

UNIVERSIDADE FEDERAL FLUMINENSE

LUCIANA BRUGIOLO GONÇALVES

**Heurísticas para o Problema de Cobertura de  
Conjuntos por Pares**

NITERÓI

2010

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

UNIVERSIDADE FEDERAL FLUMINENSE

LUCIANA BRUGIOLO GONÇALVES

# Heurísticas para o Problema de Cobertura de Conjuntos por Pares

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 em Computação. Área de concentração: Otimização Combinatória e Inteligência Artificial.

Orientadora:

Simone de Lima Martins

Co-orientador:

Luiz Satoru Ochi

NITERÓI

2010

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

G635 Gonçalves, Luciana Brugiolo.  
Heurísticas para o problema de cobertura de conjuntos por pares /  
Luciana Brugiolo Gonçalves. – Niterói, RJ : [s.n.], 2010.  
90 f.

Tese (Doutorado em Computação) - Universidade Federal  
Fluminense, 2010.

Orientadores: Simone de Lima Martins, Luiz Satoru Ochi.

1. Heurística. 2. Problema de cobertura de conjuntos por pares. 3.  
Modelo matemático. I. Título.

CDD 005.136

# Heurísticas para o Problema de Cobertura de Conjunto por Pares

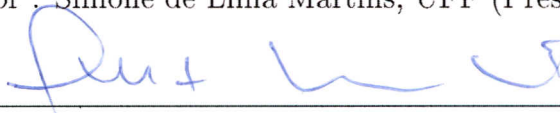
Luciana Brugiolo Gonçalves

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 em Computação.

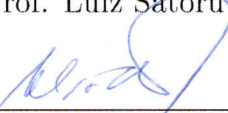
Aprovada por:



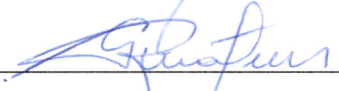
Prof<sup>a</sup>. Simone de Lima Martins, UFF (Presidente)



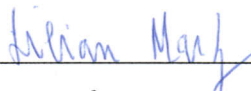
Prof. Luiz Satoru Ochi, UFF



Prof. Celso da Cruz Carneiro Ribeiro, UFF



Prof. Geraldo Robson Mateus, UFMG



Prof<sup>a</sup>. Lilian Markenzon, UFRJ

Niterói, 20 de agosto de 2010.

*À minha família, bem mais precioso que tenho.*

# Agradecimentos

Primeiramente gostaria de agradecer a Deus por me acompanhar a cada passo, por me dar forças e por não faltar em momento algum.

Gostaria de agradecer também à aqueles que me acompanham desde meus primeiros passos, Mamãe e Papai! Maria Lucia e Carlos Magno são pessoas especiais, que não medem esforços para ver a realização dos filhos e que o fazem com o coração aberto. Obrigada pelo apoio, pelo incentivo e pela dedicação. Amo vocês!

Deixo meu agradecimento também aos maninhos e a maninha. A minha amada Bruninha, que faz sentir-me um pouco mãe, sendo a mim motivo de grande alegria. Tenho que agradecer a esta pessoinha por recarregar minhas baterias a cada final de semana, dando-me forças para enfrentar as últimas etapas desta jornada. Aos membros das famílias Brugiolo e Gonçalves, que souberam entender minha ausência, agradeço pelas palavras de incentivo a cada encontro e pela torcida.

Aos meus grandes orientadores, Simone e Satoru, agradeço por cada palavra de estímulo, por cada agradável reunião, pela amizade e paciência.

Ao professor Maurício Resende agradeço pela ajuda na definição do problema tratado neste trabalho. Agradeço também aos professores que participaram da banca de defesa, professores Celso, Geraldo Robson e Lilian Markenzon. Obrigada pelos comentários e pelas valiosas contribuições. Gostaria de registrar meu agradecimento a Abílio Lucena, que contribuiu muito em sua participação na proposta de tese e que infelizmente não pode participar da defesa final. Ao professor Fábio Proti agradeço pela disponibilidade de participar da banca e pelas inúmeras consultas quanto ao término deste trabalho. Fábio, finalmente acabou!

Agradeço também a galera do suporte e das secretarias: Carlos Eduardo, Rafael, Teresa, Viviane, Ângela, Maria, Verinha, Maristér, Carlinhos e cia. Muito obrigada por tudo.

O IC é um cantinho especial da UFF! Aqui a família cresce e ganhamos irmão dos mais variados lugares. Tenho muito a agradecer aos amigos que fiz por aqui, que tanto

contribuíram, deram força e fizeram companhia. Àqueles da salinha 350C, um grupo maravilhoso que vai deixar saudades. Um agradecimento especial a Renatha e Adria, duas irmãs que ganhei durante o doutorado. Obrigada por cada dia compartilhado, cada conversa e pela ajuda nas mais diversas situações. Agradeço também ao Anand, o caçulinha, que tanto contribuiu para o desenvolvimento deste trabalho. Espero que tenhamos cuidado direitinho de você. Ao Puca agradeço pela ajuda nas mais diversas situações e por não ter me expulsado da salinha! Janine, Mário e Gustavo, obrigada por tudo. Todos vocês são especiais.

Aos vizinhos da sala 350D: irmãzinha Juliana, super Jacques, Luciene, Flávio, Cristiano, Diego, Maurício, Juliano e Thibaut. Adoraria tê-los como vizinhos de trabalho por mais tempo. Obrigada pelas inúmeras vezes que se colocaram a ajudar e pelo carinho de sempre. Ao Jacques um agradecimento especial pelas inúmeras vezes que me fez companhia pelos caminhos do Ingá, pela presteza de sempre e pelo carinho. Agradeço também a aqueles que precisa gastar minha digital ao visitá-los: Jonivan, André Renato, Tiago Neves, Aline, Alexandre, Haroldo, Luiz Merschmann, Lu Pessôa, Vinicius, Copetti, Idalmes, Aletéia, Johnny, galera de Ouro Preto, Hugo, Leonardo e tantos outros que contribuem/contribuíram para tornar o IC este lugar especial.

Há alguém a quem preciso fazer um agradecimento especial, Stênio Sã. Não tenho como agradecer pelo incentivo, tanto no início deste caminho, quando me ajudou a decidir pelo doutorado, quanto em cada etapa conquistada. Foi muito bom ter passado por este caminho tendo-o ao meu lado, podendo contar com você em todos os momentos. Nas vésperas de defesas e apresentações, quando eu mesma não me suportava, e também nas comemorações e viagens que ficarão para sempre na memória. Obrigada pela ajuda, por cada palavra de carinho e pela paciência. Adoro você!

Gostaria de agradecer também a Bruninha e Renathinha por terem me emprestado o dia de suas comemorações de mesiversário e aniversário para a defesa deste trabalho. Valeu!

Enfim, agradeço o apoio financeiro que tornou viável esta empreitada, inicialmente a CAPES e por fim a FAPERJ.



# Resumo

Esta tese trata do Problema de Cobertura de Conjuntos por Pares (PCCP), uma generalização do Problema de Cobertura de Conjuntos. Considerando  $\mathcal{U}$  o conjunto dos elementos a serem cobertos e  $\mathcal{A}$  o conjunto dos objetos, onde para cada elemento a ser coberto há uma lista composta por pares de objetos que o cobrem, no PCCP o objetivo é selecionar um subconjunto de objetos de forma que, para cada elemento, pelo menos um par de objetos pertencente a sua lista esteja contido no subconjunto de objetos selecionado. O PCCP possui aplicações na áreas de monitoramento remoto em redes de computadores e biologia computacional. Neste trabalho são apresentadas duas formulações matemáticas da literatura, assim como um novo modelo que descrevem este problema. Além disso, algoritmos baseados nas metaheurísticas *Greedy Randomized Adaptive Search Procedure* e *Iterated Local Search* são propostos. Uma heurística híbrida que combina estratégias construtivas e o modelo proposto também é avaliada. Nos experimentos com as abordagens citadas, três conjuntos de instâncias foram utilizados. O primeiro conjunto é composto por instâncias da literatura para o problema de monitoramento em redes de computadores. Nos testes envolvendo este conjunto de instâncias foi possível obter a solução ótima através da formulação proposta. Outros dois conjuntos de instâncias foram sugeridos neste trabalho. Para as instâncias onde o valor da solução ótima não foi definido, um limite superior pôde ser identificado. Utilizando as heurísticas propostas foi possível, para algumas instâncias, reduzir o limite superior definido pelo método exato.

**Palavras-chave:** Problema de Cobertura de Conjuntos por Pares, modelo matemático, heurísticas.

# Abstract

This thesis deals with the Set Cover with Pairs Problem (SCPP), a generalization of the Set Cover Problem (SCP). In the SCPP the elements must be covered by specific pairs of objects, rather than a single object. For each element to be covered there is a list composed by pairs of objects that can cover it. The objective is to choose a subset of objects in such a way that, for each element, at least one pair of objects belonging to its list is contained in the selected subset of objects. This problem has applications in remote computer network monitoring and computational biology. In this work two mathematical formulations from the literature are presented as well as a new one proposed for this problem. Also, algorithms based on the Greedy Randomized Adaptive Search Procedure and Iterated Local Search metaheuristics are proposed. A heuristic that combines constructive strategies and the new mathematical model is also evaluated. In the computational experiments, three set of instances were used. The first set is composed by literature instances for the computer network monitoring problem. In the tests involving this set of instances, it was possible to obtain the optimal solution using the proposed formulation. Another two sets of instances were developed in this work. For the instances where the value of the optimal solution was not defined, an upper bound could be identified. For some instances, using the proposed heuristics enabled to reduce the upper bound defined by the exact methods.

**Keywords:** Set Cover with Pairs Problem, mathematical formulation, heuristics.

# Glossário

BLTO	:	Busca Local Troca Objeto;
BLTP	:	Busca Local Troca Par;
BLDA	:	Busca Local <i>Drop Add</i> ;
GRASP	:	<i>Greedy Randomized Adaptive Search Procedure</i> ;
HMO	:	Heurística do Melhor Objeto;
HOMA	:	Heurística da Objeto com Maior Alcance;
HOMP	:	Heurística do Objeto com Maior Participação;
HMP	:	Heurística do Melhor Par;
HAD	:	Heurística <i>Add Drop</i> ;
ILS	:	<i>Iterated local search</i> ;
PCC	:	Problema de Cobertura de Conjuntos;
PCCP	:	Problema de Cobertura de Conjunto por Pares;
PCMM	:	Problema do Conjunto Mínimo de Monitoramento.

# Sumário

<b>Lista de Figuras</b>	<b>xi</b>
<b>Lista de Tabelas</b>	<b>xii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Problema de Cobertura de Conjunto por Pares . . . . .	2
1.2 Trabalhos Relacionados . . . . .	9
1.3 Contribuições do Trabalho . . . . .	10
1.4 Organização do Trabalho . . . . .	11
<b>2 Modelos de Programação Inteira</b>	<b>12</b>
2.1 Modelo de Gusfield - $\mathcal{F}1$ . . . . .	12
2.2 Modelo de Lancia - $\mathcal{F}2$ . . . . .	14
2.3 Modelo de Gusfield Aprimorado - $\mathcal{F}1^+$ . . . . .	14
2.4 Comparação entre os Modelos $\mathcal{F}1^+$ e $\mathcal{F}2$ . . . . .	16
2.5 Resultados Computacionais . . . . .	17
2.5.1 Descrição das Instâncias . . . . .	18
2.5.2 Comparação Empírica entre Modelos $\mathcal{F}1^+$ e $\mathcal{F}2$ . . . . .	20
<b>3 Heurísticas Construtivas e de Busca Local</b>	<b>25</b>
3.1 Heurísticas Construtivas Gulosas . . . . .	25
3.1.1 Heurística do Melhor Objeto - HMO . . . . .	26
3.1.2 Heurística do Objeto com Maior Alcance - HOMA . . . . .	26

---

3.1.3	Heurística do Objeto com Maior Participação - HOMP . . . . .	27
3.1.4	Heurística do Melhor Par - HMP . . . . .	27
3.1.5	Heurística <i>Add Drop</i> - HAD . . . . .	28
3.2	Heurísticas Construtivas Gulosas Randomizadas . . . . .	29
3.2.1	HMO, HOMA e HOMP Randomizadas . . . . .	29
3.2.2	HMP Randomizada . . . . .	30
3.2.3	HAD Randomizada . . . . .	31
3.3	Busca Local . . . . .	31
3.3.1	Busca Local Troca Objeto . . . . .	31
3.3.2	Busca Local Troca Par . . . . .	32
3.3.3	Busca Local <i>Drop Add</i> . . . . .	33
3.4	Procedimentos Complementares Pré e Pós-Construção . . . . .	34
3.4.1	Fixação de Objetos . . . . .	34
3.4.2	Remoção de Objetos Redundantes . . . . .	35
3.5	Resultados Computacionais . . . . .	36
3.5.1	Heurísticas de Construção . . . . .	36
3.5.2	Heurísticas de Busca Local . . . . .	39
<b>4</b>	<b>Metaheurísticas</b> . . . . .	<b>43</b>
4.1	<i>Greedy Randomized Adaptive Search Procedure</i> - GRASP . . . . .	43
4.2	<i>Iterated Local Search</i> - ILS . . . . .	46
4.3	Heurística Híbrida . . . . .	47
4.4	Resultados Computacionais . . . . .	49
4.4.1	GRASP . . . . .	49
4.4.2	ILS . . . . .	55
4.4.3	Heurística Híbrida . . . . .	61

