados da relaxação linear de $\mathcal{F}2$ e $\mathcal{F}2'$ para as instâncias do PRR e PRRG

	PF	RRG	P	RR
	GI	GII	GIII	GIV
MDif% Ótimo Dual	$0,\!56\%$	0,03%	$0,\!23\%$	$0,\!02\%$
MDif% Tempo (seg)	$40,\!84\%$	$226,\!02\%$	$34{,}39\%$	$165,\!39\%$
$\operatorname{MDif}\%$ Iterações	6,93%	40,77%	$16{,}64\%$	$42{,}15\%$

Os valores contidos na primeira linha da Tabela 4.4 correspondem às médias das diferenças percentuais entre os ótimos duais obidos por $\mathcal{F}2$ e $\mathcal{F}2'$; a segunda linha apresenta os valores referentes às médias das diferenças percentuais do tempo consumido por $\mathcal{F}2$ e $\mathcal{F}2'$ para obter os ótimos duais e a terceira linha apresenta as médias das diferenças percentuais entre o total de iterações realizadas por $\mathcal{F}2$ e $\mathcal{F}2'$ até a obtenção dos ótimos duais. Nas colunas GI e GIII foi computada a média da diferença percentual de 48 instâncias (15 a 60 vértices), enquanto nas colunas GII e GIV de 12 instâncias (100 a 300 vértices). Analisando os resultados apresentados na referida tabela, verifica-se que, mesmo pequena, houve melhoria na qualidade dos ótimos duais (MDif% Ótimo Dual) nos quatro grupos de instâncias com a inserção das desigualdades (4.29) e (4.30). O aumento do tempo total (MDif% Tempo) consumido por $\mathcal{F}2'$ para obter os ótimos duais para as instâncias de maiores dimensões (GII e GIV) foi, em média, 200 vezes maior do que o tempo total gasto por $\mathcal{F}2$ para as mesmas instâncias. Quando uma análise similar é realizada para os grupos de instâncias de menores dimensões (GI e GIII), verifica-se que o aumento médio do tempo total consumido por $\mathcal{F}2'$ é de aproximadamente 37%. Em relação a média do total de iterações (MDif% Iterações) que $\mathcal{F}2'$ realizou para alcançar os ótimos duais de cada instância, destaca-se um aumento percentual acima de 40% para as instâncias de GII e GIV, enquanto para GI e GIII são registrados, respectivamente, 6,93% e 16,64% de aumento em relação a quantidade de iterações realizadas por $\mathcal{F}2$.

As desigualdades válidas sugeridas em (MANIEZZO et al., 1999) para $\mathcal{F}2'$ (equações (4.29) e (4.30)) não foram adaptadas para $\mathcal{F}3$, uma vez que já estão contidas na restrição (4.23) desta formulação.

A partir dos resultados apresentados nesta seção, constatou-se experimentalmente que os limites duais obtidos por $\mathcal{F}2$ são, em média, 19,6% melhores do que os obtidos por $\mathcal{F}3$ quando são consideradas as instâncias associadas ao PRRG e 17,4% melhores para as instâncias do PRR. Quando os ótimos duais de $\mathcal{F}2'$ são comparados aos obtidos por $\mathcal{F}3$, esta diferença aumenta para cerca de 20% nas instâncias do PRRG e para cerca de 17,62% nas instâncias do PRR.

Apesar do tempo consumido por $\mathcal{F}2'$ na obtenção dos ótimos duais das 120 instâncias consideradas nos experimentos ter sido em média maior do que o tempo consumido por $\mathcal{F}2$, houve melhora na qualidade dos ótimos duais obtidos por $\mathcal{F}2'$ quando comparados aos obtidos por $\mathcal{F}2$. Portanto, $\mathcal{F}2'$ será utilizada na obtenção dos limites primais para as instâncias de ambos os problemas em estudos nesta tese.

É importante ressaltar que, assim como mencionado no início desta seção, a formulação $\mathcal{F}2$ foi utilizada para modelar o PRRG sem alterar nenhuma de suas restrições, já que optou-se por promover alterações no grafo de entrada deste problema como sugere a Seção 4.1.3.

Antes de realizar os experimentos para a obtenção dos limites primais para as instâncias do PRRG e do PRR, foram realizados alguns testes para ajustar os parâmetros do CPLEX. Os resultados destes testes são apresentados na seção a seguir.

4.3 Experimentos computacionais para o ajuste dos parâmetros do CPLEX

O CPLEX é um *software* comercial utilizado para resolver problemas de programação linear (CPLEX, 2009). O algoritmo *branch-and-cut* é usado por este *software* para resolver problemas de programação inteira e pode ter seus parâmetros configurados visando a melhoria do seu desempenho na solução de um problema específico. O ajuste dos parâmetros do *branch-and-cut* pode ser feito de forma automática por heurísticas inerentes ao *software* ou por escolhas determinadas pelo usuário.

O objetivo desta seção é identificar a melhor configuração dos parâmetros do algoritmo *branch-and-cut* que será utilizado pelo CPLEX na solução do PRRG e do PRR.

Nos experimentos apresentados a seguir foi utilizada a formulação $\mathcal{F}2'$ selecionada na Seção 4.2 e um subconjunto das instâncias descritas no Apêndice A. Foram consideradas todas as instâncias de menores dimensões (48 instâncias de GI e 48 instâncias de GIII), mais as instâncias com o total de 100 vértices (quatro instâncias de GII e quatro instâncias de GIV), totalizando 104 instâncias. As regras de redução que compõem os conjuntos propostos nas Seções 3.1.2 e 3.2.2 foram aplicadas previamente em todas as 104 instâncias utilizadas.

Os seguintes parâmetros do CPLEX foram avaliados nos experimentos reportados a seguir: *nodeselect, variableselect, branch* e *startalgorithm*. Todas as variações destes parâmetros foram comparadas com a versão *default*, cujo ajuste dos parâmetros é feito de forma automática pelo próprio *software* (somente um parâmetro foi variado de cada vez, ficando os demais com o valor *default*). As métricas utilizadas para realizar as comparações foram:

- *Tempo*: Equivale a média do tempo total de CPU (em segundos) consumido pela parametrização p do CPLEX para cada uma das instâncias até provar a otimalidade da solução ou até atingir o tempo limite de 172.800 segundos;
- Gap%: Para cada instância, a métrica Gap% equivale a diferença percentual entre o limite superior e o limite inferior obtidos pela parametrização p do CPLEX com a formulação considerada;
- *MGap%*: É a média dos valores obtidos para *Gap%* sobre todas as instâncias consideradas em um determinado experimento;

- NScore²: Para cada instância do problema, o valor NScore de uma parametrização p do CPLEX é dado pelo número das demais parametrizações que encontraram uma solução melhor do que a melhor solução obtida com a parametrização p;
- Score: Para cada parametrização p do CPLEX, o valor de Score equivale a soma dos valores NScore obtidos pelo CPLEX para cada instância com o parâmetro p.

O critério de parada adotado foi a prova da otimalidade da solução ou o tempo limite de 172.800 segundos.

A seguir são brevemente descritos cada um dos parâmetros do CPLEX que foram avaliados nos experimentos reportados nesta seção, assim como são apresentados os resultados obtidos nos referidos experimetos. Para mais detalhes sobre algum dos parâmetros abordados, sugere-se uma leitura complementar no manual do CPLEX 11.2 (CPLEX, 2009). Não é conhecida nenhuma outra fonte de consulta confiável para estas informações.

4.3.1 Parâmetro Nodeselect

O parâmetro *nodeselect* é responsável por determinar o critério que será utilizado para a escolha do vértice a ser explorado na árvore de *branch*. Os valores que podem ser assumidos por este parâmetro são:

- **Nodeselect** = **0** Adota a estratégia *Depth-first search*, fazendo com que o vértice criado mais recentemente seja explorado;
- Nodeselect = 1 Adota a estratégia Best-bound search, que explora o vértice que está ativo e fornece o melhor valor para a função objetivo. Esta estratégia é considerada default pelo CPLEX quando nenhum outro valor é especificado pelo usuário;
- Nodeselect = 2 Adota a estratégia Best-estimate search, que elege o vértice com a melhor estimativa do valor inteiro da função objetivo. Esta configuração pode ser útil em casos onde há grande dificuldade em encontrar soluções viáveis para o problema em questão ou nos casos onde a prova da otimalidade não é o fator crucial;
- Nodeselect = 3 Adota uma alternativa à estratégia *Best-estimate search*.

A Tabela 4.5 apresenta os resultados obtidos pelo CPLEX nos experimentos realizados para a escolha mais adequada do parâmetro *nodeselect* na solução do PRR e do PRRG.

² Esta métrica é uma adaptação da métrica definida por (RIBEIRO; UCHOA; WERNECK, 2002).

Foram consideradas todas as instâncias de GI e GIII e somente as instâncias de 100 vértices de GII e GIV.

			Nodes	select	
		1 (default)	0	2	3
	Tempo	11,42	8,05	11,30	$20,\!51$
\mathbf{GI}	Score	0	0	0	0
	$\mathrm{MGap}\%$	$^{0,01\%}$	$0,\!01\%$	$0,\!01\%$	$0,\!01\%$
	Tempo	$7.664,\!05$	$9.915,\!88$	$64.856,\!80$	$9.134,\!97$
\mathbf{GII}	Score	0	0	0	0
	$\mathrm{MGap}\%$	$^{0,01\%}$	$0,\!01\%$	$0,\!00\%$	$0,\!01\%$
	Tempo	$0,\!35$	$0,\!26$	0,31	$0,\!31$
GIII	Score	0	0	0	0
	$\mathrm{MGap}\%$	$^{0,01\%}$	$0,\!01\%$	$0,\!01\%$	$0,\!01\%$
	Tempo	$30,\!91$	17,94	43.218,47	$14,\!29$
\mathbf{GIV}	Score	0	0	0	0
	$\mathrm{MGap}\%$	$^{0,01\%}$	0,01%	$0,\!00\%$	$0,\!01\%$

Tabela 4.5: Resultados dos experimentos com o parâmetro nodeselect

Como pode ser observado nos resultados apresentados na Tabela 4.5 para as quatro variações do parâmetro *nodeselect*, foi provada a otimalidade das soluções obtidas para todas as 104 instâncias consideradas nos experimentos (métricas MGap% e Score). A opção *default* obteve na média o menor tempo de processamento (1.926,68 segundos). É interessante destacar que quando o parâmetro *nodeselect* assume o valor 3, o desempenho médio obtido com as 104 instâncias é consideravelmente próximo da estratégia *default* (média do tempo total é de 2.292,52 segundos). A estratégia *Best-estimate search* (*nodeselect=2*) apresentou um tempo médio de CPU muito maior do que as demais estratégias quando observados os grupos de instâncias de maiores dimensões (GII/PRRG e GIV/PRR).

4.3.2 Parâmetro Variableselect

O parâmetro *variableselect* determina como será feita a escolha da variável fracionária sobre a qual será realizada a ramificação na árvore. Esta escolha é feita após a seleção do vértice que será explorado (*nodeselect*). São cinco os valores possíveis que podem ser configurados pelo usuário para este parâmetro:

Variableselect = -1 Utiliza a regra inviabilidade mínima, que escolhe a variável com a menor parte fracionária. Esta opção pode levar a uma escolha mais rápida de uma primeira solução inteira viável, mas será geralmente mais lenta para atingir a solução inteira ótima;

- Variableselect = 0 Corresponde a estratégia *default*, onde a escolha é feita por uma heurística interna com base nas características do problema e no seu progresso.
- Variableselect = 1 Considera a regra inviabilidade máxima, que escolhe a variável com a maior parte fracionária;
- Variableselect = 2 Utiliza a regra pseudo-custo, que estima o impacto no custo da solução causado pela transformação de cada variável fracionária em variável inteira. Estas informações são usadas em uma heurística interna para a escolha da variável;
- Variableselect = 3 Considera a regra da ramificação forte, que investe esforços consideráveis na análise dos ramos em potencial na esperança de reduzir drasticamente o número de vértices que será explorado na árvore. Esta opção exige mais tempo de processamento para cada vértice, mas geralmente acaba investigando menos vértices para resolver o problema. Essa estratégia funciona particularmente bem nos problemas onde o número de variáveis binárias é significativamente maior do que o número de linhas. Também é útil quando a memória é limitada: a criação de menos vértices requer menos memória;
- Variableselect = 4 Considera a regra pseudo-redução de custos, que é uma variação computacionalmente menos intensiva do que adotada pela regra pseudo-custo (Va-riableselect=2).

A Tabela 4.6 apresenta os resultados dos experimentos para a avaliação do parâmetro variables elect na solução de 104 instâncias do PRR e do PRRG.

				Variables	elect		
		0 (default)	-1	1	2	3	4
	Tempo	11,42	8.688,15	$7.569,\!52$	11,36	34,63	$31,\!48$
\mathbf{GI}	Score	0	0	0	0	0	0
	$\mathrm{MGap}\%$	$0,\!01\%$	0,37%	$0,\!87\%$	$0,\!01\%$	0,01%	$0,\!01\%$
	Tempo	$7.664,\!05$	$172.802,\!81$	$172.802,\!63$	7.667,24	$22.695,\!32$	$45.536,\!34$
\mathbf{GII}	Score	0	5	0	0	0	0
	$\mathrm{MGap}\%$	$0,\!01\%$	5,50%	$5{,}13\%$	$0,\!01\%$	0,01%	0,56%
	Tempo	$0,\!35$	$121,\!92$	124,92	0,35	0,42	$0,\!20$
GIII	Score	0	0	0	0	0	0
	$\mathrm{MGap}\%$	$0,\!01\%$	$0,\!01\%$	$0,\!01\%$	$0,\!01\%$	0,01%	$0,\!01\%$
	Tempo	$30,\!91$	$86.557,\!27$	$86.434,\!65$	30,75	28,74	14,75
\mathbf{GIV}	Score	0	0	0	0	0	0
	MGap%	$^{0,01\%}$	$1,\!15\%$	1,36%	$0,\!01\%$	0,01%	$0,\!01\%$

Tabela 4.6: Resultados dos experimentos com o parâmetro variables elect

A partir dos resultados experimentais relacionados às seis variações do parâmetro *variableselect*, nota-se que quando a escolha da variável fracionária sobre a qual será realizada a ramificação na árvore é feita utilizando as regras de *inviabilidade mínima* (*variableselect=-1*) ou *inviabilidade máxima* (*variableselect=1*), o CPLEX não consegue provar a otimalidade da solução para todas as instâncias de GI, de GII e de GIV (métricas MGap%). Com exceção de uma das quatro instâncias de GII, onde a solução ótima não foi alcançada (vide Score=5), todos os demais casos onde a métrica MGap% é maior do que 0,01%, as soluções ótimas foram encontradas, mas a otimalidade não foi provada devido ao limite de tempo estabelecido como um dos critério de parada (172.800 segundo). Em relação ao tempo total de processamento, novamente a parametrização *default* consegue provar a otimalidade das 104 instâncias com a menor média de tempo de CPU (1.926,68 segundos). A segunda melhor alternativa para o parâmetro testado nesta seção é a opção 2, que também consegue provar a otimalidade das 104 instâncias com uma média de tempo de 1.927,42 segundos.

4.3.3 Parâmetro Branch

Após selecionar a variável para realizar a ramificação na árvore (*parâmetro variablese-lect*), é preciso determinar a direção na qual esta ramificação será realizada. O parâmetro do CPLEX que determina esta direção é o *branch*, que pode assumir os seguintes valores:

Branch = -1 Determina que a ramificação da árvore seja realizada para baixo;

Branch = 0 Este é o valor (default), onde o CPLEX decide automaticamente o sentido de cada ramificação com base nas características do problema;

Branch = 1 Determina que a ramificação da árvore seja realizada para cima.

Os resultados obtidos pelo CPLEX nos experimentos realizados para o ajuste do parâmetro *branch* são apresentados na Tabela 4.7.

			Branch	
		0 (default)	1	-1
	Tempo	11,42	$11,\!50$	$15,\!87$
\mathbf{GI}	Score	0	0	0
	$\mathrm{MGap}\%$	$^{0,01\%}$	$0{,}01\%$	$0,\!01\%$
	Tempo	$7.664,\!05$	$4.557,\!12$	$6.260,\!55$
\mathbf{GII}	Score	0	0	0
	$\mathrm{MGap}\%$	$0,\!01\%$	$0{,}01\%$	$0,\!01\%$
	Tempo	$0,\!35$	$0,\!34$	$0,\!36$
\mathbf{GIII}	\mathbf{Score}	0	0	0
	$\mathrm{MGap}\%$	$0,\!01\%$	$0,\!00\%$	$0,\!00\%$
	Tempo	$30,\!91$	12,23	$25,\!54$
\mathbf{GIV}	\mathbf{Score}	0	0	0
	$\mathrm{MGap}\%$	$^{0,01\%}$	$0,\!01\%$	$0,\!01\%$

Tabela 4.7: Resultados dos experimentos com o parâmetro branch

Analisando os resultados apresentados na Tabela 4.7, nota-se que todas as versões do CPLEX associadas a variação parâmetro *branch* obtiveram desempenho médio semelhantes. Em relação ao tempo de processamento, quando *branch*=1, obteve-se 1.145,30 segundos como média do tempo total de CPU, enquanto para a opção *default* e para *branch*=-1, obteve-se 1.926,68 e 1.575,58 segundos respectivamente. Portanto, a opção *branch*=1 (ramificação da árvore realizada para cima) se destacou sobre as demais.

4.3.4 Parâmetro Startalgorithm

Este parâmetro especifica a diretriz do algoritmo que será aplicado pelo CPLEX para resolver a relaxação inicial. A escolha dos valores para este parâmetro pode ser feita da seguinte forma:

- Startalgorithm = 0 A escolha do algoritmo que será aplicado é feita por heurísticas internas ao CPLEX;
- Startalgorithm = 1 Utiliza o algoritmo *Primal Simplex*;
- Startalgorithm = 2 Corresponde à escolha default do CPLEX. Neste caso, o algoritmo Dual Simplex é usado para resolver a relaxação inicial;

Startalgorithm = **3** Aplica o algoritmo *Network Simplex*;

Startalgorithm = 4 O algoritmo utilizado é o *Barrier with Crossover*;

Startalgorithm = 5 Utiliza o algoritmo Sifting;

Startalgorithm = 6 Utiliza de forma concorrente os algoritmos Dual Simplex, Barrier e Primal Simplex. Esta opção é permitida na raiz, mas não nos demais vértices.

A Tabela 4.8 apresenta os resultados obtidos pelo CPLEX nos experimentos que determinam a escolha mais adequada do parâmetro *startalgorithm* quando o PRR e o PRRG estão sendo modelados com a formulação $\mathcal{F}2'$. Neste experimento em particular, foram consideradas somente as instâncias com 100 vértices de GII e GIV. As métricas MGap% e Score apresentadas na referida tabela mostram que, em média, todas as sete versões do CPLEX encontraram as soluções ótimas para as 8 instâncias. Entretanto, a versão que se destacou sobre as demais foi aquela que utilizou o algoritmo *Sifting (startalgorithm=*5), consumindo em média 2.744,23 segundos de CPU. A versão automática (*startalgorithm=*0), que realiza a escolha do algoritmo que será usado para resolver a relaxação incial por heurísticas internas ao CPLEX, consumiu em média 3.847,48 segundos, sendo a segunda melhor opção para o referido parâmetro.

A partir das análises realizadas para todos os parâmetros do algoritmo branch-andcut do CPLEX testados na Seção 4.3, conclui-se que, quando considerada a formulação $\mathcal{F}2'$ aplicada às 104 instâncias utilizadas nos experimentos aqui reportados, a melhor configuração para estes parâmetros é: nodeselect=1 (estratégia default - Best-bound search); variableselect=0 (default - heurística interna ao CPLEX); branch=1 (ramificação da árvore para cima) e startalgorithm=5 (algoritmo Sifting).

A Seção 4.3.5 apresenta os resultados experimentais obtidos com o CPLEX considerando a parametrização determinada nesta seção. As soluções obtidas serão utilizadas na avaliação das heurísticas propostas no Capítulo 5 deste trabalho.

				Startalo	rorit.hm			
		0		2 (default)	3	4	5	9
	Tempo	7.664,05	9.306, 79	7.728,56	23.097, 31	21.685,79	5.462,72	7.727,95
GII	\mathbf{Score}	0	0	0	0	0	0	0
	MGap%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%
	Tempo	30,91	58,24	30,74	59,54	11,78	25,74	31,11
GIV	\mathbf{Score}	0	0	0	0	0	0	0
	MGap%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%	0,01%

Tabela 4.8: Resultados dos experimentos com o parâmetro startalgorithm

4.3.5 Resultados obtidos com a formulação $\mathcal{F}2'$

Sabe-se que uma das formas de provar a otimalidade de uma solução x para um determinado problema \mathcal{P} é encontrar uma solução x' cujo valor sirva de limite inferior $(custo(x') \leq custo(x))$ e uma solução x'' cujo valor sirva de limite superior $(custo(x'') \geq$ custo(x)), tal que custo(x') = custo(x''). Desta forma, pode-se concluir que x é uma solução ótima para \mathcal{P} . Geralmente, os algoritmos produzem uma sequência de limites superiores $(custo(x1'') > custo(x2'') > ... > custo(xp'') \ge custo(x))$ e inferiores $(custo(x1') < custo(x2') < \dots < custo(xq') \leq custo(x))$ de forma que, dado um $\varepsilon \geq 0$, pode-se definir como um critério de parada para o algoritmo a obtenção de soluções xp''e xq' que satisfazem a condição $|custo(xp'') - custo(xq')| \leq \varepsilon$. Tomando este critério de parada, consegue-se garantir que o algoritmo só termine quando o custo da solução candidata x for no máximo ε inferior a solução ótima. Analogamente, o CPLEX considera por default que a otimalidade de uma determinada solução está provada quando são obtidos limites superiores e inferiores cuja diferença percentual é de até 0,01%. Com base nestas informações, as tabelas a seguir relacionam para as instâncias com até 100 vértices: o custo da solução ótima obtida, a diferença percentual entre os limites superiores e inferiores (Gap) e o tempo de CPU (em segundos) consumido no processamento das restrições associadas ao modelo $\mathcal{F}2'$ para provar a otimalidade das soluções de 52 instâncias do PRRG e de 52 instâncias do PRR.

Todos os experimentos reportados nesta seção foram realizados com a parametrização algoritmo branch-and-cut utilizado pelo CPLEX conforme determinado na Seção 4.3, a saber: nodeselect=1 (estratégia default - Best-bound search); variableselect=0 (default heurística interna ao CPLEX); branch=-1 (ramificação para baixo) e startalgorithm=5 (algoritmo Sifting). O tempo de processamento foi limitado a 172.800 segundos (set timelimit 172.800). O conjunto de regras de redução proposto na Seção 3.2.2 foi aplicado previamente nas instâncias do PRRG (GI e GII), assim como o conjunto proposto na Seção 3.1.2 foi aplicado nas instâncias do PRRG (GIII e GIV). Os percentuais de redução de cada um dos conjuntos de vértices que compõem as instâncias do PRRG e do PRR consideradas nos experimentos com o CPLEX podem ser vistos nas Seções 3.2.3 e 3.2.3 respectivamente. Para as instâncias de menores dimensões (GI e GIII), foram confrontados os resultados obtidos com o CPLEX para as instâncias originais e reduzidas, a fim de avaliar o impacto das reduções na obtenção das soluções ótimas.

As Tabelas 4.9 e 4.10 apresentam os resultados obtidos com $\mathcal{F}2'$ para 48 instâncias do PRRG e 48 instâncias do PRR respectivamente.

GI	Custo ótimo	0	riginal	Redu	ızido
		Gap	Tempo	Gap	Tempo
gctp15_3-3-9	961.154		0.15		0.04
gctp15_3-9-3	1.036.409		0.11		0.04
gctp15_4-4-7	1.048.315		0.12		0.03
gctp15_4-6-5	1.466.511		0.27		0.12
gctp15_7-7-1	1.421.948		0.79		0.04
gctp15_9-3-3	1.795.582		0.08		0.01
gctp20_10-10-0	1.564.006		2.76		0.07
gctp20_12-4-4	2.132.347		0.14		0.01
$gctp20_4-12-4$	1.294.603		2.02		0.16
gctp20_4-4-12	1.708.966		1.57		0.01
gctp20_6-6-8	1.491.061		1.37		0.06
gctp20_6-8-6	1.525.053		2.85		0.02
gctp25_12-12-1	1 829 687		1.80		0.06
gctp25_15-5-5	1 693 310		0.69		0.01
getp25_5-15-5	1 290 023		0.92		0.09
getp25_5-5-15	978 586		2 15	_	0.20
gctp25_7-10-8	1 661 147		4 81	_	0.18
getp25_7_10_0	1.427431		1 74		
getp20_0.0.5	$1.755\ 605$		0.07		0.03
getp30_18-6-6	2104277		3.07		0,00
getp30_10-0-0	1.707.545	0.01%	5,76		0,20
getp30_6-6-18	1.506.710	0,0170	2 98		0,20
$gctp30_0 0_{10}$	1 654 180		6.51		0,10
$gctp30_9-12-3$ $gctp30_9-0_12$	1.686.404	0.01%	7 36		0,04
$gctp35_10_14_11$	1.066.695	0.01%	75.37	0.01%	6.75
$gctp35_10-14-11$	1.900.099	0,01%	22.17	0,0170	0,75
gctp35_17_17_1	2 054 434	0.01%	8.00		0,05
$gctp35_11-17-1$ $gctp35_21-7-7$	2.004.454	0,01%	14.94		
$gctp35_2117$	1 671 519	0.01%	8 31		0.40
$gctp35_7_211$	1 348 038	0,0170	3 65		0,40
gctp30 = 1721 gctp40 = 12-16-12	2.075.742	0.01%	160.44	0.01%	6.62
$gctp40_12 10 12$ $gctp40_13_13_14$	1 978 318	0,01%	14 75	0,0170	0,02
gctp40_10-10-14	2 498 674	0.01%	33.20		1.28
gctp40_20-20-0	2.490.014	0.01%	75.45	0.01%	0.81
gctp40=24-0-0 gctp40=8-24-8	1.771.596	0,0170	44.26	0,0170	0.14
gctp40_0 24 0	2 032 735	0,0070	2 604 63		0,14
gctp40=0-0-24 gctp50=10-10-30	1 585 409	0.01%	5 025 78		0,30
gctp50_10-30-10	1.000.405 1.814.287	0,01%	7 481 65		0,20
gctp50_15-20-15	2 160 737	0.01%	1 429 00	0.01%	3 34
gctp50_16-16-18	2.100.101 2.047.142	0,01%	62.07	0,0170	0.70
gctp50_10-10-10	2.047.142	0.01%	42.37		1 09
gctp50_20-20-0	2.170.000	0.01%	3 100 05	0.01%	17 10
$\frac{60000-30-10-10}{9000-30-10-10}$	2,004,400	2.45%	172 801 64	0.01%	108.10
gctp60 = 12 - 12 - 30	2.224.473	1 60%	172.801,04	0.01%	374 47
$30000 \pm 12-30-12$	2.270.014	0.01%	18 891 79	0.01%	2 08
gctn60 10-24-10	2.015.100	0.01%	1 280 60	0,0170	2,50
$\frac{19-19-22}{9}$	2.130.040 2.761.327	0.01%	399.43		1 5,54 4 74
actn60 36-12-12	2 750 783	0.01%	1 031 20	0.01%	11 10
500P00_00-12-12	2.100.100		1.001,23	0,0170	11.10
Médi	a	0,17%	8.069,28	0,01%	11,42

Tabela 4.9: Custo das soluções ótimas obtidas com a formulação $\mathcal{F}2'$ para o PRRG - GI

Legenda

O valor do limite superior é igual ao valor do limite inferior.

GIII	Custo ótimo	Or	iginal	Redi	ızido
		Gap	Tempo	Gap	Tempo
ctp15_3-3-9	1.002.222		0.08		0.00
ctp15_3-9-3	1.138.939		0.01		0.00
$ctp15_4-4-7$	1.196.129		0.01		0,00
ctp15_4-6-5	1.221.377		0.01		0.00
$ctp15_7-7-1$	1.153.884		0.00		0.00
ctp15_9-3-3	1.316.581		0.08		0.04
ctp20_10-10-0	1.251.833		0.03		0.01
ctp20_12-4-4	1.680.738	0.00%	0.47		0.08
ctp20_4-12-4	936.184		0.03		0.00
ctp20_4-4-12	1.153.040		0.42		0.00
ctp20_6-6-8	1.295.576		0.28		0.00
ctp20_6-8-6	1.262.473		0.10		0.01
ctp25_12-12-1	1.680.738		0.06		0.09
ctp25_15-5-5	1.764.081		0.61		0.11
ctp25_5-15-5	1.166.123		0.02		0.00
$ctp25_5-5-15$	1.223.360		0.54		0.02
$ctp25_7-10-8$	1.367.025		0.20		0.00
ctp25_8-8-9	1.379.150		0.78		0.00
ctp30_15-15-0	1.764.081		0.03		0.10
ctp30_18-6-6	2.023.833	0.01%	2.93		0.32
ctp30_6-18-6	1.306.845		0.04		0.00
ctp30_6-6-18	1.347.522		3,47		0,00
ctp30_9-12-9	1.379.599		0,52		0,04
ctp30_9-9-12	1.379.599		1,37		0.04
ctp35_10-14-11	1.810.009		0,86		0,01
ctp35_11-11-13	1.649.229		1,88		0,08
ctp35_17-17-1	1.771.975		0,16		$0,\!12$
$ctp35_21-7-7$	2.123.301	—	1,06		0,38
$ctp35_7-21-7$	1.662.568		0,30		0,01
$ctp35_77-21$	1.566.645	—	4,05		0,03
$ctp40_{-}12-16-12$	1.949.544	0,01%	3,94		$0,\!15$
$ctp40_{13-13-14}$	2.055.310	0,01%	4,59		0,35
$ctp40_20-20-0$	1.824.574		0,09		0,31
$ctp40_{-}24-8-8$	2.244.183	0,01%	3,52		$0,\!37$
$ctp40_{-}8-24-8$	1.393.676		0,40		0,00
$ctp40_88-24$	1.653.663	0,00%	8,61		0,11
$ctp50_{-}10_{-}10_{-}30$	1.656.656	0,01%	$199,\!12$		$0,\!17$
$ctp50_{-}10_{-}30_{-}10$	1.251.833		$0,\!57$		0,01
$ctp50_{-}15$ -20-15	1.937.583	0,01%	17,88		$0,\!19$
$ctp50_{-}16-16-18$	1.944.648	0,01%	$38,\!10$		$0,\!23$
$ctp50_{25-25-0}$	2.244.596		0,18		0,49
$ctp50_{-}30_{-}10_{-}10$	2.391.136	0,01%	$167,\!82$		2,17
$ctp60_{-}12-12-36$	1.949.532	0,01%	$40.242,\!36$		$0,\!30$
$ctp60_12$ -36-12	1.680.738	0,01%	$3,\!30$		0,09
$ctp60_{18-24-18}$	1.885.042	0,01%	$227,\!54$		0,33
$\mathrm{ctp}60_19\text{-}19\text{-}22$	1.890.330	0,01%	$3.992,\!89$		0,36
$ctp60_{-}30_{-}30_{-}0$	2.391.136	0,00%	2,01	—	$2,\!14$
$ctp60_{-}36-12-12$	2.548.788	0,01%	792,81	0,01%	7,38
Méd	ia	0.01%	792.81	0.01%	7.38

Tabela 4.10: Custo das soluções ótimas obtidas com a formulação $\mathcal{F}2'$ para o PRR - GIII

Legenda

O valor do limite superior é igual ao valor do limite inferior.

Pode-se constatar nos resultados apresentados na Tabela 4.9 que para as instâncias gctp60_12-12-36 e gctp60_12-36-12 não foi possível provar a otimalidade das soluções obtidas em até 172.800 segundos de processamento. Entretanto, quando estas instâncias são reduzidas, a otimalidade de ambas as soluções é provada em um tempo médio de 241,28 segundos. Outra constatação importante é que o tempo médio para processar as 48 instâncias é aproximadamente 700 vezes maior para as instâncias originais quando comparadas às reduzidas.

Em relação aos resultados obtidos para as instâncias associadas ao PRR que são apresentados na Tabela 4.10, observa-se que a otimalidade de todas as soluções foram provadas, tanto para as instâncias originais quanto para as reduzidas. Porém, o tempo médio de processamento foi aproximadamente 107 maior para as instâncias originais.

A Tabela 4.11 apresenta os resultados dos experimentos com o CPLEX para as instâncias do PRRG e do PRR com 100 vértices.

Tabela 4.11: Custo das soluções ótimas obtidas com $\mathcal{F}2'$ para instâncias de GII e GIV

GII reduzido	Custo	Gap	Tempo	GIV reduzido	Custo	Gap	Tempo
gctp100_20-60-20	2.677.859	0,01%	27.630,79	ctp100_20-60-20	2.232.584		0,22
gctp100_33-33-34	2.822.284	0,01%	$323,\!18$	ctp100_33-33-34	2.503.833	$0,\!01\%$	21,60
$gctp100_{50-50-0}$	3.465.968	0,01%	834,83	$ctp100_{-}50_{-}50_{-}0$	2.737.137	0,00%	10,35
$gctp100_{60-20-20}$	3.577.315	0,01%	$1.867,\!40$	$ctp100_{-}60_{-}20_{-}20_{-}20_{-}$	3.256.940	$0,\!01\%$	91,46
Média	-	0,01%	1.867,40	Média	-	0,01%	30,91
Tananda							

Legenda

Foi provada a otimalidade das soluções obtidas para as 8 instâncias apresentadas na Tabela 4.11. Vale destacar que o tempo necessário para provar a otimalidade das soluções obtidas para as instâncias associadas ao PRRG é, em média, 60 vezes superior ao tempo consumido para as do PRR.

O Apêndice C apresenta resultados computacionais com a formulação $\mathcal{F}2'$ que são complementares aos discutidos neste capítulo. Os resultados reportados no referido apêndice ressaltam o benefício da aplicação dos conjuntos de regras de redução propostos neste trabalho nos grafos associados ao PRRG e ao PRR. São mostrados os resultados obtidos a partir da relaxação linear de $\mathcal{F}2'$, destacando a diferença percentual entre os ótimos duais obtidos com os grafos originais em relação aos ótimos duais obtidos com os grafos reduzidos (Tabelas C.2 e C.1). Os valores das diferenças percentuais entre os ótimos primais e duais, tanto para os grafos reduzidos quanto para os grafos originais são apresentados nas Tabelas C.4 e C.3.

O valor do limite superior é igual ao valor do limite inferior.

4.4 Algumas relações entre o PRR e o PRRG

Esta seção apresenta como contribuições alguns resultados teóricos que visam estabelecer uma relação entre os problemas que são objetos de estudo neste trabalho. No que segue, são enunciados e provados um teorema e dois corolários decorrentes deste.

Teorema 1: Seja s uma solução viável para o PRR, então s também é viável para o PRRG.

Por hipótese, s é uma solução viável do PRR, então s é formada sobre um subconjunto de vértices de V, contém todos os vértices de T e cobre todos os vértices de W. Desta forma, como $V \subset (V \cup W)$, s também é solução do PRRG.

Pelo Teorema 1, sabe-se toda solução viável do PRR é também viável para o PRRG. Além disto, sabe-se que todas as soluções viáveis do PRRG que contém vértices de Wnão são viáveis para o PRR. Logo, pode-se representar o espaço de soluções de ambos problemas em G como mostra a Figura 4.1.



Figura 4.1: Espaço de soluções do PRR e do PRRG

Os seguintes resultados são decorrentes do Teorema 1:

Corolário 1.1: Se $s^{*'}$ é uma solução ótima do PRRG e $s^{*'}$ não contém vértices de W, então $s^{*'}$ também é uma solução ótima do PRR.

Como $s^{*'}$ não contém vértices de W, ou seja, $s^{*'}$ é uma rota formada sobre um subconjunto de vértices de V que contém todos os vértices de T e que cobre todos os vértices de W, então $s^{*'}$ é, por definição, uma solução víável do PRR. Por outro lado, se $s^{*'}$ não fosse ótima para o PRR, esta também não o seria para o PRRG.

Corolário 1.2: Dado s^* uma solução ótima do PRR e dado $s^{*'}$ uma solução ótima do PRRG, então custo $(s^{*'}) \leq custo(s^{*})$.

De fato, s^* somente difere de $s^{*'}$ por:

i) possuir um conjunto de vértices de V diferente dos contidos em $s^\ast,$ ou

ii) possuir algum vértice de W

Note que i) acontece se, e somente se, $custo(s^{*'}) = custo(s^{*})$. Se ii) acontece, então $custo(s^{*'}) \leq custo(s^{*})$, pois se $custo(s^{*'}) > custo(s^{*})$, então $s^{*'}$ não seria ótima, já que pelo *Teorema 1* toda solução viável do *PRR* é também viável para o *PRRG*.

Capítulo 5

Abordagens heurísticas propostas para o PRR e para o PRRG

O grande avanço tecnológico observado na última década, tanto em *hardware* quanto em *software*, tem possibilitado muitos progressos na solução de problemas de otimização combinatória de elevada complexidade computacional. Entretanto, mesmo com computadores capazes de processar até quadrilhões de instruções por segundo (TOP500, 2009) e com *software* que dispõem de heurísticas elaboradas capazes de auto ajustar seus parâmetros para se adequar à natureza do problema abordado (CPLEX, 2009), ainda tem sido um desafio obter soluções de qualidade e com um esforço computacional aceitável para muitos problemas de otimização associados a aplicações reais. Problemas desta natureza são, na sua maioria, caracterizados como NP-difíceis e, portanto, sabe-se que o uso exclusivo de métodos exatos usualmente limita-se à solução de instâncias de pequeno porte, fato que certamente não traduz grande parte das aplicações reais associadas a estes problemas. O caminho natural para contornar esta limitação é a utilização de métodos heurísticos, especialmente aqueles que possuem dispositivos que possibilitam escapar de ótimos locais ainda distantes de um ótimo global, combinados ou não com outras técnicas de otimização.