---

<b>5 Conclusões</b>	<b>68</b>
<b>Referências</b>	<b>70</b>
<b>Apêndice A</b>	<b>75</b>
A.1 Gerador de Instâncias . . . . .	75
A.2 Tabelas Adicionais . . . . .	76
<b>Apêndice B – Testes para calibração de parâmetros</b>	<b>88</b>
B.1 Parâmetro $p_1$ e $p_2$ da Heurística do Melhor Objeto . . . . .	88

# Lista de Figuras

1.1	Exemplo de uma instância do PCCP. . . . .	3
1.2	Uma solução para a instância do PCCP. . . . .	3
1.3	Instância $I'$ - P2CC. . . . .	4
1.4	Instância $I'$ - PCCP. . . . .	4
1.5	Cromossomos, Haplótipos e Genótipos. . . . .	5
1.6	Circuito $(m, b_1, b_2, m)$ . . . . .	8
1.7	Estrutura <i>multicast</i> . . . . .	8
3.1	Instância que apresenta elemento com único par para cobertura. . . . .	35
3.2	Instância onde elemento possui pares com objeto em comum. . . . .	35
3.3	Resultados comparativos para as instâncias do Grupo I em relação a qualidade da solução (a) e tempo de CPU (b). . . . .	41
3.4	Resultados comparativos para as instâncias do Grupo II em relação a qualidade da solução (a) e tempo de CPU (b). . . . .	42
4.1	Análise probabilística para a instância scp410_25. . . . .	54
4.2	Análise probabilística para a instância scpclr10_75. . . . .	55
4.3	Análise probabilística para a instância scp410_25. . . . .	59
4.4	Análise probabilística para a instância scpsclr10_75. . . . .	59
4.5	Comparação entre todas as abordagens para a instância scp410_25 . . . .	66
4.6	Comparação entre todas as abordagens para a instância scpsclr10_75 . . .	67

# Lista de Tabelas

2.1	Características das Instâncias do Grupos I. . . . .	18
2.2	Características das Instâncias do Grupos II. . . . .	19
2.3	Características das Instâncias do Grupo III. . . . .	20
2.4	Resultados dos modelos $\mathcal{F}1^+$ e $\mathcal{F}2$ para as instâncias do Grupo I. . . . .	22
2.5	Resultados dos modelos $\mathcal{F}1^+$ e $\mathcal{F}2$ para as instâncias do Grupo II. . . . .	24
2.6	Resultados dos modelos $\mathcal{F}1^+$ e $\mathcal{F}2$ para as instâncias do Grupo III. . . . .	24
3.1	Heurísticas Construtivas. . . . .	37
3.2	Heurísticas Construtivas - Instâncias do Grupo I. . . . .	38
3.3	Heurísticas Construtivas - Instâncias do Grupo II. . . . .	39
4.1	Versões do GPCCP. . . . .	49
4.2	Comparação entre quatro versões do GRASP para 15% das Instâncias do Grupo II. . . . .	50
4.3	Comparação entre três versões do GRASP - 420 Instâncias do Grupo II. . . . .	51
4.4	Comparação entre três versões do GRASP - 420 Instâncias do Grupo II por conjunto. . . . .	52
4.5	Comparação entre três versões do GRASP - Instâncias do Grupo III. . . . .	52
4.6	Comparação entre três versões do GRASP - Instâncias do Grupo III por conjunto. . . . .	53
4.7	Comparação entre ILS e G-DATO - Instâncias do Grupo II. . . . .	56
4.8	Comparação entre ILS e G-DATO - Instâncias do Grupo II por conjunto. . . . .	57
4.9	Comparação entre ILS e G-DATO - Instâncias do Grupo III. . . . .	57
4.10	Comparação entre ILS e G-DATO - Instâncias do Grupo III por conjunto. . . . .	58



---

4.11	Comparação entre ILS por tempo e G-DATO - Instâncias do Grupo III. . .	60
4.12	Comparação entre ILS por tempo e G-DATO - Instâncias do Grupo III por conjunto. . . . .	60
4.13	Comparação ILS com diferentes critérios de parada - Instâncias do Grupo III.	61
4.14	Comparação entre ILS, G-DATO e Híbrido - Instâncias do Grupo II. . . .	61
4.15	Comparação entre ILS, G-DATO e Híbrido - Instâncias do Grupo II por conjunto. . . . .	63
4.16	Comparação entre ILS, G-DATO e Híbrido - Instâncias do Grupo III. . . .	64
4.17	Comparação entre ILS, G-DATO e Híbrido - Instâncias do Grupo III por conjunto. . . . .	65
4.18	Comparação entre ILS por tempo, G-DATO e Híbrido - 144 Instâncias do Grupo III. . . . .	65
4.19	Comparação entre ILS por tempo, G-DATO e Híbrido - 133 Instâncias do Grupo III. . . . .	65
A.1	Média das soluções obtidas pelas versões GRASP. . . . .	77
A.2	Tempo médio de CPU, em segundos, consumido pelas versões GRASP. . .	78
A.3	ILS utilizando BLTO. . . . .	80
A.4	GRASP utilizando BLTO. . . . .	81
A.5	GRASP utilizando BLTP e BLTO. . . . .	82
A.6	Heurística Híbrida. . . . .	83
A.7	Resumo para 65 instâncias do Grupo II. . . . .	84
A.8	Comparação das heurísticas para as instâncias do Grupo III. . . . .	85
B.1	Calibrando parâmetros da heurística HMO - Grupo I. . . . .	89
B.2	Calibrando parâmetros da heurística HMO - Grupo II. . . . .	89

# Capítulo 1

## Introdução

O Problema de Cobertura de Conjuntos (PCC) é um problema bastante explorado na área de Pesquisa Operacional e Otimização [11, 12, 15, 17, 59]. Muitas aplicações práticas podem ser modeladas como PCC, como por exemplo: escalonamento de condutores, locação de unidades emergenciais, balanceamento em linha de montagem [16, 55, 58].

No modelo básico do PCC, dado o conjunto  $\mathcal{U}$  dos elementos a serem cobertos, normalmente chamado de conjunto universo, e um segundo conjunto  $\mathcal{Q}$  composto por subconjuntos de elementos de  $\mathcal{U}$ , o objetivo do PCC é cobrir todos os elementos de  $\mathcal{U}$  utilizando o menor número possível de subconjuntos pertencentes a  $\mathcal{Q}$ . Um dado elemento  $e \in \mathcal{U}$  é dito coberto por um subconjunto  $Q_i \in \mathcal{Q}$  se  $e \in Q_i$ . Uma coleção  $S \subseteq \mathcal{Q}$  é uma solução do PCC se todo elemento de  $\mathcal{U}$  é coberto por pelo menos um dos subconjuntos em  $S$ , ou seja,  $\bigcup_{Q_i \in S} Q_i = \mathcal{U}$ . Quando a cada subconjunto  $Q_i \in \mathcal{Q}$  está associado um custo  $c_i$ , o objetivo do PCC é selecionar um subconjunto  $S \subseteq \mathcal{Q}$  onde  $\bigcup_{Q_j \in S} Q_j = \mathcal{U}$  de tal forma que a soma dos custos destes objetos seja mínima, ou seja, deseja-se minimizar  $\sum_{Q_i \in S} c_i$ .

Na literatura é possível encontrar muitas variações do PCC modelando diferentes aplicações. Dentre as variações mais abordadas pode-se citar o Problema de Cobertura Parcial, onde a cobertura de parte dos elementos do conjunto  $\mathcal{U}$  pode ser negligenciada [1, 23], e o Problema da Multi-cobertura de Conjuntos [5, 14, 34, 38, 60], onde para cada elemento  $e \in \mathcal{U}$  existe um determinado número de subconjuntos  $l_e$  que devem cobri-lo. Neste último caso, quando para todos os elementos de  $\mathcal{U}$  utiliza-se um mesmo valor para  $l_e$ , ou seja,  $l_e = k$  para todo  $e \in \mathcal{U}$ , denomina-se o problema como Problema da  $k$ -Cobertura de Conjuntos (PkCC) [6, 8, 47, 48].

Neste trabalho é abordado um modelo de cobertura que difere dos mencionados anteriormente, o Problema de Cobertura de Conjunto por Pares (PCCP). Considere um

conjunto de objetos  $\mathcal{A}$ , onde a cada objeto  $a_i \in \mathcal{A}$  está associado um custo  $c_i$ , e um conjunto de elementos a serem cobertos  $\mathcal{U}$ . Nesta variante, diferente do PCC, a cobertura dos elementos de  $\mathcal{U}$  está associada a pares de objetos. Para cada elemento  $u \in \mathcal{U}$  está definido um conjunto formado por pares de objetos  $\{a_i, a_k\}$  que juntos cobrem  $u$ , onde  $a_i, a_k \in \mathcal{A}$ . O objetivo do PCCP é determinar um subconjunto de objetos  $S \subseteq \mathcal{A}$  de custo mínimo de forma que, para cada elemento  $u \in \mathcal{U}$ , exista pelo menos um par de objetos em  $S$  capaz de cobrir  $u$ . Este problema foi definido por Hassin e Segev [35] como uma generalização do Problema de Cobertura de Conjuntos, conhecidamente NP-difícil [24].

## 1.1 Problema de Cobertura de Conjunto por Pares

De acordo com Hassin e Segev [35], o Problema de Cobertura de Conjunto por Pares (PCCP) pode ser definido com o auxílio de uma função de cobertura. Considere os conjuntos  $\mathcal{U} = \{u_1, \dots, u_n\}$  e  $\mathcal{A} = \{a_1, \dots, a_m\}$ , onde  $\mathcal{U}$  representa o conjunto dos elementos a serem cobertos e  $\mathcal{A}$  o conjunto dos objetos que, aos pares, estão associados a subconjuntos de elementos de  $\mathcal{U}$ . A função de cobertura  $\mathcal{C} : \mathcal{A} \times \mathcal{A} \rightarrow 2^{\mathcal{U}}$  especifica, para cada par de objetos  $\{a_i, a_j\} \subseteq \mathcal{A}$ , o subconjunto dos elementos de  $\mathcal{U}$  que são cobertos por este par. Sendo  $c_i$  um custo não negativo associado ao objeto  $a_i \in \mathcal{A}$ , o PCCP tem por objetivo encontrar uma coleção de objetos  $S \subseteq \mathcal{A}$  de tal forma que  $\bigcup_{\{a_i, a_j\} \subseteq S} \mathcal{C}(a_i, a_j) = \mathcal{U}$  onde o custo dado por  $c(S) = \sum_{a_i \in S} c_i$  seja minimizado.

Na Figura 1.1 é possível observar uma instância do PCCP. Neste exemplo, tem-se  $\mathcal{U} = \{u_1, u_2, u_3, u_4, u_5\}$  e  $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$ , onde os arcos na figura indicam a relação de cobertura entre pares de objetos e elementos. Para cada um dos pares de objetos de  $\mathcal{A}$ , a função de cobertura é dada por:  $\mathcal{C}(a_1, a_2) = \{u_2, u_3\}$ ,  $\mathcal{C}(a_1, a_3) = \{u_1, u_3\}$ ,  $\mathcal{C}(a_1, a_4) = \emptyset$ ,  $\mathcal{C}(a_2, a_3) = \{u_1, u_2\}$ ,  $\mathcal{C}(a_2, a_4) = \{u_1, u_3, u_4, u_5\}$ ,  $\mathcal{C}(a_3, a_4) = \{u_3, u_4, u_5\}$ . Assim, como apresentado na Figura 1.2, uma possível solução é  $S = \{a_2, a_3\} \cup \{a_3, a_4\} = \{a_2, a_3, a_4\}$ , visto que  $\mathcal{C}(a_2, a_3) \cup \mathcal{C}(a_3, a_4) = \mathcal{U}$  onde o custo da solução  $S$  é dado por  $c(S) = c_2 + c_3 + c_4$ .

A partir desta definição é possível verificar que o Problema de Cobertura de Conjuntos (PCC) é um caso específico do PCCP, como apresentado em [35]. Para se obter o PCC a partir do PCCP, basta associar cada objeto a um subconjunto dos elementos de  $\mathcal{U}$ . Considerando que  $\mathcal{C}' : \mathcal{A} \rightarrow 2^{\mathcal{U}}$  representa os elementos de  $\mathcal{U}$  pertencentes ao subconjunto associado a um elemento de  $\mathcal{A}$ , basta definir a função de cobertura como  $\mathcal{C}(i, j) = \mathcal{C}'(i) \cup \mathcal{C}'(j)$ . Desta forma, os resultados quanto a complexidade, conhecidos para o PCC, se estendem para o PCCP.

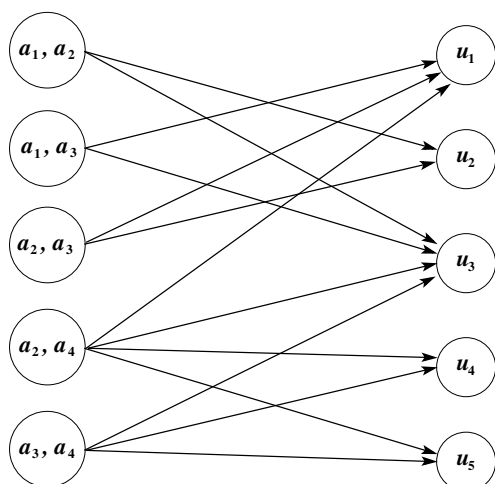


Figura 1.1: Exemplo de uma instância do PCCP.

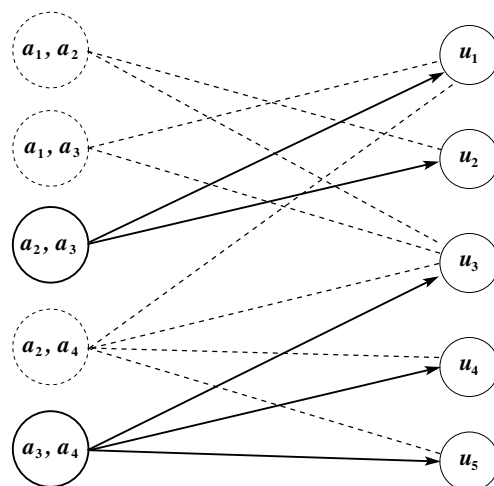
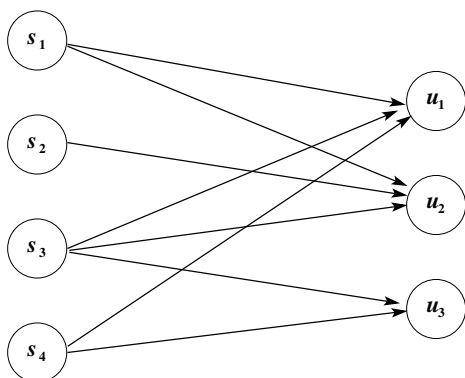
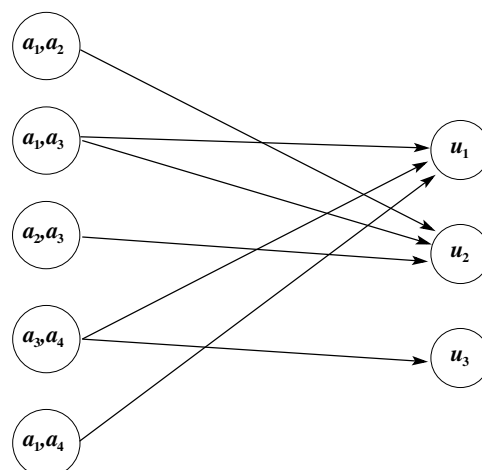


Figura 1.2: Uma solução para a instância do PCCP,  $S = \{a_2, a_3, a_4\}$ .

Outro problema que pode ser tratado como o PCCP é o Problema da  $k$ -Cobertura de Conjuntos para  $k = 2$  (P2CC). Para o P2CC considere o conjunto  $\mathcal{U}$ , composto pelos elementos a serem cobertos, e  $\mathcal{Q}$  como um conjunto composto por subconjuntos de elementos de  $\mathcal{U}$ . Uma solução do P2CC é definida por um conjunto  $S \subseteq \mathcal{Q}$  onde cada elemento  $u_i \in \mathcal{U}$  está contido em pelo menos dois subconjuntos  $s \in S$ . Para que o P2CC seja tratado como o Problema de Cobertura de Conjunto por Pares, cada subconjunto  $s \in \mathcal{Q}$  é associado a um objeto  $a \in \mathcal{A}$ . Para o conjunto  $\mathcal{A}$  definido desta forma, a função de cobertura deve ser estabelecida como  $\mathcal{C}(a_i, a_j) = a_i \cap a_j$  para todo  $i \neq j$ .

Na Figura 1.3 é apresentada uma instância do P2CC, denominada instância  $I'$ , onde  $\mathcal{U} = \{u_1, u_2, u_3\}$  e  $\mathcal{Q} = \{s_1, s_2, s_3, s_4\}$  sendo  $s_1 = \{u_1, u_2\}$ ,  $s_2 = \{u_2\}$ ,  $s_3 = \{u_1, u_2, u_3\}$  e  $s_4 = \{u_1, u_3\}$ . Na figura, um arco de  $s_i$  para  $u_k$  representa que o subconjunto  $s_i$  contém o elemento  $u_k$ . Dado o conjunto  $\mathcal{A}$  e a função de cobertura como definida no parágrafo anterior, tem-se que esta instância pode ser apresentada como uma instância do PCCP, como pode ser observado na Figura 1.4. Considerando que  $s_i \in \mathcal{Q}$  corresponde a  $a_i \in \mathcal{A}$ , ou seja  $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$ , tem-se que  $\mathcal{C}(a_1, a_2) = \{u_2\}$ ,  $\mathcal{C}(a_1, a_3) = \{u_1, u_2\}$ ,  $\mathcal{C}(a_1, a_4) = \{u_1\}$ ,  $\mathcal{C}(a_2, a_3) = \{u_2\}$  e  $\mathcal{C}(a_3, a_4) = \{u_1, u_3\}$ . Para os demais pares  $\{a_i, a_j\} \subset \mathcal{A}$ ,  $\mathcal{C}(a_i, a_j) = \emptyset$ . Desta forma, uma possível solução  $S' = \{s_2, s_3, s_4\}$  para o P2CC e equivalente a solução  $S = \{\{a_2, a_3\} \cup \{a_3, a_4\}\} = \{a_2, a_3, a_4\}$  do PCCP.

O Problema de Cobertura Mínima de Sub-strings - PCMS [36] (*Minimum Substring Cover Problem*) é outro problema onde um caso particular pode ser visto como o PCCP. O PCMS pode ser descrito da seguinte forma: Dado um conjunto de strings  $\mathcal{U}$  e um conjunto  $\mathcal{Z}$  composto por sub-strings de  $\mathcal{U}$ , um conjunto  $\mathcal{Z}' \subset \mathcal{Z}$  cobre  $\mathcal{U}$  se toda string em  $\mathcal{U}$  puder

Figura 1.3: Instância  $I'$  - P2CC.Figura 1.4: Instância  $I'$  - PCCP.

ser escrita como a concatenação de sub-strings em  $\mathcal{Z}'$ . Dado um inteiro  $l$ , uma  $l$ -cobertura de  $\mathcal{U}$  ( $l$ -PCMS) é um conjunto  $\mathcal{Z}'' \subset \mathcal{Z}$ , onde para cada  $s \in \mathcal{U}$  existem  $p$  sub-strings em  $\mathcal{Z}''$  que concatenadas resultam em  $s$ , sendo  $p \leq l$ . Por exemplo, para  $\mathcal{U} = \{ 'a', 'aab', 'aba' \}$  e  $\mathcal{Z} = \{ 'a', 'b', 'aa', 'ab', 'ba', 'aab', 'aba' \}$ , o conjunto  $\mathcal{Z}_1'' = \{ 'a', 'b' \}$  é uma 3-cobertura de  $\mathcal{U}$  enquanto  $\mathcal{Z}_2'' = \{ 'a', 'ab' \}$  é uma 2-cobertura. Assim, o  $l$ -PCMS para  $l = 2$  pode ser descrito como o PCCP. Hermelin et al. [36] destacam que este problema merece particular consideração por tratar identificação de prefixos e sufixos.

Duas aplicações para o Problema de Cobertura de Conjuntos por Pares merecem destaque: O Problema de Inferência de Haplótipos, no contexto de biologia computacional, e o Problema de Conjunto Mínimo de Monitoramento, que motivou o desenvolvimento desta tese.

- **Problema de Inferência de Haplótipos por Parcimônia Pura**  
(*Haplotype Inference by Pure Parsimony*)

O organismo humano é diplóide, isto é, seu DNA é organizado em pares de cromossomos, onde cada cromossomo é composto por uma sequência de nucleotídeos. O mapeamento do genoma humano mostrou que qualquer duas cópias de DNA diferem uma da outra em aproximadamente 1% de nucleotídeos [29, 61].

O polimorfismo de um único nucleotídeo (*Single Nucleotide Polymorphism* - SNP) é uma posição específica de um nucleotídeo cujo conteúdo mostra uma variabilidade estatisticamente significativa dentro de uma população. Uma posição é considerada um SNP se, para uma pequena parte da população (aproximadamente 5%), um certo nucleotídeo

é observado (alelo menos frequente) enquanto para o restante da população, outro nucleotídeo é verificado (alelo mais frequente). Para diferenciar estes dois alelos, 0 (zero) será utilizado para indicar que o alelo menos frequente foi observado em uma dada posição e 1 (um) para indicar a presença do alelo mais frequente.

Denomina-se *haplótipo* o conjunto de SNPs de um dos cromossomos de um indivíduo, representado por um vetor de SNPs composto por zeros e uns. Os *haplótipos* são importantes como preditores de várias doenças. Entretanto, identificar o par de *haplótipos* de um indivíduo é uma tarefa cara e difícil. Normalmente o que se tem de cada indivíduo é o seu *genótipo*.

O *genótipo* de um dado indivíduo está associado às informações sobre a combinação dos SNPs dos dois alelos do indivíduo. Dados os dois *haplótipos* de um indivíduo, em uma determinada posição de um SNP, pode-se observar o alelo mais frequente em um *haplótipo* e o alelo menos frequente no outro *haplótipo* (*heterozigoto*) ou então, o mesmo alelo nos dois *haplótipos* (*homozigoto*). Para cada posição do *genótipo* tem-se três possíveis valores para representar a combinação dos *haplótipos* 0, 1 ou 2. O número 0 representa que o alelo menos frequente foi observado nos dois *haplótipos*, o número 1 representa que o alelo mais frequente foi observado em ambos *haplótipos* e, finalmente, o número 2 indica que foram observados alelos diferentes na posição do SNP. Dado um conjunto de *genótipos*, métodos computacionais têm sido utilizados para deduzir os *haplótipos* associados a esta população (*Inferência de Haplótipos*).

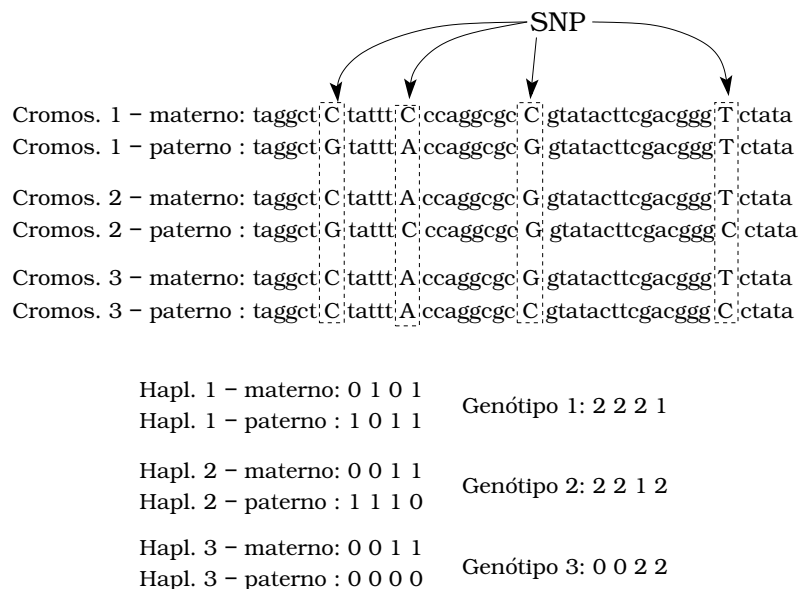


Figura 1.5: Cromossomos, Haplótipos e Genótipos.

Na Figura 1.5, extraída do trabalho de Lancia et al., 2009 [41], é possível observar o fragmento de uma dada região do DNA de três diferentes indivíduos, formado por três pares de cromossomos. Neste exemplo tem-se quatro posições SNP, onde em cada SNP dois nucleotídeos podem ser observados. Desta forma, os haplótipos são representados por um vetor 0, 1 de quatro posições onde cada posição indica se o alelo mais frequente ou se o menos frequente foi observado. Como o genótipo traz informação referente ao par de haplótipos, na figura é apresentado o genótipo associado a cada par de haplótipos dos três indivíduos.

Diz-se que um par de haplótipos  $h'$  e  $h''$  resolve um genótipo  $g$  composto por  $n$  SNPs, onde  $i = \{1, \dots, n\}$ , quando  $g_i = h'_i \oplus h''_i$  para cada um dos SNPs de  $g$ . Representando  $h_i$  como o alelo observado no  $i$ -ésimo SNP de  $h$  e  $g_i$  a informação associada ao SNP da posição  $i$  do genótipo  $g$ , a função abaixo apresenta como é realizado o cômputo de  $h'_i \oplus h''_i$ .

$$h'_i \oplus h''_i = \begin{cases} 0, & \text{se } h'_i = 0 \text{ e } h''_i = 0 \\ 1, & \text{se } h'_i = 1 \text{ e } h''_i = 1 \\ 2, & \text{se } h'_i \neq h''_i \end{cases}$$

Para um genótipo onde  $k$  SNPs são heterozigotos, ou seja,  $g_i = 2$  em  $k$  posições, existem  $2^{k-1}$  pares de haplótipos que o resolvem. Por exemplo, para o Genótipo 3 da Figura 1.5,  $g = \langle 0, 0, 2, 2 \rangle$ , onde  $k = 2$ , tem-se  $2^{2-1} = 2$  possíveis pares de haplótipos que o resolvem. Além do par de haplótipos apresentado da figura,  $(\langle 0, 0, 0, 0 \rangle, \langle 0, 0, 1, 1 \rangle)$ , outro possível par que o resolve seria  $(\langle 0, 0, 0, 1 \rangle, \langle 0, 0, 1, 0 \rangle)$ .