Este capítulo apresenta como contribuições algoritmos heurísticos propostos nesta tese para a solução aproximada do PRR e do PRRG, incluindo heurísticas de construção (Seção 5.1), busca local (Seção 5.2), algumas variações da meta-heurística GRASP (Seção 5.5), com e sem mecanismos de memória e, um algoritmo baseado da meta-heurística *Iterated Local Search* (Seção 5.6). Experimentos computacionais com as abordagens propostas são apresentados ao longo do capítulo. Testes estatísticos (ANOVA e *t-Student*) são realizados sobre os resultados obtidos a fim de avaliar o desempenho das heurísticas apresentadas.

5.1 Heurísticas de Construção

Esta seção apresenta as heurísticas construtivas propostas para gerar soluções iniciais para o PRR/PRRG. São propostos cinco procedimentos heurísticos, sendo: dois procedimentos gulosos aleatorizados (*AdicaoSucessiva* e *RemocaoSucessiva*) que são adaptações das heurísticas *SoluçãoInicial_ADD* e *SoluçãoInicial_DROP* descritas em (MOTTA, 2001), mais três novos procedimentos baseados em heurísticas construtivas clássicas do PCV (*IMP-PRR*, *IMD-PRR* e *IMB-PRR*). As seções de 5.1.1 a 5.1.5 abordam detalhadamente cada um destes procedimentos.

5.1.1 Heurística AdicaoSucessiva

A heurística construtiva *AdicaoSucessiva* consiste basicamente de um procedimento construtivo guloso aleatorizado que, a cada iteração, seleciona um único vértice aleatoriamente em uma lista restrita de candidatos e o insere na rota parcial. Este processo se repete até que a rota construída seja viável para o problema (PRR ou PRRG).

O Algoritmo 1 apresenta o pseudocódigo da heurística construtiva AdicaoSucessiva, que inicia a partir da leitura dos parâmetros de entrada (linha 1), que são:

- instancia: corresponde à cardinalidade dos conjuntos $T, W \in V \setminus T$; uma matriz de distâncias e uma matriz de cobertura;
- problema: como este trabalho busca soluções tanto para o PRR quanto para o PRRG, este parâmetro determina o problema que será resolvido;
- α: parâmetro que controla a restritividade da Lista Restrita de Candidatos (*LRC*), cujo valor varia no intervalo [0,1];
- *criterioInsercao*: critério que determina em que posição da rota atual os vértices selecionados na *LRC* serão inseridos;
- g(.): função gulosa que avalia o benefício de inserir cada vértice da lista de candidatos na rota atual.

O parâmetro *criterioInsercao*, que corresponde ao critério que será adotado para a inserção de um novo vértice na rota, pode assumir os seguintes valores: *econômico* ou *aleatório*. O primeiro critério escolhe um par de vértices (v_i, v_j) da rota atual cujo custo de inserir v_k entre v_i e v_j seja mínimo. O segundo critério escolhe aleatoriamente da rota atual o par de vértices (v_i, v_j) entre os quais v_k será inserido.

Algoritmo 1 Heurística construtiva: AdicaoSucessiva

1: $leDadosEntrada(instancia, problema, \alpha, g(.), criterioInsercao)$ 2: $Rota = \emptyset$; 3: $LRC = \emptyset;$ 4: LC = inicializaListaCandidatosAdd(instancia, problema, g(.));5: while (*Rota* inviável) do 6: $s_i = \min\{g(v_k) \mid v_k \in LC\};\$ $s_f = max\{g(v_k) \mid v_k \in LC\};$ 7: $LRC = \{ v_k \in LC \mid g(v_k) \le s_i + \alpha(s_f - s_i) \};$ 8: novoVertice = selectionaVerticeAleatoriamente(LRC);9: Rota = insereVerticeEscolhido(Rota, novoVertice, criterioInsercao);10: $LC = LC - \{novoVertice\};$ 11: 12:atualizaAvaliacaoAdd(LC, novoVertice, g(.));13: end while 14: removeVerticesSobressalentes(Rota); 15: return Rota;

A partir dos dados de entrada, a lista de candidatos (LC) é criada, associando a cada vértice v_k que a compõe, o benefício $g(v_k)$ de inserí-lo na rota atual (linha 4). Inicialmente, a LC contém todos os vértices de V se o problema em questão for o PRR ou todos os vértices de $V \cup W$ se o problema for o PRRG.

Para o procedimento AdicaoSucessiva, a função $g : \aleph \to \Re$ é definida da seguinte forma (fórmula 5.1):

$$g(v_k) = \begin{cases} |V \cup W| - \tau \times (\Psi + |W|), \ \forall v_k \in T; \\ |V \cup W| - (\tau \times \Psi), \ \forall v_k \in (V \setminus T) \cup W. \end{cases}$$
(5.1)

onde Ψ é o número de vértices de W que v_k cobre e $\tau > 0$.

De acordo com a fórmula 5.1, quanto menor for o valor associado ao vértice v_k pela função $g(v_k)$, maior o benefício associado à inserção deste vértice na rota atual.

Uma vez inicializada a lista de candidatos (linha 4), a cada iteração o algoritmo verifica o maior e o menor valor de $g(v_k)$, com $v_k \in LC$ (linhas 6 e 7) e, a partir destes valores, constrói a lista restrita de candidatos (linha 8). Assim como mencionado anteriormente, a restritividade da LRC é controlada pelo parâmetro de entrada $\alpha \in [0, 1]$. Quando $\alpha = 0$, a LRC é apresentada no seu modo mais restrito e somente os melhores benefícios serão considerados (menores valores de $g(v_k)$), isto é, preferencialmente os vértices de Tque mais cobrem vértices de W. O vértice escolhido aleatoriamente na LRC (linha 9) é inserido na rota atual a partir do critério de inserção passado como parâmetro (linha 10): econômico e aleatório. Após a inserção do novo vértice na rota atual, a LC é atualizada para refletir a última alteração realizada na rota atual (o vértice recém inserido em Rota é removido da LC (linha 11) e o valor de g(.) dos vértices atingidos pela inserção do novo vértice é recalculado (linha 12)).

Experimentos computacionais preliminares mostraram o benefício de se realizar um pós-processamento na rota construída pelo procedimento AdicaoSucessiva. Neste sentido, foi implementado um filtro para soluções construídas, que tem como objetivo remover os vértices que se tornaram sobressalentes¹ ao longo do processo de construção da solução inicial. O procedimento removeVerticesSobressalentes() que implementa o filtro (linha 14), inicia o processo de remoções sucessivas dos vértices sobressalentes da rota a partir da escolha aleatória de um vértice $v_i \in T$ e a partir deste v_i , percorre a rota tentando remover os vértices desnecessários à satisfação da condição de viabilidade do problema que está sendo resolvido (PRR ou PRRG). Alguns resultados encontrados na literatura incentivam a utilização de filtros para evitar que os procedimentos de busca local sejam aplicados às soluções de baixa qualidade.

A Figura 5.1 exemplifica a execução da heurística construtiva AdicaoSucessiva na obtenção de uma solução inicial para o PRR, considerando a instância representada na Figura 5.1A. Os parâmetros de entrada considerados na execução exemplificada são: criterioInsercao = econômico, $\alpha = 0,3$ e a função g(.) sendo definida pela equação 5.1 $com \tau = 1$. A Figura 5.1B mostra a rota parcial após a inserção de cada um dos vértices de T que, pelos valores atribuídos a τ e a α , foram os primeiros vértices escolhidos para inserção na seguinte ordem: 4 (com q(4) = 16), 2 (com q(2) = 16), 0 (com q(0) = 14), 5 (com g(5) = 16), 1 (com g(1) = 16) e 3 (com g(3) = 16). Na sequência (Figuras $5.1C, 5.1D \in 5.1E$), foram selecionados os vértices 22 (inserido entre os vértices 4 e 5), 17 (inserido entre os vértices 1 e 2) e 14 (inserido entre os vértices 17 e 2). Após a inserção do vértice 14, a rota torna-se viável, já que contém todos os vértices de T, os vértices 6, 7 e 8 estão cobertos pelo vértice 22 (o vértice 7 também é coberto pelo vértice 5), o vértice 10 é coberto pelos vértices 0 e 1, os vértices 9 e 11 são cobertos pelo vértice 17 e o vértice 12 é coberto pelo vértice 14. O procedimento removeVerticesSobressalentes() (linha 14) iniciou a varredura da rota a partir do vértice 3, mas não conseguiu remover nenhum vértice, pois caso alguma remoção ocorresse, a rota se tornaria inviável.

¹ Define-se vértices sobressalentes neste contexto como sendo todos os vértices v_i da rota atual que estão cobrindo algum vértice $v_j \in W$ que já se encontra coberto por um outro vértice v_k $(i \neq k)$ também presente na rota.



Figura 5.1: Passo a passo da construção de uma solução inicial para CTP a partir da heurística AdicaoSucessiva

5.1.2 Heurística RemocaoSucessiva

O procedimento *RemocaoSucessiva*, assim como o *AdicaoSucessiva*, é um procedimento construtivo guloso aleatorizado que, a cada iteração, constrói uma rota viável para o problema que está sendo resolvido (PRR/PRRG). Entretanto, ao contrário deste último procedimento apresentado, a *RemocaoSucessiva* parte de uma solução inicial contendo todos os vértices de V (quando o problema resolvido for o PRR) ou contendo todos os vértices de $V \cup W$ (quando o problema for o PRRG). A cada iteração, um vértice é selecionado para a remoção enquanto estas remoções não inviabilizarem a rota. A função $f : \aleph \to \Re$, que avalia o benefício de remover um vértice da rota, é definida pela fórmula 5.2:

$$f(v_k) = (\text{número de vértices de W que } v_k \text{ cobre, } v_k \in V \setminus T)$$
(5.2)

O Algoritmo 2 apresenta o pseudocódigo da heurística de construção RemocaoSucessiva, sendo considerados como dados de entrada do procedimento os parâmetros instancia, problema, α , f(.) e criterioInsercao (linha 1).

Algoritmo 2 Heurística construtiva: *RemocaoSucessiva*

1: $leDadosEntrada(instancia, problema, \alpha, f(.), criterioInsercao);$ 2: Rota = constroiRotaAleatoria(instancia, problema, criterioInsercao);3: $LRC = \emptyset$; 4: LC = inicializaListaCandidatosDrop(instancia, problema, f(.));5: while $(LC \neq \emptyset)$ do $s_i = \min\{f(v_k) \mid v_k \in LC\};\$ 6: $s_f = max\{f(v_k) \mid v_k \in LC\};\$ 7: $LRC = \{ v_k \in LC \mid f(v_k) \le s_i + \alpha(s_f - s_i) \};$ 8: verticeEscolhido = selectionaVerticeAleatoriamente(LRC);9: if (remocaoPossivel(verticeEscolhido, Rota)) then 10: $Rota = Rota \ominus \{verticeEscolhido\};$ 11: $LC = LC - \{verticeEscolhido\};$ 12:13:else $LC = LC - \{verticeEscolhido\};$ 14:end if 15:16: end while 17: return Rota:

A inicialização de *Rota* (linha 2) é feita a partir do procedimento *constroiRotaAleatoria*(), que recebe os parâmetros *instancia*, *problema* e *criterioInsercao*. O parâmetro *problema* para o referido procedimento pode assumir os seguintes valores:

- PRR: A partir deste valor, o procedimento *constroiRotaAleatoria*() constrói uma rota, vértice a vértice, até que todos os vértices de V estejam na rota;
- PRRG: A partir deste valor, o procedimento constroiRotaAleatoria() constrói uma rota, vértice a vértice, até que todos os vértices de $V \cup W$ estejam na rota.

Em cada uma das situações descritas nos itens anteriores, os vértices são escolhidos aleatoriamente e em seguida inseridos na rota de acordo com o critério determinado pelo parâmetro *criterioInsercao* (os valores assumidos por este parâmetro são os mesmos descritos na Seção 5.1.1: econômico ou aleatório). O algoritmo que descreve o procedimento *constroiRotaAleatoria*() não será apresentado devido a sua trivialidade.

A inicialização da lista de candidatos (LC) é feita a partir dos vértices $v_k \in Rota$, com $v_k \notin T$ e f(.) definida de acordo como a fórmula 5.2 (linha 4). Em seguida, a cada iteração e enquanto LC for não vazia, é verificado o maior e o menor benefício associado aos vértices de LC (linhas 6 e 7), sendo a LRC construída a partir destes benefícios e do valor de α (linha 8). Tendo sido selecionado aleatoriamente na LRC (linha 9), verticeEscolhido é removido² de Rota e de LC somente se não inviabilizar a solução corrente (linhas 10 a 12). Caso contrário, verticeEscolhido é removido somente da LC (linha 14).

A Figura 5.2 exemplifica a execução da heurística construtiva *RemocaoSucessiva* na obtenção de uma solução inicial para o PRR quando é considerado o grafo apresentado na Figura 5.1A.

A Figura 5.2A apresenta a solução construída pelo procedimento constroiRotaAleatoria() com o parâmetro criterioInsercao = econômico e problema = PRR. A partir desta rota, a seguinte sequência de vértices é selecionada para a remoção (um vértice por iteração): 13, 16, 18, 14, 20, 15, 17, 19, 22 e 21. Entretanto, como a remoção dos vértices 15, 22 e 21 inviabilizariam a rota, nenhum destes vértices foi removido (a remoção do vértice 15 deixaria o vértice 12 descoberto pela rota; como o vértice 19 foi removido antes do vértice 22, a remoção deste deixaria os vértices 6 e 8 descobertos; o vértice 17 foi removido antes do 21 e, portanto, a remoção deste último deixaria os vértices 9 e 11 descobertos). Logo, somente os vértices 13 (Figura 5.2B), 16 (Figura 5.2C), 18 (Figura 5.2D), 14 (Figura 5.2E), 20 (Figura 5.2F), 17 (Figura 5.2G) e 19 (Figura 5.2H) foram removidos.

 $^{^2~}$ A operação de
notada por \ominus indica a remoção de um vértice da rota, seguida da reco
nexão da rota e atualização do custo da mesma.



Figura 5.2: Passo a passo da construção de uma solução inicial para CTP a partir da heurística RemocaoSucessiva

5.1.3 Heurística IMP-PRR

A heurística IMP-PRR (Inserção Mais Próxima para o PRR) é uma adaptação da heurística construtiva Inserção Mais Próxima proposta originalmente para o PCV (RO-SENKRANTZ; STEARNS; LEWIS, 1977).

O Algoritmo 3 apresenta o pseudocódigo da heurística IMP-PRR, que consiste em obter uma solução viável para o PRR ou para o PRRG a partir da escolha aleatória de um vértice obrigatório (quando T for não vazio) ou a partir do vértice que mais cobre vértices de W (linhas 4 e 6).

Algoritmo 3 Heurística construtiva: *IMP-PRR*

1: *leDadosEntrada(instancia, problema)*; 2: $Rota = \emptyset;$ 3: if $(T \neq \emptyset)$ then Rota = inicializaRotaPorVerticeDeT();4: 5: else Rota = inicializaRotaPorVerticeQueMaisCobreWs();6: 7: end if 8: LDispT = inicializaListaDisponiveisEmT();9: while (LdispT $\neq \emptyset$) do $v_k = \{v_i \in LdispT \mid c_{v_iv_j} \in minimo, v_j \in Rota\};$ 10:Encontrar a aresta (v_m, v_n) em Rota tal que $c_{v_m v_k} + c_{v_k v_n} - c_{v_m v_n}$ seja mínimo; 11: $Rota = insereVerticeNaPosicao(v_k, Rota, v_m, v_n);$ 12: $LdispT = LdispT - \{v_k\};$ 13:14: end while 15: LDisp = inicializaListaDisponiveis();16: while (*Rota* inviável) do $v_z = \{v_i \in Ldisp \mid c_{v_iv_j} \notin minimo, v_j \in Rota\};$ 17:Encontrar a aresta (v_p, v_q) em Rota tal que $c_{v_p v_z} + c_{v_z v_q} - c_{v_p v_q}$ seja mínimo; 18: $Rota = insereVerticeNaPosicao(v_z, Rota, v_p, v_q);$ 19:20: $Ldisp = Ldisp - \{v_z\};$ 21: end while 22: removeVerticesSobressalentes(Rota);

23: return Rota;

A lista LdispT, que contém todos os vértices de T ainda não inseridos em Rota, é criada logo após a inicialização da rota (linha 8). Enquanto existirem vértices obrigatórios fora da rota (linha 9), um vértice de LdispT é escolhido para ser inserido em Rota da seguinte forma: calcula-se a distância entre cada um dos vértices de LdispT e todos os vértices de Rota. Se $c_{v_kv_j}$, tal que $v_k \in LdispT$ e $v_j \in Rota$, é a menor entre todas as distâncias calculadas, então o vértice v_k é escolhido para ser inserido em Rota (linha 10).

A posição em que v_k será inserido é determinada pelo par de vértices $(v_m, v_n) \in Rota$ que produzir o menor acréscimo no custo total de *Rota*, isto é, após selecionar k em LdispT, este é inserido entre os vértices v_m e $v_n \in Rota$ se $c_{v_mv_k} + c_{v_kv_n} - c_{v_mv_n}$ for mínimo (linhas 11 e 12).

Tendo inserido todos os vértices obrigatórios na rota $(LdispT = \emptyset)$, uma lista (Ldisp)com todos os vértices de $v_i \in V \setminus T$ é criada (linha 15). Se a rota parcial já for viável para o problema que está sendo resolvido (linha 16), então o procedimento inicia um processo de remoções sucessivas com o objetivo de remover de *Rota* os vértices sobressalentes (linha 22). Caso contrário, um processo de seleção de vértices em *Ldisp* e posterior inserção em *Rota* é realizado até que *Rota* se torne viável (linhas 16 a 21). Vale ressaltar que esse processo é semelhante ao realizado sobre *LdispT*, conforme descrito no parágrafo anterior. O procedimento *IMP-PRR*, cujo pseudocódigo é apresentado no Algoritmo 3, termina realizando um processo de remoções sucessivas, assim como realizado nos Algoritmos 1 e 2 (linha 22).

A Figura 5.3 exemplifica a execução da heurística construtiva *IMP-PRR* na obtenção de uma solução inicial do PRR para o grafo apresentado na Figura 5.3A, que é o mesmo considerado no passo a passo das heurísticas *AdicaoSucessiva* e *RemocaoSucessiva* (Figuras 5.1 e 5.2).

A rota é inicializada pela escolha do vértice 4. Logo após a criação de LdispT (lista que contém todos os vértices de T ainda não inseridos na rota), os seguintes vértices são escolhidos nesta sequência para serem inseridos na rota parcial: 3 (iniciando a rota 4-3-4), 2 (inserido entre os vértices 3 e 4), 0 (inserido entre os vértices 4 e 3), 1 (inserido entre os vértices 0 e 3) e 5 (inserido entre os vértices 4 e 0). A rota criada a partir dos vértices de T pode ser vista na Figura 5.3B.

A lista LDisp é inicializada com todos os vértices de $V \setminus T$, já que a rota criada sobre os vértices de T (Figura 5.3B) ainda não é viável para o PRR. A partir desta lista, é selecionada a seguinte sequência de vértices para serem inseridos na rota (um vértice por vez): 20 (inserido entre os vértices 5 e 0 - Figura 5.3C), 13 (inserido entre os vértices 2 e 4 - Figura 5.3D), 21 (inserido entre os vértices 1 e 3 - Figura 5.3E), 19 (inserido entre os vértices 4 e 5 - Figura 5.3F) e 14 (inserido entre os vértices 3 e 2 - Figura 5.3G). Após a inserção do vértice 14, a rota torna-se viável para o PRR. O procedimento removeVerticesSobressalentes() é iniciado a partir da escolha do vértice $2 \in T$, removendo em seguida os vértices 13 e 20 (Figura 5.3H).



Figura 5.3: Passo a passo da construção de uma solução inicial para CTP a partir da heurística IMP-PRR

5.1.4 Heurística IMD-PRR

A ideia geral da heurística construtiva *IMD-PRR* (Inserção Mais Distante para o PRR) é obter uma solução viável para o *PRR* ou para o *PRRG* a partir da adaptação da heurística construtiva Inserção Mais Distante proposta originalmente para o PCV (RO-SENKRANTZ; STEARNS; LEWIS, 1977). Esta seção apresenta três propostas de adaptações desta heurística a partir da variação do critério de escolha dos vértices que serão inseridos em *Rota*. O Algoritmo 4 apresenta o pseudocódigo da heurística *IMD-PRR*.

```
Algoritmo 4 Heurística construtiva: IMD-PRR
 1: leDadosEntrada(instancia, problema);
 2: Rota = \emptyset;
 3: if (T \neq \emptyset) then
      Rota = inicializaRotaPorVerticeDeT();
 4:
 5: else
      Rota = inicializaRotaPorVerticeQueMaisCobreWs();
 6:
 7: end if
8: LDispT = inicializaListaDisponiveisEmT();
9: while (LdispT \neq \emptyset) do
      v_k = \{v_i \in LdispT \mid c_{v_iv_l} \in maximo, v_l \in Rota\};
10:
      Encontrar a aresta (v_i, v_j) em Rota tal que c_{v_i v_k} + c_{v_k v_j} - c_{v_i v_j} seja mínimo;
11:
12:
      Rota = insereVerticeNaPosicao(v_k, Rota, v_i, v_j);
      LdispT = LdispT - \{v_k\};
13:
14: end while
15: LDisp = inicializaListaDisponiveis();
16: while (Rota inviável) do
17:
      v_z = \{v_i \in Ldisp \mid c_{v_iv_l} \in maximo, v_l \in Rota\};
      Encontrar a aresta (v_i, v_j) em Rota tal que c_{v_i v_z} + c_{v_z v_j} - c_{v_i v_j} seja mínimo;
18:
      Rota = insereVerticeNaPosicao(v_z, Rota, v_i, v_j);
19:
      Ldisp = Ldisp - \{v_z\};
20:
21: end while
22: removeVerticesSobressalentes(Rota);
23: return Rota;
```

A construção da solução inicial para o PRR ou para o PRRG na adaptação da Inserção Mais Distante apresentada no Algoritmo 4, inicia pela escolha aleatória de um vértice obrigatório (quando $T \neq \emptyset$) (linha 4). Partindo deste vértice, escolhe-se a cada iteração o vértice obrigatório $v_k \in DispT$ cuja distância para algum vértice da rota atual seja máxima (linha 10). O vértice k é inserido na rota atual entre os vértices (v_i, v_j) $\in Rota$ que produzirem o menor acréscimo no custo desta rota (linhas 11 e 12). Quando não existirem mais vértices de T disponíveis para inserção, isto é, quanto $LDispT = \emptyset$, verifica-se se a rota atual ainda é inviável (linha 16). Caso a inviabilidade seja comprovada, realiza-se um novo processo de escolha e inserção de vértices, porém considerando os vértices de $V \setminus T$ (ou $V \setminus T \cup W$ para o PRRG) disponíveis para a inserção (linhas 17 a 20). Este processo encerra quando *Rota* se torna viável.

Além da adaptação apresentada no Algoritmo 4, onde o vértice escolhido para inserção a cada iteração é aquele que está mais distante da rota atual, são propostas mais duas variações da heurística de construção Inserção Mais Distante do PCV, assim como mostram os itens a seguir:

- IMD_{MinMax}-PRR: Seleciona para inserção o vértice cuja distância mínima à rota atual seja máxima. Para implementar esta variação, basta substituir as linhas 10 e 17 do Algoritmo 4 por:
 - Para cada vértice k em Ldisp/LdispT e para cada vértice v_j da rota, selecionar o vértice v_k para inserção se o valor MINIMO de $c_{v_k v_j}$ for MAXIMO;
- IMD_{MaxMin}-PRR: Seleciona para inserção o vértice cuja distância máxima ao ciclo atual seja mínima. Para implementar esta variação, basta substituir as linhas 10 e 17 do Algoritmo 4 por:
 - Para cada vértice k em Ldisp/LdispT e para cada vértice v_j da rota, selecionar o vértice v_k para inserção se o valor MAXIMO de $c_{v_k v_j}$ for MINIMO;

A Figura 5.4 exemplifica a execução da heurística construtiva *IMD-PRR* na obtenção de uma solução inicial para PRR, considerando o grafo apresentado na Figura 5.4A. Na sequência de quadros que seguem a referida figura, tem-se na Figura 5.4B a rota criada sobre os vértices de T, que foram escolhidos na seguinte ordem para compor esta rota: 4, 1, 2, 5, 0 e 3. Os demais vértices inseridos na rota até que esta se tornasse viável para o PRR foram: 16 (Figura 5.4C), 15 (Figura 5.4D), 21 (Figura 5.4E), 18 (Figura 5.4F), 20 (Figura 5.4G), 17 (Figura 5.4H) e 22 (Figura 5.4I). O procedimento removeVerticesSobressalentes() inicia a varredura da rota em busca de vértices sobressalentes para remoção a partir do vértice $2 \in T$, removendo em seguida os vértices 18, 16, 20 e 21 (Figura 5.4J).

As Figuras 5.5 e 5.6 exemplificam, respectivamente, a execução da heurística construtiva IMD_{MinMax} -PRR e IMD_{MaxMin} -PRR na obtenção de uma solução inicial para o PRR. Os comentários sobre cada uma destas ilustrações encontram-se após a Figura 5.6.



Figura 5.4: Passo a passo da construção de uma solução inicial para CTP a partir da heurística IMD-PRR



Figura 5.5: Passo a passo da construção de uma solução inicial para CTP a partir da heurística IMD_{MinMax} -PRR



Figura 5.6: Passo a passo da construção de uma solução inicial para CTP a partir da heurística IMD_{MaxMin} -PRR

Na execução da heurística IMD_{MinMax} -PRR ilustrada na Figura 5.5, a rota é inicializada pela escolha do vértice 4, inserindo na sequência os vértices: 3, 2, 0, 1, 5, 19, 13, 14, 18, 15, 16 e 17. São removidos pelo procedimento removeVerticesSobressalentes() os vértices 18, 13, 16 e 14 (Figura 5.5J).

A Figura 5.6 ilustra uma execução da heurística IMD_{MaxMin} -PRR com a rota inicializada pelo vértice 4. Na sequência, são inseridos os vértices: 1, 0, 5, 2, 3, 20, 21, 16, 15, 17, 22. Após construir a rota viável (Figura 5.6H), o procedimento removeVertices-Sobressalentes() inicia pelo vértice 2 e elimina os vértices 17, 20 e 16 (Figura 5.6I).

5.1.5 Heurística IMB-PRR

O procedimento construtivo apresentado no Algoritmo 5 baseia-se na heurística Inserção Mais Barata proposta originalmente para o PCV (ROSENKRANTZ; STEARNS; LEWIS, 1977). A inicialização da rota no procedimento *IMB-PRR* (Inserção Mais Barata para o PRR) pode ser feita de duas formas distintas, assim como mostram os itens a seguir:

- PorNumWsCobertos: Escolhem-se os três vértices que cobrem mais vértices de W, sendo estes inseridos na rota na medida em que são escolhidos de forma a causar o menor acréscimo no custo total da rota;
- Escolha Aleatoria EmT: Escolhem-se os três vértices obrigatórios de forma aleatória (quando |T| ≥ 3), sendo estes inseridos na rota na medida em que são escolhidos de forma a causar o menor acréscimo no custo total da rota. Se |T| < 3, completar os três vértices com a escolha aleatória de vértices de V \ T (caso PRR) ou vértices de (V \ T) ∪ W (caso PRRG).

A escolha de uma das duas formas de inicializar a rota no Algoritmo 5 é realizada através do parâmetro de entrada *criterioInicializacao* (linha 1).

Tendo a rota inicial composta por três vértices selecionados conforme o critério de inicialização passado como parâmetro (linha 3), o algoritmo inicia um processo de inserções de vértices obrigatórios (enquanto $LdispT \neq \emptyset$), sendo selecionado a cada iteração aquele que estiver mais próximo da rota corrente (linhas 5 a 9). Se após a inserção de todos os vértices obrigatórios a rota ainda for inviável, o algoritmo inicia um novo processo de inserções de vértices de $V \setminus T$ (caso PRR) ou vértices de $(V \setminus T) \cup W$ (caso PRRG), até que *Rota* se torne viável (linhas 11 a 15). Na linha 16 é realizado um processo de remoção de vértices sobressalentes, assim como os demais algoritmos construtivos já abordados.
Algoritmo 5 Heurística construtiva: IMB-PRR

```
1: leDadosEntrada(instancia, problema, criterioInicializacao);
 2: Rota = \emptyset;
 3: Rota = inicializaRota(criterioInicializacao);
 4: LDispT = inicializaListaDisponiveisEmT();
 5: while (LdispT \neq \emptyset) do
      v_k = \{v_t \in LdispT \mid (c_{v_iv_t} + c_{v_tv_j} - c_{v_iv_j}) \text{ seja mínimo, com } v_i, v_j \in Rota\};
 6:
      Rota = insereVerticeNaPosicao(v_k, Rota, v_i, v_j);
 7:
      LdispT = LdispT - \{v_k\};
 8:
 9: end while
10: LDisp = inicializaListaDisponiveis();
11: while (Rota inviável) do
      v_z = \{v_l \in Ldisp \mid (c_{v_iv_l} + c_{v_lv_j} - c_{v_iv_j}) \text{ seja mínimo, com } v_i, v_j \in Rota\};
12:
      Rota = insereVerticeNaPosicao(v_z, Rota, v_i, v_j);
13:
      Ldisp = Ldisp - \{v_z\};
14:
15: end while
16: removeVerticesSobressalentes(Rota);
17: return Rota;
```

A Figura 5.7 exemplifica a execução da heurística construtiva *IMB-PRR* na obtenção de uma solução inicial do PRR para o grafo apresentado na Figura 5.7A.



Figura 5.7: Passo a passo da construção de uma solução inicial para CTP a partir da heurística IMB-PRR

O parâmetro criterioInicialização considerado no exemplo da Figura 5.7 foi inicializado com a opção PorNumWsCobertos. Os três primeiros vértices selecionados para compor a rota a partir deste critério foram: 21, 22 e 0 (Figura 5.7B). Na sequência, os vértices de T que ainda estavam fora da rota parcial, foram inseridos na seguinte ordem: 1, 5, 3, 4 e 2 (Figura 5.7C). Como após a inserção de todos os vértices de T a rota ainda encontrava-se inviável para o PRR, foi inserido o vértice 14 para cobrir o vértice 12 que ainda estava descoberto pela rota (Figura 5.7D). Nenhuma remoção foi realizada, pois a remoção de qualquer vértice inviabilizaria a rota construída.

5.2 Heurísticas de Busca Local

Uma busca local tem como objetivo tentar promover melhorias em uma solução corrente de forma iterativa, substituindo sucessivamente a solução corrente pela melhor da sua vizinhança, terminando esta substituição sucessiva somente quando não existirem soluções aprimorantes na vizinhança que está sendo investigada. Neste sentido, visando melhorar a qualidade das soluções produzidas pelas heurísticas construtivas descritas na Seção 5.1, são propostas algumas heurísticas de busca local para o PRR e para o PRRG descritas a seguir.

5.2.1 Heurística k-AddDrop

A heurística k-AddDrop é uma extensão da heurística $Busca_ADD$ proposta em (MOTTA, 2001) para o PRRG. A vizinhança de uma solução s explorada pela heurística k-AddDrop é formada por todas as soluções que podem ser alcançadas a partir de k inserções de vértices em s, seguidas de remoções sucessivas que ocorrem enquanto for possível remover vértices de s sem inviabilizá-la.

O Algoritmo 6 apresenta o pseudocódigo da busca k-AddDrop, iniciada a partir da leitura dos parâmetros de entrada (linha 1): solucaoInicial, que corresponde à solução obtida por uma das heurísticas construtivas apresentadas na Seção 5.1 e k, que corresponde a quantidade de vértices que serão arranjados a partir de LDispAdd. Estes arranjos de k vértices formarão os movimentos da vizinhança a ser explorada. A lista LDispAdd é composta de todos os vértices que estão fora da rota inicial (linha 4) e a lista LDispDrop é composta por todos os vértices não obrigatórios que formam a rota inicial (linha 5).

A vizinhança k-AddDrop de *inicialRota* é explorada a partir de cada arranjo de k vértices obtidos a partir de LDispAdd (linhas 6 a 21).

Algoritmo 6 Heurística de busca local: k-AddDrop 1: *leDadosEntrada(solucaoInicial, k)* 2: *inicialRota* = *solucaoInicial*; 3: melhorRota = inicialRota;4: LDispAdd = inicializaListaDisponiveisAdd(solucaoInicial); 5: LDispDrop = inicializaListaDisponiveisDrop(solucaoInicial);6: for $((v_i, ..., v_k) \in LDispAdd))$ do atualRota = inicialRota;7: $atualRota = atualRota \oplus insereKverticesNaRota((v_i, ..., v_k), atualRota);$ 8: while $(LDispDrop \neq \emptyset)$ do 9: 10: $v_{i} = selectionaVerticeAleatoriamente(LDispDrop);$ if $(remocaoNaoInviabiliza(v_i, atualRota))$ then 11: 12: $atualRota = atualRota \ominus \{v_i\};$ $LDispDrop = LDispDrop - \{v_i\};$ 13:else 14: $LDispDrop = LDispDrop - \{v_i\};$ 15:end if 16:end while 17:18:if ((custo(atualRota) < custo(melhorRota))) then 19:melhorRota = atualRota;end if 20:21: end for 22: **return** melhorRota;

O método insere Kvertices NaRota($(v_i, ..., v_k)$, atual Rota) insere os k vértices do arranjo em atual Rota, um após o outro, na sequência em que aparecem no arranjo. A posição em que cada vértice v_i deste arranjo é inserido em atual Rota é determinada pelo par de vértices $(v_p, v_q) \in atual Rota$, tal que $c_{v_pv_i} + c_{v_iv_q} - c_{v_pv_q}$) seja mínimo (linha 8). A operação denotada por \oplus representa o acréscimo dos k vértices do arranjo $(v_i, ..., v_k)$ em atual Rota e a atualização do custo da mesma após cada inserção.

Após concluídas as k inserções, um processo de remoção de vértices de atualRota é iniciado (linhas 9 a 17). A partir de LDispDrop, tenta-se remover os vértices não obrigatórios diferentes dos recém inseridos. Caso algum vértice recém inserido não cubra nenhum vértice de W, sua remoção é permitida (vide vértice 13 na Figura 5.9). Vale ressaltar que um vértice $v_j \in LDispDrop$ só é removido de atualRota se não inviabilizála (linhas 11 a 16). A operação denotada por \ominus (linha 12) representa a remoção do vértice da rota e posterior atualização do custo da mesma. A ordem dos vértices escolhidos para possível remoção é feita de forma aleatória, já que cada remoção realizada influencia as remoções posteriores.

Para cada uma das soluções investigadas na vizinhança k-AddDrop, é verificada a possibilidade de atualização da solução incumbente (linhas 18 a 20). É importante res-

saltar que, independente da qualidade da solução vizinha obtida, a busca sempre retorna a solução inicial fornecida como semente (linha 7), que aqui está sendo denotada por *solucaoInicial (best improvement)*. Após investigar toda a vizinhança, a busca encerra retornando a melhor solução encontrada.

Experimentos computacionais preliminares com um subconjunto das instâncias geradas neste trabalho (vide Apêndice A) mostraram que não é vantajoso considerar k > 3, pois o tempo de processamento aumenta muito e a qualidade da solução incumbente nem tanto quando comparada à qualidade das soluções obtidas com as vizinhanças k = 2e k = 3. Desta forma, serão consideradas três vizinhanças distintas da heurística k-AddDrop, através da variação do valor de k: 1-AddDrop (insere 1 vértice em s e remove enquanto for possível), 2-AddDrop (insere 2 vértices em s e remove enquanto for possível) e 3-AddDrop (insere 3 vértices em s e remove enquanto for possível).

A Figura 5.8 exemplifica a execução de uma iteração da heurística de busca local 1-AddDrop.



Figura 5.8: Passo a passo de uma iteração da busca local 1-AddDrop realizada em uma solução inicial construída pela heurística IMD_{MinMax} -PRR

A solução inicial ilustrada na Figura 5.8 para o PRR corresponde à rota gerada a partir da heurística construtiva IMD_{MinMax} -PRR (Figura 5.8A) ilustrada inicialmente na Figura 5.5. A solução investigada nesta vizinhança é obtida a partir da inserção do vértice 21 (Figura 5.8B), seguida da remoção do vértice 17 (nenhum outro vértice foi removido para manter a viabilidade da rota).

A Figura 5.9 ilustra a execução de uma iteração da busca local 2-AddDrop, com a solução inicial gerada a partir da heurística construtiva IMD_{MinMax} -PRR (Figura 5.9A).



Figura 5.9: Passo a passo de uma iteração da busca local 2-AddDrop realizada em uma solução inicial construída pela heurística IMD_{MinMax} -PRR

A solução investigada na vizinhança 2-AddDrop é obtida a partir da inserção do par de vértices (13,14) (Figuras 5.9B e 5.9C). O processo de remoções inicia a partir após a inserção do vértice 14 e elimina da rota os vértices 15 e 13 (Figuras 5.9D e 5.9E).

5.2.2 Heurística k-DropAdd

A heurística k-DropAdd, analogamente à heurística k-AddDrop, é uma extensão da heurística $Busca_DROP$ proposta em (MOTTA, 2001) para o PRRG. A vizinhança de uma solução s explorada pela heurística k-DropAdd é formada por todas as soluções que podem ser alcançadas a partir de k remoções de vértices em s, seguidas de um processo de adições caso s tenha ficado inviável (as adições só ocorrem até que a viabilidade de s seja restabelecida).

O Algoritmo 7 apresenta o pseudocódigo da heurística busca local k-DropAdd.