Assim, para uma população com  $n$  indivíduos, partindo de um conjunto de  $n$  genótipos composto por  $m$  posições associadas às SNPs, o Problema da Inferência de Haplótipos (PIH) busca determinar um conjunto de haplótipos que, aos pares, resolvem todos os genótipos.

O Problema de Inferência de Haplótipos de uma população, como definido em [40], pode ser descrito da seguinte forma: Dado um conjunto de genótipos  $G$  com  $n$  elementos, determinar um conjunto de haplótipos  $H$  tal que cada genótipo  $g \in G$  é resolvido por dois haplótipos  $h', h'' \in H$ .

Entre os diferentes objetivos para este problema tem-se a Parcimônia Pura, onde no Problema de Inferência de Haplótipos por Parcimônia Pura (PIHPP) busca-se determinar qual é o menor conjunto de haplótipos que, combinados em pares, resolvem o conjunto de genótipos observado. Ou seja, o objetivo é que  $H$  tenha a menor cardinalidade possível.

Para tratar o PIHPP como o Problema de Cobertura de Conjuntos por Pares, deve-

se associar o conjunto de genótipos  $G$  ao conjunto  $\mathcal{U}$  e, além disso, associar o conjunto de Haplótipos  $H$  ao conjunto de objetos  $\mathcal{A}$ . Assim, conhecendo-se, para cada genótipo  $g$ , os pares de haplótipos que o resolve, a função de cobertura pode ser definida como  $\mathcal{C}(h', h'') = h' \oplus h''$ .

Na Seção 2 é apresentado um modelo matemático, descrito em [40], que mostra o PIHPP como um problema de programação linear inteira.

### • Problema do Conjunto Mínimo de Monitoramento

O Problema do Conjunto Mínimo de Monitoramento (PCMM) tem sua origem no problema de monitoramento remoto em redes de comunicação. Esta aplicação envolve técnicas específicas para monitorar caminhos de uma rede utilizando para isto um conjunto de *hosts* de monitoramento.

Dado o interesse de monitorar o desempenho de caminhos em Redes Privadas Virtuais (VPN), mais especificamente estimando acessibilidade, perda de pacotes e atraso num caminho, em [9] é apresentada uma técnica para realização destas medições.

Considerando as limitações das tecnologias atuais, como *traceroute* ou *ping*, que realizam medições de circuitos ao invés de caminhos, e dada a possibilidade dos pacotes de sondagem (*probe packets*) não seguirem exatamente o caminho que se deseja analisar, em [31] é apresentada uma estratégia baseada no uso de túneis e topologia *multicast* para tratar o problema. Por esta estratégia, pacotes de sondagem gerados com o propósito de monitoramento são direcionados ao longo do caminho do tráfego do cliente usando túneis.

Para melhor ilustrar o problema, considere a Figura 1.6 onde os roteadores  $b_1$  e  $b_2$ , localizados nos extremos dos caminhos desta rede (*edge routers*), fornecem ao cliente acesso a uma rede VPN. Estes roteadores são equipados com transmissores e receptores de pacotes de sondagem que seguem o caminho do tráfego do cliente.

Nesta abordagem, túneis são construídos entre os nós  $b_1$  e  $b_2$  e o nó de monitoramento  $m$ . Com isso, os pacotes de sondagem percorrem o circuito  $(m, b_1, b_2, m)$  seguindo o tráfego do cliente entre os nós  $b_1$  e  $b_2$ . Contudo, o desempenho verificado é uma composição do desempenho de três sub-caminhos  $(m, b_1)$ ,  $(b_1, b_2)$  e  $(b_2, m)$ . Desta forma, é necessária a fatoração da parcela referente aos sub-caminhos  $(m, b_1)$  e  $(b_2, m)$  para, efetivamente, se obter o desempenho associado ao segmento  $(b_1, b_2)$ .

Para contornar estes obstáculos, Yu Gu et al. [30, 31] apresentam uma abordagem baseada na tomografia da rede para inferir o desempenho destes sub-caminhos. A ideia



é sobrepor uma rede *multicast* de medição. Nesta abordagem são necessários dois nós de monitoramento  $m_1$  e  $m_2$  para cada nó *branch*. Na topologia virtual *multicast* de medição, como pode ser visto na Figura 1.7, dois túneis precisam ser configurados entre os nós  $m_1$  e  $b$  (chamados  $m_{1,1}$  e  $m_{1,2}$ ) e um outro túnel deve conectar  $m_2$  e  $b$ . Assim, quando um pacote de sondagem é enviado de  $m_1$  para  $b$  através do túnel  $m_{1,1}$ , um roteamento *multicast* em  $b$  enviará uma cópia para  $m_1$  e  $m_2$ .

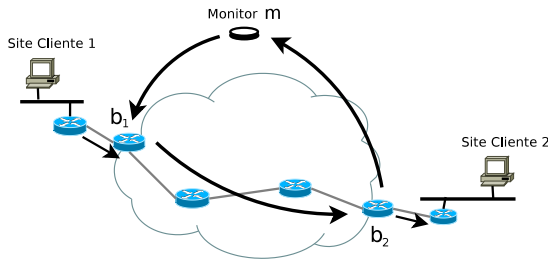


Figura 1.6: Circuito  $(m, b_1, b_2, m)$ .

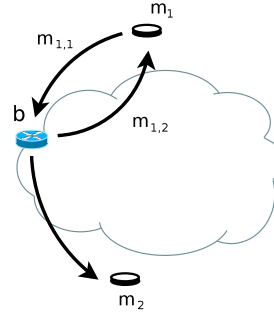


Figura 1.7: Estrutura *multicast*.

Para este tipo de análise é fundamental que as medidas sejam independentes entre os diferentes *links*. Se os caminhos  $(b, m_1)$  e  $(b, m_2)$  não são disjuntos, o nó  $b$  não será um ponto de ramificação na estrutura *multicast* proposta, causando desvio nas medição realizada. Assim, uma condição necessária para esta abordagem é a disjunção entre os caminhos físicos  $(b, m_1)$  e  $(b, m_2)$ .

Como o objetivo é realizar a medição para cada roteador de borda, o Problema do Conjunto Mínimo de Monitoramento pode ser descrito como definir um conjunto mínimo de nós de monitoramento de tal forma que, para cada nó *branch*  $b$ , haverá pelo menos dois nós de monitoramento  $m_1$  e  $m_2$  onde os caminhos  $(b, m_1)$  e  $(b, m_2)$  sejam disjuntos.

Associando esta aplicação ao Problema de Cobertura de Conjuntos por Pares, o conjunto de *branches* corresponde ao conjunto universo  $\mathcal{U}$  e o conjunto de candidatos a nós de monitoramento corresponde ao conjunto de objetos  $\mathcal{A}$ . Assim, a função de cobertura  $\mathcal{C}(m_1, m_2)$ , definida em função de pares de candidatos a nós de monitoramento, indicará o conjunto de *branches* onde os caminhos que conectam estes *branches* aos monitores  $m_1$  e  $m_2$  são disjuntos.

No PCMM, um mesmo custo é associado a cada nó candidato a monitor. Assim, o objetivo deste problema é determinar o conjunto mínimo de monitores que possibilitem o monitoramento de todos os *branches*. Neste trabalho também é abordado o modelo onde diferentes custos são associados aos monitores. Desta forma, o objetivo do problema é

encontrar o subconjunto de nós candidatos a nós de monitoramento, capazes de cobrir todos os *branches*, cuja a soma do custo destes nós seja mínima.

Uma outra questão que deve ser considerada é a interseção entre os conjuntos de *branches*  $B$  e de monitores  $M$ . Ou seja, dentre os *branches* a serem monitorados podem existir alguns que tenham a capacidade de funcionar como nós de monitoramento. Para este subconjunto de nós, o monitoramento pode ser realizado de duas formas: ou seleciona-se um par de outros nós para monitorá-lo, ou o próprio nó é suficiente para seu monitoramento.

Para resolver o problema da interseção entre os conjuntos, de forma que o PCMM possa ser tratado como o PCCP, Breslau et al. [9] sugere um artifício para manipular os nós da interseção. Para cada roteador  $k \in B \cap M$  são inseridos dois nós especiais  $z_1$  e  $z_2$  no conjunto  $M$ , com custo  $c_{z_1} = c_{z_2} = 0$ , e  $x_0$  no conjunto  $B$ . O conjunto  $P_{x_0} = \{\{z_1, z_2\}\}$ , onde  $P_{x_0}$  representa o conjunto dos possíveis pares de monitores que podem monitorar  $x_0$ , e o par  $\{k, z_1\}$  (ou o par  $\{k, z_2\}$ ) é incluído no conjunto  $P_k$ . Para cobrir  $x_0$ , obrigatoriamente os elementos do par  $\{z_1, z_2\}$  devem fazer parte da solução. Desta forma, se  $k$  for inserido na solução, o *branch*  $k$  estará coberto já que os nós do par  $\{k, z_1\}$  (ou do par  $\{k, z_2\}$ ) estarão na solução.

## 1.2 Trabalhos Relacionados

O Problema de Cobertura de Conjuntos por Pares (PCCP) foi proposto em 2005 por Hassin e Segev [35]. Neste trabalho os autores apresentam um algoritmo guloso para o PCCP que consiste em uma heurística iterativa onde, a cada etapa, um nó ou um par de nós é inserido na solução. O algoritmo seleciona o elemento ou o par de elementos mais eficiente, onde a avaliação considera a razão entre o custo de inserção e a quantidade de novos elementos cobertos. Esta abordagem gera resultados de boa qualidade, mas apresenta elevado custo computacional devido à necessidade de análise de todos os possíveis pares, além de cada elemento individualmente. Neste trabalho os autores utilizam um modelo de programação linear para descrever o PCCP, modelo este utilizado anteriormente para descrever o Problema de Inferência de Haplótipos por Parcimônia Pura [40].

O primeiro modelo matemático para o Problema de Inferência de Haplótipos por Parcimônia Pura (PIHPP) foi apresentado por Gusfield [32, 33]. Nestes trabalhos, o modelo é apresentado na forma de um exemplo, sendo apresentado em sua forma cartesiana por Brown e Harrower [10]. Este modelo, que possui um número exponencial de restrições, pode ser utilizado para descrever o Problema de Cobertura de Conjunto por Pares.

Outros modelos para o PIHPP que exploram características de cada SNP foram desenvolvidos [7, 10, 40]. Em [40] é possível verificar que o PCCP além de NP-difícil é também APX-hard. No trabalho de Lancia e Serafini [41] é apresentado um algoritmo de *branch-and-bound* com geração de linhas e colunas para o PIHPP (*Branch-and-cut-and-price*). Esta abordagem tem como base um modelo de cobertura de conjuntos, um modelo relaxado para o Problema de Inferência de Haplótipos. Vários trabalhos na literatura abordam o PIHPP [13].

No trabalho de Breslau et al. [9] são apresentadas algumas abordagens para o Problema do Conjunto Mínimo de Monitoramento (PCMM). O mesmo modelo matemático utilizado para descrever o PCCP [40] foi utilizado para descrever o PCMM. Além do modelo, Breslau et al. [9] descrevem uma heurística construtiva para o PCMM, tratando-se basicamente de uma simplificação do algoritmo apresentado no trabalho de Hassin e Segev [35]. Um algoritmo genético, que emprega a técnica de *random keys* [3, 46] para representação das soluções e uma heurística que explora características das instâncias também são propostas pelos autores para determinar os objetos que farão parte da solução.

### 1.3 Contribuições do Trabalho

O propósito inicial deste trabalho foi desenvolver abordagens capazes de determinar boas soluções para o Problema do Conjunto Mínimo de Monitoramento, uma aplicação na área de redes de telecomunicações que pode ser modelada como Problema de Cobertura de Conjuntos por Pares. Durante o desenvolvimento deste trabalho outros problemas que podem ser modelados como o PCCP foram identificados.

Para o Problema de Cobertura de Conjunto por Pares cinco heurísticas de construção foram desenvolvidas e três algoritmos de busca local. Além disto, neste trabalho são apresentadas abordagens exatas, heurísticas baseadas em GRASP e ILS e também uma heurística híbrida que podem ser aplicadas a qualquer problema que possa ser modelado como o PCCP. Parte dos resultados apresentados neste trabalho estão publicados em [26, 27, 28].

## 1.4 Organização do Trabalho

No Capítulo 2 são apresentados dois modelos de programação inteira da literatura para o Problema de Cobertura de Conjuntos por Pares. Além destes, um terceiro modelo, que pode ser visto como um aprimoramento de um dos modelos da literatura, é proposto. Uma comparação entre o modelo da literatura e o modelo proposto é apresentada. Ao final deste capítulo são apresentados experimentos computacionais que confirmam os resultados comparativos entre os modelos. Três grupos de instâncias são utilizados nos testes. O primeiro grupo trata-se de um conjunto de instâncias da literatura proposto para o Problema do Conjunto Mínimo de Monitoramento. Um segundo grupo de instâncias, derivadas das instâncias do primeiro grupo, são apresentadas. Por fim, um terceiro grupo com instâncias derivadas de instâncias da literatura para o Problema de Cobertura de Conjuntos são propostas.

Heurísticas de construção e busca local são apresentadas no Capítulo 3. Inicialmente são apresentadas as versões gulosas das heurísticas de construção e, em seguida, versões randomizadas destas heurísticas são sugeridas. São descritas cinco heurísticas de construção, sendo que em quatro destas a solução é obtida de forma incremental, onde a cada iteração objetos são incluídos na solução. Uma quinta abordagem funciona de forma inversa, onde parte-se de uma solução com redundância e, em seguida, removem-se objetos da solução com objetivo de diminuir seu custo mantendo a viabilidade. Além das heurísticas de construção, três algoritmos de busca local são apresentados. Testes comparativos entre estas abordagens são apresentados no final do capítulo.

No Capítulo 4 são apresentadas algumas abordagens que fazem uso das heurísticas apresentadas no Capítulo 3. Inicialmente é proposto um algoritmo baseado na metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP). Além deste, uma solução utilizando a metaheurística *Iterated local search* (ILS) é apresentada para o PCCP. Uma abordagem híbrida que combina as heurísticas de construção e o modelo de programação linear proposto neste trabalho é também apresentado neste capítulo.

Por fim, no Capítulo 5, são apresentadas as conclusões deste trabalho e algumas sugestões para continuidade desta pesquisa.

# Capítulo 2

## Modelos de Programação Inteira

O primeiro modelo de programação inteira que pode ser utilizado para descrever o Problema de Cobertura de Conjuntos por Pares foi apresentado por Gusfield [32] através de um exemplo. Este modelo, aqui denominado  $\mathcal{F}1$ , foi formalmente descrito no trabalho de Brown e Harrower [10].

Em Hassin e Segev [35], onde é proposto o PCCP, um outro modelo de programação inteira é apresentado. Este mesmo modelo foi utilizado por Breslau et al. [9] para descrever o Problema de Conjunto Mínimo de Monitoramento. Como no trabalho de Lancia et al. [40] este já havia sido utilizado para descrever o Problema de Inferência de Haplótipos, neste trabalho o modelo será denominado Modelo de Lancia ou simplesmente  $\mathcal{F}2$ .

Neste capítulo é proposto um terceiro modelo, obtido a partir do Modelo de Gusfield. Neste modelo, aqui denominado Modelo de Gusfield Aprimorado ou  $\mathcal{F}1^+$ , as restrições do modelo original são combinadas de forma a obter restrições mais fortes para o PCCP. Verificando a literatura é possível notar que este modelo pode ser visto também como uma simplificação do modelo apresentado por Bertolazzi et al. [7] para o Problema de Inferência de Haplótipos. Adicionalmente é proposta uma desigualdade válida que pode ser incorporada aos modelos aqui analisados.

### 2.1 Modelo de Gusfield - $\mathcal{F}1$

Nesta seção será apresentado o modelo proposto por Gusfield [32] da forma como descrito no trabalho de Brown e Harrower [10]. O Modelo de Gusfield ( $\mathcal{F}1$ ) foi proposto para tratar o Problema da Inferência de Haplótipos, mas pode ser utilizado para modelar o PCCP.

Como definido no capítulo anterior, considere os conjuntos  $\mathcal{U}$  e  $\mathcal{A}$ , onde  $\mathcal{U}$  representa

o conjunto dos elementos a serem cobertos e  $\mathcal{A}$  o conjunto dos objetos que aos pares estão associados a subconjuntos de elementos de  $\mathcal{U}$ , onde a cada  $j \in \mathcal{A}$  existe um custo associado  $c_j \geq 0$ . A função de cobertura  $\mathcal{C}(j, k)$  especifica, para cada par de objetos  $\{j, k\} \subseteq \mathcal{A}$ , um subconjunto dos elementos de  $\mathcal{U}$  que são cobertos por este par.

Para cada elemento  $e \in \mathcal{U}$ , define-se  $P_e$  como um conjunto composto por pares  $\{j, k\}$  onde  $j, k \in \mathcal{A}$  e  $\mathcal{C}(j, k) \supset \{e\}$ , ou seja,  $P_e$  é o conjunto composto pelos pares de objetos que cobrem  $e$ . Considerando  $w_{ijk}$  uma variável binária que assume valor 1 se o par  $\{j, k\} \in P_i$  foi escolhido para cobrir o elemento  $i$  e 0 caso contrário, e ainda  $x_j$  uma variável binária que indica, para cada elemento  $j \in \mathcal{A}$ , se este foi selecionado para participar da solução ( $x_j = 1$ ), ou não ( $x_j = 0$ ), o modelo  $\mathcal{F}1$  pode ser escrito como segue.

$$\text{Minimizar } \sum_{j \in \mathcal{A}} c_j x_j, \quad \text{Sujeito a:} \quad (2.1)$$

$$\sum_{\{j, k\} \in P_i} w_{ijk} = 1, \quad \forall i \in \mathcal{U}; \quad (2.2)$$

$$w_{ijk} \leq x_j, \quad \forall i \in \mathcal{U} \text{ e } \forall j, k \in P_i; \quad (2.3)$$

$$w_{ijk} \leq x_k, \quad \forall i \in \mathcal{U} \text{ e } \forall j, k \in P_i; \quad (2.4)$$

$$x_j \in \{0, 1\}, \quad \forall j \in \mathcal{A}; \quad (2.5)$$

$$w_{ijk} \in \{0, 1\}, \quad \forall i \in \mathcal{U} \text{ e } \forall \{j, k\} \in P_i. \quad (2.6)$$

Como definido na Função (2.1), o objetivo é selecionar um subconjunto de objetos de custo mínimo tal que todos os elementos de  $\mathcal{U}$  sejam cobertos. Considerando-se  $c_i = 1$ , como no Problema da Inferência de Haplótipos, o objetivo torna-se determinar o conjunto de tamanho mínimo de objetos que cobrem todos os elementos.

O conjunto de restrições (2.2) indica que para cada elemento a ser coberto um par de objetos será selecionado. As restrições (2.3) e (2.4) garantem que a seleção de um par de objetos só é possível se ambos os objetos estiverem na solução. As restrições (2.5) e (2.6) indicam o domínio das variáveis do modelo.

As variáveis  $w_{ijk}$  são necessárias para se ter um modelo de programação inteira. Este modelo poderia ser escrito apenas em função da variável  $x_j$ . Para isto as restrições (2.2), (2.3) e (2.4) poderiam ser substituídas por  $\sum_{\{j, k\} \in P_i} x_j x_k \geq 1$ , para todo  $i \in \mathcal{U}$ .

## 2.2 Modelo de Lancia - $\mathcal{F}2$

No trabalho de Hassin e Segev [35] é apresentado um modelo que descreve o Problema de Cobertura de Conjuntos por Pares. Este mesmo modelo foi utilizado anteriormente por Lancia et al. [40] para tratar o Problema de Inferência de Haplótipos, neste trabalho o modelo será denominado Modelo de Lancia ou simplesmente  $\mathcal{F}2$ .

Considere os conjuntos  $\mathcal{U}$ ,  $\mathcal{A}$  e  $P_e$  como definidos na seção anterior, e também a variável binária  $x_i$  indicando se o objeto  $i \in \mathcal{A}$  foi selecionado para fazer parte da solução. No Modelo  $\mathcal{F}2$  será utilizada uma variável binária  $y_{jk}$ , onde  $\{j, k\} \subseteq \mathcal{A}$ , que assumirá valor 1 se o par  $\{j, k\}$  for selecionado para cobrir algum dos elementos de  $\mathcal{U}$ , e 0 caso contrário. Utilizando as variáveis  $x_i$  e  $y_{jk}$ , o Modelo de Lancia pode ser descrito da seguinte forma:

$$\text{Minimizar } \sum_{j \in \mathcal{A}} c_j x_j, \quad \text{Sujeito a:} \quad (2.7)$$

$$y_{jk} \leq x_j, \quad \forall j \in \mathcal{A} \text{ e } \forall k \in \mathcal{A}; \quad (2.8)$$

$$y_{jk} \leq x_k, \quad \forall j \in \mathcal{A} \text{ e } \forall k \in \mathcal{A}; \quad (2.9)$$

$$\sum_{\{j,k\} \in P_i} y_{jk} \geq 1, \quad \forall i \in \mathcal{U}; \quad (2.10)$$

$$x_i \in \{0, 1\}, \quad \forall i \in \mathcal{A}; \quad (2.11)$$

$$y_{jk} \in \{0, 1\}, \quad \forall \{j, k\} \in \mathcal{A}. \quad (2.12)$$

A função objetivo deste modelo é idêntica a função do modelo apresentado na seção anterior. As restrições (2.8) e (2.9) asseguram que as variáveis  $y_{jk}$  somente poderão ser positivas se as variáveis  $x_j$  e  $x_k$  forem iguais a 1. Assim, somente será possível utilizar um par de objetos se ambos estiverem na solução. As restrições (2.10) estão associadas à cobertura dos elementos  $i \in \mathcal{U}$ . Para estes elementos, a cobertura se dá pela seleção de pelo menos um dos pares do conjunto  $P_i$ , que representa os pares que podem cobrir um elemento  $i$ . As restrições (2.11) e (2.12) indicam o domínio das variáveis envolvidas na modelagem.

## 2.3 Modelo de Gusfield Aprimorado - $\mathcal{F}1^+$

Nesta seção é proposta uma formulação para o PCCP que faz uso de variáveis estendidas, apresentadas na Seção 2.2. Considerando  $w_{ijk}$  uma variável binária que assume valor 1 se

o par  $\{j, k\} \in P_i$  foi escolhido para cobrir o elemento  $i$  e 0 caso contrário, e dado o conjunto de objetos  $T_i \subseteq \mathcal{A}$  que compõem os pares contidos em  $P_i$ , ou seja,  $T_i = \bigcup_{\{j,k\} \in P_i} \{j, k\}$ , o Modelo de Gusfield Aprimorado, ou simplesmente  $\mathcal{F}1^+$ , pode ser expresso com segue.

$$\text{minimizar } \sum_{i \in \mathcal{A}} c_i x_i \quad \text{Sujeito a:} \quad (2.13)$$

$$\sum_{\{j,k\} \in P_i} w_{ijk} = 1, \quad \forall i \in \mathcal{U} \quad (2.14)$$

$$\sum_{k \in T_i | j \neq k} w_{ijk} \leq x_j, \quad \forall i \in \mathcal{U}, \forall j \in T_i \quad (2.15)$$

$$x_i \in \{0, 1\}, \quad \forall i \in \mathcal{U} \quad (2.16)$$

$$w_{ijk} \in \{0, 1\}, \quad \forall i \in \mathcal{U}, \forall \{j, k\} \subset P_i \quad (2.17)$$

A função objetivo (2.13) é a mesma dos modelos anteriores. As restrições (2.14) asseguram que cada elemento  $i \in \mathcal{U}$  deve ser coberto por exatamente um par  $\{j, k\} \in P_i$ . As restrições (2.15) asseguram que, caso o objeto  $j$  seja selecionado para cobrir um elemento  $i$  formando par com qualquer outro objeto em  $T_i$ , então  $x_j$  necessariamente estará incluído na solução. O conjunto de restrições (2.16) e (2.17) definem o domínio das variáveis.

No modelo  $\mathcal{F}1^+$  as restrições (2.15) foram obtidas a partir da combinação das restrições (2.3) e (2.4) do Modelo de Gusfield ( $\mathcal{F}1$ ), com objetivo de reforçar a formulação. Como o lado direito das restrições permaneceram constantes, é fácil notar que o somatório introduzido nas restrições (2.15) deixa a formulação  $\mathcal{F}1^+$  mais forte que o modelo  $\mathcal{F}1$ .

Neste trabalho é proposta também uma desigualdade válida que pode ser incorporada aos modelos analisados. Uma desigualdade válida é uma restrição que é atendida por todas as soluções inteiras de um modelo. Uma desigualdade válida pode ser inserida com o objetivo de aumentar o limite inferior durante a resolução dos modelos de programação inteira de minimização. Deve-se destacar que as desigualdades válidas (2.18) podem ser adicionadas tanto aos modelos  $\mathcal{F}1$  e  $\mathcal{F}1^+$  quanto em  $\mathcal{F}2$ . Essas desigualdades obrigam que, para cada elemento  $i \in \mathcal{U}$ , devem haver pelo menos dois objetos  $j \in T_i$  incluídos na solução.

$$\sum_{j \in T_i} x_j \geq 2, \quad i \in \mathcal{U}. \quad (2.18)$$



## 2.4 Comparação entre os Modelos $\mathcal{F}1^+$ e $\mathcal{F}2$

Descrever um problema utilizando um modelo de Programação Linear Inteira Mista (PLIM) permite, em princípio, que estes problemas sejam resolvidos de maneira exata através da utilização de pacotes de *software* resolvidores de PLIM, como CPLEX [39], por exemplo. O desempenho dos resolvidores de PLIM, que utilizam métodos baseados em *branch-and-cut*, dependem fortemente da formulação utilizada. Com objetivo de verificar qual das restrições é mais forte, nesta seção é realizada uma comparação entre o Modelo de Gusfield Aprimorado  $\mathcal{F}1^+$  e o Modelo de Lancia  $\mathcal{F}2$ .