Alg	goritmo 7 Heurística de busca local: k-DropAdd
1:	leDadosEntrada(solucaoInicial, k)
2:	inicialRota = solucaoInicial;
3:	melhorRota = inicialRota;
4:	LDispAdd = inicializaListaDisponiveisAdd(solucaoInicial);
5:	LDispDrop = inicializaListaDisponive isDropAdd(solucaoInicial);
6:	for $((v_i,, v_k) \in LDispDrop))$ do
7:	atualRota = inicialRota;
8:	$removeKverticesDeRota((v_i,, v_k), atualRota);$
9:	while $(ehInviavel(atualRota))$ do
10:	$v_j = verificaInviabilidade(atualRota);$
11:	$atualRota = atualRota \oplus selectionaVerticeParaViabilizar(v_j, LDispAdd);$
12:	end while
13:	if $((custo(atualRota) < custo(melhorRota)))$ then
14:	melhorRota = atualRota;
15:	end if
16:	end for
17:	return melhorRota;

A diferença entre os algoritmos que implementam as buscas k-AddDrop (Algoritmo 6) e k-DropAdd (Algoritmo 7) começa na inicialização da lista de vértices disponíveis para a remoção (linha 5). Nesta última busca, os vértices obrigatórios são considerados em LDispDrop e, consequentemente, também nos arranjos que serão formados a partir desta lista. Experimentos computacionais mostraram que esta estratégia permite que a busca escape de alguns ótimos locais, pois ao remover um ou mais vértices de T, estes não são necessariamente reinseridos na mesma posição da rota em que foram removidos.

Após a inicialização das listas LDispAdd e LDispDrop (linhas 4 e 5), inicia-se o processo de exploração da vizinhança k-DropAdd de inicialRota (linha 6). A remoção de cada vértice do arranjo ($v_i, ..., v_k$) é feita pelo método $removeKverticesDeRota((<math>v_i, ..., v_k$), atual-Rota), que também atualiza o custo de atualRota a cada remoção realizada (linha 8). Se a rota resultante das remoções for inviável, o método verificaInviabilidade(atualRota)(linha 10) retorna em v_j o vértice associado à primeira inviabilidade identificada, isto é, retorna um vértice de T, caso seja identificada a sua ausência em atualRota ou retorna um vértice de W, caso seja identificado que este ainda não está sendo coberto por atualRota.

O método selecionaVerticeParaViabilizar $(v_j, LDispAdd)$ é responsável por selecionar um vértice em LDispAdd para viabilizar a rota de acordo com a indicação de v_j . Caso a restauração da viabilidade de *atualRota* em relação a v_j não seja possível a partir de LDispAdd, então é permitido que os vértices recém removidos sejam utilizados para esta finalidade (linha 11). A operação denotada por \oplus representa a inserção de um vértice em *atualRota* de modo a produzir o menor acréscimo no custo desta. O processo de inserção de vértices continua até que *atualRota* seja viável (linhas 9 a 12).

Assim como acontece na heurística k-AddDrop, para cada uma das soluções investigadas na vizinhança k-DropAdd, é verificada a possibilidade de atualização da solução incumbente (linhas 13 a 15). A busca encerra retornando a melhor solução encontrada na vizinhança investigada.

A Figura 5.10 exemplifica a execução de uma iteração da heurística de busca local 1-DropAdd, tendo como solução inicial para o PRR a rota gerada a partir da heurística construtiva IMD_{MinMax} -PRR ilustrada na Figura 5.10A.

A busca inicia a execução pela remoção do vértice 15 (Figura 5.10B). Como a solução fica inviável após esta remoção, pois o vértice $12 \in W$ fica descoberto, é preciso inserir um novo vértice (diferente do recém removido, sempre que possível) para cobrir o vértice que ficou descoberto. Após a inserção do vértice 14, a solução torna-se novamente viável para o PRR (Figura 5.10C).



Figura 5.10: Passo a passo de uma iteração da busca local 1-DropAdd realizada em uma solução inicial construída pela heurística IMD_{MinMax} -PRR

A Figura 5.11 exemplifica a execução de uma iteração da heurística de busca local 2-DropAdd, tendo como solução inicial para o PRR a rota gerada pela heurística construtiva IMD_{MinMax}-PRR (Figura 5.11A).

A iteração da busca local 2-DropAdd exemplificada na Figura 5.11, inicia pela remoção do par de vértices (2,15). A rota resultante das remoções é inviável para o PRR e, portanto, é necessário iniciar um processo para viabilizá-la: reinserir os vértices de T ou cobrir os vértices de W descobertos pela rota atual. No caso da rota em questão, é necessário reinserir o vértice 2 (Figura 5.11D) e um outro vértice da vizinhança do vértice 12. Como a reinserção do vértice que acabou de ser removido só ocorre quando não há outra possibilidade para viabilizar a rota, o vértice 14 é inserido para cobrir o vértice 12 (Figura 5.11E).



Figura 5.11: Passo a passo de uma iteração da busca local 2-DropAdd realizada em uma solução inicial construída pela heurística IMD_{MinMax} -PRR

5.2.3 Heurística 20pt-PRR

A heurística 2Opt-PRR (2-Optimal para o PRR) apresentada nesta seção é uma adaptação da heurística 2-Optimal do PCV (LINS; KERNIGHAN, 1973) e baseia-se na busca por melhores soluções na vizinhança de uma determinada rota r realizando todas as possíveis trocas de duas arestas não adjacentes, quebrando o circuito em dois caminhos desconexos e então reconectando esses caminhos da outra maneira possível se, e somente se, esta outra maneira possível reduzir o custo da rota r.

Para reduzir a complexidade da heurística 20pt-PRR, considerou-se a rota associada ao PRR/PRRG como sendo direcionada para realizar a escolha dos pares de arestas não adjacentes a serem removidas da rota. Desta forma, a quantidade de pares de arestas a serem analisados diminui e consequentemente, o custo total para realizar a busca a partir da vizinhança definida por esta heurística também diminui. Vale ressaltar que a escolha das arestas para reconectar os caminhos não considera nenhuma orientação na rota e por isto, nenhuma solução é deixada de ser analisada a partir da adaptação aqui proposta.

A Figura 5.12 exemplifica a execução de uma iteração da heurística de busca local 2Opt-PRR, tendo como solução inicial para o PRR a rota gerada a partir da heurística construtiva IMD_{MinMax} -PRR ilustrada inicialmente na Figura 5.5.



Figura 5.12: Passo a passo de uma iteração da busca local 2Opt-PRR realizada em uma solução inicial construída pela heurística IMD_{MinMax} -PRR

A Figura 5.12A ilustra a rota quando a orientação é considerada para esta. O par de arestas (5,0) e (4,19) é removido da solução inicial, quebrando o circuito em dois caminhos desconexos: $19 \rightarrow 5 e 0 \rightarrow 1 \rightarrow 17 \rightarrow 3 \rightarrow 15 \rightarrow 2 \rightarrow 4$. Uma das possibilidades para reconectar estes dois caminhos é inserir as arestas (4,5) e (19,0), o que implicará na alteração da orientação atribuída à aresta (19,5) para (5,19). Note que, como a solução resultante é pior do que a solução inicial, esta é descartada e uma nova possibilidade para a troca dos pares de arestas poderá realizada.

O algoritmo que descreve a heurística 20pt-PRR não será apresentado neste trabalho, pois difere do algoritmo da proposta original (LINS; KERNIGHAN, 1973) apenas na obtenção dos pares de arestas a serem removidos.

5.2.4 Heurística VNS-PRR

A meta-heurística VNS (Variable Neighborhood Search) proposta por Nenad Mladenović e Pierre Hansen (MLADENOVIĆ; HANSEN, 1997) fundamenta-se na troca sistemática de estruturas de vizinhança para explorar o espaço de soluções. Contrariamente a outras meta-heurísticas baseadas em métodos de busca local, o VNS não segue uma trajetória, mas explora vizinhanças gradualmente mais distantes da solução corrente e intensifica a busca em torno de uma nova solução somente se um movimento de melhora for realizado.

Dada uma solução s, entende-se por vizinhança de s como sendo o conjunto formado por todas as soluções s' que podem ser obtidas através da aplicação de uma modificação elementar na solução s, sendo esta modificação comumente chamada de movimento. Portanto, $\{N_1, N_2, N_3, ..., N_{k_{max}}\}$ pode ser considerado um conjunto finito de estruturas de vizinhanças onde $N_k(s)$ é o conjunto de soluções pertencentes a k-ésima vizinhança de s.

De acordo com Hansen e Mladenović (2003), a meta-heurística VNS explora sistematicamente as seguintes observações:

- (1) Um mínimo local de uma estrutura de vizinhança não necessariamente o é em outra;
- (2) Um mínimo global é um mínimo local em todas as possíveis estruturas de vizinhanças;
- (3) Para muitos problemas, os mínimos locais de uma ou mais vizinhanças são relativamente próximos uns dos outros.

Vale ressaltar que a terceira observação corresponde a um fato empírico e infere que um ótimo local frequentemente provê alguma informação sobre o ótimo global.

O procedimento genérico da meta-heurística VNS inicia a partir de um ótimo local (solução inicial), normalmente obtido por um procedimento de construção. Um conjunto de estruturas de vizinhança e um critério de parada são parâmetros que devem ser definidos previamente nesta meta-heurística. A melhor solução (solução incumbente) é inicializada como sendo a solução inicial e, enquanto o critério de parada estabelecido não for satisfeito, a primeira estrutura de vizinha é considerada e uma nova solução é encontrada aleatoriamente nesta vizinhança a partir da solução inicial. Uma busca local é aplicada na nova solução para obter um ótimo local. Se o custo deste ótimo local for menor do que o custo da solução incumbente, então a solução incumbente e a solução inicial são atualizadas e a primeira estrutura de vizinhança passa a ser considerada. Caso contrário, a estrutura de vizinhança seguinte à vizinhança atual é considerada. Quando a última estrutura de vizinhança é atingida $(k = k_{max})$ sem que uma solução promissora (melhor que a solução incumbente) seja encontrada, a busca recomeça na primeira vizinhança até que a condição de parada seja satisfeita. Esta condição pode ser considerada como o tempo máximo de execução, um número máximo de iterações ou número máximo de iterações desde a última melhora. No final de todo o processo, a solução incumbente é retornada.

A heurística VNS-PRR (VNS para o PRR), que segue os princípios da meta-heurística VNS descrita no parágrafo anterior, é apresentada no Algoritmo 8.

Alg	goritmo 8 Heurística de busca local: VNS-PRR
1:	$entrada(instancia, problema, rotaConstrutivo, \{V_1, V_2,, V_{k_{max}}\}, criterioParada);$
2:	atualRota = rotaConstrutivo;
3:	melhorRota = atualRota;
4:	$rotaVizinha = \emptyset;$
5:	while (critério de parada não atingido) do
6:	k = 1;
7:	while $(k < k_{max})$ do
8:	$rotaVizinha_1 = encontraVizinho(atualRota, V_k);$
9:	$rotaVizinha_2 = realizaBusca(V_k, rotaVizinha_1);$
10:	if $(custo(rotaVizinha_2) < custo(melhorRota))$ then
11:	$atualRota = rotaVizinha_2;$
12:	melhorRota = atualRota;
13:	k = 1;
14:	else
15:	k = k + 1;
16:	end if
17:	end while
18:	end while
19:	return melhorRota:

O Algoritmo 8 inicia a partir da leitura dos parâmetros (linha 1): instancia, problema, rotaConstrutivo, estruturas de vizinhanças ($\{V_1, V_2, ..., V_{k_{max}}\}$), criterioParada. Os três primeiros parâmetros são comuns a todas as buscas apresentadas nesta seção; o quarto parâmetro corresponde às estruturas de vizinhanças que serão gradativamente exploradas e, o quinto parâmetro, relaciona-se ao critério de parada que será utilizado, podendo ser o número máximo de iterações, o número máximo de iterações sem melhora, o tempo máximo de CPU, entre outros.

A linha 8 obtém aleatoriamente uma solução $(rotaVizinha_1)$ na vizinhança k de atualRota. Esta estratégia tem como finalidade evitar a ciclagem da busca, pois quando são usados processos determinísticos no lugar de processos estocásticos, a ciclagem pode ocorrer. Uma busca na vizinhança k da solução $rotaVizinha_1$ é realizada na linha 9, retornando para $rotaVizinha_2$ a melhor solução obtida nesta busca. Se o custo da nova solução $(rotaVizinha_2)$ for melhor do que a solução incumbente (linha 10), a solução atual (atualRota) e a solução incumbente (melhorRota) são atualizadas (linhas 11 e 12) e a busca é intensificada em torno da nova solução atual, voltando à primeira estrutura de vizinhança explorada (linha 13). Caso a atualização da solução incumbente não ocorra, a vizinhança k + 1 é explorada se $k < K_{max}$ e se o critério de parada não tiver sido atingido.

Muitas aplicações reportadas na literatura, assim como em (RIBEIRO; UCHOA; WER-NECK, 2002), (RIBEIRO; SOUZA, 2002) e (BRIMBERG et al., 2000), têm obtido bons resultados substituindo a busca local realizada internamente no VNS (linha 9 do Algoritmo 8) por um procedimento baseado na meta-heurística VND (*Variable Neighborhood Descent*), que é uma variante da meta-heurística VNS onde a troca de estruturas de vizinhança é feita de forma determinística. Seguindo esta tendência, a seção a seguir apresenta a heurística 9 que está sendo proposta neste trabalho como uma das alternativas de busca local para o PRR/PRRG.

5.2.4.1 Heurística VND-PRR

A meta-heurística VND (Variable Neighborhood Descent) (HANSEN; MLADENOVIĆ; PEREZ-BRITOS, 2001) baseia-se na observação (1) descrita anteriormente na Seção 5.2.4 e visa a exploração do espaço de busca realizando a troca de estruturas de vizinhança de forma determinística. O Algoritmo 9 apresenta o pseudocódigo da heurística VND-PRR proposta neste trabalho.

Algoritmo 9 Heurística de busca local: VND-PRR

1: $leDadosEntrada(instancia, problema, rotaConstrutivo, \{V_1, V_2, ..., V_{k_{max}}\});$ 2: atualRota = rotaConstrutivo; 3: k = 1: 4: while $(k < k_{max})$ do $rotaVizinha = realizaBusca(V_k, atualRota);$ 5: if (custo(rotaVizinha) < custo(atualRota)) then 6: 7: atualRota = rotaVizinha;k = 1: 8: 9: else 10:k = k + 1;end if 11:12: end while 13: return atualRota;

Os parâmetros de entrada do Algoritmo 9 diferem dos parâmetros do Algoritmo 8 por não haver a necessidade de estabelecer um critério de parada para o VND, uma vez que a busca é encerrada quando não existirem soluções aprimorantes em nenhuma das vizinhanças da solução atual.

A busca inicia a partir da solução fornecida como parâmetro de entrada (rotaConstrutivo) e a exploração do espaço de busca em torno desta solução parte da primeira estrutura de vizinhança (k = 1). Se uma solução aprimorante for encontrada em $V_1(atualRota)$ (linha 6), a solução incumbente é atualizada (linha 7) e a busca retorna à primeira estrutura de vizinhança (linha 8). Caso contrário, a busca passa a ser realizada na estrutura de vizinhança seguinte (linha 10). Enquanto for possível encontrar uma solução aprimorante em qualquer uma das vizinhanças da solução corrente, a busca prossegue retornando a primeira estrutura de vizinhança.

As seguintes estruturas de vizinhança são propostas para as heurísticas VNS-PRR e VND-PRR:

Estrutura de vizinhança 1:

1DropAdd (\mathcal{V}_1) - Um vértice $i \in V$ é removido da solução. Se esta remoção implicar na inviabilidade da solução, um processo de inserções sucessivas é iniciado considerando os seguintes casos:

(a) Se $i \in T$, então i é reinserido na solução entre os vértices (k,h) que produzirem o menor acréscimo no custo total da rota;

(b) Se $i \in V \setminus T$, um vértice $j \in V \setminus T$ é aleatoriamente selecionado em $S_l - \{i\}$, onde $l \in W$ é o vértice que deixou de ser coberto após a remoção do vértice $i \in S_l = \{u \in V \mid c_{lu} \leq d\}$. Se $S_l - \{i\} = \emptyset$ o vértice i é reinserido na solução entre os vértices (k,h) que produzirem o menor acréscimo no custo total da rota. Este processo é repetido até que a viabilidade da solução seja restabelecida.

2DropAdd (\mathcal{V}_2) - Esta vizinhança é similar à (\mathcal{V}_1) , exceto que 2 vértices são removidos antes de eventualmente iniciar o processo de inserções.

3DropAdd (\mathcal{V}_3) - Neste caso, 3 vértices são removidos antes de eventualmente iniciar o processo de inserções.

4DropAdd (\mathcal{V}_3) - Neta vizinhança, 4 vértices são removidos antes de eventualmente iniciar o processo de inserções.

Estrutura de vizinhança 2:

1AddDrop (\mathcal{V}_1) - Um vértice $v_i \in V$ é inserido na solução. Um processo de remoções sucessivas é iniciado para tentar remover os vértices que podem ter se tornado desnecessário à cobertura de algum vértice de W após a inserção do vértice v_i . Logo, um vértice $v_j \neq v_i$ só sé removido da rota se v_j não inviabilizar a rota.

2AddDrop (\mathcal{V}_2) - Esta vizinhança é similar à (\mathcal{V}_1), exceto que 2 vértices são inseridos antes de iniciar o processo de remoções.

3AddDrop (\mathcal{V}_3) - Neste caso, 3 vértices são inseridos antes de iniciar o processo de remoções.

4AddDrop (\mathcal{V}_4) - Nesta vizinhança, 4 vértices são inseridos antes de iniciar o processo de remoções.

5.3 Metodologia dos experimentos

A fim de avaliar as heurísticas propostas neste trabalho para o PRR e para o PRRG, utilizou-se dois testes estatísticos: ANOVA (*ANalysis Of VAriance*) e *t-Student*. Uma breve descrição destes testes é apresentada no Apêndice D.

O teste ANOVA foi utilizado com o objetivo de verificar se existe diferença estatisticamente significativa entre as heurísticas propostas. Caso não sejam identificadas diferenças significativas entre as variâncias no teste de ANOVA, ainda é possível verificar se há uma diferença sistemática atuando entre as heurísticas quando comparadas duas a duas. A distribuição de probabilidade *t-Student* é amplamente utilizada para esta comparação, assim como descreve o Apêndice D.

Um nível de significância³ deve ser previamente determinado com o objetivo de estabelecer uma margem de erro máxima aceitável. Esta margem de erro é utilizada para calcular o coeficiente de confiança, que permite determinar qual será o nível de confiabilidade dos resultados que serão apresentados.

Nos experimentos desta tese, a média aritmética μ_h foi calculada a partir do custo das soluções obtidas em todas as execuções da heurística h, para cada uma das instâncias que compõem a amostra considerada. Uma outra maneira de calcular a média μ_h seria utilizar somente o custo da melhor solução obtida pela heurística h para cada uma das instâncias da amostra. Porém, neste caso todas as demais execuções da heurística h estariam sendo desconsideradas no cálculo da média, fato que poderia conduzir os testes estatísticos a resultados inconsistentes acerca do desempenho global das heurísticas.

Considerando amostras aleatórias e independentes de κ populações⁴ e denotando-se as κ médias aritméticas por $\mu_1, \mu_2, ..., \mu_{\kappa}$, conduziu-se o teste da análise de variância (ANOVA para um fator) a fim de verificar a seguinte hipótese nula (H_0):

 H_0 : Todas as κ populações têm a mesma média, isto é, $\mu_1 = \mu_2 = \dots = \mu_{\kappa}$;

 H_1 : Nem todas as κ populações têm a mesma média.

Após a determinação das hipóteses que serão testadas $(H_0 \in H_1)$, calcula-se o valor de F_{calc} a partir das variâncias entre os grupos e dentro dos grupos, considerando o nível

³ É aceitável estatisticamente um nível de significância em torno de 5%.

 $^{^4~}$ As κ populações podem ser vistas como κ heurísticas aplicadas a um determinado conjunto de n instâncias.

de significância pré-estabelecido. Os grupos equivalem às heurísticas que estão sendo consideradas no experimento que está sendo avaliado.

Para obter o valor F tabelado (F_{tab}) na Tabela de Distribuição F é necessário calcular os graus de liberdade do numerador e denominador da referida tabela. Como considera-se no teste ANOVA a comparação de κ grupos, existem $\kappa - 1$ graus de liberdade associados às variâncias entre grupos (graus de liberdade do numerador). Tendo uma amostra de tamanho n (n instâncias consideradas no experimento para cada heurística), sabe-se que cada um dos κ grupos contribui com n-1 graus de liberdade. Logo, existem $n-\kappa$ graus de liberdade associados às variâncias dentro dos grupos (graus de liberdade do denominador).

Tendo F_{calc} e F_{tab} , rejeita-se a hipótese nula quando F_{calc} é maior do que F_{tab} . Quando H_0 for rejeitada pela estatística do teste F, pode-se concluir com o nível de significância pré-estabelecido, que pelo menos uma das médias difere das demais. Por outro lado, caso H_0 não seja rejeitada, é possível afirmar que as variâncias não são estatisticamente diferentes, que as amostras pertencem à mesma população e provavelmente estão normalmente distribuídas. Em quaisquer destes casos, sem realizar mais testes estatísticos ou uma análise visual dos dados, não se pode determinar quais das heurísticas se destacam. É necessária a utilização de uma segunda técnica estatística para uma avaliação mais precisa das heurísticas. Neste trabalho, o teste *t-Student* foi a segunda técnica considerada para comparar as heurísticas duas a duas.

Para realizar a comparação das heurísticas duas a duas usando o teste t-Student, é necessário obter o valor de t tabelado (t_{tab}) na Tabela de Distribuição t e calcular um valor t_{calc} para todas as heurísticas combinadas duas a duas. Logo, se existem κ heurísticas, serão calculados $\frac{\kappa^2 - \kappa}{2}$ valores t_{calc} . O valor de t_{calc} é calculado a partir da fórmula D.2 descrita no Apêndice D. Para obter t_{tab} , é necessário calcular o grau de liberdade do numerador e do denominador da Tabela de Distribuição t. O grau de liberdade do numerador é dado pelo tamanho da amostra (quantidade de instâncias). Para calcular o grau de liberdade do denominador é necessário estabelecer um nível de significância para o referido teste. Se a hipótese nula que será testada utiliza afirmação da igualdade entre as médias das duas heurísticas consideradas ($H_0: \mu_1 = \mu_2$), então será possível testar as seguintes hipóteses alternativas: $H_1: \mu_1 > \mu_2$; $H_1: \mu_1 < \mu_2$; $H_1: \mu_1 \neq \mu_2$. Se a hipótese alternativa considerada for $H_1: \mu_1 \neq \mu_2$, então o grau de liberdade do denominador é igual ao nível de significância estabelecido. Caso contrário, é igual a metade do nível de significância estabelecido, já que apenas considerará uma das extremidades da calda da curva de distribuição t. A tomada de decisão a partir do valor calculado t_{calc} da estatística t seguiu duas estratégias: a clássica de Neyman e Pearson, com um nível de significância de 0,025% e a de Fisher, considerando o valor descritivo de p^5 (ambas as estratégias utilizadas são brevemente descritas no Apêndice D).

Primeiramente, utilizou-se a abordagem de Neyman e Pearson para a tomada de decisão. Nesta abordagem, se o valor de t_{calc} for menor ou igual ao valor de t_{tab} , então rejeita-se H_0 . Caso contrário, utiliza-se uma segunda abordagem, a de Fisher, que compara o valor de t_{calc} com o valor p, que corresponde a probabilidade de obter uma amostra, no mínimo, tão extrema quanto a amostra dada. Se $t_{calc} > p$, então rejeita-se H_0 . Após comparar todas as heurísticas do experimento duas a duas usando o teste t-Student, propõe-se neste trabalho a consolidação dos resultados obtidos em uma matriz quadrada $T_{\kappa \times \kappa}$, onde:

$$\begin{split} T[i][j] &= -, \, \forall i = j \quad \Rightarrow \quad \text{Não há comparação da heurística com a própria;} \\ T[i][j] &= H_0, \, \forall i \neq j \quad \Rightarrow \quad \text{Comparando a heurística } i \text{ com a } j, \, H_0 \text{ foi aceita;} \\ T[i][j] &= H_1, \, \forall i \neq j \quad \Rightarrow \quad \text{Comparando a heurística } i \text{ com a } j, \, H_0 \text{ foi rejeitada.} \end{split}$$

O objetivo desta tabela é fornecer uma visão global das diferenças estatísticas identificadas pelo teste *t-Student*, para então tentar determinar quais das heurísticas se destacaram. Quando esta consolidação não foi suficiente para obter um resultado conclusivo, foram consideradas algumas das métricas descritas em (RESENDE et al., 2010) para complementar a análise dos dados:

- Best: Para cada instância, Best é o valor da melhor solução obtida entre todas as execuções dos algoritmos que estão sendo comparados;
- #Best: Para cada algoritmo A, #Best equivale ao número de instâncias para as quais o valor da melhor solução obtida com o algoritmo A é igual a Best;
- Dif%: Para cada algoritmo A, Dif% é a diferença percentual relativa entre o valor da melhor solução encontrada pelo algoritmo A e Best;
- *MDif%*: É a média de *Dif%* sobre todas as instâncias consideradas em um determinado experimento;

Esta seção descreveu a metodologia que será utilizada para avaliar todas as heurísticas propostas nesta tese. O nível confiança determinado para os testes estatísticos ANOVA e *t-Student* foi de 99,99% e 99,975% respectivamente, para todos os experimentos avaliados.

⁵ O valor p é a probabilidade de obter uma amostra, no mínimo, tão extrema quanto a amostra dada. Se p é muito pequeno, então deve-se rejeitar a hipótese nula.

A seção a seguir apresenta o resultado da avaliação das heurísticas de construção e busca local.

O nível de significância que estabelece a margem de erro máxima aceitável no teste ANOVA foi estabelecido em 0,01%, assim como no teste *t-Student* foi estabelecido em 0,025%. Todas as análises realizadas nesta tese com estas estatísticas utilizaram estes valores.

5.4 Experimentos computacionais com as heurísticas de construção e de busca local

Esta seção apresenta os resultados dos testes empíricos realizados com as heurísticas de construção e de busca local propostas neste trabalho para o PRR e para o PRRG. O objetivo desta seção é selecionar a melhor heurística construtiva entre as propostas na Seção 5.1 e, a partir desta, selecionar entre as buscas locais 2-DropAdd, 3-DropAdd, 2-AddDrop e 3-AddDrop, aquela que obteve o melhor desempenho. Tanto a melhor heurística construtiva quanto a melhor heurística de busca local identificadas neste experimento serão utilizadas para compor versões do GRASP básico, juntamente com as buscas locais baseadas na meta-heurística VNS que foram propostas na Seção 5.2.4.

Apesar de existirem outros trabalhos na literatura que abordam os problemas em estudo nesta tese, nenhuma instância ou código fonte dos algoritmos foi cedido pelos outros autores para que fosse possível realizar uma comparação com as abordagens heurísticas existentes. Portanto, os experimentos realizados neste trabalho consideram 72 instâncias para o PRRG (GI e GII) e 72 instâncias para o PRR (GIII e GIV), com o conjunto de vértices do grafo variando de 15 a 1000 unidades, sendo todas estas 144 instâncias geradas como descrito no Apêndice A.

Os cinco primeiros números primos foram tomados como sementes para a geração de números aleatórios, sendo estas sementes iguais a 3, 5, 7, 11 e 13. O algoritmo Mersenne Twister, proposto por Matsumoto e Nishimura (MATSUMOTO; NISHIMURA, 1998), foi usado como gerador de números pseudoaleatórios (randômicos) em todos os algoritmos propostos nesta tese. Assim, cada algoritmo (construção e busca local) executou cinco vezes para cada uma das instâncias, considerando sementes distintas.

Todas as heurísticas descritas neste capítulo foram implementadas utilizando a linguagem de programação C++, com o compilador g++ (versão 4.4.1 - 4Ubuntu9) e foram executadas em um computador com processador Intel® CoreTM i7 CPU 870, com 2.93 GHz de frequência, 8 MB de memória cache e 8 GB de memória RAM. Como sistema operacional foi utilizada a distribuição x86_64 GNU/Linux Ubuntu, com kernel na versão 2.6.31-20-generic.

Experimento computacional com as heurísticas de construção

Assim como descrito na Seção 5.3, para comparar o desempenho dos algoritmos propostos na obtenção de uma solução inicial para o PRR e para o PRRG, foi realizado um estudo de inferência estatística nos dados obtidos no experimento computacional que considerou as seguintes heurísticas construtivas:

- **HC-1:** IMB-PRR, com a inicialização da rota feita de acordo com o critério *PorNumWsCobertos*;
- HC-2: IMD-PRR;
- **HC-3:** IMD_{MaxMin}-PRR;
- **HC-4:** IMD_{MinMax}-PRR;
- HC-5: IMP-PRR;
- **HC-6:** IMB-PRR, com a inicialização da rota feita de acordo com o critério *EscolhaAleatoriaEmT*;
- HC-7: AdicaoSucessiva;
- HC-8: RemocaoSucessiva.

Cada heurística executou 5 vezes para cada instância, sendo cada execução associada a uma das sementes definidas para a geração de números aleatórios. Foram consideradas as 72 instâncias geradas para o PRRG (GI e GII) e as 72 instâncias geradas para o PRR (GIII e GIV). Os testes foram realizados para as instâncias originais e reduzidas. As instâncias de GI e GII foram reduzidas considerando a sequência 1 para a aplicação das regras propostas na Seção 3.2.2, já as instâncias de GIII e GIV foram reduzidas considerando a sequência 1 para a aplicação das regras propostas na Seção 3.1.2.

Experimentos preliminares foram realizados para avaliar o desempenho das heurísticas construtivas AdicaoSucessiva e RemocaoSucessiva quando o parâmetro α era variado. O objetivo destes experimentos era identificar, mesmo que de forma empírica, o melhor valor do parâmetro α para o conjunto de instâncias consideradas. Todas as 144 instâncias geradas neste trabalho foram utilizadas neste experimento e nenhuma regra de redução foi aplicada a estas instâncias. Foram testados vinte e um valores para o parâmetro α ,

que variou no intervalo [0, 1], a um fator de 0,05. Cada uma das heurísticas executou cinco vezes para cada valor α , sendo cada execução associada a uma das sementes descritas anteriormente. Portanto, cada um dos algoritmos obteve 105 soluções para cada instância. Considerando as métricas MDif% e #Best, concluiu-se que o melhor desempenho das heurísticas AdicaoSucessiva e RemocaoSucessiva foi alcançado com $\alpha = 0, 15$ e $\alpha = 0, 25$ respectivamente.

O teste de ANOVA para as heurísticas construtivas considerou as oito heurísticas enumeradas anteriormente, com as heurísticas AdicaoSucessiva e RemocaoSucessiva tendo o valor do parâmetro α fixado em 0, 15 e 0, 25 respectivamente.

Sendo μ_i a média aritmética do custo das soluções obtidas em todas as execuções da heurística HC-i para cada uma das instâncias que compõem a amostra do experimento, as hipóteses verificadas no teste de ANOVA para as heurísticas construtivas foram:

- H_0 : A qualidade das soluções obtidas não depende da heurística construtiva utilizada, ou seja, $\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6 = \mu_7 = \mu_8$;
- H_1 : A qualidade das soluções depende da heurística utilizada, ou seja, nem todas as μ_{κ} médias são iguais, com $\kappa = 1, 2, ..., 8$.

Resultados obtidos para as instâncias originais e reduzidas do PRRG

A amostra considerada era composta de todas as instâncias de GI e GII, totalizando 72 instâncias. Utilizando-se a estatística F na amostra composta das instâncias originais, obteve-se o valor calculado F_{calc} igual a 0,01896. Com este resultado e considerando um grau de confiança de 99,95%, a hipótese nula H_0 não foi rejeitada. Quando foi considerada a amostra composta das instâncias reduzidas, obteve-se o valor calculado F_{calc} igual a 0,08241, que também não rejeita a hipótese nula H_0 . Portanto, conclui-se com base nestas informações que as amostras pertencem a populações com o mesmo valor médio e que quaisquer diferenças encontradas são, provavelmente, devidas ao acaso, não sendo identificada pelo teste de ANOVA nenhuma diferença estatisticamente significativa entre as oito heurísticas construtivas. O teste estatístico *t-Student* foi utilizado para verificar se existia diferença sistemática atuando entre as heurísticas construtivas quando comparadas duas a duas.

O teste *t-Student* foi aplicado para comparar as oito heurísticas construtivas duas a duas, totalizando 28 comparações, assim como previsto na Seção 5.3. Este cálculo foi realizado tanto com a amostra composta das instâncias originais quanto das reduzidas.

Comparando os 28 valores de t_{calc} obtidos para cada par de heurísticas em ambas as amostras com o valor de t_{tab} , assim como sugere a abordagem de Pearson e Neyman, não foi possível rejeitar a hipótese nula em nenhum dos 56 casos (28 para as instâncias originais e 28 para as reduzidas). O mesmo ocorreu quando estes valores de t_{calc} foram comparados com p, assim como sugere a abordagem de Fisher. Como em nenhum dos casos analisados houve rejeição da hipótese nula, não há necessidade de consolidar os resultados do teste t-Student na tabela proposta na Seção 5.3.

Com uma margem de erro de no máximo 0,01%, conclui-se com base nos dois testes estatísticos considerados, que a qualidade média das soluções obtidas não depende da heurística construtiva utilizada, assim como determina a hipótese nula estabelecida. Logo, para que a escolha da melhor heurística construtiva não fosse realizada de forma aleatória entre as oito propostas avaliadas pelos testes de ANOVA e *t-Student*, foram utilizadas as métricas descritas na Seção 5.3. A Tabela 5.1 sumariza os dados destas métricas tanto para as instâncias originais quanto para as instâncias reduzidas associadas ao PRRG (GI e GII).

Heurísticas	#I	Best	MDif% Tempo Méd		fédio (seg)	
Construtivas	Original	Reduzido	Original	Reduzido	Original	Reduzido
HC-1	35	40	$7,\!22\%$	$5{,}39\%$	$0,\!02$	$0,\!01$
HC-2	17	27	$6,\!11\%$	$5,\!01\%$	$0,\!17$	$0,\!01$
HC-3	43	47	$8,\!06\%$	$6,\!96\%$	$0,\!17$	$0,\!01$
HC-4	48	50	$7{,}53\%$	6,70%	$0,\!18$	$0,\!00$
HC-5	41	43	7,70%	7,04%	$0,\!19$	0,01
HC-6	63	70	1,73%	$0,\!65\%$	$0,\!02$	$0,\!00$
HC-7	48	48	$3{,}09\%$	$1,\!92\%$	$0,\!07$	$0,\!05$
HC-8	55	55	$2,\!55\%$	$2,\!32\%$	$0,\!17$	$0,\!14$

Tabela 5.1: Comparação do desempenho das heurísticas construtivas para o PRRG considerando as instâncias originais e reduzidas

Analisando a Tabela 5.1, pode-se constatar que a heurística HC-6 se destaca entre as demais, tanto para as instâncias originais quanto para as instâncias reduzidas. O número de instâncias para as quais HC-6 obteve o valor da melhor solução foi maior do que todas as outras sete heurísticas. Os menores valores associados a métrica MDif% também são observados para HC-6. Em relação ao tempo de processamento médio para construir as 72 soluções, nota-se, como era esperado, uma redução significativa nestes tempos quando são consideradas as instâncias reduzidas e que, em média, os menores tempos são observados para HC-1 e HC-6, que são as duas variações da heurística IMB-PRR apresentada na Seção 5.1.5. Logo, conclui-se com base nas observações descritas, que a heurística HC-6 será utilizada como procedimento de construção para as demais abordagens propostas nesta tese para o PRRG.

Assim como mostrado na Seção 4.3.5, das 72 instâncias associadas ao PRRG consideradas no experimento aqui reportado, são conhecidas 52 soluções ótimas. Tendo estas informações, realizou-se um comparativo entre as melhores soluções obtidas pelas oito heurísticas, tanto para as instâncias originais, quanto para as instâncias reduzidas e as soluções ótimas conhecidas. Neste comparativo foi possível constatar que a média das diferenças percentuais entre as melhores soluções obtidas para as instâncias originais e a solução ótima foi de 1,58% e foram encontradas 30 soluções ótimas. Em relação às instâncias reduzidas, estes valores são, respectivamente, 1,07% e 32. O resultado completo desta comparação pode ser visto na Tabela E.2 do Apêndice E.

Resultados obtidos para as instâncias originais e reduzidas do PRR

Um procedimento análogo ao realizado para o PRRG descrito anteriormente foi também realizado para o PRR. Neste caso, a amostra considerada era composta de todas as instâncias de GIII e GIV, totalizando 72 instâncias.

Aplicando a estatística F na amostra composta das instâncias originais, obteve-se o valor calculado F_{calc} igual a 0,02129. Tendo sido estabelecido previamente um grau de confiança de 99,95%, a hipótese nula H_0 não foi rejeitada. Para a amostra composta das instâncias reduzidas, obteve-se o valor calculado F_{calc} igual a 0,146664, que também não consegue rejeitar a hipótese nula H_0 . Logo, assim como para o PRRG, conclui-se que as amostras consideradas para o PRR pertencem a populações com o mesmo valor médio e que quaisquer diferenças encontradas são, provavelmente, devidas ao acaso, não sendo identificada pelo teste de ANOVA nenhuma diferença estatisticamente significativa entre as oito heurísticas construtivas. O teste estatístico t-Student também foi utilizado para comparar as heurísticas construtivas duas a duas.

O cálculo do teste *t-Student* foi realizado tanto com a amostra composta das instâncias originais quanto das reduzidas e quando comparados os 28 valores de t_{calc} obtidos para cada par de heurísticas em ambas as amostras com o valor de t_{tab} (abordagem de Pearson e Neyman), não foi possível rejeitar a hipótese nula em nenhum dos casos 56 casos (28 para as instâncias originais e 28 para as reduzidas). De forma análoga, quando estes valores de t_{calc} foram comparados com p (abordagem de Fisher), também não houve rejeição da hipótese nula em nenhum dos casos. Portanto, não se faz necessário consolidar os resultados do teste *t-Student* na tabela proposta na Seção 5.3. Conclui-se com base nos resultados apresentados pelas estatísticas F e t, que a qualidade média das soluções obtidas não depende de nenhuma das oito heurísticas construtivas consideradas quando estas são utilizadas na obtenção de soluções iniciais para o PRR. Logo, a escolha da melhor heurística construtiva para o PRR foi realizada com base nas métricas descritas na Seção 5.3. A Tabela 5.2 sumariza os dados destas métricas tanto para as instâncias originais quanto para as instâncias reduzidas associadas ao PRR (GIII e GIV).

Heurísticas	#Best		#Best MDif%		Tempo Médio (seg)	
Construtivas	Original	Reduzido	Original	Reduzido	Original	Reduzido
HC-1	32	40	$7,\!51\%$	$5{,}38\%$	$0,\!01$	0,00
HC-2	48	60	3,78%	$2,\!05\%$	$0,\!01$	0,00
HC-3	30	32	7,01%	6,75%	$0,\!00$	0,00
HC-4	31	35	$7,\!31\%$	6,96%	$0,\!00$	0,00
HC-5	35	37	7,76%	$7,\!03\%$	$0,\!01$	0,00
HC-6	53	65	$^{3,15\%}$	1,82%	$0,\!00$	0,00
HC-7	27	52	$8,\!03\%$	2,57%	$0,\!04$	0,03
HC-8	57	62	2,93%	1,85%	$0,\!04$	0,03

Tabela 5.2: Comparação do desempenho das heurísticas construtivas para o PRR considerando as instâncias originais e reduzidas

A partir dos dados consolidados na Tabela 5.2, quando observados os valores das métricas MDif% e #Best, pode-se constatar que a heurística construtiva HC-6 obteve o melhor desempenho para os grafos reduzidos, enquanto a heurística HC-8 obteve o melhor desempenho para os grafos originais. Em relação aos tempos médios de processamento, nota-se que estes são menores do que quando estas oito heurísticas são aplicadas ao PRRG (vide tempo na Tabela 5.1).

Em relação às 72 instâncias associadas ao PRR consideradas no experimento aqui reportado, são conhecidas 52 soluções ótimas, assim como pode ser constatado na Seção 4.3.5. Comparando as melhores soluções obtidas pelas oito heurísticas (instâncias originais e reduzidas) e as soluções ótimas conhecidas, constatou-se que a média das diferenças percentuais entre as melhores soluções obtidas para as instâncias originais e a solução ótima foi de 0,44% e foram encontradas 31 soluções ótimas. Em relação às instâncias reduzidas, estes valores são, respectivamente, 0,12% e 45. O resultado completo desta comparação pode ser visto na Tabela E.1 do Apêndice E.

Consolidando os resultados dos testes realizados com as oito heurísticas construtivas para as instâncias originais e reduzidas do PRR e do PRRG, nota-se que a heurística HC-6 obteve o melhor desempenho médio relacionado às métricas MDif% e #Best para as instâncias originais e reduzidas do PRRG e para as instâncias reduzidas do PRR quando comparado aos resultados das demais heurísticas, enquanto obteve um desempenho pior do que a heurística HC-8 somente para as instâncias originais do PRR. Logo, a heurística HC-6 será utilizada como procedimento de construção para as demais propostas deste capítulo.

Experimento computacional com as heurísticas de busca local

O objetivo deste experimento é avaliar desempenho médio de quatro heurísticas de busca local, visando selecionar a melhor delas para ser utilizada nas heurísticas GRASP que serão apresentadas na Seção 5.5.1. As seguintes heurísticas de busca local serão consideradas neste experimento: 2-DropAdd, 3-DropAdd, 2-AddDrop e 3-AddDrop. Todas estas buscas foram descritas na Seção 5.2. A heurística 2Opt-PRR foi aplicada no final de cada uma das buscas com o objetivo de tentar melhorar a qualidade da solução obtida pela busca local através da alteração da ordem de visita de alguns vértices da rota. Vale lembrar que, a partir dos resultados apresentados com as heurísticas de construção, a heurística HC-6 foi utilizada para fornecer a solução inicial de todas as buscas locais.

Cada uma das heurísticas de busca local consideradas neste experimento executou 5 vezes para cada instância, sendo cada execução associada a uma das sementes definidas para a geração de números aleatórios. Foram utilizadas 64 instâncias do PRRG (GI e GII) e 64 instâncias do PRR (GIII e GIV), com o total de vértices variando de 15 a 500. Os testes foram realizados somente para as instâncias reduzidas. As reduções das instâncias de GI e GII foram obtidas considerando a sequência 1 para a aplicação das regras propostas na Seção 3.2.2, já as reduções das instâncias de GIII e GIV utilizaram a sequência 1 para a aplicação das regras propostas na Seção 3.1.2.

Sendo μ_{2AD} , μ_{3AD} , μ_{2DA} e μ_{3DA} respectivamente as médias aritméticas dos custos das soluções obtidas em todas as execuções das heurísticas 2-AddDrop, 3-AddDrop, 2-DropAdd e 3-DropAdd para cada uma das instâncias que compõem a amostra do experimento, as hipóteses verificadas no teste de ANOVA para as heurísticas de busca local foram:

- H_0 : A qualidade das soluções obtidas não depende da heurística de busca local utilizada, ou seja, $\mu_{2AD} = \mu_{3AD} = \mu_{2DA} = \mu_{3DA};$
- H_1 : A qualidade das soluções depende da heurística de busca local utilizada, ou seja, nem todas as μ_{κ} médias são iguais, com $\kappa = 2AD, 3AD, 2DA, 3DA$.

Considerando as buscas locais aplicadas às instâncias do PRRG (amostra com 64 instâncias reduzidas de GI e GII) e utilizando-se a estatística F, obteve-se o valor calculado F_{calc} de 0,61332. Com este resultado e considerando um grau de confiança de 99,95%, a hipótese nula H_0 não foi rejeitada. Logo, o teste de ANOVA não identificou diferença estatisticamente significativa entre as quatro heurísticas de busca local consideradas para esta amostra.

O teste estatístico t-Student foi utilizado para verificar se existem diferenças entre as heurísticas de busca local quando comparadas duas a duas. Este teste foi aplicado para um grau de confiança de 99,975% e a tomada de decisão para refutar ou não a hipótese nula (H_0) foi feita seguindo as estratégias de Pearson/Neyman e Fisher. Como foram constatadas diferenças sistemáticas atuando para algumas heurísticas quando comparadas duas a duas, o resultado do teste t-Student com as quatro heurísticas de busca local foi sumarizado na Tabela 5.3, assim como proposto na Seção 5.3.

Tabela 5.3: Resultado do teste *t-Student* com as heurísticas de busca local - Grafos Reduzidos - PRRG

Erro $0,025\%$	2-AddDrop	2-DropAdd	3-AddDrop	3-DropAdd
2-AddDrop	_	H_0	H_1	H_1
2-DropAdd	H_0	—	H_1	H_1
3-AddDrop	H_1	H_1	_	H_0
3-DropAdd	H_1	H_1	H_0	_

Legenda:

 H_0 A hipótese nula **não foi** rejeitada.

 H_1 A hipótese nula **foi** rejeitada.

A partir do resultado apresentado na Tabela 5.3, pode-se verificar que foram identificadas diferenças estatisticamente significativas entre as heurísticas consideradas, sendo facilmente percebido que as heurísticas 3-DropAdd e 3-AddDrop se destacaram sobre as demais (2-DropAdd e 2-AddDrop). Portanto, analisando os valores médios obtidos por estas heurísticas para a amostra considerada nos experimentos, identifica-se que ambas obtiveram um desempenho médio melhor do que as demais heurísticas, fato que descarta a escolha das heurísticas 2-DropAdd e 2-AddDrop para compor as versões GRASP que serão apresentadas a seguir. Para saber qual das heurísticas que se destacaram no teste ttinha um melhor desempenho quando comparadas entre si, foram utilizadas as métricas descritas na Seção 5.3 e os resultados foram sumarizados na Tabela 5.4. Tabela 5.4: Comparação do desempenho das heurísticas de busca local para as instâncias reduzidas do PRRG

Busca Local	#Best	MDif%
3-DropAdd	61	0,78%
3-AddDrop	43	$3,\!46\%$

A partir dos dados da Tabela 5.4 nota-se que a busca 3-DropAdd encontrou 18 melhores soluções (#Best) a mais do que a busca 3-AddDrop. Em relação a métrica MDif%, a diferença entre as buscas é de 2,68%, sendo o pior desempenho atribuído a busca 3-AddDrop.

As estatísticas F e t também foram aplicadas nos resultados do experimento realizado com as quatro buscas locais aplicadas às instâncias reduzidas do PRR (amostra com 64 instâncias reduzidas de GIII e GIV). Utilizando-se a estatística F, obteve-se o valor calculado F_{calc} de 0,00035. Considerando um grau de confiança de 99,95%, a hipótese nula H_0 não foi rejeitada para esta amostra. Logo, assim como para o PRRG, o teste de ANOVA não identificou diferença estatisticamente significativa entre as referidas heurísticas de busca local.

A comparação duas a duas considerando as quatro buscas locais foi realizada pela estatística t, com um grau de confiança de 99,975% e a tomada de decisão para refutar ou não a hipótese nula (H_0) também foi feita seguindo as estratégias de Pearson/Neyman e Fisher. Ao sumarizar em uma tabela o resultado do teste t-Student com as quatro heurísticas de busca local tomadas duas a duas para o PRR, deparou-se com um resultado igual ao apresentado na Tabela 5.3. Logo, consolidando os resultados do teste t com os valores médios obtidos pelas quatro heurísticas de busca local para a amostra considerada para o PRR, identifica-se que as heurísticas 3-DropAdd e 3-AddDrop obtiveram um desempenho médio melhor do que as demais heurísticas, o que possibilita descartar as heurísticas 2-DropAdd e 2-AddDrop.

A fim de comparar o desempenho das heurísticas que se destacaram no teste da estatística t para o PRR, são apresentados na Tabela 5.5 os resultados obtidos utilizandose as métricas descritas na Seção 5.3. Tabela 5.5: Comparação do desempenho das heurísticas de busca local para as instâncias reduzidas do PRR

Busca Local	#Best	MDif%
3-DropAdd	64	$0,\!00\%$
3-AddDrop	46	$3{,}29\%$

Analisando os resultados apresentados na Tabela 5.5, percebe-se que a busca 3-DropAdd encontrou todas as 64 melhores soluções, contra apenas 46 relacionadas a busca 3-AddDrop. A diferença entre as buscas segundo a métrica MDif% é de 3,29%, sendo o pior desempenho novamente atribuído a busca 3-AddDrop.

E importante lembrar que os tempos consumidos pelas buscas locais aumentam proporcionalmente com o tamanho da vizinhança investigada que, por sua vez, aumentam com a cardinalidade de vértices do grafo de entrada. Desta forma, sabe-se que as buscas 2-DropAdd e 2-AddDrop são menos custosas do que as buscas 3-DropAdd e 3-AddDrop quando considerada uma mesma instância. Como o objetivo destes experimentos era avaliar a qualidade das vizinhanças das quatro buscas locais e, portanto verificar qual delas apresentava o melhor desempenho médio para ser utilizada nas heurísticas GRASP que serão apresentadas na Seção 5.5, os tempos de processamento não foram consideradas nas análises, uma vez que a estratégia *best-improving* foi usada nestes experimentos e a estratégia *first-improving* será usada quando estas buscas forem utilizadas nas heurísticas GRASP.

As buscas locais VND-PRR e *VNS&VND-PRR* não foram consideradas nos testes aqui reportados, pois todas serão testadas como buscas locais nas heurísticas GRASP propostas na Seção 5.5.1, juntamente com a heurística 3-DropAdd que se destacou nestes experimentos.

5.5 Variações da meta-heurística GRASP

A meta-heurística GRASP (*Greedy Randomized Adaptive Search Procedure*) foi proposta por Tom Feo e Maurício Resende em 1995 e consiste em um processo iterativo constituído de duas fases: uma fase de construção, na qual uma solução viável é construída, seguida de uma fase de busca local, que visa o aprimoramento da solução obtida na fase anterior (RESENDE; FEO, 1995).

A fase de construção é considerada iterativa e adaptativa, podendo também ser gulosa.

A iteratividade está associada ao fato da solução inicial ser construída elemento a elemento iterativamente. A adaptabilidade relaciona-se ao fato dos benefícios associados à adição de cada elemento serem levados de uma iteração para outra nesta fase. A componente gulosa está relacionada à seleção de cada elemento em uma lista de candidatos, conhecida na literatura como Lista Restrita de Candidatos (LRC), que é definida a partir de uma função gulosa $g(s) : LC \longrightarrow \Re$ responsável por medir o benefício de selecionar cada um dos elementos da LRC. O tamanho da lista restrita é determinado por um parâmetro $\alpha \in [0, 1]$ que controla a aleatoriedade do algoritmo. Sendo $LRC = \{s \in LC \mid g(s) \leq s_i + \alpha(s_s - s_i)\}$, com s_i e s_s sendo o menor e o maior valor associados aos elementos $s \in LC$ por g(s), quando $\alpha = 0$, o procedimento de construção torna-se guloso, pois a lista de candidatos torna-se restrita ao mínimo e quando $\alpha = 1$, a construção torna-se totalmente aleatória, já que todas as opções remanescentes são candidatas para a escolha do próximo elemento da solução.

A fase de busca local tem como finalidade refinar a solução obtida na fase de construção, uma vez que esta não necessariamente representa um ótimo local. Em linhas gerais, a fase de busca local visa a substituição da solução atual pela melhor solução de sua vizinhança. Na maioria dos casos, a busca local encerra quando nenhuma solução de melhor qualidade é encontrada na vizinhança pesquisada. Os principais aspectos responsáveis pelo sucesso de uma busca local consistem na escolha eficiente da estrutura da vizinhança, na eficiência da técnica de busca empregada nesta vizinhança e também da solução inicial fornecida para esta busca.

O Algoritmo 10 apresenta o procedimento básico da meta-heurística GRASP, destacando os módulos de construção e busca local descritos anteriormente. Como pode-se observar, os módulos de construção e busca local que compõem a meta-heurística GRASP são executados *maxIter* vezes, sendo *maxIter* um parâmetro de entrada (linha 2). A realização de múltiplas iterações do GRASP pode ser interpretada como uma seleção estratégica do espaço de soluções, o que auxilia esta meta-heurística a tentar escapar de ótimos locais ainda distantes de um ótimo global.

Apesar de serem encontrados na literatura vários trabalhos que comprovem a eficiência da meta-heurística GRASP na obtenção de soluções de boa qualidade em um tempo computacional razoável, quando comparadas a outras meta-heurísticas projetadas para a solução do mesmo problema, sabe-se que, por causa da independência entre as iterações de um GRASP Básico⁶, os benefícios de uma iteração não são considerados nas iterações

⁶ Chama-se de GRASP Básico todo procedimento desenvolvido considerando o *framework* padrão desta meta-heurística, assim como mostrado no Algoritmo 10.

10 11

subsequentes para a escolha de novas soluções ou para guiar o procedimento de busca em direção às regiões promissoras. Em outras palavras, na versão original nenhuma iteração leva em consideração o histórico de atualizações da solução incumbente ao longo de sua execução. Neste sentido, diversas variações do GRASP Básico tem sido propostas, tais como GRASP Reativo, GRASP com Reconexão de Caminhos, GRASP com memória adaptativa, entre outros (RESENDE; RIBEIRO, 2003).

AI	goritmo 10 Algoritmo basico da neuristica GRASP
1:	Definir a função gulosa $g(.)$ e maxIter;
2:	$x^* \leftarrow \emptyset;$
3:	$custo(x^*) \leftarrow \infty;$
4:	$\mathbf{for} \ \mathbf{k} = 1, \dots, \max \mathrm{Iter} \ \mathbf{do}$
5:	{Fase de Construção:}
6:	$x \leftarrow \emptyset;$
7:	Inicializar a lista de candidatos LC com todos os elementos;
8:	while $LC \neq \emptyset$ do
9:	$s_i \leftarrow min\{g(t) \mid t \in LC\};$
10:	$s_s \leftarrow max\{g(t) \mid t \in LC\};$
11:	$LRC = \{s \in LC \mid g(s) \le s_i + \alpha(s_s - s_i)\};$
12:	Selectionar s aleatoriamente na LRC ;
13:	$x \leftarrow x \cup s;$
14:	end while
15:	{Fase de Busca Local:}
16:	Aplicar busca local em x gerando uma nova solução x' ;
17:	if $custo(x') < custo(x^*)$ then
18:	$x^* \leftarrow x'$
19:	end if
20:	end for
21:	return x^* ;

A Seção 5.5.1 apresenta as propostas baseadas na meta-heurística GRASP realizadas neste trabalho, tanto as que seguem a linha do framework básico desta meta-heurística (Seção 5.5.1), quanto na versão conhecida na literatura por GRASP Reativo (Seção 5.5.2).

5.5.1 GRASP Básico

Visando obter soluções de boa qualidade em um tempo computacional razoável quando comparado aos tempos computacionais dispendidos por métodos exatos para solucionar o PRR e o PRRG (vide resultados apresentados no Apêndice C), esta seção apresenta as variações da meta-heurística GRASP propostas neste trabalho para os referidos problemas. Assim como brevemente discutido no início da Seção 5.5, uma iteração da metaheurística GRASP utiliza um procedimento de construção seguido de um procedimento de busca local. Neste sentido, a Tabela 5.6 apresenta as combinações dos procedimentos de construção e de busca local selecionados após a realização de experimentos computacionais para avaliar as heurísticas de construção e busca local propostas neste trabalho (vide Seção 5.4).

Versões do GRASP Básico	Construção	Busca Local
GRASP_Bas1	HC-6	3-DropAdd
$GRASP_Bas2$	HC-6	VNS-PRR (est. de vizinhança $1 + 3$ -DropAdd)
$GRASP_Bas3$	HC-6	VNS-PRR (est. de vizinhança $2 + 3$ -DropAdd)
GRASP_Bas4	HC-6	VNS&VND-PRR

Tabela 5.6: Versões do GRASP Básico propostas para o PRR e PRRG

A primeira coluna da Tabela 5.6 contém o nome dado a cada uma das variações da meta-heurística GRASP nela apresentada. A segunda e a terceira coluna contém respectivamente o procedimento de construção e busca local que compõem cada uma das variações que se destacaram nos experimentos da Seção 5.4 e, por este motivo, estão relacionadas na referida tabela.

A versão GRASP_Bas4 considera a estrutura de vizinhança 1 como estrutura de vizinhança do VND e considera a estrutura de vizinhança 2 como estrutura de vizinhança do VNS.

5.5.2 GRASP Reativo

O GRASP Reativo é uma variação da meta-heurística GRASP que tem como principal característica o auto ajuste do valor atribuído ao parâmetro α a partir da qualidade das soluções previamente encontradas. Desta forma, ao invés de usar um valor fixo para α (vide Algoritmo 10), esta variante seleciona o valor de α aleatoriamente em um conjunto discreto de valores { $\alpha_0, \alpha_1, \alpha_2, ..., \alpha_n$ }, onde cada valor α_i é associado a uma probabilidade p_i de ser selecionado ($p_1 = \sum_{i=1...n} p_i = 1$).

Estudos realizados em (PRAIS; RIBEIRO, 2000b) mostraram que a utilização de valores distintos de α em diferentes iterações GRASP permite que diferentes listas de candidatos restritas sejam criadas, possibilitando que diferentes soluções sejam construídas, o que poderia não ocorrer se fosse considerado um valor de α fixo.

Em linhas gerais, pode-se dizer que um GRASP Reativo incorpora mecanismos de

memória na fase de construção, fato que na prática, faz com que se um determinado valor de α for mais adaptado à uma instância, então este valor acaba possuindo uma probabilidade p_i mais alta de ser selecionado do que os demais valores. Existem algumas sugestões de estratégias para a variação do parâmetro α na literatura, assim como pode ser visto em (RESENDE; RIBEIRO, 2003), (PRAIS; RIBEIRO, 2000a) e (PRAIS; RIBEIRO, 2000b). Os algoritmos propostos neste trabalho baseiam-se na estratégia *equiprovável*, onde a distribuição de probabilidades dos valores de α permanece inalterada ao longo das iterações (5.5.2.1) e na *qualificação absoluta*, que ajusta a distribuição de probabilidades a cada bloco de iterações (5.5.2.2).

5.5.2.1 GRASP Reativo-EQ

O mecanismo de memória utilizado pelo GRASP Reativo-EQ utiliza diferentes valores do parâmetro α , ajustados de acordo com a qualidade das soluções obtidas em iterações precedentes (fase de treinamento). Ao final destas iterações, escolhe-se <u>o valor</u> de α que produziu os melhores resultados durante o treinamento para executar as iterações remanescentes. O Algoritmo 11 apresenta o procedimento GRASP Reativo-EQ proposto para o PRR e para o PRRG.

Algoritmo 11 Algoritmo da heurística GRASP Reativo-EQ proposto para o PRR/PRRG

- 1: Definir maxIteration, g(.);
- 2: A={ $\alpha_0, \alpha_1, \alpha_2, ..., \alpha_n$ };
- 3: {Inicializações:}
- 4: Q[i]=0, i=0...n {custos das soluções relacionadas aos valores de α };
- 5: $P[i] = \frac{1}{n}, i=0...n;$
- 6: β =percentual de iterações que serão usadas na fase de treinamento;
- 7: {Fase de Treinamento:}
- 8: trainingIterations = $|maxIteration \times \beta|$;
- 9: Realizar trainingIterations iterações GRASP nesta fase;
- 10: Atualizar A, Q ao longo das iterações do treino;
- 11: Atualizar o vetor de probabilidades P para os valores α selecionados no treino;

12: {Iterações GRASP após a fase de Treinamento:}

13: for k = 1,...,(maxIteration - trainingIterations) do

```
14: selecionar \alpha = \alpha_i em A considerando P;
```

- 15: rotaAtual = constroiSolucao(α_i , semente, g(.));
- 16: rotaAtual = realizaBuscaLocal(rotaAtual);
- 17: **if** custo(rotaAtual) < custo(melhorRota) **then**
- 18: melhorRota = rotaAtual;
- 19: **end if**
- 20: end for
- 21: return melhorRota;

O GRASP Reativo-EQ com estratégia equiprovável para a variação do parâmetro α inicia sua execução a partir de uma distribuição de probabilidade uniforme (linha 5). Na fase de treinamento (7 a 11), a distribuição de probabilidade é atualizada com base nas informações coletadas nestas iterações, tal como a média dos custos das soluções obtidas para cada valor de α . Procedendo a atualização desta forma, a maior probabilidade será associada ao valor α com o qual foram obtidas as melhores soluções e, consequentemente, este possuirá mais chance de ser escolhido nas iterações seguintes (linhas 12 a 20).

5.5.2.2 GRASP Reativo-QA

A atualização das probabilidades no GRASP Reativo com estratégia para variação do parâmetro α baseada na regra *qualificação absoluta* foi proposta em (PRAIS; RIBEIRO, 2000b). Neste caso, a distribuição de probabilidade é alterada periodicamente a cada bloco de iterações realizadas com base nas informações coletadas na fase de busca local. A quantidade de iterações de cada bloco corresponde a um número pré-fixado fornecido como parâmetro. O Algoritmo 12 apresenta o procedimento GRASP Reativo-QA proposto para o PRR e para o PRRG.

Algoritmo 12 Algoritmo da heurística GRASP Reativo-QA proposto para o PRR/PRR

```
1: Definir maxIteration, blockIterations, q(.);
 2: A={\alpha_0, \alpha_1, \alpha_2, ..., \alpha_n};
 3: {Inicializações:}
 4: Q[i]=0, i=0...n {custos das soluções relacionadas aos valores de \alpha};
5: P[i] = \frac{1}{n}, i=0...n;
 6: Qtde[i]=0, i=0...n
 7: for k = 1, \dots, maxIteration do
      selectionar \alpha = \alpha_i em A considerando P;
 8:
      Qtde[i] = Qtde[i]+1;
9:
      rotaAtual = constroiSolucao(\alpha_i, semente, q(.));
10:
      rotaAtual = realizaBuscaLocal(rotaAtual);
11:
12:
      if custo(rotaAtual) < custo(melhorRota) then
         melhorRota = rotaAtual;
13:
      end if
14:
      Q[i] = Q[i] + custo(rotaAtual);
15:
      if ((k \mod blockIterations) == 0) then
16:
         fazer Q[j] = Q[j] / Qtde[j], \forall j=0,...,m;
17:
         fazer Q[j] = custo(melhorRota) / Q[j], \forall j=0,...,m;
18:
         somaGeralQ = soma de todos os valores em Q;
19:
         fazer P[j] = Q[j] / somaGeralQ, \forall j=0,...,m;
20:
      end if
21:
22: end for
23: return melhorRota;
```

Os parâmetros do Algoritmo 12 são (linhas 1 e 2): a função gulosa g(.), que mede o benefício de selecionar cada um dos elementos da LRC; o número máximo de iterações GRASP que serão realizadas (maxIterations); a quantidade de iterações GRASP que serão realizadas antes de cada atualização de probabilidade (blockIterations) e o conjunto discreto de valores do parâmetro α que serão considerados (A). As probabilidades associadas a cada α_i são inicializadas seguindo uma distribuição uniforme (linha 5). A cada iteração GRASP, um valor para o parâmetro α é selecionado em A para realizar a construção da LRC, considerando a distribuição de probabilidade P (linhas 8 a 10). Uma busca local é realizada a partir da solução obtida na fase de construção (linha 11) e o custo da solução retornada por esta busca é acumulado em Q[i] (linha 15). Se uma solução aprimorante foi encontrada, a solução incumbente é atualizada (linhas 12 a 14). Caso tenham sido realizadas blockIterations iterações desde a última atualização de probabilidades, então a distribuição de probabilidade é alterada com base nas informações coletadas nestas iterações (linhas 16 a 21).

5.6 Iterated Local Search (ILS)

A meta-heurística ILS (LOURENÇO; MARTIN; STÜTZLE, 2001) pressupõe que os ótimos locais de um problema de otimização podem ser gerados a partir de perturbações na solução ótima local corrente.

Em linhas gerais, se f(.) é uma função que avalia uma solução do problema \wp (sem perda de generalidade, suponha que \wp é um problema de minimização) e S é o conjunto de todas as possíveis soluções de \wp , pode-se dizer que o objetivo da meta-heurística ILS é explorar um subconjunto de $S^* \subset S$, caminhando de um ótimo local s^* para outro próximo a este, aplicando-se uma perturbação em s^* de forma a gerar uma solução intermediária $s' \in S$. Um procedimento de busca local é então aplicado a s' para encontrar um ótimo local $s^{*'} \in S^*$. Caso $s^{*'}$ satisfaça o critério de aceitação estabelecido, este se tornará o próximo elemento do caminho a ser percorrido em S^* . Caso contrário, retorna-se à solução s^* .

O Algoritmo 13 apresenta o procedimento básico da meta-heurística ILS. Como podese observar, os módulos de busca local utilizados dentro (linha 5) e fora (linha 2) do laço principal não precisam ser necessariamente iguais. Na prática, a busca local realizada logo após a obtenção da solução inicial tende a ser mais custosa do que a realizada dentro do laço, visando obter um ótimo local de boa qualidade para iniciar o caminho (s^*) , já que esta busca será realizada apenas uma vez. O critério de aceitação (linha 6) determina quando uma nova solução deve ou não ser aceita no caminho $(s^{*'})$. Este critério pode ser usado para controlar o balanço entre a intensificação e a diversificação da busca. Se somente soluções aprimorantes forem aceitas, isto é, se $f(s^{*'}) < f(s^{*})$, então pode-se dizer que a intensificação da busca é forte.

Algoritmo 13 Algoritmo básico da meta-heurística ILS
1: $s_0 \leftarrow constroiSolucaoInicial();$
2: $s^* \leftarrow realizaBuscaLocal_1(s_0);$
3: while critério de parada não satisfeito do
4: $s' \leftarrow realizaPerturbacao(historico, s^*);$
5. $s^{*'} \leftarrow realizaBuscaLocal_2(s');$
6: if $criterioDeAceitacao(s^*, s^{*'}, historico)$ then
7: $s^* \leftarrow s^{*'};$
8: end if
9: end while
10: return s^* ;

Para que o procedimento que compõe a meta-heurística ILS encontre regiões promissoras em S, as perturbações aplicadas às soluções do caminho precisam ser suficientemente fortes, permitindo que a busca local explore diferentes soluções no espaço de busca e, suficientemente fracas, para evitar que o algoritmo se comporte como um procedimento de múltiplos reinícios aleatórios. Relatos da literatura mostram que perturbações determinísticas podem conduzir a formação de ciclos. Para evitar esta ciclagem, pode-se tornar a perturbação aleatória ou adaptativa (LOURENÇO; MARTIN; STÜTZLE, 2003).

Para a heurística ILS-PRR proposta neste trabalho para a solução do PRR e PRRG, são feitas as seguintes considerações:

- Os parâmetros de entrada da heurística são: a instância e o problema que será resolvido (no caso PRR ou PRRG);
- O critério de parada será o mesmo adotado para as heurísticas GRASP propostas: número máximo de iterações;
- A heurística IMB-PRR (HC-6) será utilizada na construção da solução inicial, já que se destacou nos experimentos realizados na Seção 5.4. Entretanto, qualquer uma das propostas de heurísticas construtivas descritas na Seção 5.1 poderia ser considerada (linha 1 do Algoritmo 13);

- A busca local externa (linha 2 do Algoritmo 13) será realizada pela heurística VNS-PRR (vide Seção 8);
- A estratégia adotada na heurística ILS-PRR para perturbar uma solução (linha 4 do Algoritmo 13) é escolhida aleatoriamente no seguinte conjunto de movimentos: inserir e remover k vértices aleatoriamente (k variando de 1 a 5) ou inserir e remover k arestas aleatoriamente (k variando de 2 a 3);
- A busca local interna (linha 5 do Algoritmo 13) será realizada pela heurística VND-PRR (vide Seção 9), utilizando a mesma estrutura de vizinhança definida para a heurística VNS-PRR que será realizada após a construção da solução inicial (linha 1 do Algoritmo 13).

5.7 Estratégia de intensificação utilizando a Reconexão por Caminhos

A Reconexão por Caminhos (RC), conhecida na literatura por *Path Relinking* (PR), foi proposta por (GLOVER, 1996) para ser utilizada como estratégia de intensificação e diversificação na Busca Tabu (BT). O objetivo desta técnica é gerar novas soluções a partir da exploração de uma trajetória que conecta soluções de boa qualidade.

Sendo s_B e s_G duas soluções de boa qualidade, respectivamente chamadas de solução base e solução guia, define-se um caminho entre s_B e s_G como sendo um espaço de soluções gerado através da seleção de movimentos que leve a solução base à solução guia. Logo, um caminho entre s_B e s_G é obtido produzindo-se uma sequência de soluções intermediárias onde, a cada iteração, s_i é obtida a partir de s_{i-1} através da inserção de movimentos que devem ser aplicados à s_B para que esta atinja s_G .

A utilização da RC nas soluções obtidas por procedimentos GRASP foi proposta em (LAGUNA; MARTI, 1999). A RC pode ser utilizada como uma estratégia de intensificação a cada ótimo local encontrado após a fase de busca local ou como uma estratégia de pós-otimização, entre os pares de soluções elite⁷.

A estratégia de RC proposta neste trabalho para o PRR e para o PRRG propõe conectar uma solução base s_B a uma solução guia s_G a partir dos seguintes passos:

⁷ O termo elite refere-se às soluções de boa qualidade que são geradas ao longo da execução de um algoritmo.
- 1. Inicialmente calcula-se a diferença simétrica $|\Delta(s_B, s_G)|$ entre a solução base e a solução guia. Esta diferença é obtida somando-se a quantidade de vértices que estão em s_B e não em s_G (vértices que serão removidos de s_B) à quantidade de vértices que estão estão em s_G e não em s_B (vértices que serão inseridos em s_B). O valor de $|\Delta(s_B, s_G)|$ corresponde a quantidade de movimentos⁸ que serão realizados para conectar s_G à s_B ;
- 2. A cada iteração, o melhor movimento de $\Delta(s_B, s_G)$ ainda não aplicado em s_B é efetuado até que se alcance a s_G . Os movimentos de $\Delta(s_B, s_G)$ correspondem à remoção dos vértices que estão em s_B e não em s_G e à inserção dos vértices que estão em s_G e não em s_B . Após a aplicação do movimento, ele é removido de $\Delta(s_B, s_G)$ e a melhor solução encontrada é tomada como a nova solução intermediária;
- 3. Quando não mais existirem movimentos em $\Delta(s_B, s_G)$, a convergência de s_B para s_G está completa e o procedimento encerra retornando a melhor solução encontrada no caminho.

Cada movimento realizado nas soluções intermediárias do caminho pode transformar uma solução viável em inviável ou inviável em viável. Em geral, somente os movimentos de remoção podem inviabilizar uma solução, enquanto os movimentos de inserção podem viabilizá-la. Portanto, propõe-se realizar a avaliação de cada movimento de $\Delta(s_B, s_G)$ da seguinte forma:

Após a realização do movimento, deve-se verificar a viabilidade da nova solução intermediária:

- Se a nova solução intermediária for viável, aplica-se o procedimento 2Opt-PRR proposto na Seção 5.2.3 nesta solução e o custo da solução retornado por este procedimento é então correspondente a avaliação do movimento associado. Caso este movimento seja o melhor movimento da iteração correspondente, a nova solução intermediária do caminho será correspondente à solução antes da aplicação do procedimento de busca local;
- Se a nova solução intermediária for inviável, então aplica-se um procedimento para viabilizá-la. Este procedimento insere os vértices de T que estiverem fora da rota de modo a gerar o menor acréscimo no custo total desta e/ou insere vértices que cubram os vértices de W descobertos. Neste último caso, escolhe-se aleatoriamente

⁸ Sabe-se que se $|\Delta(s_B, s_G)| = k$, então k + (k-1) + (k-2) + ... + 1 soluções serão visitadas no caminho.

um vértice da vizinhança do vértice de W descoberto e insere-o na rota de modo a gerar o menor acréscimo no custo total. Após viabilizar a solução, também aplica-se o procedimento 2Opt-PRR proposto na Seção 5.2.3 na solução viabilizada e o custo da solução retornada por este procedimento é então correspondente ao movimento associado. Caso este movimento seja o melhor movimento da iteração correspondente, a nova solução intermediária do caminho será correspondente à solução antes da viabilização.

Para exemplificar estratégia de RC proposta neste trabalho para o PRR e para o PRRG, considere as seguintes soluções base e guia:

 $s_B = 0 - 1 - 2 - 3 - 4 - 5 - \dots - 14 - \dots - 20 - 21 - 22$ $s_G = 0 - 1 - 2 - 3 - 4 - 5 - \dots - 14 - \dots - 17 - \dots - 19$

A diferença simétrica entre a solução base e a solução guia é dada por $|\Delta(s_B, s_G)| = 5$, pois deve-se remover da solução base os vértices 20, 21, 22 e inserir os vértices 17 e 19. Assim sendo, serão realizados cinco movimentos, o que equivale a 15 soluções visitadas no caminho. A Figura 5.13 ilustra todo o processo realizado pela estratégia de RC proposta, partindo da solução base até a solução guia descritas anteriormente.



Figura 5.13: Exemplo da utilização da estratégia de RC proposta para o PRR e para o PRRG

O movimento de remoção de um vértice está identificado na Figura 5.13 pelo sinal (-) seguido do índice do vértice que será removido. De forma análoga, o movimento de inserção está identificado pelo sinal (+) seguido do índice do vértice que será inserido.

Os movimentos destacados na cor azul, indicam que a solução obtida após o referido movimento é viável e os destacados na cor salmão indicam que a solução obtida após o movimento é inviável. As soluções descatadas em cada passo foram selecionadas como soluções intermediárias do caminho.

Sugere-se neste trabalho utilizar a RC descrita anteriormente nas soluções do PRR e do PRRG obtidas nas variações GRASP propostas como uma estratégia de intensificação. A ideia é utilizar a RC nas Φ^9 últimas iterações entre a solução retornada pela busca local e uma solução do conjunto elite selecionada aleatoriamente. A quantidade de iterações que serão consideradas para a aplicação da RC e o tamanho do conjunto elite serão definidas em experimentos preliminares.

Segundo (RIBEIRO; UCHOA; WERNECK, 2002), para realizar a escolha da alternativa mais apropriada para explorar a trajetória entre as soluções em uma RC, é preciso considerar o tempo de processamento e a qualidade da solução. Se somente uma trajetória deve ser investigada, melhores soluções tendem a ser obtidas quando a RC usa como solução inicial a solução de melhor qualidade (menor custo no caso do PRR e do PRRG), pois como a vizinhança da solução inicial (s_B) é explorada com mais profundidade do que a vizinhança da solução final (s_G) , esta estratégia oferece mais chances do algoritmo investigar mais detalhadamente a vizinhança da solução de melhor qualidade. Com base nestas informações, será utilizada neste trabalho a trajetória que adota como solução base a solução de melhor qualidade.

5.8 Experimentos computacionais com as heurísticas GRASP e ILS

Esta seção apresenta os resultados dos testes realizados com as heurísticas GRASP (incluindo as versões básicas e as versões com memória adaptativa) e com a heurística ILS propostas neste trabalho para o PRR e para o PRRG.

Os experimentos que seguem nesta seção estão organizados em quatro grupos:

Testes com duas versões do GRASP básico para a calibração do parâmetro α. O objetivo destes testes é determinar o valor para o parâmetro α que será usado nas execuções de todas as versões do GRASP básico apresentadas na Seção 5.5.1;

 $^{^9}$ Φ é um parâmetro que corresponde a um valor percentual.

- Testes com as versões do GRASP básico apresentadas na Seção 5.5.1. O objetivo destes testes é selecionar o melhor GRASP básico para ser utilizado nas versões com memória adaptativa;
- Testes com as versões adaptativas do GRASP, sendo GRASP Reativo-EQ e GRASP Reativo-QA, considerando a configuração (construção e busca local) do GRASP que se destacou no experimento anterior;
- Testes com o melhor GRASP adaptativo identificado no experimento anterior e a heurística ILS-PRR apresentada na Seção 5.6;

Nos testes para a calibração do parâmetro α para as versões do GRASP básico foram considerados vinte e um valores para o referido parâmetro, que variou no intervalo [0, 1], a um fator de 0,05. Foram consideradas neste experimento as instâncias do PRR e do PRRG com até 300 vértices, totalizando 60 instâncias para cada problema abordado. A sequência 1 para a aplicação das regras propostas na Seção 3.2.2 foram utilizadas previamente nas instâncias de GI e GII. Para as instâncias de GIII e GIV, foi considerada sequência 1 para a aplicação das regras propostas na Seção 3.1.2. Somente as versões GRASP_Bas1 e GRASP_Bas2 foram consideradas nestes testes. Cada versão executou cinco vezes para cada valor de α , sendo uma associada a cada semente. A busca local que compõe a versão GRASP_Bas2 teve como critério de parada o total de 50 iterações. Tanto o GRASP_Bas1 quanto o GRASP_Bas2 tiveram como critério de parada o total de 100 iterações (este valor foi obtido em testes preliminares). A partir dos resultados alcançados e utilizando as métricas MDif% e #Best, constatou-se que o melhor desempenho para as versões testadas foi obtido com $\alpha = 0.45$. Logo, este valor será considerado em todas as versões do GRASP básico na próxima sequência de experimentos.