**Proposição 1** *O valor da relaxação linear de  $\mathcal{F}1^+$  com as restrições (2.18) é mais forte que o valor obtido da relaxação linear de  $\mathcal{F}2$  incluindo (2.18).*

**Prova 1** *Uma vez que as variáveis  $x$  são comuns a ambas as formulações, é necessário definir apenas o relacionamento entre as variáveis  $y$  e  $w$ , como apresentado em (2.19).*

$$y_{jk} = \max_{i \in \mathcal{U}} \{w_{ijk}\} \quad (2.19)$$

As restrições (2.8) e (2.9) são também satisfeitas pois dado (2.15), que implica em  $w_{ijk} \leq x_j$ ,  $\forall i \in \mathcal{U}$  e  $\forall \{j, k\} \subset P_i$  tem-se que  $\max_{i \in \mathcal{U}} \{w_{ijk}\} \leq x_j$  e considerando (2.19) é possível verificar que  $y_{jk} \leq x_j$ , (2.8) e (2.9).

As restrições (2.10) são satisfeitas pois se  $\sum_{\{j,k\} \in P_i} w_{ijk} = 1$ ,  $i \in \mathcal{U}$  e dada a relação definida em (2.19), então pode-se concluir que  $\sum_{\{j,k\} \in P_i} y_{jk} \geq 1$ .

Por outro lado, nem sempre é possível construir uma solução viável para  $\mathcal{F}2$  em termos de  $(x^*, y^*)$  a partir do vetor solução  $(x, w)$  da relaxação linear de  $\mathcal{F}1^+$ . Considere, por exemplo, a instância  $I$ , com único elemento a ser coberto e três possíveis objetos  $\mathcal{A} = \{1, 2, 3\}$  e  $\mathcal{U} = \{4\}$  considerando  $P_4 = \{\{1, 2\}, \{1, 3\}\}$  onde os custos dos elementos de  $\mathcal{A}$  correspondem a  $c_1 = 10$ ,  $c_2 = 4$  e  $c_3 = 1$ . Para esta instância, as formulações  $\mathcal{F}1^+$  e  $\mathcal{F}2$  podem ser escritas como segue.

$$\text{minimizar } Z_{\mathcal{F}1^+}: 10x_1 + 4x_2 + x_3 \quad (2.20)$$

$$\text{minimizar } Z_{\mathcal{F}2}: 10x_1 + 4x_2 + x_3 \quad (2.28)$$

Sujeito a:

Sujeito a:

$$w_{4,1,2} + w_{4,1,3} = 1 \quad (2.21)$$

$$y_{1,2} + y_{1,3} \geq 1 \quad (2.29)$$

$$w_{4,1,2} + w_{4,1,3} \leq x_1 \quad (2.22)$$

$$y_{1,2} \leq x_1 \quad (2.30)$$

$$w_{4,1,2} \leq x_2 \quad (2.23)$$

$$y_{1,2} \leq x_2 \quad (2.31)$$

$$w_{4,1,3} \leq x_3 \quad (2.24)$$

$$y_{1,3} \leq x_1 \quad (2.32)$$

$$x_1 + x_2 + x_3 \geq 2 \quad (2.25)$$

$$y_{1,3} \leq x_3 \quad (2.33)$$

$$x_i \in \{0, 1\}, \forall i \in \mathcal{A} \quad (2.26)$$

$$x_1 + x_2 + x_3 \geq 2 \quad (2.34)$$

$$w_{ijk} \in \{0, 1\}, \forall i \in \mathcal{U}, \forall \{j, k\} \subset P_i \quad (2.27)$$

$$x_i \in \{0, 1\}, \forall i \in \mathcal{A} \quad (2.35)$$

$$y_{ij} \in \{0, 1\}, \forall \{i, j\} \subset \mathcal{A} \quad (2.36)$$

A solução da relaxação linear de  $\mathcal{F}2$  ( $LP\text{-}\mathcal{F}2$ ) é  $x_1^* = 0.5$ ,  $x_2^* = 0.5$ ,  $x_3^* = 1$ ,  $y_{1,2}^* = 0.5$  e  $y_{1,3}^* = 0.5$  com  $Z_{LP\text{-}\mathcal{F}2}^* = 8$ . Note que não é possível construir uma solução com o mesmo custo para a relaxação linear de  $\mathcal{F}2$  pois a partir de (2.21) e (2.22), tem-se que  $x_1 \geq w_{4,1,2} + w_{4,1,3} = 1$ , implicando necessariamente  $x_1 = 1$  em  $\mathcal{F}2$ . Como custo do objeto 1 é  $c_1 = 10$ , obrigatoriamente  $Z_{LP\text{-}\mathcal{F}1^+} \geq 10$  e então  $Z_{LP\text{-}\mathcal{F}1^+} > Z_{LP\text{-}\mathcal{F}2}^* = 8$ .

A solução da relaxação linear de  $\mathcal{F}2^+$  ( $LP\text{-}\mathcal{F}1^+$ ) corresponde a  $x_1^* = 1$ ,  $x_2^* = 0$ ,  $x_3^* = 1$ ,  $w_{4,1,2}^* = 0$  e  $w_{4,1,3}^* = 1$  com custo  $Z_{LP\text{-}\mathcal{F}1^+}^* = 11$ . Como os valores das variáveis são inteiros, esta é também a solução ótima da instância I.

Experimentos computacionais foram realizados para ratificar a comparação entre as formulação  $\mathcal{F}1^+$  e  $\mathcal{F}2$ . Estes resultados são apresentados próxima seção.

## 2.5 Resultados Computacionais

Antes de apresentar os resultados obtidos pelos modelos de programação inteira é necessário descrever as instâncias que são consideradas nesta tese para avaliação empírica das abordagens analisadas.

### 2.5.1 Descrição das Instâncias

Neste trabalho foram utilizados três diferentes grupos de instâncias. O primeiro grupo é composto por 560 instâncias propostas por Breslau et al. [9] para o Problema do Conjunto Mínimo e Monitoramento. Como para estas instâncias os objetos possuem custo unitário, nesta tese foi proposto um segundo grupo de instâncias, derivadas das instâncias de Breslau et al., onde custos variados foram atribuídos aos objetos. Para que os experimentos não ficassem restritos a instâncias do Problema do Conjunto Mínimo de Monitoramento, um terceiro grupo de instâncias derivadas de instâncias propostas para o Problema de Cobertura de Conjuntos também foi utilizado. Assim, três grupos de instâncias foram utilizados nos experimentos computacionais realizados.

#### - Instâncias do Grupo I

O Grupo I é composto de 560 instâncias propostas por Breslau et al. [9] para o Problema do Conjunto Mínimo de Monitoramento. Nestas instâncias considera-se que todos os objetos que consistem em candidatos a nós de monitoramento, possuem um mesmo custo.

Para melhor apresentar os resultados, as 560 instâncias são divididas em sete conjuntos, de acordo com a parcela do total de nós associada aos objetos (conjunto  $\mathcal{A}$  - monitores) e elementos a serem cobertos (conjunto  $\mathcal{U}$  - *branches*), como mostra a Tabela 2.1. Nestas instâncias, o número total de nós das redes, denotado  $|N|$ , variou entre os valores 26, 50, 100, 190, 220, 250, 300 e 558. Os conjuntos de instâncias foram rotulados como mostra a primeira coluna da tabela. Para cada um dos sete conjuntos, existem 10 instâncias para cada valor de número de nós, totalizando 80 instâncias de cada conjunto. Para identificar cada conjunto foi adotada a nomenclatura  $M_m\_B_b$  onde o número de monitores (objetos) é  $|M| = 1/m \times |N|$  e o número de *branches* (elementos a serem cobertos)  $|B| = 1/b \times |N|$ .

Grupo	N =26		N = 50		N =100		N =190		N =220		N =250		N =300		N =558	
	M	B	M	B	M	B	M	B	M	B	M	B	M	B	M	B
M1_B1	26	26	50	50	100	100	190	190	220	220	250	250	300	300	558	558
M1_B2	26	13	50	25	100	50	190	95	220	110	250	125	300	150	558	279
M1_B4	26	7	50	13	100	25	190	48	220	55	250	63	300	75	558	140
M1_B8	26	4	50	7	100	13	190	24	220	28	250	32	300	38	558	70
M2_B2	13	13	25	25	50	50	95	95	110	110	125	125	150	150	279	279
M4_B4	7	7	13	13	25	25	48	48	55	55	63	63	75	75	140	140
M8_B8	4	4	7	7	13	13	24	24	28	28	32	32	38	38	70	70
Total de instâncias	70		70		70		70		70		70		70		70	

Tabela 2.1: Características das Instâncias do Grupo I.

### - Instâncias do Grupo II

Para compor o Grupo II, foram criadas 420 instâncias derivadas daquelas propostas por Breslau et al. [9] onde  $|N| = 300$  e  $|N| = 558$ . Para cada uma das 140 instâncias, foram criadas três novas instâncias onde os custos dos objetos foram obtidos a partir de três diferentes intervalos de valores. Para cada instância criada, a cada objeto foi selecionado aleatoriamente um custo num dado intervalo. Os intervalos utilizados foram  $[1, 10]$ ,  $[1, 100]$  e  $[1, 1000]$ . Na Tabela 2.2 é possível verificar como as 420 instâncias estão distribuídas. Estas instâncias estão disponíveis em <http://labic.ic.uff.br/Instance>.

Intervalo	$ N  = 300$	$ N  = 558$
$[1, 10]$	70	70
$[1, 100]$	70	70
$[1, 1000]$	70	70

Tabela 2.2: Características das Instâncias do Grupos II.

### - Instâncias do Grupo III

Neste trabalho é proposto um terceiro grupo de instâncias, derivado de instâncias disponíveis na *OR-Library* [4] originalmente desenvolvidas para o Problema de Cobertura de Conjuntos. Foram selecionadas 48 instâncias do PCC onde, para cada uma destas, foram criadas três novas instâncias para o Problema de Cobertura de Conjuntos por Pares.

Nas instâncias para o PCC, para cada linha a ser coberta existe uma lista, que chamouse de  $L$ , composta pelas colunas que a cobre. Para gerar uma instância do PCCP, as colunas em  $L$  associadas a uma determinada linha  $i$  foram tomadas duas a duas. Cada par de colunas obtido desta maneira tem uma probabilidade de não ser inserido na nova instância. Esta probabilidade é dada por um parâmetro  $p$ .

Cada instância do SCP originou três novas instâncias do SCPP, que foram obtidas utilizando-se para o parâmetro  $p$  os valores 25%, 50% e 75%. No Anexo A é apresentado o Algoritmo 12, que descreve o procedimento de geração das instâncias. Dados os valores de  $p$ , estas instâncias podem ser divididas em três conjuntos de 48 instâncias: *scp\_25*, *scp\_50* e *scp\_75*.

Na Tabela 2.3 é possível observar as características gerais deste conjunto de instâncias. As instâncias criadas possuem entre 50 e 28160 linhas (elementos) e número de colunas variando entre 192 e 11264 (objetos). O valor do parâmetro  $p$  afeta o número de pares

da instância, indicado na tabela por #pares. Quanto menor o valor de  $p$ , maior é a quantidade de pares nas instâncias. Utilizando-se  $p = 25\%$  o número médio de pares é de 238810 pares, e  $p = 75\%$  este valor é de 77543. As novas instâncias estão disponíveis em <http://labic.ic.uff.br/Instance>.

Características		Mínimo	Máximo
#linhas		50	28160
#colunas		192	11264
#pares	scp_25	1074	1013130
	scp_50	707	670600
	scp_75	348	329537

Tabela 2.3: Características das Instâncias do Grupo III.

## 2.5.2 Comparação Empírica entre Modelos $\mathcal{F}1^+$ e $\mathcal{F}2$

Os testes apresentados nesta seção visam reforçar o resultado teórico apresentado neste capítulo. Os modelos  $\mathcal{F}1^+$  e  $\mathcal{F}2$  foram submetidos ao resolvidor CPLEX 11.2 [39], executado em uma máquina Intel(R) Core(TM)2 Quad CPU Q9550 com 2.83GHz e 8GB de memória RAM utilizando a plataforma Linux 64 bits (kernel 2.6.27-16).

Os primeiros testes foram realizados utilizando as instâncias do Grupo I. Para este grupo de instâncias, o modelo  $\mathcal{F}1^+$  foi executado até a obtenção da solução ótima. Considerando  $\delta_{\mathcal{F}1^+}(i)$  o tempo gasto pela formulação  $\mathcal{F}1^+$  para obter a solução ótima de uma instância  $i$ , o tempo limite de execução estabelecido para o modelo  $\mathcal{F}2$ , representado por  $\Delta_{\mathcal{F}2}(i)$ , foi ajustado segundo a Equação (2.37).

$$\Delta_{\mathcal{F}2}(i) = \lceil \delta_{\mathcal{F}1^+}(i) \times 5 \rceil \quad (2.37)$$

A Tabela 2.4 mostra os resultados condensados das duas formulações para as 560 instâncias propostas por Breslau et al. [9]. A coluna **Conjunto** apresenta os conjuntos de instâncias do Grupo I, a coluna **RL** representam o *gap* médio entre a relaxação linear e a solução ótima, a coluna **R-LB** corresponde ao *gap* médio entre o limite inferior (*lower bound*) obtido no nó raiz e a solução ótima, a coluna **B-LB** indica o *gap* médio entre o limite inferior ao final da execução e a solução ótima e a coluna **B-UB** indica o *gap* médio entre a melhor solução inteira encontrada e a solução ótima. Considerando *opt* o custo associado a solução ótima, para calcular o *gap* entre *opt* e uma solução de custo  $x$  a seguinte equação foi utilizada:  $gap(x, opt) = Valor\_Absoluto(opt-x)/opt$ . Para as instâncias onde a solução ótima não foi determinada, no cálculo do *gap* será considerado

o custo da melhor solução inteira obtida.