Todos os demais experimentos desta seção foram executados somente para as instâncias de maiores dimensões (100 a 1000 vértices), já que todas as soluções ótimas de GI e GIII foram alcançadas por ambas as versões GRASP nos experimentos de calibração do parâmetro α , assim como pode ser observado nas Tabelas E.4 e E.3 do Apêndice E que agrupam os resultados obtidos com GRASP_Bas1 ($\alpha = 0.45$).

Testes com as versões do GRASP básico

O objetivo deste experimento é avaliar o desempenho médio das quatro versões do GRASP básico propostas na Seção 5.5.1 visando selecionar a melhor delas para ser utilizada nas versões do GRASP Reativo propostas na seção 5.5.2. As seguintes heurísticas serão consideradas neste experimento: GRASP_Bas1, GRASP_Bas2, GRASP_Bas3 e GRASP_Bas4. A heurística 20pt-PRR foi aplicada no final de cada uma das versões GRASP, assim como foi feito nas heurísticas de busca local.

Cada uma das heurísticas consideradas neste experimento executou 5 vezes para cada instância, sendo cada execução associada a uma das sementes definidas para a geração de números aleatórios. Foram utilizadas 24 instâncias do PRRG (GII) e 24 instâncias do PRR (GIV), com o total de vértices variando de 100 a 1000. Os testes foram realizados somente para as instâncias reduzidas (GII reduzidas considerando a sequência 1 para a aplicação das regras propostas na Seção 3.2.2; GIV reduzidas considerando a sequência 1 para a aplicação das regras propostas na Seção 3.1.2). O critério de parada de cada GRASP é a realização de 100 iterações e das versões VNS é a realização de 50 iterações. A estratégia *first-improving* foi usada nas buscas locais.

Supondo μ_{Gb1} , μ_{Gb2} , μ_{Gb3} e μ_{Gb4} respectivamente as médias aritméticas dos custos das soluções obtidas em todas as execuções das heurísticas GRASP_Bas1, GRASP_Bas2, GRASP_Bas3 e GRASP_Bas4 para cada uma das instâncias que compõem a amostra do experimento, as hipóteses verificadas no teste de ANOVA para estas heurísticas foram:

- H_0 : A qualidade das soluções obtidas não depende da versão GRASP básica utilizada, ou seja, $\mu_{Gb1} = \mu_{Gb2} = \mu_{Gb3} = \mu_{Gb4};$
- H_1 : A qualidade das soluções depende da versão GRASP básica utilizada, ou seja, nem todas as μ_{κ} médias são iguais, com $\kappa = Gb1, Gb2, Gb3, Gb4$.

Considerando as referidas versões GRASP aplicadas às instâncias do PRRG (amostra com 24 instâncias reduzidas de GII) e utilizando-se a estatística F, obteve-se o valor calculado F_{calc} de 0,86152. Com este resultado e considerando um grau de confiança de 99,95%, a hipótese nula H_0 não foi rejeitada. Portanto, o teste de ANOVA não identificou diferença estatisticamente significativa entre as quatro versões GRASP consideradas para esta amostra.

Utilizando o teste estatístico *t-Student* para verificar a existência de diferenças entre as heurísticas GRASP consideradas quando comparadas duas a duas, obteve-se o seguinte resultado considerando as estratégias de Pearson/Neyman e Fisher para refutar ou não H_0 :

Erro $0,025\%$	GRASP_Bas1	GRASP_Bas2	GRASP_Bas3	GRASP_Bas4
GRASP_Bas1		H_1	H_0	H_1
$GRASP_Bas2$	H_1	—	H_1	H_0
GRASP_Bas3	H_0	H_1	—	H_1
$GRASP_Bas4$	H_1	H_0	H_1	_

Tabela 5.7: Resultado do teste *t-Student* com as heurísticas GRASP básicas aplicadas aos grafos reduzidos do PRRG

Legenda:

 H_0 A hipótese nula **não foi** rejeitada.

 H_1 A hipótese nula **foi** rejeitada.

Observando os dados apresentados na Tabela 5.7, nota-se que foram constatadas diferenças sistemáticas atuando para algumas heurísticas GRASP quando comparadas duas a duas: as versões GRASP Bas2 e GRASP Bas4 possuem diferenças sistemáticas em relação as outras duas. Para saber quais apresentaram resultado superior as demais, foram analisados os resultados médios obtidos. A partir desta análise, foi possível identificar que as versões GRASP_Bas2 e GRASP_Bas4 obtiveram um desempenho médio melhor do que as demais heurísticas, fato que descarta a escolha de GRASP_Bas1 e GRASP_Bas3 para ser utilizada nas versões do GRASP Reativo. Para saber qual das heurísticas que se destacaram no teste t tinha um melhor desempenho quando comparadas entre si, foram utilizadas as métricas descritas na Seção 5.3 e os resultados foram sumarizados na Tabela 5.8.

Tabela 5.8: Comparação do desempenho das heurísticas GRASP básicas aplicadas aos grafos reduzidos do PRRG (GII)

Busca Local	#Best	MDif%	Tempo Médio (s)
GRASP_Bas1	9	$9{,}48\%$	$1.762,\!12$
$GRASP_Bas2$	18	$2,\!56\%$	$4.152,\!18$
GRASP_Bas3	13	$8,\!98\%$	$5.423,\!02$
$GRASP_Bas4$	22	0,46%	$7.694,\!07$

Em relação aos tempos computacionais reportados na Tabela 5.8, já era esperado que a versão GRASP Bas1 consumisse menos tempo que as demais, assim como já era esperado que GRASP Bas4 consumisse mais tempo, já que emprega uma busca VND. Entretanto, analisando os valores relacionados nas métricas #Best e MDif% conclui-se que, mesmo consumindo, em média, mais tempo de CPU, é vantajosa a escolha do GRASP Bas4 para ser utilizada nas versões do GRASP Reativo quando o PRRG está sendo considerado.

Quando as versões GRASP consideradas são aplicadas às instâncias do PRR (amostra

com 24 instâncias reduzidas de GIV), utilizando-se a estatística F, obteve-se o valor calculado F_{calc} de 0, 70132. Para um grau de confiança de 99, 95%, a hipótese nula H_0 não foi rejeitada, mostrando que o teste de ANOVA não identificou diferença estatisticamente significativa entre as quatro versões GRASP consideradas. Os resultados com o teste estatístico *t-Student* para verificar a existência de diferenças entre as referidas heurísticas quando comparadas duas a duas, considerando as estratégias de Pearson/Neyman e Fisher para refutar ou não H_0 , obteve-se um resultado consolidado igual ao apresentado quando estava sendo considerada a amostra do PRRG (vide Tabela 5.8). Logo, para completar a análise sobre os resultados obtidos neste experimento, a Tabela 5.9 apresenta os dados quando são consideradas as métricas descritas na Seção 5.3.

Tabela 5.9: Comparação do desempenho das heurísticas GRASP básicas aplicadas aos grafos reduzidos do PRR (GIV)

Busca Local	#Best	MDif%	Tempo Médio (s)
GRASP_Bas1	11	$^{6,67\%}$	$1.032,\!21$
$GRASP_Bas2$	21	$1,\!36\%$	$3.452,\!02$
GRASP_Bas3	16	$4{,}08\%$	$4.213,\!19$
$GRASP_Bas4$	24	$0,\!00\%$	$5.894,\!05$

Observa-se nos dados da Tabela 5.9 que o GRASP_Bas4 encontrou a melhor solução para todas as instâncias da amostra considerada, se destacando também quando aplicado às instâncias do PRR. O GRASP_Bas2 também é a segunda melhor

Testes com as versões adaptativas do GRASP

O objetivo deste experimento é avaliar o desempenho médio das duas versões adaptativas do GRASP: GRASP Reativo-EQ e GRASP Reativo-QA. A heurística 20pt-PRR foi aplicada no final de cada uma das versões GRASP mencionadas.

As heurísticas consideradas neste experimento executaram 5 vezes para cada instância, sendo cada execução associada a uma das sementes definidas para a geração de números aleatórios. Foram utilizadas as mesmas amostras do experimento anterior (versões do GRASP básico). O critério de parada adotado nas versões do GRASP Reativo é a realização de 100 iterações e das versões VNS é a realização de 50 iterações. A estratégia *first-improving* foi usada nas buscas locais, assim como definido anteriormente.

Os parâmetros do GRASP Reativo-EQ são: o conjunto de valores de α foi composto pelos cinco melhores valores identificados nos experimentos para a calibração deste parâmetro para as versões do GRASP básico, que são 0.45, 0.35, 0.2, 0.5 e 0.15. O percentual de iterações da fase de treinamento foi determinado em experimentos preliminares com subconjunto de cinco instâncias de GII e de GIV. O valor escolhido para este parâmetro entre 10%, 15%, 30% e 50% foi igual a 30%.

Para GRASP Reativo-QA, estabeleceu-se os seguintes parâmetros: o mesmo conjunto de valores de α definido para o GRASP Reativo-EQ foi utilizado. Após a realização de experimentos preliminares com as mesmas instâncias e valores considerados no GRASP Reativo-EQ, definiu-se em 10% o percentual de iterações que serão realizadas entre as atualizações de probabilidade.

Seja μ_{GReq} e μ_{GRqa} respectivamente as médias aritméticas dos custos das soluções obtidas em todas as execuções das heurísticas GRASP Reativo-EQ e GRASP Reativo-QA para cada uma das instâncias que compõem a amostra do experimento. Portanto, as hipóteses verificadas no teste de ANOVA para estas heurísticas foram:

- H_0 : A qualidade das soluções obtidas não depende da versão GRASP Reativa utilizada, ou seja, $\mu_{GReq} = \mu_{GRqa}$;
- H_1 : A qualidade das soluções depende da versão GRASP Reativa utilizada, ou seja, nem todas as μ_{κ} médias são iguais, com $\kappa = GReq$ e GRqa.

Aplicando a estatística F sobre os dados obtidos na execução das referidas versões GRASP quando considerada a amostra composta pelas instâncias do PRRG (amostra com 24 instâncias reduzidas de GII) ou pelas instâncias do PRRG, o teste de ANOVA não identificou diferença estatisticamente significativa entre as versões GRASP consideradas. Logo, tendo sido estabelecido um grau de confiança de 99,95%, a hipótese nula H_0 não foi rejeitada. Como o teste t é um caso especial de análise de variância para teste entre dois grupos, já era de se esperar que o mesmo também não identificasse diferenças entre as heurísticas GRASP consideradas quando comparadas duas a duas. Portanto, a comparação entre o GRASP Reativo-EQ e o GRASP Reativo-QA foi feita com base nas informações consolidadas na Tabela 5.10 para o PRRG e na Tabela 5.11 para o PRR.

Como pode ser constatado nas Tabelas 5.10 e 5.11, quando o valor de α é auto ajustável ao longo das iterações do GRASP Reativo em função das soluções obtidas nas iterações precedentes, as soluções neste caso tendem a ser melhores, já que existe maior aderência da heurística às particularidades de cada instância. Ao usar diferentes valores de α em diferentes iterações, diferentes listas restritas de candidatos são geradas, eventualmente levando a construção de diferentes soluções que não seriam construídas caso fosse usado um valor fixo para α . A abordagem reativa melhora o GRASP básico em termos de robustez e qualidade das soluções produzidas, pois permite maior diversificação nas soluções geradas e uma menor dependência dos ajustes dos parâmetros.

Tabela 5.10: Comparação do desempenho das heurísticas GRASP Reativo-EQ e GRASP Reativo-QA aplicadas aos grafos reduzidos do PRRG (GII)

Busca Local	#Best	MDif%	Tempo Médio (s)
GRASP Reativo-EQ	20	$1,\!28\%$	8.019,08
GRASP Reativo-QA	23	0,13%	7.978,34

O aumento observado nos tempos de processamento, em geral, está ligado as soluções de pior qualidade que também são produzidas, tornando a busca local mais lenta nestes casos.

Tabela 5.11: Comparação do desempenho das heurísticas GRASP Reativo-EQ e GRASP Reativo-QA aplicadas aos grafos reduzidos do PRR (GIV)

Busca Local	#Best	MDif%	Tempo Médio (s)
GRASP Reativo-EQ	22	0,78%	6.401,02
GRASP Reativo-QA	24	0,00%	$6.034,\!13$

O procedimento de reconexão por caminhos foi aplicado em ambas as versões do GRASP Reativo consideradas neste experimento. Utilizando um conjunto elite de tamanho 10 e o parâmetro $\Phi = 30\%$ (as demais parametrizações estão definidas na Seção 5.7), os resultados obtidos mostraram uma melhora percentual de cerca de 9,2% nos custos médios das soluções obtidas e um aumento de cerca de 17% nos tempos de processamento. Desta forma, sugere-se utilizar a versão GRASP Reativo-QA com reconexão de caminhos para comparar com a abordagem ILS proposta.

Testes com GRASP Reativo-QA com reconexão de caminhos e a heurística ILS-PRR

O objetivo deste experimento é comparar o desempenho médio de duas abordagens heurísticas distintas propostas para o PRR e para o PRRG neste trabalho: GRASP Reativo-QA com reconexão de caminhos e a heurística ILS-PRR. A heurística 20pt-PRR foi aplicada no final de cada uma das versões GRASP mencionadas.

Ambas as heurísticas consideradas neste experimento executaram 5 vezes para cada instância, sendo cada execução associada a uma das sementes definidas para a geração de números aleatórios. Foram utilizadas as mesmas amostras dos dois experimentos anteriores (GII e GIV reduzidos, tendo 24 instâncias cada grupo). O critério de parada adotado para as duas heurísticas que estão sendo comparadas é a realização de 100 iterações e das versões VNS é a realização de 50 iterações. A estratégia *first-improving* foi usada nas buscas locais, assim como definido para todas os outros experimentos com a heurística GRASP.

Os parâmetros do Reativo-QA são os mesmos definidos no experimento anterior e os demais parâmetros da heurística ILS-PRR foram definidos na Seção 5.6.

Considerando μ_{GRqa} e μ_{ILS} respectivamente as médias aritméticas dos custos das soluções obtidas em todas as execuções das heurísticas GRASP Reativo-QA e ILS-PRR para cada uma das instâncias que compõem a amostra do experimento, as hipóteses verificadas no teste de ANOVA para estas heurísticas foram:

- H_0 : A qualidade das soluções obtidas não depende das heurísticas utilizadas, ou seja, $\mu_{GRqa} = \mu_{ILS};$
- H_1 : A qualidade das soluções depende das heurísticas utilizadas, ou seja, nem todas as μ_{κ} médias são iguais, com $\kappa = GRqa$ e *ILS*.

Seguindo a metodologia descrita na Seção 5.3 para a realização dos testes estatísticos ANOVA e *t-Student*, não foi possível rejeitar H_0 em nenhum dos referidos testes. Logo, a partir destes resultados, é possível afirmar que as variâncias não são estatisticamente diferentes, que as amostras pertencem à mesma população e provavelmente estão normalmente distribuídas.

A Tabela 5.12 sumariza os resultados do experimento quando considerada a amostra de 24 instâncias reduzidas do PRRG (GII) e a Tabela 5.13 apresenta os resultados equivalentes quando a amostra é de 24 instâncias reduzidas do PRR (GIV).

Tabela 5.12: Comparação do desempenho das heurísticas GRASP Reativo-QA com reconexão por caminhos e ILS-PRR quando aplicadas aos grafos reduzidos do PRRG (GII)

Busca Local	#Best	MDif%	Tempo Médio (s)
ILS-PRR	23	$0,\!08\%$	$10.003,\!11$
GRASP Reativo-QA	21	$0,\!29\%$	$9.376,\!17$

A partir dos resultados apresentados, constatou-se que a abordagem utilizada pelo *ILS-PRR* obteve os melhores resultados na média e também encontrou a maior quantidade de melhores soluções por instância, apesar do tempo de CPU ser ligeiramente maior do que o tempo consumido pelo GRASP Reativo-QA com reconexão por caminhos.

Tabela 5.13: Comparação do desempenho das heurísticas GRASP Reativo-QA com reconexão por caminhos e ILS-PRR quando aplicadas aos grafos reduzidos do PRR (GIV)

Busca Local	#Best	MDif%	Tempo Médio (s)
ILS-PRR	24	$0,\!00\%$	$7.971,\!34$
GRASP Reativo-QA	23	0,11%	$7.086,\!12$

Analisando o micro dado deste experimento, constatou-se que a qualidade das soluções iniciais construídas pelo GRASP Reativo-QA foi, na média, superior a qualidade das soluções iniciais construídas na heurística *ILS-PRR*. Logo, acredita-se que as perturbações aplicadas nas iterações da heurística *ILS-PRR*, provavelmente contribuíram de forma decisiva para que a busca não ficasse presa em ótimos locais de baixa qualidade.

Capítulo 6

Conclusões

Esta tese abordou duas variantes do clássico Problema do Caixeiro Viajante (PCV) ainda pouco exploradas na literatura, conhecidas como Problema de Recobrimento por Rotas (PRR) e Problema de Recobrimento por Rotas Generalizado (PRRG). Ambos são classificados como NP-difíceis (CURRENT, 1981) e, portanto, sabe-se que o uso exclusivo de métodos exatos na solução destes problemas fica restrito, em via de regra, às instâncias de pequeno porte, fato que motivou a proposta de métodos heurísticos para resolvê-los.

Foram propostas novas abordagens heurísticas, tanto para o PRR quanto para o PRRG, onde enumeram-se: cinco heurísticas construtivas, cinco heurísticas de busca local, três variantes da meta-heurística GRASP, sendo uma variante básica e duas com mecanismos de memória na fase de construção e um procedimento baseado na metaheurística ILS.

Como não é conhecida nenhuma biblioteca pública de instâncias para os problemas abordados nesta tese, experimentos computacionais empíricos foram realizados para cada uma das heurísticas propostas, considerando 144 instâncias geradas aleatoriamente, com o total de vértices dos grafos variando entre 15 e 1000. Esta biblioteca de instâncias agora encontra-se disponível no *site* http://labic.ic.uff.br/. Dois testes estatísticos (ANOVA e *t-Student*) foram utilizados sobre os resultados médios obtidos nos experimentos a fim de avaliar o desempenho das referidas heurísticas. A partir das análises realizadas, constatou-se que a abordagem utilizada pela heurística *ILS-PRR*, conseguiu atingir os melhores resultados, apesar de exigir um tempo computacional um pouco maior do que as variações da meta-heurística GRASP testadas. Também constatou-se que a qualidade das soluções iniciais construídas nas heurísticas baseadas nas variações do *GRASP Reativo* foi, na média, superior a qualidade das soluções iniciais construídas na heurística *ILS-PRR*. Desta forma, como a busca local utilizada em ambas abordagens era a mesma, conclui-se que as perturbações aplicadas nas iterações da heurística *ILS-PRR*, provavelmente contribuíram para escapar de ótimos locais de baixa qualidade.

Além das abordagens heurísticas, foram propostos neste trabalho dois novos conjuntos de regras de redução para os grafos associados ao PRR e ao PRRG. Experimentos realizados com estas regras considerando as 144 instâncias mencionadas anteriormente, mostraram que as reduções obtidas no total de vértices dos grafos foram bastante representativas, ficando em torno de 57% para os grafos associados ao PRR e 42% para os grafos associados ao PRRG. Vale ressaltar que o percentual de redução dos vértices do conjunto W aumentou significativamente com a proposta das novas regras ((R7) e (R9)), o que certamente reduziu a complexidade das buscas locais propostas. Além disto, as reduções também se mostraram muito significativas na obtenção de melhores limites duais quando comparados aos limites obtidos para as mesmas instâncias antes das reduções.

Ainda entre as contribuições relacionadas nesta tese, pode-se mencionar os resultados teóricos que relacionam os dois problemas estudados (Seção 4.4), além da prova matemática da corretude das novas regras de redução e das regras da literatura utilizadas nos dois conjuntos propostos neste trabalho (Seções 3.1.2 e 3.2.2).

Foi realizado um comparativo entre as três formulações existentes na literatura para o PRR ($\mathcal{F}1$, $\mathcal{F}2 \in \mathcal{F}3$), relacionando o número de restrições e o número de variáveis em função do tamanho do problema, onde constatou-se que a formulação $\mathcal{F}1$ produz um número de restrições consideravelmente superior às outras duas formulações ($O(2^n)$ restrições para $\mathcal{F}1 \in O(n^2)$ restrições para $\mathcal{F}2 \in \mathcal{F}3$). Um comparativo experimental entre as formulações $\mathcal{F}2 \in \mathcal{F}3$ usando o *software* CPLEX e um conjunto de 120 instâncias do PRR e do PRRG, mostrou que a formulação $\mathcal{F}2$ apresentou o melhor desempenho na obtenção dos limites duais para todas as instâncias consideradas no experimento. Utilizando estas mesmas instâncias, também foi realizada uma avaliação empírica da melhoria obtida nos valores dos limites duais quando as desigualdades válidas sugeridas em (MANIEZZO et al., 1999) são inseridas em $\mathcal{F}2$. Pelos resultados obtidos, verificou-se que, mesmo pequena, houve melhora na qualidade dos ótimos duais com a inserção das referidas desigualdades.

Como trabalhos futuros, sugere-se testar outras abordagens para a construção da solução inicial da heurística *ILS-PRR*, assim como estudar a contribuição de cada um dos componentes desta proposta na obtenção das soluções. Sugere-se também avaliar em detalhes o impacto da periodicidade da atualização das probabilidades do *GRASP*

Reativo_QA, assim como o benefício que pode ser agregado a cada uma das propostas apresentadas com a implementação de um conjunto elite de soluções. Uma outra oportunidade para a realização de pesquisas futuras diz respeito a utilização de diferentes estratégias para aplicação da reconexão por caminhos, já que neste trabalho apenas se considerou a reconexão sendo executada entre a solução incumbente retornada pela heurística e uma solução do conjunto elite selecionada aleatoriamente.

Referências

ARKIN, E. M.; HASSIN, R. Approximating algorithms for the geometric covering salesman problem. *Discrete Applied Mathematics*, v. 55, n. 3, p. 197–218, 1994. ISSN 0166-218X.

BALAS, E. The prize collecting traveling salesman problem. *Networks*, v. 19, p. 621–636, 1989.

BALAS, E.; HO, A. Set covering algorithms using cutting planes, heuristics, and subgradiente optimization: A computational study. *Math. Programming*, v. 12, p. 37–70, 1980.

BAZARAA, M. S.; JARVIS, J. J.; SHERALI, H. D. Linear programming and network flows (2nd ed.). New York, NY, USA: John Wiley & Sons, Inc., 1990. ISBN 0-471-63681-9.

BELLMORE, M.; NEMHAUSER, G. The traveling salesman problem: A survey. *Operations Research*, v. 16, n. 3, p. 538–558, 1968.

BRASILEIRO, L. A.; LACERDA, M. G. Análise do uso de sig no roteamento dos veículos de coleta de resíduos sólidos domiciliares. *Eng. Sanit. Ambient.*, v. 13, n. 4, 2008.

BRIMBERG, J.; HANSEN, P.; MLADENOVIĆ, N.; TAILLARD, E. D. Improvement and Comparison of Heuristics for Solving the Uncapacitated Multisource Weber Problem. *Operations Research*, v. 48, n. 3, p. 444–460, 2000. Disponível em: http://or.journal.informs.org/cgi/content/abstract/48/3/444>.

BRITO, L. R. Novas propostas para o Problema de Recobrimento de Rotas. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, COPPE, Rio de Janeiro, Brasil, 2005.

COLIN, E. C. Pesquisa Operacional: 170 aplicações em estratégia, finanças, logística, produção, marketing e vendas. Com enfoque em programação matemática. Rio de Janeiro, RJ, Brasil: LTC Editora, 2007.

CPLEX. *IBM ILOG CPLEX*. 2009. Acesso em: 05/abril/2010. Disponível em: http://www-2000.ibm.com/software/integration/optimization/cplex/>.

CURRENT, J. Multiobjective Design of Transportation Networks. Tese (Doutorado) — Department of Geography and Environmental Engineering, The Johns Hopkins University, 1981.

CURRENT, J.; HASAN, P.; ROLLAND, E. Efficient algorithms for solving the shortest covering path problem. *Transportation Science*, v. 28, p. 317–327, 1994.

CURRENT, J.; REVELLE, C.; COHON, J. The minimum-covering/shortest-path problem. *Decision Sciences*, v. 19, n. 3, p. 490–503, 1988.

CURRENT, J. R.; SCHILLING, D. A. The covering salesman problem. *Transportation Science*, v. 23, p. 208–213, 1989.

CURRENT, J. R.; SCHILLING, D. A. The median tour and maximal covering tour problems: Formulations and heuristics. *European Journal of Operational Research*, v. 73, n. 1, p. 114–126, February 1994. Disponível em: http://ideas.repec.org/a/eee/ejores/v73y1994i1p114-126.html>.

DETOFENO, T. C. Otimização de rotas de coleta de resíduos sólidos urbanos, utilizando técnicas de pequisa operacional. Dissertação (Mestrado) — Departamento de Matemática, Universidade Federal do Paraná, Curitiba, Paraná, 2009.

DORIGO, M. Optmization, Learning and Natural Algorithms. Tese (Doutorado) — Politecnico di Milano, 1992.

FINKE, G.; CLAUS, A.; GUNN, E. A two-commodity network flow approach to the traveling salesman problem. *Congress. Numerantium*, John Wiley & Sons, Chichester, v. 41, p. 167–178, 1984.

FISCHETTI, M.; SALAZAR, J.; TOTH, P. The symmetric generalized travelling salesman polytope. *Networks*, v. 26, n. 2, p. 113–123, 1995.

GAVISH, B.; GRAVES, S. C. The travelling salesman problem and related problems. In: *Working Paper OR-078.* Cambridge, MA, USA: Massachusetts Institute of Technology, Operations Research Center, 1978.

GENDREAU, M.; HERTZ, A.; LAPORTE, G. New insertion and postoptimization procedures for the traveling salesman problem. *Oper. Res.*, INFORMS, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, USA, v. 40, n. 6, p. 1086–1094, 1992. ISSN 0030-364X.

GENDREAU, M.; LAPORTE, G.; SEMET, F. The Covering Tour Problem. OPERATIONS RESEARCH, v. 45, n. 4, p. 568–576, 1997. Disponível em: <http://or.journal.informs.org/cgi/content/abstract/45/4/568>.

GEOFFRION, A. Lagrangean relaxation and its uses in integer programming. v. 2, p. 82–114, 1974.

GLOVER, F. Scatter search and star paths: Beyond the genetic metaphor. OR. Spektrum, v. 17, p. 125–137, 1995.

GLOVER, F. Tabu search and adaptive memory programing - advances, applications and challenges. In: BARR, R.; HELGASON, R.; KENNINGTON, J. (Ed.). Interfaces in Computer Science and Operations Research. Boston: KLUWER ACADEMIC PUBLISHERS, 1996. p. 1–75.

GLOVER, F. A template for scatter search and path relinking. In: *AE 97: Selected Papers from the Third European Conference on Artificial Evolution*. London, UK: Springer-Verlag, 1998. p. 3–54. ISBN 3-540-64169-6.

HACHICHA, M.; HODGSON, M. J.; LAPORTE, G. Heuristics for the multi-vehicle covering tour problem. *Computers and Operations Research*, v. 27, p. 29–42, 2000.

HANSEN, P.; MLADENOVIĆ, N.; PEREZ-BRITOS, D. Variable neighborhood decomposition search. *Journal of Heuristics*, Kluwer Academic Publishers, Hingham, MA, USA, v. 7, n. 4, p. 335–350, 2001. ISSN 1381-1231.

JOZEFOWIEZ, N.; SEMET, F.; TALBI, E. The bi-objective covering tour problem. *Comput. Operational Research*, Elsevier Science Ltd., Oxford, UK, UK, v. 34, n. 7, p. 1929–1942, 2007. ISSN 0305-0548.

KUBIK, P. Heuristic Solution Aproachs for the Covering Tour Problem. Tese (Doutorado) — University of Vienna, 2007.

LABBE, M.; LAPORTE, G. Maximizing user convenience and postal service efficiency in post box location. *Belgian J. Opns. Res. Statist. and Computer Sci.*, v. 26, p. 21–35, 1986.

LAGUNA, M.; MARTI, R. Grasp and path relinking for 2-layer straight line crossing minimization. *INFORMS J. on Computing*, INFORMS, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, USA, v. 11, n. 1, p. 44–52, 1999. ISSN 1526-5528.

LAPORTE, G.; MARTELLO, S. The selective traveling salesman problem. *Discrete* Applied Mathematics, v. 26, p. 93–207, 1990.

LESSING, L.; DUMITRESCU, I.; STÜTZLE, T. A comparison between aco algorithms for the set covering problem. In: *ANTS Workshop*. Berlin, Heidelberg: Springer-Verlag, 2004. (Lecture Notes in Computer Science, v. 3172), p. 1–12. ISBN 3-540-22672-9.

LIMA, R. S.; SILVA, A. N. R. Um parâmetro urbano global como referência para análises locais em modelos de locação-alocação. *Pesquisa Operacional*, v. 24, n. 3, 2004.

LINS, S.; KERNIGHAN, B. W. An effective heuristic algorithm for the traveling salesman problem. *Oper. Res.*, v. 21, p. 498–516, 1973.

LOURENÇO, H. R.; MARTIN, O.; STÜTZLE, T. A beginner's introduction to iterated local search. *Proceedings of the 4th Metaheuristics International Conference (MIC)*, v. 1, p. 1–6, 2001.

LOURENÇO, H. R.; MARTIN, O.; STÜTZLE, T. Iterated local search. In: GLOVER, F.; KOCHENBERGER, G. A. (Ed.). *Handbook of Metaheuristics*. Norwell, MA, USA: KLUWER ACADEMIC PUBLISHERS, 2003, (International Series in Operations Research & Management Science, v. 57). p. 321–353. ISBN 1402072635.

LYRA, A. R. de. O Problema de Recobrimento de Rotas com Coleta de Prêmios: Regras de Redução, Formulação Matemática e Heurísticas. Dissertação (Mestrado) — Instituto de Computação, Universidade Federal Fluminense, Niterói, Brasil, 2004.

MANIEZZO, V.; BALDACCI, R.; BOSCHETTI, M.; ZAMBONI, M. Scatter Search Methods for The Covering Tour Problem. 1999. Technical Report, Scienze dell'Informazione, University of Bologna, Italy, June 1999. MANIEZZO, V.; BALDACCI, R.; BOSCHETTI, M.; ZAMBONI, M. Scatter search methods for the covering tour problem. In: *Metaheuristic Optimization via Memory and Evolution*. New York, NY, USA: Springer-Verlag, 2005. cap. 3.

MARCHIORI, E.; STEENBEEK, A. An evolutionary algorithm for large scale set covering problems with application to airline crew scheduling. In: *Scheduling, in Real World Applications of Evolutionary Computing. Lecture Notes in Computer Science.* Berlin, Heidelberg: Springer-Verlag, 2000. p. 367–381.

MATSUMOTO, M.; NISHIMURA, T. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, ACM, New York, NY, USA, v. 8, n. 1, p. 3–30, 1998. ISSN 1049-3301.

MLADENOVIĆ, N. Variable neighborhood search - a new metaheuristic for combinatorial optimization. *Optimization Days*, Montreal, p. 112, 1995.

MLADENOVIĆ, N.; HANSEN, P. Variable neighborhood search. *Computers and Operations Research*, Elsevier Science Ltd., Oxford, UK, UK, v. 24, n. 11, p. 1097–1100, 1997. ISSN 0305-0548.

MOTTA, L. C. S. Novas abordagens para o Problema de Recobrimento de Rotas. Dissertação (Mestrado) — Instituto de Computação, Universidade Federal Fluminense, Niterói, Brasil, 2001.

OLIVEIRA, W. A. de. Construção de rotas para patrulhamento urbano preventivo. Dissertação (Mestrado) — Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas, Campinas, São Paulo, 2008.

OPPONG, J. R.; HODGSON, M. J. Spatial acessibility to health care facilities in suhum district, ghana. *Professional Geographer*, v. 46, p. 199–209, 1994.

ORMAN, A. J.; WILLIAMS, H. P. A Survey of Different Integer Programming Formulations of the Travelling Salesman Problem. London School of Economics and Political Science, London, 2004.

PRAIS, M.; RIBEIRO, C. Parameter variation in GRASP procedures. *Investigación Operativa*, v. 9, p. 1–20, 2000.

PRAIS, M.; RIBEIRO, C. C. Reactive grasp: An application to a matrix decomposition problem in tdma traffic assignment. *INFORMS J. on Computing*, INFORMS, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, USA, v. 12, n. 3, p. 164–176, 2000. ISSN 1526-5528.

RAMESH, T. Traveling purchaser problem. Oper. Res., v. 18, n. 2, p. 78-91, 1981.

REINELT, G. Tsplib - a traveling salesman problem library. ORSA - Journal on Computing, p. 376–384, 1991.

RESENDE, M.; RIBEIRO, C. Greedy randomized adaptive search procedures. In: GLOVER, F.; KOCHENBERGER, G. A. (Ed.). *Handbook of Metaheuristics*. Norwell, MA, USA: KLUWER ACADEMIC PUBLISHERS, 2003, (International Series in Operations Research & Management Science, v. 57). p. 219–249. ISBN 1402072635. RESENDE, M. G. C.; FEO, T. A. Greedy randomized adaptative search procedures. *Journal of Global Optimization*, p. 1–27, 1995.

RESENDE, M. G. C.; MARTÍ, R.; GALLEGO, M.; DUARTE, A. Grasp and path relinking for the max-min diversity problem. *Comput. Oper. Res.*, Elsevier Science Ltd., Oxford, UK, UK, v. 37, n. 3, p. 498–508, 2010. ISSN 0305-0548.

RIBEIRO, C. C.; SOUZA, M. C. Variable neighborhood search for the degree-constrained minimum spanning tree problem. *DAM*, v. 118, n. 1-2, p. 43–54, 2002.

RIBEIRO, C. C.; UCHOA, E.; WERNECK, R. F. A hybrid grasp with perturbations for the steiner problem in graphs. *INFORMS J. on Computing*, INFORMS, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, USA, v. 14, n. 3, p. 228–246, 2002. ISSN 1526-5528.

ROSENKRANTZ, D. J.; STEARNS, R. E.; LEWIS, P. M. An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, v. 6, n. 3, p. 563–581, 1977.

SILVA, M. de S. A. Problema de Recobrimento de Rotas com Coleta de Prêmios. Dissertação (Mestrado) — Instituto de Computação, Universidade Federal Fluminense, Niterói, Brasil, 2009.

TOP500. The 34th TOP500 List. 2009. Acesso em: 10/março/2010. Disponível em: http://www.top500.org/lists/2009/11>.

APÊNDICE A - Instâncias para o PRR e para o PRRG

Este apêndice apresenta a descrição das instâncias para o PRR e para o PRRG propostas neste trabalho. A geração destas instâncias fez-se necessário uma vez que é desconhecida a existência de qualquer biblioteca pública para ambos problemas. Portanto, a Seção A.1 detalha o processo da geração das instâncias e a Seção A.2 apresenta o formato do arquivo de dados, a regra de formação dos nomes e a representação das estruturas de uma instância para o PRR/PRRG.

A.1 Grupos de instâncias gerados

Com o objetivo de apresentar uma significativa diversidade de instâncias, tanto para o PRR quanto para o PRRG, foram geradas aleatoriamente 144 instâncias, variando-se a cardinalidade de cada um dos conjuntos que compõem os vértices do grafo.

As coordenadas dos vértices foram obtidas utilizando o algoritmo Mersenne Twister, proposto por Matsumoto e Nishimura (MATSUMOTO; NISHIMURA, 1998). Este algoritmo é um gerador de números pseudo-aleatórios (randômicos) que tem licença gratuita para todo o tipo de uso e possui um período de $2^{19937-1}$. Nele os dados estão equidistribuídos em 623 dimensões para uma precisão de 32 *bits* (gera 1 bilhão de números em ponto flutuante de 64 *bits*). Cada coordenada foi gerada aleatoriamente em um retângulo de dimensão $[0, 640] \times [0, 480]$, sendo a data/hora do sistema operacional considerada como semente para o algoritmo.

Os conjuntos T, $W \in V \setminus T$ foram definidos considerando os |T| primeiros pares de coordenadas (x, y) gerados para compor os vértices do conjunto T, os seguintes |W| pares para compor os vértices do conjunto W e os pares remanescentes para compor os vértices do conjunto $V \setminus T$. As 144 instâncias foram divididas em 4 grupos:

- *Grupo I:* Formado por 48 instâncias para o PRRG, com $|V \cup W|$ variando de 15 a 60 vértices;
- •Grupo II: Formado por 24 instâncias para o PRRG, com |V ∪ W| variando de 100 a 1.000 vértices;
- •Grupo III: Formado por 48 instâncias para o PRR, com |V ∪ W| variando de 15 a 60 vértices;
- •Grupo IV: Formado por 24 instâncias para o PRR, com |V ∪ W| variando de 100 a 1.000 vértices.

Para os grupos I e II, a distância de cobertura associada a cada uma das instâncias foi definida pela seguinte equação:

$$d = max\{min\{c_{ij} \mid i \in W \ e \ j \in V \cup W, \ \forall i \neq j\}\}$$
(A.1)

Da forma como a equação A.1 foi definida, assegura-se que todo vértice $i \in W$ estará coberto por pelo menos um vértice $j \in V \cup W$, já que uma solução viável do PRRG pode conter vértices de W.

Para as instâncias que compõem os grupos III e IV, a distância de cobertura foi determinada pela seguinte equação:

$$d = \max\{\min\{c_{ij} \mid i \in W e \ j \in V, \ \forall i \neq j\}\}$$
(A.2)

A equação A.2 garante que todo vértice $i \in W$ será coberto por pelo menos um vértice $j \in V$, uma vez que uma solução viável associada ao PRR é formada somente por vértices de V.

Vale ressaltar que, assim como mostrado na Seção 4.4, toda solução viável para o PRR também é viável para o PRRG. Por conseguinte, todas as instâncias que compõem os grupos III e IV admitem solução para ambos os problemas. Entretanto, a correspondência equivalente não pode ser feita para as instâncias dos grupos I e II, que só têm a viabilidade assegurada para o PRRG.

As 144 instâncias foram geradas através de um programa desenvolvido na linguagem de programação C++, com o compilador g + + (versão 4.2.4). O ambiente de teste foi composto por um computador com processador Intel T2250 Dual Core de 1,73GHz, 2GB de memória RAM DDR2 de 533MHz, tendo como sistema operacional a distribuição

GNU/Linux Ubuntu, com o kernel na versão 2.6.24-19-generic#1-smp.

A Tabela A.1 apresenta a distribuição percentual do total dos vértices do grafo nos conjuntos $T, W \in V \setminus T$ adotada na geração das 144 instâncias.

Tabela A.1: Cardinalidade de cada conjunto que compõe as instâncias geradas

	Distribuição percentual dos vértices das instâncias								
	Grupos	I e III		Grupos I	I e IV				
$n = \{1,, n\}$	5, 20, 25, 30	$0, 35, 40, 50, 60\}$	$m = \{1$	00, 200, 300	$,500,700,1000\}$				
T	$ T \qquad W \qquad V \setminus T $			W	$ V \setminus T $				
$\lfloor 20\% n \rfloor$	$\lfloor 20\% n \rfloor$	n - (T + W)							
$\lfloor 20\% n \rfloor$	$\lfloor 60\% n \rfloor$	n - (T + W)	$\lfloor 20\% m \rfloor$	$\lfloor 60\% m \rfloor$	m - (T + W)				
$\lfloor 30\% n \rfloor$	$\lfloor 40\% n \rfloor$	n - (T + W)		—					
$\lfloor 33\%n \rfloor$	$\lfloor 33\%n \rfloor$	n - (T + W)	$\lfloor 33\% m \rfloor$	$\lfloor 33\% m \rfloor$	m - (T + W)				
$\lfloor 50\% n \rfloor$	$\lfloor 50\% n \rfloor$	n - (T + W)	$\lfloor 50\% m \rfloor$	$\lfloor 50\% m \rfloor$	m - (T + W)				
$\lfloor 60\% n \rfloor$	$\lfloor 20\% n \rfloor$	n - (T + W)	$\lfloor 60\% m \rfloor$	$\lfloor 20\% m \rfloor$	m - (T + W)				

Legenda:

— Nenhuma instância foi gerada com a distribuição percentual relacionada.

n = |V| + |W|, ou seja, n é o total de vértices para as instâncias dos Grupos I e III.

m=|V|+|W|,ou seja, mé o total de vértices para as instâncias dos Grupos II e IV.

Os valores percentuais apresentados na Tabela A.1 referem-se a quantidade percentual dos vértices de cada conjunto em relação ao total de vértices do grafo. O objetivo da distribuição proposta é obter instâncias com características distintas através da variação da cardinalidade dos conjuntos de vértices que as compõem.

Para exemplificar a distribuição proposta na Tabela A.1, considere um grafo com um total de 35 vértices (grupo I ou III) e uma distribuição dada por $|T| = \lfloor 33\% n \rfloor$, $|W| = \lfloor 33\% n \rfloor$ e $|V \setminus T| = n - (|T| + |W|)$. Portanto, $|T| = \lfloor 33\% \times 35 \rfloor = 11$, $|W| = \lfloor 33\% \times 35 \rfloor = 11$ e $|V \setminus T| = 35 - 22 = 13$.

Todas as 144 instâncias propostas neste trabalho, como também o programa que as gerou, foram disponibilizados para acesso público no seguinte endereço eletrônico:

http://labic.ic.uff.br/AutoIndex/index.php?dir=CTP%26GCTP/

Espera-se que, com a disponibilização pública desta biblioteca de instâncias, possam surgir outras pesquisas que não só melhorem os resultados aproximados obtidos neste trabalho, como também consiga-se provar a otimalidade da solução de mais instâncias.

A.2 Nomenclatura das instâncias e representação das estruturas de dados

Cada instância gerada como descrito na Seção A.1, tanto para o PRR quanto para o PRRG, foi gravada em um arquivo de dados que contém as informações sobre total de vértices do grafo, o total de vértices obrigatórios (T), o total de vértices a serem cobertos (W), o total de vértices opcionais $(V \setminus T)$, a distância de cobertura, os índices de cada vértice e suas coordenadas $x \in y$. O formato deste arquivo é mostrado no quadro a seguir:

23			→ Total de vértices do grafo
6			$\neg \neg \rightarrow$ Total de vértices de obrigatórios (T)
7			$\neg \neg \rightarrow$ Total de vértices de a serem cobertos (W)
10			+ Total de vértices de opcionais $(V \setminus T)$
0	425	60	$- \rightarrow$ Índice do vértice - Coordenada x - Coordenada y
1	469	65	
2	310	415	
3	275	320	
4	139	300	
5	300	15	
6	139	140	
7	315	103	
8	304	166	
9	566	215	
10	486	35	
11	432	208	
12	477	361	
13	231	310	
14	374	371	
15	486	405	
16	11	271	
17	464	209	
18	231	474	
19	210	248	
20	321	17	
21	468	130	
22	190	120	
140000			→ Distância de cobertura

A nomenclatura das instâncias geradas obedeceu a seguinte regra de formação:

$Problema A_B_C_D$

onde:

Problema refere-se a sigla do problema associado à instância em questão: *ctp* ou *gctp*; **A** refere-se à cardinalidade do grafo;

B refere-se à cardinalidade do conjunto T;

 \mathbf{C} refere-se à cardinalidade do conjunto W;

D refere-se à cardinalidade do conjunto $V \setminus T$.

Considerando a instância representada no arquivo de dados ilustrado anteriormente e a regra de formação apresentada, o nome da referida instância seria $ctp23_6-7-10$ se esta fosse associada ao PRR ou $gctp23_6-7-10$ se esta fosse associada ao PRRG. As siglas dos problemas utilizadas na citada regra de formação correspondem aos nomes originalmente referenciados na literatura, ou seja, ctp para o Covering Tour Problem e gctp para o Generalized Covering Tour Problem.

Para representar os dados de uma instância nas implementações realizadas neste trabalho, utilizou-se um vetor para armazenar as coordenadas dos vértices, 5 variáveis inteiras para armazenar o total de vértices do grafo, o total de vértices de obrigatórios (T), o total de vértices a serem cobertos (W), o total de vértices opcionais $(V \setminus T)$ e a distância de cobertura (a distância de cobertura foi armazenada como um número inteiro, considerando três casas decimais de precisão, para evitar problemas com os arredondamentos automáticos).

As distâncias entre os pares de vértices do grafo foram armazenadas em uma matriz triangular inferior, já que $c_{ij} = c_{ji}, \forall i \neq j$.

Visando reduzir a complexidade dos algoritmos propostos, foram utilizados duas estruturas para mapear a cobertura dos vértices de W, já que estas informações são frequentemente verificadas por todos os algoritmos:

Mapeamento 1: Relaciona os vértices de *W* com os vértices da sua região de cobertura;

Mapeamento 2: Relaciona os vértices do grafo com os vértices de W que estes cobrem.

Para facilitar o entendimento destas duas estruturas, considere a instância representada no arquivo de dados ilustrado no início desta seção. Se esta for uma instância do PRR $(ctp23_6-7-10)$, as duas estruturas para mapear a cobertura dos vértices de W seriam criadas como mostra a Figura A.1.

Mapeamento 1 - PRR					
$W_i \in W$	r .	$v_{j}\!\in\!V$			
6	\rightarrow	19	22		
7	→	20	22		
8	→	19	22		
9	→	17	21		
10	→	21			
11	→	17	21		
12	\rightarrow	14	15		

Mapeamento 2 - PRR							
$v_j \in V$	7	$W_i\!\!\in\!$	W				
0	$0 \rightarrow$		10				
1	\rightarrow	10					
2	\rightarrow	null	,				
3	\rightarrow	null					
4	\rightarrow	null					
5	\rightarrow	7					
13	\rightarrow	null					
14	\rightarrow	12					
15	\rightarrow	12					
16	\rightarrow	null					
17	\rightarrow	9	11				
18	\rightarrow	null					
19	\rightarrow	6	8				
20	\rightarrow	7					
21	\rightarrow	9	10				
22	\rightarrow	6	7				

11

8

Figura A.1: Estruturas que representam a cobertura dos vértices de W da instância $ctp23_6-7-10$ do PRR

Observe na Figura A.1 que a estrutura chamada de *Mapeamento 1* é a responsável por associar cada vértice $w_i \in W$ aos vértices $v_j \in V$ tal que $c_{ij} \leq d$. A estrutura identificada como *Mapeamento 2* associa cada vértice $v_j \in V$ aos vértices $w_i \in W$ que ele cobre.

A Figura A.2 ilustra como seriam criadas as estruturas que mapeiam a cobertura dos vértices de W se o mesmo arquivo de dados fosse associado ao PRRG. Note que, a estrutura chamada *Mapeamento 1* considera na cobertura de cada $w_i \in W$, os vértices de V e os vértices de W, incluindo o próprio w_i . Analogamente, a estrutura representada pelo *Mapeamento 2*, associa cada vértice $v_j \in V \cup W$ aos vértices $w_i \in W$ que são cobertos por v_j .

Mapeamento 1 - PRRG							
w _i ∈ W	T	\mathbf{v}_{j}	$v_j \in V \cup W$				
6	→	6	19	22			
7	\rightarrow	0	5	7	8	20	22
8	→	7	8	11	19	22	
9	→	9	11	17	21		
10	\rightarrow	0	1	10	21		
11	→	8	9	11	17	21	
12	→	12	14	15			•

. .

мар	eame	nto 2 -	PRRG	
∈V∪	W	Wi∈	W	
0	\rightarrow	7	10	
1	\rightarrow	10		
2	\rightarrow	null		
3	\rightarrow	null		
4	\rightarrow	null		
5	\rightarrow	7		
6	\rightarrow	6		
7	\rightarrow	7	8	
8	\rightarrow	7	8	11
9	\rightarrow	9	11	
10	\rightarrow	10		
11	\rightarrow	8	9	11
12	\rightarrow	12		
13	\rightarrow	null		
14	\rightarrow	12		
15	\rightarrow	12		
16	\rightarrow	null		
17	\rightarrow	9	11	
18	\rightarrow	null		
19	\rightarrow	6	8	
20	\rightarrow	7		
21	\rightarrow	9	10	11
22	\rightarrow	6	7	8
		$ \begin{array}{c} Mapeame \\ Mapeame \\ \in V \cup W \\ \hline 0 \\ \hline 0 \\ \hline 1 \\ \hline 1 \\ \hline 2 \\ \hline 1 \\ \hline 2 \\ \hline 1 \\ \hline 2 \\ \hline 1 \\ \hline 3 \\ \hline - 1 \\ \hline 7 \\ \hline - 1 \\ \hline - 1 \\ \hline 7 \\ \hline - 1 $	Mapeamento 2 - \bigcirc V \bigcirc W $\heartsuit_i \in$ \bigcirc N \neg \neg 1 \rightarrow 10 2 \rightarrow $null$ 3 \rightarrow $null$ 3 \rightarrow $null$ 4 \rightarrow $null$ 5 \rightarrow 7 6 \rightarrow 6 7 \rightarrow 7 8 \rightarrow 7 9 \rightarrow 9 10 \rightarrow 10 11 \rightarrow 8 12 \rightarrow 12 13 \rightarrow $null$ 14 \rightarrow 12 15 \rightarrow 12 16 \rightarrow $null$ 17 \rightarrow 9 18 \rightarrow $null$ 19 \rightarrow 6 20 \rightarrow 7 21 \rightarrow 9	Mapeamento 2 - PRRG $\in V \cup W$ $w_i \in W$ $0 \rightarrow 7$ 10 $1 \rightarrow 10$ 2 $2 \rightarrow null$ 3 $3 \rightarrow null$ 4 $4 \rightarrow null$ $5 \rightarrow 7$ $6 \rightarrow 6$ $7 \rightarrow 7$ $6 \rightarrow 6$ $7 \rightarrow 7$ 8 $9 \rightarrow 9$ $10 \rightarrow 10$ $11 \rightarrow 8$ $10 \rightarrow 10$ $11 \rightarrow 8$ $12 \rightarrow 12$ $13 \rightarrow null$ $14 \rightarrow 12$ $15 \rightarrow 12$ $16 \rightarrow null$ $17 \rightarrow 9$ $16 \rightarrow null$ $17 \rightarrow 9$ $18 \rightarrow null$ $19 \rightarrow 6$ $20 \rightarrow 7$ $21 \rightarrow 9$ 10 $22 \rightarrow 6$

Figura A.2: Estruturas que representam a cobertura dos vértices de W da instância gctp23_6-7-10 do PRRG

É importante ressaltar que, tanto para o PRR quanto para o PRRG, podem existir vértices em V que não cobrem nenhum vértice em W, o que justifica o termo null que é encontrado no Mapeamento 2 das Figuras A.1 e A.2.

Como descrito no Capítulo 1, uma solução viável do PRR é representada por um ciclo sobre subconjunto de V (para o PRRG, considere $V \cup W$), que contém todos os vértices do subconjunto $T \subseteq V$ e cobre cada vértice do conjunto W. A Figura A.3 ilustra uma solução viável para a instância *ctp23_6-7-10*.



Figura A.3: Representação gráfica de uma rota viável para a instância ctp23_6-7-10

Uma rota ou solução do PRR/PRRG foi representada neste trabalho por um vetor de pares ordenados (v_i, v_j) , onde v_i representa o vértice que antecede o vértice v_k na rota e v_j o vértice que o precede. O vértice v_k é representado pelo índice k do vetor. A solução ilustrada na Figura A.3 seria representada neste vetor de pares ordenados da seguinte forma:

5	1	0	21	14	3	2	4	3	22	22	0	-1	-1	21	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	14	4	5
	0		1	2	2		3	4	4	5		•	••	14	4	1	5	1	6	1	7	1	8	1	9	2	0	2	21	2	2

Os vetores que representam as soluções do PRR/PRRG têm tamanho fixo ($|V \cup W|$ compartimentos), sendo o compartimento dos vértices que estão fora da rota inicializados com -1. Portanto, quando uma solução estiver associada ao PRR, os compartimentos referentes aos vértices de W serão sempre inicializados com -1, pois não fazem parte de nenhuma solução viável deste problema.

APÊNDICE B – Experimentos computacionais com as regras de redução propostas

Este apêndice apresenta o resultado completo dos experimentos computacionais realizados para avaliar o impacto da redução dos grafos associados ao PRR e ao PRRG, quando são utilizados os conjuntos de regras de redução propostos neste trabalho.

Nos experimentos envolvendo as 72 instâncias do PRR, foram consideradas seis sequências distintas para a aplicação das quatro regras de redução que compõem o conjunto proposto na Seção 3.1.2:

Sequência 1: $R7 \rightarrow R6 \rightarrow R5 \rightarrow R4$ Sequência 2: $R6 \rightarrow R7 \rightarrow R5 \rightarrow R4$ Sequência 3: $R6 \rightarrow R7 \rightarrow R5 \rightarrow R4$ Sequência 4: $R5 \rightarrow R7 \rightarrow R6 \rightarrow R4$ Sequência 5: $R5 \rightarrow R6 \rightarrow R7 \rightarrow R4$ Sequência 6: $R7 \rightarrow R5 \rightarrow R6 \rightarrow R4$

Os experimentos envolvendo as 72 instâncias do PRRG foram realizados considerando as seguintes sequências para a aplicação das três regras de redução que compõem o conjunto apresentado na Seção 3.2.2:

Sequência 1: $R9 \rightarrow R8 \rightarrow R4$ Sequência 2: $R8 \rightarrow R9 \rightarrow R4$ Sequência 3: $R4 \rightarrow R8 \rightarrow R9$ Sequência 4: $R8 \rightarrow R4 \rightarrow R9$

As tabelas a seguir foram organizadas pelo problema analisado (PRR ou PRRG), pelos grupos de instâncias (Grupo I, II, III e IV) e pelas sequências de aplicação das regras que compõem cada conjunto testado. Vale ressaltar que, para facilitar a comparação dos resultados, todas as tabelas apresentam a cardinalidade dos conjuntos de vértices dos grafos, antes e depois das reduções.

Instâncias do	G	rafo O	riginal		[] JP7	7 B6	R5 R4	1	[] [Pf	B7	R5 R4	1
Crupo III			W	V\T		$\frac{1}{ \mathbf{T} }$	100, 104			$ \mathbf{r} $	100, 104	
Grupo III								V \ 1				
ctp15_3-3-9	15	3	3	9	4	4	0	0	4	4	0	0
ctp15_3-9-3	15	3	9	3 -	(4	1	2	(4	1	2
ctp15_4-4-7	15	4	4	7	6	6	0	0	6	6	0	0
ctp15_4-6-5	15	4	6	5	7	7	0	0	7	7	0	0
ctp15_7-7-1	15	7	7	1	7	7	0	0	7	7	0	0
ctp15_9-3-3	15	9	3	3	10	10	0	0	10	10	0	0
ctp20_10-10-0	20	10	10	0	10	10	0	0	10	10	0	0
ctp20_12-4-4	20	12	4	4	12	12	0	0	12	12	0	0
ctp20_4-12-4	20	4	12	4	5	5	0	0	5	5	0	0
ctp20_4-4-12	20	4	4	12	7	7	0	0	7	7	0	0
$ctp20_{-6-6-8}$	20	6	6	8	7	7	0	0	7	7	0	0
ctp20_6-8-6	20	6	8	6	7	7	0	0	7	7	0	0
$ctp25_12-12-1$	25	12	12	1	12	12	0	0	12	12	0	0
ctp25_15-5-5	25	15	5	5	15	15	0	0	15	15	0	0
$ctp25_{-}5-15-5$	25	5	15	5	6	6	0	0	6	6	0	0
ctp25_5-5-15	25	5	5	15	10	7	1	2	10	7	1	2
$ctp25_{-}7_{-}10_{-}8$	25	7	10	8	8	8	0	0	8	8	0	0
ctp25_8-8-9	25	8	8	9	9	9	0	0	9	9	0	0
$ctp30_{15-15-0}$	30	15	15	0	15	15	0	0	15	15	0	0
$ctp30_18-6-6$	30	18	6	6	19	19	0	0	19	19	0	0
$ctp30_{-}6_{-}18_{-}6$	30	6	18	6	8	8	0	0	8	8	0	0
ctp30 = 6-6-18	30	6	6	18	7	7	0	0	7	7	0	0
$ctp30_912-9$	30	9	12	9	10	10	0	0	10	10	0	0
$ctp30_9-9-12$	30	9	9	12	10	10	0	0	10	10	0	0
ctp35_10-14-11	35	10	14	11	14	14	0	0	14	14	0	0
$ctp35_11-11-13$	35	11	11	13	15	15	0	0	15	15	0	0
$\operatorname{ctp35}_{-17-17-1}$	35	17	17	1	17	17	0	0	17	17	0	0
${ m ct}{ m p35}_{-}{ m 21}_{-}{ m 7}_{-}{ m 7}$	35	21	7	7	24	24	0	0	24	24	0	0
${ m ct}{ m p35}_{-}$ 7-21-7	35	7	21	7	10	10	0	0	10	10	0	0
${ m ct}{ m p35}_{-}$ 7-7-21	35	7	7	21	14	11	1	2	14	11	1	2
ctp40_12-16-12	40	12	16	12	16	13	1	2	16	13	1	2
$ctp40_13-13-14$	40	13	13	14	20	17	1	2	20	17	1	2
$ctp40_20-20-0$	40	20	20	0	20	20	0	0	20	20	0	0
$ctp40_24-8-8$	40	24	8	8	25	25	0	0	25	25	0	0
ct p 40 - 8 - 24 - 8	40	8	24	8	9	9	0	0	9	9	0	0
ct p 40 - 8 - 8 - 24	40	8	8	24	16	13	1	2	16	13	1	2
$ctp50_{-}10-10-30$	50	10	10	30	17	13	1	3	17	13	1	3
$ctp50_{-}10-30-10$	50	10	30	10	10	10	0	0	10	10	0	0
$ctp50_{-}15_{-}20_{-}15_{-}$	50	15	20	15	17	17	0	0	17	17	0	0
$ctp50_{-}16_{-}16_{-}18$	50	16	16	18	18	18	0	0	18	18	0	0
ctp50 = 25 - 25 - 0	50	25	25	0	25	25	0	0	25	25	0	0
$ctp50_{-}30_{-}10_{-}10$	50	30	10	10	30	30	0	0	30	30	0	0
$ctp60_{-}12-12-36$	60	12	12	36	22	16	1	5	22	16	1	5
$ctp60_12-36-12$	60	12	36	12	12	12	0	0	12	12	0	0
$ctp60_18-24-18$	60	18	24	18	19	19	0	0	19	19	0	0
ctp60_19-19-22	60	19	19	22	20	20	0	0	20	20	0	0
$ctp60_{-}30_{-}30_{-}0$	60	30	30	0	30	30	0	0	30	30	0	0
$ctp60_{-}36-12-12$	60	36	12	12	36	36	0	0	36	36	0	0

Tabela B.1: Redução das instâncias do PRR (Grupo III) - Sequências 1 e 2 das regras de redução

Instâncias do	G	rafo O	riginal		{₿€	B5	R7 R4	1	{B!	B7	R6 R4	1
Grupo III			W	V\T		$ \mathbf{T} $	W		$ V \perp W $	$ \mathbf{T} $	W	
atplf 2.2.0		9	2			4				4	1	
$ctp_{15} = 5 - 5 - 9$	15	3 9	3 0	9	0	4	1	บ ว	0	4	1	0
ctp15 = 5 - 5 - 5	15	- 3	3	7	0	-4 6	2	2 0	8	4 6	2	2 0
ctp15 = 4-4-7	15	4	4	5	0		2	0	10		2	0
$ctp15_4-0-3$	15	4	7	1	10	7	0	0	10	7	0	0
$ctp15_{-7-1}$	15	0	2	2	11	10	1	0	11	10	1	0
ctp13 = 9-3-3	20	10	10	0	10	10	1	0	10	10	1	0
$ctp_{20} = 10 - 10 - 0$	20	10	10	4	10	10	0	0	10	10	0	0
$ctp_{20} = 12 - 4 - 4$	20	12	4	4	12	12	1	0	12	12	1	0
$ctp_{20} = 4 - 12 - 4$	20	4	12	4	10	5	1	0	10	5	1 2	0
$ctp_{20} = 4 - 4 - 12$	20	4	4	12	10	7	- J - 1	0	10	7	- J - 1	0
ctp20=0-0-8	20	6	0	0 6	0	7	1	0	8	7	1	0
$ctp_{20} = 0-0-0$	20	10	19	1	10	10	1	0	10	10	1	0
$ctp_{25} = 12 - 12 - 1$	20	12	12	I E	12	12	0	0	12	12	0	0
$ctp_{25} = 15 - 5 - 5$	20	10	15		15	10	1	0	15	10 6	1	0
$ctp_{25} = 5 - 15 - 5$	20	5	10	15	19	0	1	0	19	0	1	0
ctp25 = 5 - 5 - 15	20	5	10	15	12	1		2	12	1	- 3 - 1	2
ctp2b = 7 - 10 - 8	25	(10	8	9	8	1	0	9	8	1	0
ctp25 = 8-8-9	25	8	8	9	10	9	1	0	10	9	1	0
$ctp_{30} = 15 - 15 - 0$	30	10	15	0 C	15	10	1	0	15	10	1	0
ctp30 = 18 - 6 - 6	30	18	0	0	20	19	1	0	20	19	1	0
ctp30_6-18-6	30	0	18	0	10	8	2	0	10	8	2	0
ctp30_6-6-18	30	6	0	18	8	10	1	0	8	10	1	0
ctp30_9-12-9	30	9	12	9	11	10	1	0	11	10	1	0
ctp30_9-9-12	30	9	9	12	11	10	1	0	11	10	1	0
ctp35_10-14-11	35	10	14	11	18	14	4	0	18	14	4	0
ctp35_11-11-13	35	11	11	13	19	15	4	0	19	15	4	0
ctp35_17-17-1	35	17	17	1	17	17	0	0	17	17	0	0
ctp35_21-7-7	35	21	1	7	27	24	3	0	27	24	3	0
ctp35_7-21-7	35	(21	(13	10	3	0	13	10	3	0
$ctp_{35} - (-21)$	35	10	10	21	18	11	5	2	18		5	2
ctp40_12-16-12	40	12	10	12	17	13	2	2	17	13	2	2
ctp40_13-13-14	40	13	13	14	24	17	5	2	24	17	5	2
$ctp40_20-20-0$	40	20	20	0	20	20	0	0	20	20	0	0
$ctp40_24-8-8$	40	24	8	8	26	25	1	0	26	25	1	0
ctp40_8-24-8	40	8	24	8	10	9		0	10	9		0
ctp40_8-8-24	40	8	8	24	21	13	6	2	21	13	6	2
ctp50_10-10-30	50	10	10	30	20	13	4	3	20	13	4	3
ctp50_10-30-10	50	10	30	10	10	10	0	0	10	10	0	0
ctp50_15-20-15	50	15	20	15	19	17	2	0	19	17	2	0
ctp50_16-16-18	50	16	16	18	20	18	2	0	20	18	2	0
ctp50_25-25-0	50	25	25	0	25	25	0	0	25	25	0	0
ctp50_30-10-10	50	30	10	10	30	30	0	0	30	30	0	0
ctp60_12-12-36	60	12	12	36	26	16	5	5	26	16	5	5
ctp60_12-36-12	60	12	36	12	12	12	0	0	12	12	0	0
ctp60_18-24-18	60	18	24	18	20	19	1	0	20	19	1	0
ctp60_19-19-22	60	19	19	22	21	20	1	0	21	20	1	0
ctp60_30-30-0	60	30	30	0	30	30	0	0	30	30	0	0
ctp60_36-12-12	60	36	12	12	36	36	0	0	36	36	0	0

Tabela B.2: Redução das instâncias do PRR (Grupo III) - Sequências 3 e 4 das regras de redução

Instâncias do	G	rafo O	riginal		∫ B⊧	E B 6	R7 R4	ι	∫B2	7 B 5	R6 R4	ι
Crupo III			W		$ \mathbf{V} + \mathbf{W} $	$ \mathbf{T} $				$\frac{100}{100}$	100, 104	
Grupo III												
ctp15_3-3-9	15	3	3	9	5	4	1	0	4	4	0	0
ctp15_3-9-3	15	3	9	3	8	4	2	2	(4	1	2
ctp15_4-4-7	15	4	4	7	8	6	2	0	6	6	0	0
ctp15_4-6-5	15	4	6	5	10	7	3	0	7	7	0	0
ctp15_7-7-1	15	7	7	1	7	7	0	0	7	7	0	0
ctp15_9-3-3	15	9	3	3	11	10	1	0	10	10	0	0
ctp20_10-10-0	20	10	10	0	10	10	0	0	10	10	0	0
ctp20_12-4-4	20	12	4	4	12	12	0	0	12	12	0	0
$ctp20_4-12-4$	20	4	12	4	6	5	1	0	5	5	0	0
ctp20_4-4-12	20	4	4	12	10	7	3	0	7	7	0	0
$ctp20_{-}6-6-8$	20	6	6	8	8	7	1	0	7	7	0	0
$ctp20_{-6-8-6}$	20	6	8	6	8	7	1	0	7	7	0	0
$ctp25_12-12-12$	25	12	12	1	12	12	0	0	12	12	0	0
$ctp25_15-5-5$	25	15	5	5	15	15	0	0	15	15	0	0
${ m ct}{ m p25}_{-}5$ -15-5	25	5	15	5	7	6	1	0	6	6	0	0
$ctp25_{-}5-5-15$	25	5	5	15	12	7	3	2	10	7	1	2
ctp25 = 7 - 10 - 8	25	7	10	8	9	8	1	0	8	8	0	0
$ctp25_{-}8-8-9$	25	8	8	9	10	9	1	0	9	9	0	0
$ctp30_{-}15_{-}15_{-}0$	30	15	15	0	15	15	0	0	15	15	0	0
$ctp30_18-6-6$	30	18	6	6	20	19	1	0	19	19	0	0
$ctp30_{-}6-18-6$	30	6	18	6	10	8	2	0	8	8	0	0
ct p 30 = 6 - 6 - 18	30	6	6	18	8	7	1	0	7	7	0	0
ct p 30 - 9 - 12 - 9	30	9	12	9	11	10	1	0	10	10	0	0
$ct p 30_{-}9 - 9 - 12$	30	9	9	12	11	10	1	0	10	10	0	0
ctp35_10-14-11	35	10	14	11	18	14	4	0	14	14	0	0
$ctp35_11-11-13$	35	11	11	13	19	15	4	0	15	15	0	0
ctp35 = 17 - 17 - 1	35	17	17	1	17	17	0	0	17	17	0	0
$ctp35_21-7-7$	35	21	7	7	27	24	3	0	24	24	0	0
${ m ct}{ m p35}_{-}$ 7-21-7	35	7	21	7	13	10	3	0	10	10	0	0
${ m ct}{ m p35}_{-}$ 7-7-21	35	7	7	21	18	11	5	2	14	11	1	2
$\operatorname{ctp40}_{-12}$ -16-12	40	12	16	12	17	13	2	2	16	13	1	2
ctp40_13-13-14	40	13	13	14	24	17	5	2	20	17	1	2
$\operatorname{ctp40}_{-20-20-0}$	40	20	20	0	20	20	0	0	20	20	0	0
$ctp40_24-8-8$	40	24	8	8	26	25	1	0	25	25	0	0
$ctp40_{-8-24-8}$	40	8	24	8	10	9	1	0	9	9	0	0
ctp40_8-8-24	40	8	8	24	21	13	6	2	16	13	1	2
${ m ctp50}_{-10-10-30}$	50	10	10	30	20	13	4	3	17	13	1	3
$ctp50_{-}10-30-10$	50	10	30	10	10	10	0	0	10	10	0	0
$ctp50_{-}15_{-}20_{-}15$	50	15	20	15	19	17	2	0	17	17	0	0
$ctp50_{-}16-16-18$	50	16	16	18	20	18	2	0	18	18	0	0
ctp50 = 25 - 25 - 0	50	25	25	0	25	25	0	0	25	25	0	0
$ctp50_{-}30_{-}10_{-}10$	50	30	10	10	30	30	0	0	30	30	0	0
$ctp60_{-}12-12-36$	60	12	12	36	26	16	5	5	22	16	1	5
$ctp60_{-}12-36-12$	60	12	36	12	12	12	0	0	12	12	0	0
$ctp60_18-24-18$	60	18	24	18	20	19	1	0	19	19	0	0
ctp60 = 19-19-22	60	19	19	22	21	20	1	0	20	20	0	0
$ctp60_{-}30_{-}30_{-}0$	60	30	30	0	30	30	0	0	30	30	0	0
$ctp60_36-12-12$	60	36	12	12	36	36	0	0	36	36	0	0

Tabela B.3: Redução das instâncias do PRR (Grupo III) - Sequências 5 e 6 das regras de redução

	T/	0	8	0	0	0	0	6	6	0	0	9	10	21	22	0	0	12	14	0	2	34	33	0	2
<u>{4}</u>															_				_				_		_
R5, F	M	0		0	0	0	0	က	4	0	0	2	က	~	×	0	0	ŝ	л С	0		15	12	0	
5, R7,	T	20	34	50	60	100	120	41	74	150	181	61	66	110	167	250	301	141	231	350	421	202	331	500	603
${\rm R}$	N + N	20	43	50	60	100	120	53	87	150	181	69	112	139	197	250	301	156	250	350	424	251	376	500	606
	$ V \setminus T $	0	×	0	0	0	0	6	6	0	0	9	10	21	22	0	0	12	14	0	2	34	33	0	2
$35, \mathrm{R4}$	M	0	-1	0	0	0	0	က	4	0	0	2	e C	×	×	0	0	က	S	0	1	15	12	0	-1
, R6, I	T	20	34	50	60	100	120	41	74	150	181	61	66	110	167	250	301	141	231	350	421	202	331	500	603
{R7	V + W	20	43	50	60	100	120	53	87	150	181	69	112	139	197	250	301	156	250	350	424	251	376	500	606
	$ V \setminus T $	20	34	0	20	0	40	40	68	0	60	60	102	100	170	0	100	140	238	0	140	200	340	0	200
iginal	M	60	33	50	20	100	40	120	66	150	60	180	66	300	165	250	100	420	231	350	140	009	330	500	200
rafo Oı	Τ	20	33	50	60	100	120	40	66	150	180	60	66	100	165	250	300	140	231	350	420	200	330	500	600
5	V + W	100	100	100	100	200	200	200	200	300	300	300	300	500	500	500	500	200	700	700	700	1000	1000	1000	1000
Instâncias do	Grupo IV	ctp100-20-60-20	ctp100-33-33-34	ctp100-50-50-0	${ m ctp100-60-20-20}$	$ctp200_{-}100_{-}100_{-}0$	$\operatorname{ctp200-120-40-40}$	${ m ctp200-40-120-40}$	${ m ctp200-66-68-}$	${ m ctp300}_{-150-150-0}$	$ctp300_{-}180_{-}60_{-}60$	${ m ctp300-60-180-60}$	ctp300-99-99-102	$ctp500_{-}100_{-}300_{-}100$	${ m ctp500}_{-165-165-170}$	ctp500_250-250-0	${ m ctp500-300-100-100}$	$ctp700_{-}140_{-}420_{-}140$	${ m ctp700-231-231-238}$	ctp700_350-350-0	${ m ctp700_{-}420}{ m -}140{ m -}140$	ctp1000_200-600-200	$ctp1000_{-330-330-340}$	ctp1000-500-500-0	ctp1000_600-200-200

Tabela B.4: Redução das instâncias do PRR (Grupo IV) - Sequências 1 e 2 das regras de redução

Tabela B.5: Redı	ıção das i	nstân	cias d	o PRR	(Grupo]	- (V)	Sequê	ncias 3	e 4 das re	egras	de rec	lução
Instâncias do	IJ	rafo O	riginal		{B(3, R5, I	37, R4		{R ^r	6, R7, F	$\{6, R4\}$	
Grupo IV	$ \mathbf{V} + \mathbf{W} $	H	M	$ V \setminus T $	$ \mathbf{V} + \mathbf{W} $	T	M	$ V \setminus T $	N + N	T	M	$ V \setminus T $
${ m ctp100-20-60-20}$	100	20	60	20	20	20	0	0	20	20	0	0
ctp100_33-33-34	100	33	33	34	44	34	2	×	44	34	2	×
${\rm ctp100-50-50-0}$	100	50	50	0	50	50	0	0	50	50	0	0
${ m ctp100-60-20}$	100	60	20	20	60	60	0	0	60	60	0	0
${ m ctp200_{-}100_{-}100_{-}0}$	200	100	100	0	100	100	0	0	100	100	0	0
$\operatorname{ctp200_120-40-40}$	200	120	40	40	120	120	0	0	120	120	0	0
${ m ctp200-40-120-40}$	200	40	120	40	54	41	4	6	54	41	4	6
${ m ctp200-66-66-68}$	200	66	66	68	95	74	12	6	95	74	12	6
ctp300_150-150-0	300	150	150	0	150	150	0	0	150	150	0	0
$\operatorname{ctp300_{-}180-60-60}$	300	180	60	60	182	181	-	0	182	181	-1	0
${ m ctp300-60-180-60}$	300	60	180	60	20	61	ŝ	9	70	61	က	9
${ m ctp300_{-}99-99-102}$	300	66	66	102	112	66	ŝ	10	112	66	ĉ	10
${ m ctp500}_{-100-300-100}$	500	100	300	100	149	110	18	21	149	110	18	21
${ m ctp500}_{-165-165-170}$	500	165	165	170	199	167	10	22	199	167	10	22
${ m ctp500}_{-}250-250-0$	500	250	250	0	250	250	0	0	250	250	0	0
${ m ctp500-300-100-100}$	500	300	100	100	302	301	Ţ	0	302	301	1	0
$\operatorname{ctp700_{-}140^{-}420^{-}140}$	200	140	420	140	157	141	4	12	157	141	4	12
ctp700_231-231-238	700	231	231	238	250	231	ъ	14	250	231	ъ	14
${ m ctp700_{-350-350-0}}$	200	350	350	0	350	350	0	0	350	350	0	0
${ m ctp700_{-}420}{ m -}140{ m -}140$	700	420	140	140	425	421	2	2	425	421	2	2
${ m ctp1000-200-600-200}$	1000	200	600	200	253	202	17	34	253	202	17	34
${ m ctp1000}_{-330-330-340}$	1000	330	330	340	377	331	13	33	377	331	13	33
${ m ctp1000-500-500-0}$	1000	500	500	0	500	500	0	0	500	500	0	0
ctp1000_600-200-200	1000	600	200	200	609	603	4	2	609	603	4	2

ãõ
reduc
de 1
gras
s rej
4 da
3 e ^z
ias
uênc
Seq
Ν
odn
G
\sim
PRR (
lo PRR (
cias do PRR (
stâncias do PRR (
s instâncias do PRR (
o das instâncias do PRR (
ução das instâncias do PRR (
Redução das instâncias do PRR (
3.5: Redução das instâncias do PRR (
la B.5: Redução das instâncias do PRR (