Além dos *gaps*, na Tabela 2.4 são apresentados também os tempos de CPU consumidos pelo resolvidor utilizando cada um dos modelos, para resolver o problema de programação linear (**LP**), para resolver a raiz da árvore de *branch-and-cut* (**Raiz**) e o tempo total utilizado para indicar a melhor solução inteira obtida (**Total**).

Para cinco instâncias do conjunto M1-B1, a execução da formulação  $\mathcal{F}2$  foi interrompida por falta de memória. Então, para este conjunto, a Tabela 2.4 apresenta os resultados referentes às demais 75 instâncias deste conjunto.

Como pode ser observado na tabela, para todos os conjuntos de instâncias do Grupo I os *gaps* obtidos por  $\mathcal{F}1^+$  são consideravelmente menores do que aqueles obtidos por  $\mathcal{F}2$ . Enquanto utilizando-se  $\mathcal{F}1^+$  foi possível encontrar as soluções ótimas para todas as instâncias, utilizando-se a formulação  $\mathcal{F}2$  foram obtidas soluções ótimas apenas para 144 instâncias dentro do tempo limite estabelecido, que corresponde a 25.9% do total de instâncias. Além disso, apesar de ter sido definido um tempo de CPU grande em relação ao tempo consumido pelo modelo  $\mathcal{F}1^+$ , o valor médio de **B-LB** obtido por  $\mathcal{F}2$  foi de 47.56% e em 376 instâncias o tempo limite não foi suficiente para finalizar o processamento do nó raiz.

Além de apresentar um valor de **B-LB** alto, o *gap* entre a melhor solução inteira utilizando o modelo  $\mathcal{F}2$  e a solução ótima é, em média, de 260,56%, indicando que, em média, a diferença entre o custo da solução inteira obtida pelo modelo  $\mathcal{F}2$  e a solução ótima, é superior ao dobro do custo da solução ótima obtida pelo modelo  $\mathcal{F}1^+$ .



Considerando as mesmas métricas observadas para as instâncias do Grupo I, a Tabela 2.5 traz os resultados obtidos para as instâncias do Grupo II. Para este grupo de instâncias, também foi definido o tempo limite de execução para o modelo  $\mathcal{F}2$  de acordo com a Equação (2.37).

Com a utilização do modelo  $\mathcal{F}1^+$  foi possível indicar a solução ótima para todas as 420 instâncias do Grupo II, enquanto utilizando-se  $\mathcal{F}2$  foi possível para apenas 30 instâncias. A partir do modelo  $\mathcal{F}2$ , em apenas 107 destas instâncias o resolvidor foi capaz de determinar uma solução fracionária no tempo limite estabelecido. Dado que cada conjunto que compõe o Grupo II contém 60 instâncias, a coluna **#Inst** da Tabela 2.5 apresenta o número de instâncias de cada conjunto onde a formulação  $\mathcal{F}2$  conseguiu obter uma solução. As colunas **R-LB**, **B-LB** e **B-UB** associadas à  $\mathcal{F}2$  consideram apenas estas instâncias.

Como pode ser observado, para os conjuntos M1\_B4 e M1\_B8 a formulação  $\mathcal{F}2$  não foi capaz de determinar nenhuma solução no tempo limite. Comparando o *gap* da relaxação linear, obtido para todas as instâncias, nota-se que a formulação  $\mathcal{F}1^+$  tem um *gap* menor que o *gap* apresentado por  $\mathcal{F}2$ . Para 357 instâncias, o valor da relaxação linear obtido com a utilização de  $\mathcal{F}1^+$  coincide com o valor da solução ótima.

Para as instâncias do Grupo III, os resultados são apresentados na Tabela 2.6. Para este grupo de instâncias, foi estabelecido, para os dois modelos, um tempo limite de execução de dez horas. Os *gaps* apresentados na Tabela 2.6 são referentes à melhor solução encontrada a partir dos modelos utilizados, não se tratando, em todos os casos, da solução ótima da instância.

Utilizando-se a formulação  $\mathcal{F}1^+$  foram obtidas soluções ótimas para 85 instâncias, enquanto utilizando-se a  $\mathcal{F}2$  foram obtidas soluções ótimas somente para 11 instâncias. Para o conjunto de instâncias com a maior quantidade de pares (conjunto *scp\_25*), a formulação  $\mathcal{F}2$  não foi capaz de provar a otimalidade de nenhuma instância, sendo todas as execuções interrompidas devido ao tempo máximo de CPU (36000 segundos). Em nenhuma instância a melhor solução inteira obtida a partir de  $\mathcal{F}2$  possui custo inferior ao custo da melhor solução inteira de  $\mathcal{F}1^+$ , visto que a média do valor de **B-UB** associado a  $\mathcal{F}1^+$  é zero. A partir da Tabela 2.6 fica claro que  $\mathcal{F}1^+$  é superior a  $\mathcal{F}2$ , visto que a mesma produz melhores resultados em todas as métricas verificadas neste experimento.





# Capítulo 3

## Heurísticas Construtivas e de Busca Local

Neste capítulo são apresentados os algoritmos construtivos e de busca local para o PCCP desenvolvidos neste trabalho. Na Seção 3.1 são apresentadas as heurísticas construtivas gulosas, algoritmos capazes de iterativamente construir uma solução tendo como ponto de partida apenas os dados da instância a ser resolvida. Neste tipo de abordagem, a cada iteração, um novo elemento é incorporado na solução que está sendo construída até que a mesma se torne completa e viável. A cada etapa realiza-se uma avaliação dos elementos candidatos a participarem da solução e, aquele melhor avaliado, segundo uma função de avaliação, é inserido na solução.

Para que os procedimentos de construção sejam capazes de determinar soluções distintas, uma estratégia bastante utilizada é a introdução de elementos aleatórios no procedimento de construção. Na Seção 3.2 deste capítulo pode-se verificar como a aleatoriedade foi introduzida nos procedimentos de construção apresentados na Seção 3.1.

Além dos procedimentos de construção, neste capítulo são descritas três heurísticas de busca local, visando aprimorar as soluções obtidas a partir das heurísticas de construção.

### 3.1 Heurísticas Construtivas Gulosas

Nesta seção são apresentadas heurísticas determinísticas capazes de construir uma solução para o Problema de Cobertura de Conjuntos por Pares. Considerando  $\mathcal{A}$  o conjunto de objetos, onde cada objeto de  $a \in \mathcal{A}$  possui associado um custo  $c_a$ , e  $\mathcal{U}$  o conjunto dos elementos  $e \in \mathcal{U}$  que devem ser cobertos,  $P_e$  o conjunto dos pares  $\{i, j\} \subset \mathcal{A}$  onde  $\mathcal{C}(i, j) \supset \{e\}$  e ainda  $T_e$  o conjunto dos objetos dos pares que cobrem  $e$ , ou seja,  $T_e = \cup_{\{i, j\} \in P_e} \{i, j\}$ , as heurísticas construtivas podem ser descritas como segue.

### 3.1.1 Heurística do Melhor Objeto - HMO

Este algoritmo é uma simplificação do algoritmo guloso proposto por Hassim e Segev [35]. Enquanto na heurística original os elementos são avaliados individualmente e aos pares, na abordagem proposta é considerada apenas a inserção de um objeto a cada iteração. Outra diferença refere-se à função utilizada na avaliação do ganho com a inserção de um objeto na solução.

Sendo  $S \subset \mathcal{A}$  uma solução parcial para o PCCP, a cada iteração são avaliados os objetos  $a \in \mathcal{A} \setminus S$ . Para avaliar o ganho com a inserção de um objeto  $a$  considera-se o número de novos elementos que serão cobertos por  $S \cup \{a\}$ , definido por  $\mathcal{C}(S \cup \{a\}) \setminus \mathcal{C}(S)$ . Além disto, na função de avaliação considera-se o custo  $c_a$  associado ao objeto e também o número de elementos  $e \in \mathcal{U}$  ditos parcialmente cobertos. Um elemento  $e$  é dito parcialmente coberto por  $S \cup \{a\}$  se  $a \in T_e$  mas  $e \notin \mathcal{C}(S \cup \{a\})$ , ou seja, o elemento que forma par com  $a$  em  $P_e$  não pertence a  $S$ .

Considere  $U_a$  o conjunto dos elementos não cobertos pela solução parcial  $S$  onde o objeto  $a$  pertence a pelo menos um dos pares que podem cobri-los. Dados os conjuntos  $U'$  e  $U''$  onde  $U' \cup U'' = U_a$ , sendo  $U'$  o conjunto de elementos cobertos por  $S \cup \{a\}$  e  $U''$  o conjunto composto pelos elementos parcialmente cobertos por  $S \cup \{a\}$ , na avaliação do ganho associado ao objeto  $a$  é considerada a cardinalidade dos conjuntos  $U'$  e  $U''$ . Com objetivo de priorizar a cardinalidade de  $U'$  em relação à cardinalidade de  $U''$ , são incluídos na avaliação  $p_1$  e  $p_2$  como pesos associados aos conjuntos  $U'$  e  $U''$  respectivamente, como mostra a Equação 3.1.

$$f(a) = \frac{p_1 \times |U'| + p_2 \times |U''|}{c_a} \quad (3.1)$$

No Algoritmo 1 é possível verificar cada uma das etapas da Heurística do Melhor Objeto (HMO). Como pode ser observado na linha 1, o conjunto solução  $S$  encontra-se inicialmente vazio. Então, enquanto o conjunto  $S$  não cobre todos os elementos de  $\mathcal{U}$  (linha 2), a cada iteração um objeto  $j \in \mathcal{A} \setminus S$  é inserido no conjunto solução. Para selecionar o objeto a ser inserido, os candidatos são avaliados de acordo com a Equação 3.1. Aquele objeto com maior valor associado à equação é inserido em  $S$ , na linha 4.

### 3.1.2 Heurística do Objeto com Maior Alcance - HOMA

Nesta seção é apresentada outra simplificação do algoritmo guloso proposto em [35], similar ao apresentado na seção anterior. A diferença está na função de avaliação dos objetos.

---

**Algoritmo 1** *HMO* ( $p_1, p_2$ ).
 

---

- 1:  $S \leftarrow \emptyset$ ;
  - 2: **enquanto**  $\mathcal{C}(S) \neq \mathcal{U}$  **faça**
  - 3:    $j \leftarrow \operatorname{argmax}_{j \in \mathcal{A} \setminus S} f(j)$
  - 4:    $S \leftarrow S \cup \{j\}$ ;
  - 5: **fim enquanto**
- 

No algoritmo HOMA, Heurística do Objeto com Maior Alcance, para selecionar um entre os objetos candidatos a participarem do conjunto solução  $S$  é utilizada outra função de avaliação, apresentada na Equação (3.2). Na avaliação de cada objeto candidato  $a$  são considerados o custo deste objeto  $c_a$  e também o número de elementos em  $\mathcal{U}$  ainda não cobertos por  $S$  onde  $a \in T_e$ . Os demais passos do algoritmo são compartilhados por ambas as abordagens.

$$f(a) = \frac{|Q_a|}{c_a}, \text{ onde } Q_a = \{e \in \mathcal{U} | e \notin \mathcal{C}(S) \text{ e } T_e \supset \{a\}\} \quad (3.2)$$

### 3.1.3 Heurística do Objeto com Maior Participação - HOMP

A Heurística do Objeto com Maior Participação (HOMP) também compartilha a estrutura do Algoritmo 1, utilizando uma função diferente para classificar os objetos candidatos a serem inseridos na solução. Na classificação dos objetos, a HOMP utiliza a Equação 3.3, onde é considerado o custo associado ao objeto e também a participação deste objeto nos pares dos elementos ainda não cobertos por  $S$ .

Considere novamente, para um dado objeto  $a \in \mathcal{A}$ ,  $Q_a = \{e \in \mathcal{U} | e \notin \mathcal{C}(S) \text{ e } T_e \supset \{a\}\}$ . Para cada  $e \in Q_a$  será definido o conjunto  $P_e^a$  como o conjunto dos pares  $\{i, j\} \in P_e$  onde  $i = a$  ou  $j = a$ . Assim, na Equação 3.3 é considerado, para um determinado objeto  $a$ , seu custo de inserção  $c_a$  e também, para cada elemento  $e \in Q_a$ , o percentual dos pares deste elemento que incluem o objeto  $a$ .

$$f(a) = \frac{\sum_{e \in Q_a} |P_e^a| / |P_e|}{c_a} \quad (3.3)$$

### 3.1.4 Heurística do Melhor Par - HMP

A Heurística do Melhor Par utiliza uma abordagem diferente da utilizada pelas heurísticas apresentadas nas seções anteriores. A idéia foi baseada na heurística da inserção mais barata [11]. Como pode ser observado no Algoritmo 2, a cada iteração um elemento  $e \in \mathcal{U}$  ainda não coberto é selecionado e um dos pares de objetos de  $P_e$  é selecionado para

ser incluído no conjunto solução  $S$ .