Tabela B.6: Redı	ıção das i	nstân	cias d	o PRR	. (Grupo]	- (V)	Sequê	ncias 5	e 6 das r	egras	de re	lução
Instâncias do	0	rafo Oı	riginal		{R:	5, R6, I	37, R4		{R7	7, R5, I	(6, R4)	
Grupo IV	V + W	Τ	M	$ V \setminus T $	$ \mathbf{V} + \mathbf{W} $	T	M	$ V \setminus T $	$ \mathbf{V} + \mathbf{W} $	L	Μ	$ V \setminus T $
ctp100-20-60-20	100	20	60	20	20	20	0	0	20	20	0	0
ctp100_33-33-34	100	33	33	34	44	34	2	×	43	34	-1	8
${\rm ctp100-50-50-0}$	100	50	50	0	50	50	0	0	50	50	0	0
${ m ctp100-60-20-20}$	100	60	20	20	60	60	0	0	60	60	0	0
${ m ctp200_{-}100{-}100{-}0}$	200	100	100	0	100	100	0	0	100	100	0	0
$\operatorname{ctp200_{-}120-40-40}$	200	120	40	40	120	120	0	0	120	120	0	0
${ m ct}{ m p200}_{-40}$ -120-40	200	40	120	40	54	41	4	6	53	41	က	6
${ m ctp200-66-66-68}$	200	66	66	68	95	74	12	6	87	74	4	6
$ctp300_{-}150_{-}150_{-}0$	300	150	150	0	150	150	0	0	150	150	0	0
$\operatorname{ctp300_{-}180-60-60}$	300	180	60	60	182	181	1	0	181	181	0	0
$\operatorname{ctp300-60-180-60}$	300	60	180	60	20	61	°	9	69	61	2	9
${ m ctp300-99-99-102}$	300	66	66	102	112	66	ŝ	10	112	66	ŝ	10
${ m ctp500}_{-100-300-100}$	500	100	300	100	149	110	18	21	139	110	×	21
${ m ctp500}_{-165-165-170}$	500	165	165	170	199	167	10	22	197	167	×	22
${ m ct}{ m p500}_{-}250{ m -}250{ m -}0$	500	250	250	0	250	250	0	0	250	250	0	0
${ m ctp500-300-100-100}$	500	300	100	100	302	301	1	0	301	301	0	0
${ m ctp700_{-}140-420-140}$	700	140	420	140	157	141	4	12	156	141	e	12
ctp700_231-231-238	700	231	231	238	250	231	ъ	14	250	231	ю	14
${ m ctp700_{-350-350-0}}$	200	350	350	0	350	350	0	0	350	350	0	0
${ m ctp700_{-}420\text{-}140\text{-}140}$	700	420	140	140	425	421	2	2	424	421	Ч	2
${ m ctp1000-200-600-200}$	1000	200	600	200	253	202	17	34	251	202	15	34
$ctp1000_{-330-330-340}$	1000	330	330	340	377	331	13	33	376	331	12	33
${ m ctp1000-500-500-0}$	1000	500	500	0	500	500	0	0	500	500	0	0
ctp1000_600-200-200	1000	600	200	200	609	603	4	2	606	603	1	2

gras de reducão	
7) - Sequências 5 e 6 das re	
Grupo IV	
ubela B.6: Reducão das instâncias do PRR (

$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	Instâncias do	G	rafo O	riginal		1	R9 R8	B4}		1	R8 B9	B B 4 }	
$\begin{array}{ c c c c c c c c c c c c c c c c c c c$	Grupo I			W	V\T		T	W	V\T		T	$ \mathbf{w} $	
gtcl 1.1. 3.9.3 1.0 3 3 9 3 1.2 3 5 4 1.2 3 5 4 1.2 3 5 4 1.2 3 5 4 1.2 3 5 4 1.2 3 5 4 1.2 3 5 4 1.1 4 4 3 5 1.1 4 4 3 1.1 4 4 3 1.1 4 4 3 1.1 4 4 1.1 1.1 4 4 3 3 1.1 9 1 1.1 1.1 9 7 1 1 1.1	motp15 2.2.0		2	2			2	<u></u>	2		2	<u> '' </u>	
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	gctp15_3-3-9	15	3	0	3	12	3	5	3 4	12	3	5	
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	gctp15 = 3 - 3 - 3	15		3	7	12		1	4	12		1	4
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	gctp15 = 4-4-7	15	4	4	5	9	4	1	4	9	4		4
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$gctp15_4-0-3$	15	4		1	11	4 7	41	J 1	11	4	1	1
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$gctp15_{-7-1}$	15		1	2	9	1	1	1	9	0	1	1
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	gctp13 = 9-3-3	10	10	10	0	11	9	1	1 9	11	9		1
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	gctp20_10-10-0	20	10	10	0	10	10	1	2	10	10		2
$\begin{array}{c} gc p 2 0 - 4 + 12 + 4 & 1 & 1 & 1 & 4 & 4 & 1 & 1 & 3 & 4 & 4 & 1 & 3 \\ gc p 2 0 - 6 + 4 + 12 & 2 & 4 & 4 & 1 & 2 & 8 & 4 & 1 & 3 & 8 & 4 & 1 & 3 \\ gc p 2 0 - 6 - 6 & 2 & 0 & 6 & 6 & 8 & 6 & 8 & 6 & 1 & 1 & 1 \\ gc p 2 0 - 6 - 6 & 2 & 0 & 6 & 8 & 6 & 8 & 6 & 1 & 1 & 1 \\ gc p 2 0 - 5 - 6 & 2 & 0 & 6 & 1 & 1 & 1 & 1 & 1 & 8 & 6 & 6 & 1 & 1 \\ gc p 2 - 1 - 1 & 2 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 &$	gctp20 = 12 - 4 - 4	20	12	4	4	12	12	4	7	12	12	4	0
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	gctp20 = 4 - 12 - 4	20	4	12	4	10	4	4	1	10	4	4	1
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	gctp20 = 4-4-12	20	4	4	12	0	4	1	ა ე	0	4		ა ე
gctp20-0-8-0 20 0 8 6 8 6 1 1 1 8 6 1 gctp25-121-21 25 15 5 5 15 15 0 00 12 12 0 0 gctp25-5-55 25 5 5 5 15 13 5 2 6 14 5 3 6 144 7 3 44 144 7 3 44 14 7 3 3 21 15 3 3 21 15 3 3 21 15 3 3 21 15 3 3 21 15 3 3 21 15 3 3 21 15 3 3 21 15 3 3 21 15 3 3 21 15 3 3 21 15 3 3 21 15 16 3 5	gctp20_6-6-8	20	0	0	8	12	0	ن ۱	ن ۱	12	b C	3	3
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	gctp20_6-8-6	20	6	8	0	8	0	1	1	8	0		1
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	gctp25_12-12-1	25	12	12	1	12	12	0	0	12	12	0	0
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	gctp25_15-5-5	25	15	5	5	15	15	0	0	15	15	0	0
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	gctp25_5-15-5	25	5	15	5	14	5	3	6	14	5	3	6
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	gctp25_5-5-15	25	5	5	15	13	5	2	6	13	5	2	6
$\begin{array}{c} gctp 25 = 8 - 8 - 9 & 25 & 8 & 8 & 9 & 10 & 8 & 1 & 10 & 8 & 1 & 10 \\ gctp 30 = 15 - 15 - 0 & 30 & 15 & 15 & 0 & 21 & 15 & 3 & 3 & 21 & 15 & 3 & 3 \\ gctp 30 = 6 - 18 & 30 & 6 & 18 & 6 & 17 & 6 & 5 & 6 & 17 & 6 & 5 & 6 \\ gctp 30 = 6 - 18 & 30 & 6 & 6 & 18 & 14 & 6 & 3 & 5 & 14 & 6 & 3 & 5 \\ gctp 30 = 9 - 9 - 12 & 30 & 9 & 9 & 12 & 9 & 11 & 9 & 1 & 11 & 9 & 1 & 11 \\ gctp 30 = 9 - 9 - 12 & 30 & 9 & 9 & 12 & 18 & 9 & 3 & 6 & 18 & 9 & 3 & 6 \\ gctp 35 = 10 - 14 - 11 & 35 & 11 & 11 & 11 & 26 & 10 & 7 & 9 & 26 & 10 & 7 & 9 \\ gctp 35 = 10 - 14 - 11 & 35 & 11 & 11 & 13 & 11 & 11 & 0 & 0 & 0 \\ gctp 35 = 17 - 17 - 1 & 35 & 7 & 7 & 7 & 23 & 21 & 1 & 1 & 23 & 21 & 1 & 1 \\ gctp 35 = 2 - 7 - 7 & 35 & 7 & 7 & 21 & 7 & 20 & 7 & 4 & 4 \\ gctp 35 = -7 - 21 & 35 & 7 & 7 & 21 & 7 & 20 & 7 & 4 & 9 \\ gctp 35 = -7 - 7 & 35 & 7 & 7 & 21 & 31 & 12 & 7 & 12 \\ gctp 40 = 12 - 16 - 2 & 40 & 12 & 16 & 12 & 31 & 12 & 7 & 12 \\ gctp 40 = -13 - 14 & 40 & 13 & 13 & 14 & 21 & 13 & 3 & 5 \\ gctp 40 = -20 - 0 & 40 & 20 & 20 & 0 & 28 & 20 & 2 & 6 \\ gctp 40 = -24 - 8 & 40 & 8 & 24 & 8 & 17 & 8 & 4 & 5 \\ gctp 40 = -8 - 24 & 4 & 8 & 8 & 26 & 24 & 1 & 1 \\ gctp 50 = -10 - 3 & 50 & 10 & 30 & 10 & 12 & 10 & 3 & 3 & 3 & 6 & 33 \\ gctp 50 = -10 - 10 - 30 & 50 & 10 & 30 & 10 & 12 & 10 & 1 & 12 & 10 & 1 \\ gctp 50 = -15 - 0 & 5 & 20 & 15 & 28 & 15 & 5 & 8 & 28 & 15 & 5 \\ gctp 50 = -16 - 18 & 50 & 16 & 18 & 27 & 16 & 4 & 7 \\ gctp 50 = -16 - 18 & 50 & 10 & 10 & 36 & 30 & 3 & 3 & 36 & 33 & 36 & 3 & 3 \\ gctp 60 = -12 - 23 & 6 & 12 & 12 & 36 & 12 & 12 & 16 & 39 & 12 & 11 & 16 \\ gctp 60 = -18 - 24 - 18 & 1$	gctp25_7-10-8	25	7	10	8	14	7	3	4	14	7	3	4
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	gctp25_8-8-9	25	8	8	9	10	8	1	1	10	8	1	1
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	$gctp30_{15-15-0}$	30	15	15	0	21	15	3	3	21	15	3	3
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	gctp30_18-6-6	30	18	6	6	20	18	1	1	20	18	1	1
gctp30-6-6-1830666181463514635gctp30-9-1230912911911119111gctp35-10-14-1135101411261079261079gctp35-11-11-133511111311111001111100gctp35-17-73521772321111232111gctp35-7.21-735721772074920749gctp35-7.21-735772114725144725gctp40-12-16-124012161231127123112712gctp40-20-20-04020200282026282026gctp40-24-8-84088241885518855gctp50-10-30-1050103010121011121011gctp50-16-16-1850161618271647271647gctp50-12-50-155015201528155828 <t< td=""><td>gctp30_6-18-6</td><td>30</td><td>6</td><td>18</td><td>6</td><td>17</td><td>6</td><td>5</td><td>6</td><td>17</td><td>6</td><td>5</td><td>6</td></t<>	gctp30_6-18-6	30	6	18	6	17	6	5	6	17	6	5	6
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	$g \operatorname{ctp} 30 = 6 - 6 - 18$	30	6	6	18	14	6	3	5	14	6	3	5
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	gctp30_9-12-9	30	9	12	9	11	9	1	1	11	9	1	1
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	gctp30_9-9-12	30	9	9	12	18	9	3	6	18	9	3	6
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	gctp35_10-14-11	35	10	14	11	26	10	7	9	26	10	7	9
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	gctp35_11-11-13	35	11	11	13	11	11	0	0	11	11	0	0
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	gctp35_17-17-1	35	17	17	1	25	17	4	4	25	17	4	4
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	gctp35_21-7-7	35	21	7	7	23	21	1	1	23	21	1	1
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	gctp35_7-21-7	35	7	21	7	20	7	4	9	20	7	4	9
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	gctp35_7-7-21	35	7	7	21	14	7	2	5	14	7	2	5
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	gctp40_12-16-12	40	12	16	12	31	12	7	12	31	12	7	12
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	gctp40_13-13-14	40	13	13	14	21	13	3	5	21	13	3	5
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	gctp40_20-20-0	40	20	20	0	28	20	2	6	28	20	2	6
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	gctp40_24-8-8	40	24	8	8	26	24	1	1	26	24	1	1
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	gctp40_8-24-8	40	8	24	8	17	8	4	5	17	8	4	5
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	gctp40_8-8-24	40	8	8	24	18	8	5	5	18	8	5	5
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	gctp50_10-10-30	50	10	10	30	20	10	3	7	20	10	3	7
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	gctp50_10-30-10	50	10	30	10	12	10	1	1	12	10	1	1
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	gctp50_15-20-15	50	15	20	15	28	15	5	8	28	15	5	8
gctp50_25-25-0 50 25 25 0 33 25 2 6 33 25 2 6 gctp50_30-10-10 50 30 10 10 36 30 3 36 30 3 3 gctp60_12-12-36 60 12 12 36 39 12 8 19 39 12 8 19 gctp60_12-36-12 60 12 36 12 39 12 11 16 39 12 11 16 gctp60_18-24-18 60 18 24 18 29 18 4 7 29 18 4 7 gctp60_19-19-22 60 19 19 22 33 19 6 8 33 19 6 8 gctp60_30-30-0 60 30 30 0 40 30 5 5 40 30 5 5	gctp50_16-16-18	50	16	16	18	27	16	4	7	27	16	4	7
gctp50_30-10-10 50 30 10 10 36 30 3 36 30 3 3 gctp60_12-12-36 60 12 12 36 39 12 8 19 39 12 8 19 gctp60_12-36-12 60 12 36 12 39 12 11 16 39 12 11 16 gctp60_18-24-18 60 18 24 18 29 18 4 7 29 18 4 7 gctp60_19-19-22 60 19 19 22 33 19 6 8 33 19 6 8 gctp60_30-30-0 60 30 30 0 40 30 5 5 40 30 5 5	gctp50_25-25-0	50	25	25	0	33	25	2	6	33	25	2	6
gctp60_12-12-36 60 12 12 36 39 12 8 19 39 12 8 19 gctp60_12-36-12 60 12 36 12 39 12 11 16 39 12 11 16 gctp60_18-24-18 60 18 24 18 29 18 4 7 29 18 4 7 gctp60_19-19-22 60 19 19 22 33 19 6 8 33 19 6 8 33 19 6 8 5 5	gctp50_30-10-10	50	30	10	10	36	30	3	3	36	30	3	3
gctp60_12-36-12 60 12 36 12 39 12 11 16 39 12 11 16 gctp60_18-24-18 60 18 24 18 29 18 4 7 29 18 4 7 gctp60_19-19-22 60 19 19 22 33 19 6 8 33 19 6 8 gctp60_30-30-0 60 30 30 0 40 30 5 5 40 30 5 5	gctp60_12-12-36	60	12	12	36	39	12	8	19	39	12	8	19
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	gctp60_12-36-12	60	12	36	12	39	12	11	16	39	12	11	16
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	gctp60_18-24-18	60	18	24	18	29	18	4	7	29	18	4	7
sctp60 30-30-0 60 30 30 0 40 30 5 5 40 30 5 5	gctp60_19-19-22	60	19	19	22	33	19	6	8	33	19	6	8
	gctp60_30-30-0	60	30	30	0	40	30	5	5	40	30	5	5
gctp60_36-12-12 60 36 12 12 38 36 1 1 38 36 1 1	gctp60_36-12-12	60	36	12	12	38	36	1	1	38	36	1	1

Tabela B.7: Redução das instâncias do PRRG (Grupo I) - Sequências 1 e 2 das regras de redução

Instâncias do	G	rafo O	riginal		1	R4 R8	8 B9}		1	R8 R 4	1 B9}	
Grupo I			W	V\T	V + W	T	W	V\T				V\T
motp15 2.2.0		2	2			2				2		4
$gctp15_{-}3-3-9$	15	3	3	9	9	3	2	4 F	9	3	Z E	4
gctp15_5-9-5	15	3	9	3 7	10	3	1	0	10	3	1	0
gctp15 = 4-4-7	15	4	4		13	4	1	8	13	4	1	8
gctp15_4-0-5	15	4	0	D 1	12	4	4	4	12	4	4	4
gctp15_7-7-1	15	1	1	1	15	1	1	1	15	1	1	(
gctp15_9-3-3	15	9	3	3	14	9	1	4	14	9	1	4
gctp20_10-10-0	20	10	10	0	20	10	1	9	20	10	1	9
gctp20_12-4-4	20	12	4	4	17	12	0	5	17	12	0	5
gctp20_4-12-4	20	4	12	4	20	4	4	12	20	4	4	12
gctp20_4-4-12	20	4	4	12	12	4	1	7	11	4	1	6
$gctp20_{-}6-6-8$	20	6	6	8	15	6	3	6	15	6	3	6
$gctp20_{-6-8-6}$	20	6	8	6	18	6	1	11	18	6	1	11
gctp25_12-12-1	25	12	12	1	25	12	0	13	25	12	0	13
$gctp25_{-}15_{-}5_{-}5$	25	15	5	5	21	15	0	6	21	15	0	6
$gctp25_{-}5-15-5$	25	5	15	5	25	5	3	17	25	5	3	17
$gctp25_{-}5-5-15$	25	5	5	15	14	5	2	7	14	5	2	7
$gctp25_{-}7-10-8$	25	7	10	8	22	7	3	12	20	7	3	10
gctp25_8-8-9	25	8	8	9	23	8	1	14	23	8	1	14
gctp30_15-15-0	30	15	15	0	30	15	3	12	30	15	3	12
gctp30_18-6-6	30	18	6	6	26	18	1	7	25	18	1	6
gctp30_6-18-6	30	6	18	6	30	6	5	19	30	6	5	19
gctp30_6-6-18	30	6	6	18	23	6	3	14	22	6	3	13
gctp30_9-12-9	30	9	12	9	30	9	1	20	30	9	1	20
gctp30_9-9-12	30	9	9	12	25	9	3	13	24	9	3	12
gctp35_10-14-11	35	10	14	11	31	10	7	14	30	10	7	13
gctp35_11-11-13	35	11	11	13	35	11	0	24	35	11	0	24
gctp35-17-17-1	35	17	17	1	34	17	4	13	34	17	4	13
gctp35_21-7-7	35	21	7	7	30	21	1	8	30	21	1	8
gctp35_7-21-7	35	7	21	7	35	7	4	24	35	7	4	24
gctp35_7-7-21	35	7	7	21	25	7	2	16	25	7	2	16
gctp40_12-16-12	40	12	16	12	38	12	7	19	37	12	7	18
gctp40 13-13-14	40	13	13	14	39	13	3	23	38	13	3	22
gctp40 20-20-0	40	20	20	0	40	20	2	18	40	20	2	18
gctp40 24-8-8	40	24	8	8	34	24	- 1	9	34	24	-	9
gctp40 8-24-8	40	8	24	8	39	8	4	27	39	8	4	27
gctp40_8-8-24	40	8	8	24	20	8	5	7	20	8	5	
gctp50 = 10-10-30	50	10	10	30	33	10	3	20	20	10	3	14
gctp50 = 10 - 10 - 30 - 10	50	10	30	10	50	10	1	39	50	10	1	39
gctp50 = 10-30-10 gctp50 = 15/20/15	50	15	20	15	46	15	5	26	45	15	5	25
gctp50 = 15 - 20 - 15 gctp50 = 16 - 16 - 18	50	16	16	18	40	16	4	20	40	16	4	20
getp50_10-10-18	50	25	25	10	50	25	4 0	24	50	25	4 0	24
gctp30=23-23-0	50	20 20	20	10		20	2	20 19		20	2	20 10
getp60 10-10-10	50 60	ას 19	10	10	40	ას 19	3 0	12	40	30 19	3 0	12
gctp60_12-12-30	60	12	12	30 10	40	12	8	20	40	12	ð 11	20
gctp60_12-36-12	00	12	30	12	50	12		33 25	54	12	11	31 24
gctp60_18-24-18	00	18	24	18	57	18	4	35	50	18	4	34
gctp60_19-19-22	60	19	19	22	45	19	6	20	44	19	6	19
gctp60_30-30-0	60	30	30	0	60	30	5	25	60	30	5	25
gctp60_36-12-12	60	36	12	12	57	36		20	57	36		20

Tabela B.8: Redução das instâncias do PRRG (Grupo I) - Sequências 3 e 4 das regras de redução
4}	V V	7 32	8 14	6	5	ۍ ۳	6	3 54	7 14	7 18	8	0 94	9 29	1 130	6 58	7 32	8	7 211	7 33	4 13	1 9	5 232	0 35	2 4	11
8, R9, R	T V	20 1	33	50 8	60	100	120	40 2	99	150 7	180	60 3	66	100 8	165 2	250 1	300 (140 8	231 7	350 4	420 4	200 8	330 1	500	600 1
{R	V + W	69	55	67	02	106	131	117	87	175	193	184	137	311	249	299	314	438	271	367	433	517	375	506	691
	$ V \setminus T $	32	14	6	2	ŝ	9	54	14	18	×	94	29	130	58	32	×	211	33	13	6	232	35	4	11
$, R4 \}$	$ \mathbf{W} $	17	×	×	ъ	ŝ	ъ	23	7	2	ъ	30	6	81	26	17	9	87	2	4	4	85	10	2	10
R9, R8	L	20	33	50	60	100	120	40	66	150	180	60	66	100	165	250	300	140	231	350	420	200	330	500	600
	$ \mathbf{V} + \mathbf{N} $	69	55	67	20	106	131	117	87	175	193	184	137	311	249	299	314	438	271	367	433	517	375	506	694
	$ V \setminus T $	20	34	0	20	0	40	40	68	0	60	60	102	100	170	0	100	140	238	0	140	200	340	0	006
iginal	M	60	33	50	20	100	40	120	66	150	60	180	66	300	165	250	100	420	231	350	140	600	330	500	006
rafo Oı	$ \mathbf{T} $	20	33	50	60	100	120	40	66	150	180	60	66	100	165	250	300	140	231	350	420	200	330	500	600
	V + W	100	100	100	100	200	200	200	200	300	300	300	300	500	500	500	500	200	700	200	700	1000	1000	1000	1000
Instâncias do	Grupo II	gctp100-20-60-20	gctp100-33-33-34	gctp100-50-50-0	gctp100_60-20-20	$gctp200_{-}100_{-}100_{-}0$	gctp200_120-40-40	gctp200_40-120-40	gctp200_66-66-68	gctp300_150-150-0	gctp300_180-60-60	gctp300-60-180-60	gctp300_99-99-102	gctp500-100-300-100	gctp500_165-165-170	gctp500_250-250-0	gctp500_300-100-100	$gctp700_{-}140_{-}420_{-}140$	gctp700_231-231-238	gctp700_350-350-0	gctp700_420-140-140	gctp1000_200-600-200	gctp1000_330-330-340	gctp1000-500-500-0	ectn1000 600-200-200

Tabela B.9: Redução das instâncias do PRRG (Grupo II) - Sequências 1 e 2 das regras de redução

Tabela B.10: Redu	ıção das ii	nstâne	cias d	o PRR(G (Grupo	- (II)	Sequ	ências 3	s e 4 das r	egras	de re	dução
Instâncias do	5	rafo Or	iginal		}	R4, R8	$, R9 \}$		}	R8, R4	$, R9 \}$	
Grupo II	V + W	T	M	$ V \setminus T $	V + W	L	M	$ V \setminus T $	N + N	T	M	V T
gctp100-20-60-20	100	20	09	20	100	20	17	63	66	20	17	62
gctp100-33-33-34	100	33	33	34	93	33	×	52	92	33	8	51
gctp100-50-50-0	100	50	50	0	100	50	×	42	100	50	×	42
gctp100_60-20-20	100	60	20	20	87	60	ъ	22	87	60	S	22
gctp200_100-100-0	200	100	100	0	200	100	ŝ	67	200	100	က	97
gctp200-120-40-40	200	120	40	40	176	120	ъ	51	173	120	ъ	48
gctp200-40-120-40	200	40	120	40	199	40	23	136	199	40	23	136
gctp200_66-66-68	200	66	66	68	191	99	7	118	191	66	7	118
gctp300_150-150-0	300	150	150	0	300	150	7	143	300	150	2	143
gctp300_180-60-60	300	180	60	60	275	180	ъ	90	273	180	ъ	88
gctp300_60-180-60	300	60	180	60	298	60	30	208	298	60	30	208
gctp300_99-99-102	300	66	66	102	294	66	6	186	291	66	6	183
gctp500_100-300-100	500	100	300	100	495	100	81	314	493	100	81	312
gctp500_165-165-170	500	165	165	170	465	165	26	274	458	165	26	267
$gctp500_{-}250_{-}250_{-}0$	500	250	250	0	500	250	17	233	500	250	17	233
gctp500-300-100-100	500	300	100	100	457	300	9	151	453	300	9	147
gctp700_140-420-140	200	140	420	140	696	140	87	469	691	140	87	464
gctp700_231-231-238	700	231	231	238	694	231	7	456	693	231	7	455
gctp700_350-350-0	700	350	350	0	700	350	4	346	200	350	4	346
gctp700_420-140-140	700	420	140	140	674	420	4	250	670	420	4	246
gctp1000-200-600-200	1000	200	600	200	997	200	85	712	995	200	85	710
gctp1000_330-330-340	1000	330	330	340	980	330	10	640	974	330	10	634
gctp1000-500-500-0	1000	500	500	0	1000	500	2	498	1000	500	2	498
gctp1000-600-200-200	1000	600	200	200	926	600	10	316	919	600	10	309

e 4 das regras de redução
I) - Sequências 3
Π
(Grupo
tedução das instâncias do PRRG
: R
B.1(
abela

APÊNDICE C – Resultados complementares dos experimentos com o CPLEX

Este apêndice apresenta resultados dos experimentos computacionais realizados para o PRR e para o PRRG com a formulação $\mathcal{F}2$, acrescida das desigualdades válidas sugeridas em (MANIEZZO et al., 1999) para melhorar o valor dos limites inferiores. Vale lembrar que, nestas condições, a formulação $\mathcal{F}2$ é denotada por $\mathcal{F}2'$, assim como especificado no Capítulo 4.

Os resultados reportados a seguir complementam os discutidos no Capítulo 4 e ressaltam o benefício da aplicação dos conjuntos de regras de redução para o PRR^1 e para o $PRRG^2$, propostos respectivamente nas Seções 3.1.2 e 3.2.2 deste trabalho.

Os experimentos deste apêndice foram realizados com a parametrização algoritmo branch-and-cut utilizado pelo CPLEX como definido na Seção 4.3. O tempo de processamento foi limitado a 172.800 segundos em todos os testes. O ambiente computacional utilizado foi o mesmo descrito na Seção 4.2.

Nas Tabelas C.1 e C.2 são organizados os resultados obtidos a partir da relaxação linear de $\mathcal{F}2'$, dos quais são apresentados para os grafos originais os valores dos ótimos duais (3^{*a*} coluna), o total de iterações utilizadas pelo CPLEX para a obtenção deste ótimo (4^{*a*} coluna) e o tempo total de execução (5^{*a*} coluna). As três últimas colunas apresentam os resultados equivalentes para os grafos reduzidos. Também é apresentada a diferença percentual entre os ótimos duais obtidos com os grafos originais em relação aos ótimos duais obtidos com os grafos reduzidos (2^{*a*} coluna).

Finalizando este apêndice, as Tabelas C.3 e C.4 apresentam um comparativo entre os valores das soluções inteiras ótimas e os valores das soluções ótimas da relaxação linear. A 2^a coluna apresenta a diferença percentual entre estas soluções para os grafos originais e a 3^a coluna apresenta a mesma diferença para os grafos reduzidos.

¹ A sequência de regras utilizadas na redução dos grafos associados ao PRR foi {R7, R6, R5, R4}

² Para o PRRG, a sequência de regras utilizada na redução dos grafos foi {R9, R8, R4}

	Original				Re	eduzido			
Grupo III	Dif%	Ótimo Dual	Iter	Tempo	Ótimo Dual	Iter	Tempo		
ctp15 3-3-9	9,86%	912.242,00	115	0,00	1.002.222,00	9	0,00		
ctp15_3-9-3	1,08%	1.037.171,00	38	$0,\!00$	1.048.362,73	39	$0,\!00$		
ctp15 4-4-7	2,45%	1.154.726,75	138	0,00	1.183.052,00	25	0,00		
ctp15 4-6-5	0,61%	1.211.941,67	91	$0,\!00$	1.219.318,40	32	$0,\!00$		
ctp15 7-7-1	22,08%	851.188,50	86	0,00	1.039.139,00	43	0,00		
ctp15 9-3-3	5,42%	1.218.922,29	207	0,01	1.284.941,81	81	0,00		
ctp20 10-10-0	6,34%	1.147.437,24	147	0,00	1.220.193,81	73	0,00		
ctp20 12-4-4	9,13%	1.463.495,86	330	0,01	1.597.165,26	235	0,01		
ctp20 4-12-4	20,85%	774.693,56	71	0,00	936.184,00	16	0,00		
ctp20 4-4-12	26,39%	887.912,47	315	0,01	1.122.207,16	34	0,00		
ctp20 6-6-8	29,04%	983.791,11	226	0,01	1.269.495,60	27	0,00		
ctp20_6-8-6	16,06%	1.076.484,12	171	0,01	1.249.357,38	32	0,00		
ctp25 12-12-1	6,49%	1.499.818,86	273	0,01	1.597.165,26	235	0,01		
ctp25 15-5-5	6.79%	1.541.217,28	570	0.02	1.645.874.96	317	0.01		
ctp25 5-15-5	12.88%	1.026.351.00	83	0,00	1.158.525,25	21	0,00		
ctp25 5-5-15	10,04%	969.158,39	377	0,02	1.066.455,50	108	0,00		
ctp25 7-10-8	21,13%	1.097.962.61	277	0,01	1.329.943.86	40	0,00		
ctp25_8-8-9	21.36%	1.111.915,22	334	0.01	1.349.451,06	46	0,00		
ctp30 15-15-0	4.74%	1.571.436,78	323	0,01	1.645.874,96	317	0,01		
ctp30 18-6-6	3.90%	1.830.504.28	895	0.04	1.901.970.50	557	0.02		
ctp30_6-18-6	7.23%	1.189.188.56	185	0.01	1.275.108.00	51	0.00		
ctp30_6-6-18	- * -	- * -	- *	- * -	- * -	- * -	- * -		
ctp30 9-12-9	22.97%	1.096.200.00	393	0.02	1.347.959.81	80	0.00		
ctp30 9-9-12	36.30%	988.962.11	515	0.02	1.347.959.81	80	0.00		
ctp35 10-14-11	12.39%	1.597.933.43	688	0.03	1.795.915.39	279	0.01		
	14.44%	1.422.598.68	666	0.03	1.627.954.24	333	0.01		
ctp35 17-17-1	0.54%	1.683.829.65	538	0.02	1.692.932.71	369	0.01		
ctp35 21-7-7	9.88%	1.851.726,95	1283	0.07	2.034.622,18	1061	0.05		
ctp35 7-21-7	10,23%	1.491.244,29	345	0,01	1.643.853,75	75	0,00		
ctp35 7-7-21	14.41%	1.195.100,61	974	0.06	1.367.267,12	246	0.01		
ctp40 12-16-12	24.80%	1.449.786,79	794	0,04	1.809.328,23	345	0,01		
ctp40 13-13-14	14.94%	1.664.940,89	1115	0.04	1.913.701.26	515	0.02		
ctp40_20-20-0	0.35%	1.697.877,03	618	0,02	1.703.888,06	658	0.02		
ctp40 24-8-8	7.21%	2.021.842,10	1477	0.09	2.167.536,66	975	0.05		
ctp40 8-24-8	37,34%	993.189,18	312	0,01	1.364.051,27	53	0,00		
ctp40 8-8-24	16,63%	1.329.657,76	1371	0,08	1.550.757,08	293	0,01		
ctp50 10-10-30	28,13%	1.174.340,31	2039	0,18	1.504.642,92	330	0,01		
ctp50 10-30-10	36,21%	895.801,27	513	0,02	1.220.193,81	73	0,00		
ctp50 15-20-15	12,59%	1.616.230,88	1230	0,06	1.819.660,34	388	0,01		
ctp50 16-16-18	15,11%	1.603.690,03	1506	0,11	1.845.955,60	498	0,02		
ctp50 25-25-0	1,33%	2.139.449,66	1283	0,06	2.167.959,32	949	0,04		
ctp50_30-10-10	6,15%	2.139.180,08	2808	$0,\!24$	2.270.644,13	1309	0,08		
ctp60 12-12-36	37,08%	1.302.898,07	2624	$0,\!27$	1.785.961,67	627	0,02		
ctp60 12-36-12	39,94%	1.141.295,37	772	0,03	1.597.165,26	235	0,01		
ctp60 18-24-18	$22,\!39\%$	1.431.584,35	1902	$0,\!16$	1.752.056,44	575	0,01		
ctp60 19-19-22	28,42%	1.367.705,99	2497	$0,\!19$	1.756.395,62	573	0,01		
ctp60_30-30-0	0,47%	2.260.071,14	1633	0,09	2.270.644, 13	1309	0,08		
ctp60_36-12-12	7,39%	2.230.354,92	3907	0,41	$2.395.241,\!94$	2159	$0,\!14$		
Média	14,93%		830,96	0,05		355,85	0,01		

Tabela C.1: Ótimos duais obtidos com a formulação $\mathcal{F}2'$ para o PRR - Grupo III

Legenda

- \star — Não foi encontrada nenhuma solução para relaxação linear de ${\cal F}2'$

	Original				Re Re	eduzido	
Grupo I	Dif%	Ótimo Dual	Iter	Tempo	Ótimo Dual	Iter	Tempo
gctp15_3-3-9	25,74%	468.487,50	211	$0,\!05$	$589.094,\!50$	60	0,00
gctp15_3-9-3	9,86%	729.511,27	278	$0,\!01$	801.421,68	168	0,01
$gctp15_4-4-7$	$17{,}52\%$	850.326,00	205	0,01	999.338,89	77	0,00
$gctp15_4-6-5$	$20,\!02\%$	998.284,62	237	$0,\!01$	1.198.091, 91	151	0,00
gctp15_7-7-1	$29{,}31\%$	990.510, 39	252	0,01	1.280.849,33	110	0,00
gctp15_9-3-3	$12,\!50\%$	$1.383.262,\!67$	282	$0,\!01$	$1.556.108,\!57$	167	0,00
$gctp20_{10-10-0}$	$56,\!38\%$	930.843,73	455	$0,\!02$	$1.455.674{,}50$	241	0,01
$gctp20_{12-4-4}$	10,56%	$1.907.258,\!68$	500	$0,\!02$	$2.108.597,\!67$	254	0,01
$gctp20_4-12-4$	22,40%	850.432,59	397	$0,\!02$	1.040.933,75	206	0,01
$gctp20_4-4-12$	58,98%	1.014.132,33	440	$0,\!02$	1.612.235,44	66	0,00
$gctp20_6-6-8$	$14,\!66\%$	$1.065.007,\!59$	473	$0,\!02$	$1.221.151,\!63$	186	0,01
$gctp20_6-8-6$	$46,\!36\%$	963.173,71	429	$0,\!02$	$1.409.663,\!14$	95	0,00
$gctp25_{12-12-1}$	11,88%	1.545.886,32	747	$0,\!04$	$1.729.491,\!45$	227	0,01
$gctp25_15$ -5-5	$11,\!43\%$	1.440.696,05	823	$0,\!04$	1.605.355,04	297	0,01
$gctp25_5$ -15-5	$10,\!38\%$	799.007, 61	638	$0,\!03$	$881.975,\!07$	202	0,01
$\operatorname{gctp}{25}_{5}$ -5-15	$24,\!99\%$	610.466,93	579	$0,\!03$	763.037,70	193	0,01
$gctp25_7-10-8$	$23,\!09\%$	1.087.288,45	738	$0,\!04$	1.338.386,95	253	0,01
$gctp25_8-8-9$	$23{,}51\%$	1.095.626,05	609	$0,\!03$	1.353.243,86	141	0,00
$gctp30_{15-15-0}$	6,29%	$1.646.503,\!18$	1401	$0,\!09$	$1.750.009,\!67$	671	0,03
$gctp30_18$ -6-6	$13,\!45\%$	1.768.889,57	1260	$0,\!08$	2.006.796,19	561	0,02
$gctp30_6-18-6$	$35,\!31\%$	828.918,46	885	$0,\!04$	1.121.582,48	373	0,01
$gctp30_6-6-18$	- * -	- * -	- * -	- * -	- * -	- * -	- * -
$gctp30_9-12-9$	$37,\!66\%$	1.119.534,78	899	$0,\!06$	$1.541.132,\!63$	175	0,00
$gctp30_9-9-12$	$17,\!57\%$	1.173.966,46	1278	$0,\!09$	1.380.248,33	454	0,02
$gctp35_10-14-11$	6,26%	1.289.352,92	1553	$0,\!11$	1.370.045,72	1058	0,06
$gctp35_{11-11-13}$	36,06%	877.170,44	1252	$0,\!08$	1.193.460,00	99	0,00
gctp35_17-17-1	$19,\!63\%$	$1.524.322,\!12$	1778	$0,\!13$	1.823.479,47	905	0,05
$gctp35_21-7-7$	7,69%	2.016.378,45	1863	$0,\!14$	2.171.423,05	782	0,04
$gctp35_7-21-7$	$22,\!22\%$	1.093.035,98	1428	$0,\!12$	$1.335.903,\!04$	492	0,02
gctp35_7-7-21	$_{32,30\%}$	858.799,76	1432	$0,\!13$	1.136.181,32	225	0,01
$gctp40_{12-16-12}$	$23,\!66\%$	1.300.671,38	1987	$0,\!18$	$1.608.422,\!12$	1334	0,11
gctp40_13-13-14	19,86%	1.487.902,45	2077	$0,\!17$	1.783.427,11	592	0,03
$gctp40_{20-20-0}$	5,43%	2.149.580,86	2130	$0,\!18$	2.266.390,79	1229	0,07
gctp40_24-8-8	30,30%	1.748.285,49	2449	$0,\!20$	2.278.056,28	975	0,05
gctp40_8-24-8	47,88%	1.070.677,28	1873	$0,\!15$	$1.583.281,\!27$	410	0,02
gctp40_8-8-24	37,36%	1.212.703,96	1841	$0,\!14$	1.665.806, 61	359	0,01
gctp50_10-10-30	40,29%	946.638,79	2426	$0,\!28$	1.328.042,53	504	0,02
gctp50_10-30-10	43,74%	1.262.233,08	2974	$0,\!37$	1.814.287,00	186	0,01
$gctp50_{15-20-15}$	$21,\!32\%$	1.464.534,01	3343	$0,\!40$	1.776.769,08	1252	0,08
gctp50_16-16-18	28,46%	1.489.564,59	3136	$0,\!38$	1.913.482,29	1028	0,06
$gctp50_{25-25-0}$	$20,\!13\%$	1.716.530,12	3608	$0,\!43$	2.062.097,96	1650	0,11
gctp50_30-10-10	12,26%	2.039.388,70	4133	$0,\!54$	2.289.361,31	2157	0,13
gctp60_12-12-36	15,91%	1.319.429,37	4970	0,92	1.529.345,47	1965	0,21
gctp60_12-36-12	27,21%	1.239.675,64	4947	0,81	1.577.016,35	2177	0,18
gctp60_18-24-18	33,00%	1.399.639,22	4279	0,71	$1.861.536,\!67$	1260	0,07
gctp60_19-19-22	20,39%	1.543.110,05	5197	$0,\!87$	1.857.767,83	1845	0,15
gctp60_30-30-0	7,30%	2.225.174,31	7515	1,20	2.387.525,63	2944	0,23
36-12-12	5,71%	2.453.858,91	6073	0,95	2.594.022,29	2654	0,26
Média	23, 49%	-	$1878,\!30$	0,22	-	$710,\!85$	0,05

Tabela C.2: Ótimos duais obtidos com a formulação $\mathcal{F}2'$ para o PRRG - Grupo I

Legenda

* — Não foi encontrada nenhuma solução para relaxação linear de ${\cal F}2'$

Instâncias	Grafo Original	Grafo Reduzido
do Grupo III	Ótimo PPLIxLP	Ótimo PPLIxLP
ctp15 3-3-9	8,98%	0,00%
ctp15 3-9-3	8,94%	7,95%
ctp15 4-4-7	3,46%	1,09%
ctp15 4-6-5	0,77%	0,17%
ctp15 7-7-1	26,23%	9.94%
ctp15 9-3-3	7,42%	2,40%
ctp20 10-10-0	8,34%	2,53%
ctp20 12-4-4	12,93%	4,97%
ctp20 4-12-4	17,25%	0,00%
ctp20 4-4-12	22,99%	$2,\!67\%$
ctp20 6-6-8	24,07%	2,01%
ctp20_6-8-6	14,73%	1,04%
ctp25 12-12-1	10,76%	4,97%
ctp25 15-5-5	12,63%	6,70%
ctp25 5-15-5	11,99%	0,65%
ctp25 5-5-15	20,78%	12,83%
ctp25 7-10-8	19.68%	2,71%
ctp25_8-8-9	19.38%	2.15%
ctp30 15-15-0	10,92%	6,70%
ctp30 18-6-6	9,55%	6,02%
ctp30 6-18-6	9,00%	2,43%
ctp30_6-6-18	- * -	-*-%
ctp30 9-12-9	$20{,}54\%$	2,29%
ctp30 9-9-12	$28,\!32\%$	$2,\!29\%$
ctp35_10-14-11	11,72%	0,78%
ctp35_11-11-13	13,74%	$1,\!29\%$
ctp35_17-17-1	4,97%	4,46%
$ctp35_{21-7-7}$	12,79%	4,18%
ctp35_7-21-7	$10,\!30\%$	$1,\!13\%$
ctp35_7-7-21	23,72%	12,73%
ctp40_12-16-12	$25{,}63\%$	$7,\!19\%$
ctp40_13-13-14	18,99%	6,89%
ctp40_20-20-0	6,94%	$6{,}61\%$
ctp40_24-8-8	9,91%	3,42%
ctp40_8-24-8	28,74%	$2,\!13\%$
ctp40_8-8-24	$19{,}59\%$	6,22%
ctp50_10-10-30	$29,\!11\%$	9,18%
ctp50_10-30-10	28,44%	2,53%
$ctp50_{15-20-15}$	$16{,}59\%$	6,09%
ctp50_16-16-18	17,53%	5,08%
ctp50_25-25-0	$4,\!68\%$	3,41%
ctp50_30-10-10	10,54%	5,04%
ctp60_12-12-36	$33,\!17\%$	8,39%
ctp60_12-36-12	$32,\!10\%$	4,97%
ctp60_18-24-18	24,06%	7,05%
ctp60_19-19-22	$27,\!65\%$	7,09%
ctp60_30-30-0	5,48%	5,04%
36-12-12		6,02%
Média	17,89%	4,43%

Tabela C.3: Comparação dos resultados da formulação $\mathcal{F}2'$ obtidos para o PRR - Instâncias originais e reduzidas do Grupo III

Legenda

- * -

Não foi encontrada nenhuma solução para relaxação linear de $\mathcal{F}2'$

Instâncias	Grafo Original	Grafo Reduzido
do Grupo I	Ótimo PPLIxLP	Ótimo PPLIxLP
gctp15 3-3-9	51,26%	38,71%
gctp15 3-9-3	$29,\!61\%$	22,67%
gctp15 4-4-7	18,89%	4,67%
gctp15 4-6-5	31,93%	18,30%
gctp15 7-7-1	30,34%	9,92%
gctp15 9-3-3	22,96%	13,34%
gctp20 10-10-0	40,48%	6,93%
gctp20 12-4-4	10,56%	1,11%
gctp20 4-12-4	34,31%	19,59%
gctp20 4-4-12	40,66%	5,66%
gctp20 6-6-8	28,57%	18,10%
gctp20 6-8-6	36,84%	7,57%
gctp25 12-12-1	15,51%	5,48%
gctp25 15-5-5	14,92%	5,19%
gctp25 5-15-5	38,06%	31,63%
gctp25 5-5-15	$37,\!62\%$	22,03%
gctp25 7-10-8	34,55%	19,43%
gctp25 8-8-9	23,24%	5,20%
gctp30 15-15-0	6,21%	0,32%
gctp30 18-6-6	15,94%	4,63%
gctp30 6-18-6	51,46%	34,32%
gctp30 6-6-18	- * -	- * -
gctp30 9-12-9	32,32%	6,83%
gctp30 9-9-12	30,39%	18,15%
gctp35 10-14-11	34,44%	30,34%
gctp35 11-11-13	31,62%	6,97%
gctp35 17-17-1	$25,\!80\%$	11,24%
gctp35 21-7-7	17,32%	10,97%
gctp35 7-21-7	$34,\!61\%$	20,08%
$gctp35_7-7-21$	36,29%	15,72%
gctp40_12-16-12	$37,\!34\%$	22,51%
gctp40_13-13-14	24,79%	9,85%
gctp40_20-20-0	$13,\!97\%$	9,30%
gctp40_24-8-8	27,08%	4,99%
gctp40_8-24-8	39,56%	10,63%
gctp40_8-8-24	40,34%	18,05%
gctp50_10-10-30	40,29%	16,23%
gctp50_10-30-10	30,43%	0,00%
$gctp50_{15-20-15}$	$_{32,22\%}$	17,77%
gctp50_16-16-18	27,24%	6,53%
$gctp50_{25-25-0}$	20,92%	5,00%
gctp50_30-10-10	20,16%	10,38%
gctp60_12-12-36	40,69%	31,25%
gctp60_12-36-12	$45,\!60\%$	30,79%
gctp60_18-24-18	32,70%	10,49%
$gctp60_{19-19-22}$	29,75%	15,42%
gctp60_30-30-0	19,42%	13,54%
$gctp60_36-12-12$	10,79%	5,70%
Média	29,57%	13,90%
Legenda	· · · · · · · · · · · · · · · · · · ·	· · · · ·

Tabela C.4: Comparação dos resultados da formulação $\mathcal{F}2'$ obtidos para o PRRG - Instâncias originais e reduzidas do Grupo I

* — Não foi encontrada nenhuma solução para relaxação linear de $\mathcal{F}2'$

APÊNDICE D - Testes de inferência estatística

O presente apêndice realiza uma breve descrição dos testes estatísticos utilizados para avaliar o desempenho dos algoritmos propostos no Capítulo 5 para resolver de forma aproximada o PRR e o PRRG.

D.1 Análise de variância (ANOVA)

A análise de variância (*ANalysis Of VAriance* - ANOVA) é um teste estatístico que visa fundamentalmente a comparação de médias provenientes de grupos¹ diferentes (VI-EIRA, 2006) (LEVINE, 2008).

Segundo Levine (2008), para realizar um teste ANOVA para igualdade entre as médias aritméticas de populações², subdivide-se a variação total em dois componentes de variância: aquela que é decorrente da diferença entre os grupos (isto é, variância entre as heurísticas) e aquela que é decorrente da diferença dentro dos grupos. Se a variância entre os grupos for maior do que dentro dos grupos, isso pode indicar que existe uma diferença significativa entre os grupos analisados, ou seja, as variâncias são estatisticamente diferentes e algum tratamento se destaca sobre os demais (existe uma heurística com variância estatisticamente diferente das demais).

O teste de hipóteses é uma regra de decisão utilizada para aceitar ou rejeitar uma hipótese estatística com base em elementos amostrais (LEVINE, 2008) (VIEIRA, 2006). Basicamente são descritas duas hipóteses:

•Hipótese nula, representada por H_0 ;

•Hipótese alternativa, representada por H_1 .

¹ No contexto deste trabalho, os grupos equivalem às heurísticas propostas que, estatisticamente, também são chamadas de tratamentos.

² Quando a ANOVA é utilizada para testar a igualdade entre médias aritméticas de algoritmos heurísticos, entende-se uma população como sendo um algoritmo heurístico aplicado a um conjunto de instâncias para o problema que está sendo resolvido.

Geralmente a hipótese alternativa representa a suposição que se deseja provar e a hipótese nula é formulada com o propósito de ser rejeitada. Conseguindo rejeitar a hipótese nula (H_0) , a hipótese alternativa deverá ser aceita (H_1) .

Os pressupostos da ANOVA são:

- •Os κ grupos representam populações cujos valores são selecionados de maneira aleatória independente, seguem uma distribuição normal e possuem variâncias iguais;
- •A hipótese nula (H_0) baseia-se no fato de que não existe diferença nas médias aritméticas das populações, isto é, $\mu_1 = \mu_2 = \dots = \mu_{\kappa}$;
- •A hipótese alternativa baseia-se no fato de que nem todas as κ médias aritméticas das populações são iguais.

Deve-se destacar que, pelos pressupostos da ANOVA, a variável de interesse deve ter distribuição normal (teste paramétrico) e os grupos analisados devem ser independentes entre si. Para verificar se as amostras estão normalmente distribuídas é necessário saber se uma ou mais delas podem ser consideradas como extraídas da mesma população. Como a distribuição normal é determinada pela média e pelo desvio padrão (ou variância), se estas estatísticas não diferirem significativamente entre si, pode-se aceitar que elas foram extraídas da mesma população. Precisa-se ainda determinar se as diferenças entre as médias e variâncias amostrais devem-se somente às variações aleatórias ou se os dados vêm de populações onde elas são diferentes. Mesmo que seja possível concluir que estas estatísticas amostrais são diferentes, ainda será necessário decidir se estas diferenças podem ser consideradas como de importância significativa (estatisticamente diferentes).

Supondo que amostras aleatórias e independentes de κ populações estejam disponíveis, com θ observações cada e denotando-se as κ médias por $\mu_1, \mu_2, ..., \mu_{\kappa}$, uma análise de variância para um fator (*One-way*) pode ser conduzida para se testar a hipótese nula descrita da seguinte forma:

 H_0 : Todas as κ populações têm a mesma média, isto é, $\mu_1 = \mu_2 = \dots = \mu_{\kappa}$;

 H_1 : Nem todas as κ populações têm a mesma média.

A estatística F testa a hipótese nula de que todas as populações têm a mesma média, ou seja, H_0 é a hipótese de que as amostras pertencem à populações com o mesmo valor médio e que quaisquer diferenças encontradas são, provavelmente, devidas ao acaso. A hipótese alternativa baseia-se no fato de que quaisquer diferenças detectadas são devidas ao efeito dos tratamentos (heurísticas). Sendo MQE a média dos quadrados entre os grupos (variância entre as médias provenientes dos grupos) e MQD a média dos quadrados dentro dos grupos (variância dentro dos grupos), pode-se definir a estatística do teste Fda seguinte forma:

$$F = \frac{MQE}{MQD} \tag{D.1}$$

A tabela de distribuição F permite expressar todos os possíveis valores desta razão entre as variâncias. A razão F aumentará muito quando as médias dos grupos forem muito diferentes (LEVINE, 2008) (VIEIRA, 2006). Caso H_0 não seja rejeitada, então não é possível inferir que há uma diferença sistemática atuando que explique a diferença entre os valores amostrais melhor do que a ação do acaso. Neste caso, verifica-se que as variâncias não são estatisticamente diferentes (médias homocedásticas), o que é uma forte indicação que as amostras pertencem a mesma população e estão normalmente distribuídas. Com base nesta análise, pode-se concluir que se enquadram neste caso as distribuições em que os possíveis valores da variável aleatória tenham a mesma probabilidade de ocorrência.

Quando não forem detectadas diferenças significativas entre as variâncias no teste de ANOVA, ainda é possível verificar se há uma diferença sistemática atuando entre os tratamentos (heurísticas) tomados dois a dois, isto é, ainda é possível verificar se estas médias diferem estatisticamente. Para tal, a distribuição de probabilidade estatística *t*-*Student* é amplamente utilizada para a comparação entre duas médias de populações com distribuição normal e amostras independentes em testes de hipóteses (LEVINE, 2008), assim como é descrito na seção a seguir.

D.2 Teste *t*-Student

A distribuição de probabilidade estatística denominada *t-Student* permite realizar a comparação entre duas médias de populações com distribuição normal e amostras independentes. Se as médias diferirem estatisticamente segundo esta distribuição, pode-se afirmar que as amostras não são homogêneas e que há uma diferença sistemática atuando, o que indica diferenças entre os tratamentos (heurísticas, no caso deste trabalho), apesar de não terem sido detectadas diferenças significativas entre as variâncias (teste de ANOVA).

Segundo Levine (2008), para utilizar o teste estatístico t-Student é necessário supor

que os grupos têm distribuição normal ou aproximadamente normal e que o desvio padrão (variância) dos dois grupos é semelhante. Com base nestas pressuposições, o teste estatístico *t-Student* permite testar a hipótese nula $H_0: \mu_1 = \mu_2$, com as seguintes possíveis hipóteses alternativas: $H_1: \mu_1 > \mu_2$; $H_1: \mu_1 < \mu_2$; $H_1: \mu_1 \neq \mu_2$, dependendo do problema que está sendo proposto. A hipótese nula anteriormente descrita é equivalente à: $H_0: \mu_1 - \mu_2 = 0$, onde μ_1 e μ_2 são médias de populações distintas.

A distribuição *t-Student* foi escolhida como base para as análises dos algoritmos propostos neste trabalho por ser uma medida de distribuição de probabilidade estatística amplamente utilizada para a comparação de médias amostrais em testes de hipóteses. Além disso, este teste estatístico é um caso especial de análise de variância para teste entre dois grupos com variáveis quantitativas (ANOVA de um fator) (LEVINE, 2008).

Conforme mencionado no final da Seção D.1, a utilização do teste estatístico t-Student é recomendada quando a normalidade e a igualdade das variâncias são verificadas, o que poderia ser demonstrado pela aplicação de um teste F, tendo como resultado a não rejeição da hipótese nula (H_0) de igualdade entre médias. É possível aplicar este teste estatístico considerando a alternativa unilateral (quando $H_1 : \mu_1 > \mu_2$ ou $H_1 : \mu_1 < \mu_2$) e/ou bilateral $(H_1 : \mu_1 \neq \mu_2)$ para análise da área de rejeição de H_0 . Portanto, deve-se confrontar o valor observado (valor do t calculado ou t_{calc}) com a área de rejeição de H_0 para um nível de significância pré-fixado. É aceitável estatisticamente ter o nível de significância em torno de 5%. Se t_{calc} cair na área de rejeição de H_0 , opta-se pela rejeição da hipótese nula. O valor t_{calc} pode ser obtido pela seguinte equação:

$$t_{calc} = \frac{X_1 - X_2}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \tag{D.2}$$

Sendo X_1 e X_2 estimadores de médias populacionais de duas amostras, S_1^2 e S_2^2 as variâncias amostrais e $(n_1 + n_2 - 2)$ o número de graus de liberdade necessário para o cálculo de t_{calc} com a distribuição *t-Student*, pode-se obter o valor de S_p através da equação:

$$S_p = \sqrt{\frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}}$$
(D.3)

Na abordagem de Neyman e Pearson fixa-se antecipadamente a hipótese nula (H_0) , a alternativa (H_1) e o nível de significância (margem de erro), para então tomar a decisão. Deve-se confrontar o valor de t_{calc} com a área de rejeição de H_0 considerando o nível de significância pré-estabelecido (margem de erro). Caso t_{calc} esteja na área de rejeição de H_0 , a hipótese nula deverá ser rejeitada.

Na abordagem de Fisher especifica-se uma hipótese inicial, equivalente a H_0 de Neyman e Pearson mas, para tomada de decisão, deve-se realizar o experimento, calcular a probabilidade de ocorrência do valor p^3 , a probabilidade de ocorrência do valor observado (ou de um valor mais extremo da estatística do teste) em uma curva de probabilidade especificada. A decisão estatística será tomada com base no valor desta probabilidade. Se o valor de p for considerado pequeno, rejeita-se a hipótese nula (H_0) e se o valor de pfor considerado grande, então rejeita-se a hipótese alternativa (H_1).

A distribuição *t-Student* também pode ser utilizada quando as amostras são pequenas, ou seja, podendo o número de elementos amostrais ser inferior a 30 (LEVINE, 2008).

 $^{^3~}$ O valor p é a probabilidade de obter uma amostra, no mínimo, tão extrema quanto a amostra dada.

APÊNDICE E – Resultados complementares dos experimentos com as heurísticas

Este apêndice apresenta alguns resultados complementares dos experimentos computacionais realizados neste trabalho para avaliar as heurísticas propostas para o PRR e para o PRRG. Outros resultados foram disponibilizados para acesso público no seguinte endereço eletrônico: http://labic.ic.uff.br.

As tabelas que seguem neste apêndice estão organizadas pelo problema que foi considerado (PRR ou PRRG) e pelo tipo de heurística que foi testada.

Instâncias	Cplex	Instâncias Ori	ginais	Instâncias Red	uzidas
GIII e GIV	Solução ótima	Melhor solução	Dif%	Melhor solução	Dif%
gctp15_3-3-9	1.002.222	1.002.222	0.00%	1.002.222	0.00%
gctp15_3-9-3	1.138.939	1.138.939	0,00%	1.138.939	0,00%
gctp15_4-4-7	1.196.129	1.196.129	0,00%	1.196.129	0,00%
gctp15_4-6-5	1.221.377	1.221.377	0,00%	1.221.377	0,00%
gctp15_7-7-1	1.153.884	1.153.884	0,00%	1.153.884	0,00%
gctp15_9-3-3	1.316.581	1.316.581	0,00%	1.316.581	0,00%
gctp20_10-10-0	1.251.833	1.251.833	0,00%	1.251.833	0,00%
gctp20_12-4-4	1.680.738	1.680.738	0,00%	1.680.738	0,00%
$gctp20_4-12-4$	936.184	936.184	0,00%	936.184	0,00%
$gctp20_4-4-12$	1.153.040	1.153.040	0,00%	1.153.040	0,00%
$gctp20_66-8$	1.295.576	1.295.576	0,00%	1.295.576	0,00%
$gctp20_68-6$	1.262.473	1.262.473	0,00%	1.262.473	0,00%
$gctp25_12-12-1$	1.680.738	1.680.738	0,00%	1.680.738	0,00%
$gctp25_{15-5-5}$	1.764.081	1.764.081	0,00%	1.764.081	0,00%
$gctp25_{5-15-5}$	1.166.123	1.166.123	0,00%	1.166.123	0,00%
$gctp25_55-15$	1.223.360	1.224.325	0,08%	1.223.360	0,00%
$gctp25_710-8$	1.367.025	1.367.025	0,00%	1.367.025	0,00%
$gctp25_88-9$	1.379.150	1.379.150	0,00%	1.379.150	0,00%
$gctp30_{15-15-0}$	1.764.081	1.764.081	0,00%	1.764.081	0,00%
$gctp30_{18-6-6}$	2.023.833	2.044.770	1,02%	2.023.833	0,00%
$gctp30_{6-18-6}$	1.306.845	1.306.845	0,00%	1.306.845	0,00%
$gctp30_66-18$	1.347.522	1.347.522	0,00%	1.347.522	0,00%
$gctp30_9-12-9$	1.379.599	1.379.599	0,00%	1.379.599	0,00%
$gctp30_9-9-12$	1.379.599	1.379.599	0,00%	1.379.599	0,00%
gctp35_10-14-11	1.810.009	1.837.538	1,50%	1.810.009	0,00%
$gctp35_11-11-13$	1.649.229	1.679.066	1,78%	1.649.229	0,00%
$gctp35_17-17-1$	1.771.975	1.771.975	0,00%	1.771.975	0,00%
$gctp35_21-7-7$	2.123.301	2.153.163	1,39%	2.123.301	0,00%
$gctp35_7-21-7$	1.662.568	1.662.568	0,00%	1.662.568	0,00%
$gctp35_77-21$	1.566.645	1.573.535	0,44%	1.566.645	0,00%
$gctp40_{12}-16-12$	1.949.544	1.954.170	$0,\!24\%$	1.949.544	0,00%
$gctp40_{-}13$ -13-14	2.055.310	2.074.415	0,92%	2.071.205	0,77%
$gctp40_20-20-0$	1.824.574	1.824.574	0,00%	1.824.574	0,00%
$gctp40_24-8-8$	2.244.183	2.274.045	1,31%	2.244.183	0,00%
$gctp40_{-}8-24-8$	1.393.676	1.393.676	0,00%	1.393.676	0,00%
$gctp40_{-}8-8-24$	1.653.663	1.662.332	0,52%	1.653.663	0,00%
$gctp50_{10-10-30}$	1.656.656	1.661.282	$0,\!28\%$	1.656.656	0,00%
$gctp50_10-30-10$	1.251.833	1.256.459	0,37%	1.251.833	0,00%
$gctp50_{15-20-15}$	1.937.583	1.937.583	0,00%	1.937.583	0,00%
gctp50_16-16-18	1.944.648	1.944.648	0,00%	1.944.648	0,00%
$gctp50_{25-25-0}$	2.244.596	2.274.458	1,31%	2.274.458	1,31%
gctp50_30-10-10	2.391.136	2.438.548	1,94%	2.431.031	1,64%
$gctp60_12-12-36$	1.949.532	1.961.481	0,61%	1.952.634	0,16%
$gctp60_12-36-12$	1.680.738	1.680.738	0,00%	1.680.738	0,00%
gctp60_18-24-18	1.885.042	1.893.926	0,47%	1.885.042	0,00%
gctp60_19-19-22	1.890.330	1.890.330	0,00%	1.890.330	0,00%
gctp60_30-30-0	2.391.136	2.431.031	1,64%	2.391.136	0,00%
gctp60_36-12-12	2.548.788	2.583.219	1,33%	2.548.788	0,00%
gctp100_20-60-20	2.232.584	2.232.584	0,00%	2.232.584	0,00%
gctp100_33-33-34	2.503.833	2.541.475	1,48%	2.533.919	1,19%
gctp100_50-50-0	2.737.137	2.742.309	0,19%	2.742.104	0,18%
gctp100_60-20-20	3.256.940	3.391.532	3,97%	3.285.442	0,87%

Tabela E.1: Comparação entre as melhores soluções obtidas pelas oito heurísticas construtivas para o PRR e as soluções ótimas conhecidas

Instâncias	Cplex	Instâncias Ori	ginais	Instâncias Red	uzidas
GI e GII	Solução ótima	Melhor solução	Dif%	Melhor solução	Dif%
ctp15_3-3-9	961.154	961.153	0.00%	961.153	0.00%
ctp15_3-9-3	1.036.409	1.036.409	0,00%	1.036.409	0,00%
ctp15_4-4-7	1.048.315	1.048.315	0.00%	1.048.315	0,00%
ctp15_4-6-5	1.466.511	1.466.511	0.00%	1.466.511	0.00%
ctp15_7-7-1	1.421.948	1.421.948	0.00%	1.421.948	0.00%
ctp15_9-3-3	1.795.582	1.795.582	0.00%	1.795.582	0.00%
ctp20_10-10-0	1.564.006	1.564.006	0.00%	1.564.006	0.00%
ctp20_12-4-4	2.132.347	2.132.347	0.00%	2.132.347	0.00%
$ctp20_4-12-4$	1.294.603	1.294.603	0.00%	1.294.603	0.00%
ctp20_4-4-12	1.708.966	1.708.966	0.00%	1.708.966	0.00%
ctp20_6-6-8	1.491.061	1.564.237	4.68%	1.530.415	2.57%
ctp20_6-8-6	1.525.053	1.525.053	0.00%	1.525.053	0.00%
ctp25_12-12-1	1 829 687	1 829 687	0.00%	1 829 687	0.00%
ctp25 = 12 + 12 + 12 ctp25 = 15 - 5 - 5	1.693.310	1.693.310	0.00%	1.693.310	0.00%
$ctp25_5-15-5$	1.290.023	1.290.023	0.00%	1.290.023	0.00%
$ctp25_{-5}$ 10 0 $ctp25_{-5}$ 5-5-15	978 586	978 586		978 586	0.00%
$ctp25_{-5} = 5 = 5 = 10$ $ctp25_{-7} = 10-8$	1 661 147	1 661 147	0.00%	1 661 147	0.00%
ctp25_8-8-9	1.001.111 1.427.431	1.001.111 1.427.431		1.001.111 1.427.431	
ctp20 = 0.015	1 755 605	1 755 605	0,00%	1 755 605	0,00%
ctp30_18-6-6	2 104 277	2 116 803	0,0070	2 116 803	0,0070
$ctp30_{6-18-6}$	1 707 545	1 707 545	0,00%	1 707 545	0,00%
$ctp30_{-6.6,18}$	1.506.710	1.506.710	0,0070	1 506 710	0,0070
$ctp30_012_0$	1.654.180	1.602.415	2 26%	1.654.180	0,0070
$ctp30_9-12-9$	1.686.404	1.686.404	2,2070	1.686.404	0,0070
$ctp35 \pm 10.14.11$	1.066.605	2 040 444	3.61%	2 040 444	3.61%
$ctp35_{11-11-13}$	1 282 834	1 282 834	0.01%	1 282 834	0.00%
$ctp35_{17}171$	2.054.434	2 007 386	2.05%	2 007 386	2.05%
$ctp35_17-17-1$ $ctp35_2177$	2.034.434	2.097.380	2,0570	2.097.380	2,0570
$ctp35_21-7-7$	1 671 510	1.671.510	0,9470	1 671 510	0,2070
$ctp35_7-21-7$	1 3/8 038	1 3/8 038	0,0070	1 3/8 038	0,0070
$ctp35_{-7-21}$	1.548.058 2.075.742	2 1 48 243	2 37%	2 1 27 106	2 420%
$ctp40_12-10-12$ $atp 40_12, 12, 14$	2.075.742	2.140.243	3,3770	2.127.190	2,4270
$ctp40_10_10_10_14$	2 408 675	2 626 103	4 85%	2 566 803	2 65%
$ctp40_20-20-0$	2.490.015	2.020.105	4,0070	2.300.805	2,00%
$ctp40_24-8-8$	2.397.045	2.397.045	0,0070	2.397.045	0,0070
$ctp40_0-24-8$	2 032 735	1.771.590 2.032.734		1.771.090 2.032.734	0,0070
$ctp40_0-0-24$	1 585 400	1 500 947	0,0070	1 5 95 409	0,0070
$ctp50_10-10-30$ $ctp50_10_30_10$	1.000.409	1.399.247	0,0170	1.303.400	0,0070
$ctp50_{15} - 10 - 30 - 10$	2 160 737	2 201 664	5 71%	2 256 288	1 23%
$ctp50_{15}-20-15$	2.100.757	2.291.004 2.047.142	0,7170	2.230.288	4,2370
$ctp50_10-10-18$	2.047.142	2.047.142	2 6 9 07	2.047.142	0,0070
$ctp50_20-20-0$	2.170.030	2.203.081	3,0070	2.220.214	2,2470
$ctp30_30-10-10$	2.334.400	2.072.793	4,4370	2.371.010	0,0770
$ctp00_12-12-30$	2.224.475	2.220.390	0,1070	2.220.090	0,1870
$ctp00_12-30-12$	2.278.014	2.301.199	3,0070	2.301.199	3,5070
ctp60_10-24-10	2.079.730	2.140.070	2,0070	2.110.100	1,02/0 2,010%
$ctp60_19-19-22$	2.190.040	2.212.097	5 44%	2.240.104	2,21/0 2.970%
ctp60_36_19_19	2.701.327	2.320.134	1 61 02	2.020.001	2,2770 1 1502
$c_{100} = 30 - 12 - 12$	2.130.103	2.005.102	5 62%	2.102.109	5 55 970
ctp100_20-00-20	2.011.039	2.001.101	5,0370	2.000.000	1 1 2 07
$c_{tp100} = 33 - 33 - 34$	2.022.204	3.003.200 3.717.659	6 77%	2.940.202 3.708 K9K	6 540%
ctp100_50-50-0	3 577 215	3.717.002	6 86%	3.700.020	6 86%
- 00-20-20	1 0.011.010	0.040.004	0,0070	0.040.004	0,0070

Tabela E.2: Comparação entre as melhores soluções obtidas pelas oito heurísticas construtivas para o PRRG e as soluções ótimas conhecidas

Instâncias	Cplex	Instâncias Red	uzidas
GIII e GIV	Solução ótima	Melhor solução	Dif%
ctp15_3-3-9	1.002.222	1.002.222	0.00%
ctp15_3-9-3	1.138.939	1.138.939	0.00%
ctp15_4-4-7	1.196.129	1.196.129	0,00%
ctp15_4-6-5	1.221.377	1.221.377	0.00%
ctp15_7-7-1	1.153.884	1.153.884	0.00%
ctp15_9-3-3	1.316.581	1.316.581	0.00%
ctp20_10-10-0	1.251.833	1.251.833	0.00%
ctp20_12-4-4	1.680.738	1.680.738	0.00%
ctp20_4-12-4	936.184	936.184	0.00%
ctp20_4-4-12	1.153.040	1.153.040	0.00%
ctp20_6-6-8	1.295.576	1.295.576	0,00%
ctp20_6-8-6	1.262.473	1.262.473	0.00%
ctp25_12-12-1	1.680.738	1.680.738	0.00%
ctp25_15-5-5	1.764.081	1.764.081	0,00%
ctp25_5-15-5	1.166.123	1.166.123	0,00%
$ctp25_55-15$	1.223.360	1.223.360	0,00%
ctp25_7-10-8	1.367.025	1.367.025	0,00%
ctp25_8-8-9	1.379.150	1.379.150	0,00%
ctp30_15-15-0	1.764.081	1.764.081	0,00%
ctp30_18-6-6	2.023.833	2.023.833	0,00%
ctp30_6-18-6	1.306.845	1.306.845	0,00%
ctp30_6-6-18	1.347.522	1.347.522	0,00%
ctp30_9-12-9	1.379.599	1.379.599	0,00%
ctp30_9-9-12	1.379.599	1.379.599	0,00%
ctp35_10-14-11	1.810.009	1.810.009	0,00%
ctp35_11-11-13	1.649.229	1.649.229	0,00%
ctp35_17-17-1	1.771.975	1.771.975	0,00%
$ctp35_21-7-7$	2.123.301	2.123.301	$0,\!00\%$
$ctp35_7-21-7$	1.662.568	1.662.568	$0,\!00\%$
$ctp35_77-21$	1.566.645	1.566.645	$0,\!00\%$
$ctp40_{12}$ -16-12	1.949.544	1.949.544	$0,\!00\%$
$ctp40_{13-13-14}$	2.055.310	2.055.310	$0,\!00\%$
ctp40_20-20-0	1.824.574	1.824.574	$0,\!00\%$
$ctp40_{24-8-8}$	2.244.183	2.244.183	$0,\!00\%$
$ctp40_8-24-8$	1.393.676	1.393.676	$0,\!00\%$
$ctp40_88-24$	1.653.663	1.653.663	$0,\!00\%$
ctp50_10-10-30	1.656.656	1.656.656	$0,\!00\%$
$ctp50_10-30-10$	1.251.833	1.251.833	$0,\!00\%$
$ctp50_{15-20-15}$	1.937.583	1.937.583	$0,\!00\%$
ctp50_16-16-18	1.944.648	1.944.648	$0,\!00\%$
ctp50_25-25-0	2.244.596	2.244.596	$0,\!00\%$
$ctp50_30-10-10$	2.404.830	2.404.830	$0,\!00\%$
ctp60_12-12-36	1.949.532	1.949.532	$0,\!00\%$
ctp60_12-36-12	1.680.738	1.680.738	$0,\!00\%$
ctp60_18-24-18	1.885.042	1.885.042	$0,\!00\%$
ctp60_19-19-22	1.890.330	1.890.330	0,00%
ctp60_30-30-0	2.391.136	2.391.136	0,00%
ctp60_36-12-12	2.548.788	2.548.788	0,00%
ctp100_20-60-20	2.232.584	2.232.584	0,00%
ctp100_33-33-34	2.503.833	2.503.833	0,00%
ctp100_50-50-0	2.737.137	2.745.120	$0,\!29\%$
ctp100_60-20-20	3.256.940	3.299.334	$1,\!28\%$

Tabela E.3: Comparação entre as melhores soluções obtidas pelo GRASP_Bas
1 para o PRR e as soluções ótimas conhecidas

Instâncias	Cplex	Instâncias Red	uzidas
GI e GII	Solução ótima	Melhor solução	Dif%
gctp15_3-3-9	961.154	961.153	0.00%
gctp15_3-9-3	1.036.409	1.036.409	0.00%
gctp15_4-4-7	1.048.315	1.048.315	0.00%
gctp15_4-6-5	1.466.511	1.466.511	0.00%
gctp15_7-7-1	1.421.948	1.421.948	0.00%
gctp15_9-3-3	1.795.582	1.795.582	0.00%
gctp20_10-10-0	1.564.006	1.564.006	0.00%
gctp20_12-4-4	2.132.347	2.132.347	0.00%
gctp20_4-12-4	1.294.603	1.294.603	0.00%
gctp20_4-4-12	1.708.966	1.708.966	0.00%
gctp20_6-6-8	1.491.061	1.491.061	0.00%
gctp20_6-8-6	1.525.053	1.525.053	0.00%
gctp25_12-12-1	1.829.687	1.829.687	0.00%
gctp25 15-5-5	1.693.310	1.693.310	0.00%
$gctp25_5-15-5$	1.290.023	1.290.023	0.00%
$gctp25_5-5-15$	978.586	978.586	0.00%
gctp25 7-10-8	1.661.147	1.661.147	0.00%
gctp25 8-8-9	1.427.431	1.427.431	0.00%
gctp30_15-15-0	1 755 605	1 755 605	0.00%
gctp30_18-6-6	2.104.277	2.104.276	0.00%
gctp30_6-18-6	1 707 545	1 707 545	0.00%
gctp30_6-6-18	1.506.710	1.506.710	0.00%
$gctp30_9-12-9$	1.654.180	1.654.180	0.00%
gctp30_9-9-12	1.686.404	1.686.404	0.00%
gctp35_10-14-11	1.966.695	1.966.695	0.00%
gctp35_11-11-13	1.282.834	1.282.834	0.00%
gctp35 17-17-1	2.054.434	2.054.434	0.00%
gctp35_21-7-7	2.438.858	2.438.858	0.00%
$gctp35_7-21-7$	1.671.519	1.671.519	0.00%
$gctp35_7-7-21$	1.348.038	1.348.038	0.00%
gctp40_12-16-12	2.075.742	2.075.742	0.00%
gctp40_13-13-14	1.978.318	1.978.318	0.00%
gctp40_20-20-0	2.498.675	2.498.675	0.00%
gctp40_24-8-8	2.397.645	2.397.645	0,00%
gctp40_8-24-8	1.771.596	1.771.596	0,00%
gctp40_8-8-24	2.032.735	2.032.735	0,00%
gctp50_10-10-30	1.585.409	1.585.409	0,00%
gctp50_10-30-10	1.814.287	1.814.287	0,00%
gctp50_15-20-15	2.160.737	2.160.737	0,00%
gctp50_16-16-18	2.047.142	2.047.142	0,00%
gctp50_25-25-0	2.170.630	2.170.630	0,00%
$gctp50_30-10-10$	2.554.406	2.554.406	0,00%
$gctp60_12-12-36$	2.224.473	2.224.473	0,00%
$gctp60_12-36-12$	2.278.614	2.278.614	0,00%
$gctp60_{18-24-18}$	2.079.736	2.079.736	0,00%
$gctp60_19-19-22$	2.196.545	2.196.545	0,00%
$gctp60_30-30-0$	2.761.327	2.761.327	0,00%
$gctp60_36-12-12$	2.750.783	2.750.783	0,00%
$gctp100_20-60-20$	2.677.859	2.703.845	0,96%
$gctp100_33-33-34$	2.822.284	2.845.817	0,83%
$gctp100_{50-50-0}$	3.465.968	3.561.591	$2,\!68\%$
$gctp100_{60-20-20}$	3.577.315	3.647.102	1,91%

Tabela E.4: Comparação entre as melhores soluções obtidas pelo GRASP_Bas1 para o PRRG e as soluções ótimas conhecidas