Considerando  $P_e$  o conjunto dos possíveis pares de objetos que podem cobrir o elemento  $e \in \mathcal{U}$ , o primeiro passo do algoritmo é ordenar os elementos de forma ascendente de acordo com a cardinalidade  $P_e$  (linha 1), obtendo a lista ordenada  $\hat{U}$ . E, como apresentado na linha 2, o conjunto  $S$  está inicialmente vazio, de forma idêntica às heurísticas anteriores.

---

**Algoritmo 2 HMP**


---

```

1:  $\hat{U} \leftarrow \text{Ordenar}(\mathcal{U})$ ;
2:  $S \leftarrow \emptyset$ ;
3: enquanto  $(\mathcal{C}(S) \neq \mathcal{U})$  faça
4:    $e \leftarrow \text{Selecionar\_elemento}(\hat{U})$ 
5:    $\{a_1, a_2\} \leftarrow \text{Selecionar\_melhor\_par}(P_e)$ ;
6:    $S \leftarrow S \cup \{a_1, a_2\}$ ;
7:   Atualizar  $(\hat{U}, S)$ ;
8: fim enquanto

```

---

Depois desta etapa de inicialização, enquanto o conjunto  $S$  não caracteriza uma solução para o problema, ou seja, enquanto  $\mathcal{C}(S) \neq \mathcal{U}$ , a cada iteração um dos elementos  $e \in \hat{U}$  é selecionado na linha 4, respentando-se a ordem estabelecida na linha 1 do algoritmo.

Na avaliação dos pares de objetos presentes em  $P_e$  é considerado o custo dos objetos envolvidos e também a quantidade de novos elementos cobertos  $(\mathcal{C}(S \cup \{a_1, a_2\}) \setminus \mathcal{C}(S))$ . A Equação 3.4 apresenta esta função de avaliação.

$$g(a_1, a_2) = (\mathcal{C}(S \cup \{a_1, a_2\}) \setminus \mathcal{C}(S)) / (c_{a_1} + c_{a_2}) \quad (3.4)$$

Vale ressaltar que o custo de um dos objetos do par  $\{a_1, a_2\}$  pode ser desconsiderado, caso este já pertença ao conjunto  $S$ . Na linha 5, o par de objetos com maior valor associado a  $g(a_1, a_2)$  será selecionado e, no próximo passo, inserido no conjunto  $S$ . Após a inclusão, o conjunto  $\hat{U}$  é atualizado (linha 7). Nesta etapa são removidos de  $\hat{U}$  todos os elementos cobertos, visto que a inserção de um par pode ocasionar a cobertura de outros elementos além do elemento  $e$  considerado na iteração corrente.

### 3.1.5 Heurística *Add Drop* - HAD

Quando comparada com as abordagens apresentadas nas seções anteriores, verifica-se que a Heurística *Add-Drop* funciona de forma inversa às demais. Como pode ser observado na linha 1 do Algoritmo 3, inicialmente o conjunto solução engloba todos os objetos do

conjunto  $\mathcal{A}$ . Desta forma, a solução  $S$  é, durante todo o procedimento, uma solução completa.

---

**Algoritmo 3** *HAD*


---

```

1:  $S \leftarrow \mathcal{A}$ ;
2:  $A' \leftarrow \text{Ordenar}(\mathcal{A})$ ;
3: enquanto ( $A' \neq \emptyset$ ) faça
4:    $a \leftarrow \text{Retirar\_objeto}(A')$ ;
5:    $resp \leftarrow \text{Verificar\_remoção}(a)$ ;
6:   se ( $resp = \text{true}$ ) então
7:      $S \leftarrow S \setminus \{a\}$ ;
8:   fim se
9: fim enquanto

```

---

Na linha 2, os objetos  $a \in \mathcal{A}$  são ordenados de modo não decrescente de acordo com o custo  $c_a$  associado a cada um destes. Nesta heurística é verificada a opção de retirar cada um dos objetos  $a \in A'$  do conjunto  $S$ , ou seja, o conjunto  $A'$  é composto por aqueles objetos cuja possibilidade de remoção ainda não foi verificada.

Na linha 4, um objeto  $a$  é selecionado segundo a ordem estabelecida na linha 2 e removido do conjunto  $A'$ . Em seguida, na linha 5, é verificado se a remoção do objeto  $a$  do conjunto  $S$  fará com que algum elemento de  $\mathcal{U}$  fique descoberto. Se a remoção não causar inviabilidade na solução, este objeto é removido de  $S$  na linha 7. O algoritmo termina quando não houver nenhum candidato no conjunto  $A'$  (linha 3).

## 3.2 Heurísticas Construtivas Gulosas Randomizadas

Nesta seção descreve-se como a aleatoriedade foi introduzida nos algoritmos de construção apresentados neste trabalho. A randomização é importante para possibilitar que os algoritmos construtivos gerem soluções diferentes a cada execução.

### 3.2.1 HMO, HOMA e HOMP Randomizadas

Como as heurísticas HMO, HOMA e HOMP diferem apenas quanto a função de avaliação dos elementos candidatos a serem inseridos na solução, o fator de aleatoriedade é inserido de forma similar para estas abordagens. O algoritmo 4 ilustra como funciona a versão randomizada destas heurísticas, onde o parametro  $\alpha$  indica o grau de aleatoriedade do algoritmo.

Inicialmente, o conjunto solução encontra-se vazio (linha 1). Considerando  $LC$  a

**Algoritmo 4** *Heurísticas Randomizadas - HMO, HOMA e HOMP* ( $\alpha$ )

---

```

1:  $S \leftarrow \emptyset$ ;
2:  $LC \leftarrow \mathcal{A}$ 
3: enquanto ( $\mathcal{C}(S) \neq \mathcal{U}$ ) faça
4:    $LRC \leftarrow$  Criar_Lista_Restrita_Candidatos ( $LC, f$ );
5:    $j \leftarrow$  Seleção_aleatória ( $LRC$ );
6:    $S \leftarrow S \cup \{j\}$ ;
7:    $LC \leftarrow$  Atualizar ( $LC, S$ );
8: fim enquanto

```

---

lista de candidatos formada por todos os objetos que podem ser introduzidos na solução parcial  $S$ , inicialmente  $LC$  é formada por todos os objetos pertencentes a  $\mathcal{A}$ . Da mesma forma como apresentado na seção anterior, a seleção do próximo objeto a ser inserido na solução é determinada pela avaliação dos objetos candidatos de acordo com uma função de avaliação  $f$ , neste caso Equações 3.1, 3.2 e 3.3.

Como apresentado na linha 4 do algoritmo, a Lista Restrita de Candidatos ( $LRC$ ) é formada pelos objetos com maior valor associado a função de avaliação. Para determinar a quantidade de objetos de  $LC$  que serão incorporados a  $LRC$  é utilizado um parâmetro  $\alpha \in [0, 1]$ . A  $LRC$  será formada pelos  $\alpha \times |LC|$  elementos melhor avaliados de  $LC$ . O elemento a ser introduzido na solução parcial é selecionado aleatoriamente entre os objetos pertencentes a  $LRC$ , na linha 5. Após a inserção do objeto selecionado (linha 6), a  $LC$  é atualizada na linha 7. Nesta atualização, além de se retirar o objeto  $j$  da lista  $LC$ , outros objetos que não cobrem os elementos de  $\mathcal{U} \setminus \mathcal{C}(S)$  também são retirados.

### 3.2.2 HMP Randomizada

Na Heurística do Melhor Par, a aleatoriedade é inserida no processo de seleção do elemento a ser coberto, linha 4 do Algoritmo 2. A lista de candidatos  $LC$  é inicialmente composta por todos os elementos de  $\mathcal{U}$  e, a cada iteração, a Lista Restrita de Candidatos  $LRC$  é formada pelos elementos de  $LC$  ainda não cobertos com menor valor associado a  $|P_e|$ . Da mesma forma como apresentado na seção anterior, a quantidade de elementos de  $LRC$  será limitada por um parâmetro  $\alpha$ . A seleção do par a ser incluído na solução e demais passos do algoritmo são idênticos aos apresentados na Seção 3.1.4.

### 3.2.3 HAD Randomizada

Na Heurística *Add Drop*, o procedimento de seleção do objeto a ser removido a cada iteração (linha 4 do Algoritmo 3) foi randomizada. Na versão randomizada desta abordagem, durante a inicialização do algoritmo, a Lista de Candidatos (*LC*) é preenchida com todos os objetos do conjunto  $\mathcal{A}$ . E, a cada iteração, a Lista Restrita de Candidatos (*LRC*) é formada pelos objetos  $a \in LC$  com maior custo  $c_a$ . O número de elementos de *LRC* é, da mesma forma como as demais heurísticas randomizadas, limitada pelo parâmetro  $\alpha$ . Um dos objetos de *LRC* é selecionado aleatoriamente e, em seguida, verifica-se se é possível remover o mesmo do conjunto  $S$  sem inviabilizar a solução. Após esta análise, o objeto é removido da *LC*. O algoritmo termina quando *LC* estiver vazia.

## 3.3 Busca Local

O objetivo do procedimento de busca local é, de forma iterativa, melhorar a qualidade de uma solução substituindo, a cada etapa, a solução corrente por uma solução de menor custo que se encontre na sua vizinhança.

A busca na vizinhança pode ser implementada utilizando a estratégia *best-improving* ou *first-improving*. No caso da estratégia *best-improving* todos os vizinhos de uma solução são avaliados e a solução corrente é substituída pelo vizinho de menor custo. Já na estratégia *first-improving*, a solução corrente é substituída pela primeira solução encontrada na vizinhança com custo inferior ao custo da solução atual.

Nas próximas subseções são apresentados dois algoritmos de busca local que implementam a estratégia *best-improving*, Busca Local Troca Objeto e Busca Local Troca Par, e uma estratégia que utiliza a abordagem *first-improving*, Busca Local *Drop Add*.

### 3.3.1 Busca Local Troca Objeto

A vizinhança analisada por este algoritmo de busca local considera pequenos ajustes causados pela remoção de um único objeto do conjunto solução  $S$  e posterior reconstrução da solução evitando a utilização do objeto removido. O Algoritmo 5, que recebe como parâmetro de entrada uma solução obtida por um dos algoritmos construtivos, descreve a heurística Busca Local Troca Objeto - BLTO.

Considerando  $S^*$  a melhor solução observada durante todo o procedimento de busca,



**Algoritmo 5** *BLTO* ( $S$ )

---

```

1:  $S^* \leftarrow S$ ;
2: enquanto (melhorar solução  $S^*$ ) faça
3:    $S'' \leftarrow S^*$ ;
4:   para todo objeto  $j \in S^*$  faça
5:      $S' \leftarrow S^* \setminus \{j\}$ ;
6:      $S' \leftarrow$  Reconstrução_com_penalização ( $S', j$ );
7:     se ( $\text{custo}(S') < \text{custo}(S'')$ ) então
8:        $S'' \leftarrow S'$ ;
9:     fim se
10:  fim para
11:  se ( $\text{custo}(S'') < \text{custo}(S^*)$ ) então
12:     $S^* \leftarrow S''$ ;
13:  fim se
14: fim enquanto

```

---

inicialmente  $S^*$  receberá a solução passada como parâmetro (linha 1). A cada iteração, linhas 2 a 14,  $S''$  armazena a melhor solução na vizinhança de  $S^*$  e  $S'$  representa uma solução temporária.

Com objetivo de determinar a melhor solução na vizinhança de  $S^*$ , cada objeto presente no conjunto solução é removido (linha 5) e a solução é reconstruída utilizando uma das heurísticas construtivas apresentadas neste trabalho. Na reconstrução, há uma penalização para evitar que o objeto removido seja reinserido na solução (linha 6).

Dentre as heurísticas construtivas apresentadas neste trabalho, apenas HAD não pode ser utilizada na reconstrução. Isto porque esta é uma heurística inversa, onde todos os objetos são inicialmente inseridos na solução e depois verifica-se a possibilidade de remover cada um deles.

Por se tratar de um procedimento que adota a estratégia *best-improving*, a cada iteração toda a vizinhança é analisada (linhas 4 a 10) e  $S''$  armazenará a melhor solução na vizinhança de  $S^*$ . Na linha 12 ocorre a atualização de  $S^*$ , caso tenha sido possível encontrar uma solução  $S''$  na vizinhança com custo inferior ao custo de  $S^*$ . Caso a atualização ocorra, a busca reiniciará na vizinhança da nova solução  $S^*$ . O algoritmo termina quando toda a vizinhança é analisada e nenhuma solução melhor que  $S^*$  for encontrada.

### 3.3.2 Busca Local Troca Par

A Busca Local Troca Par (BLTP) é muito semelhante à estratégia BLTO. A diferença é que, na BLTP, um par de objetos é removido da solução  $S^*$  a cada iteração. Por per-

mitir movimentos mais significativos, a heurística BLTP percorre um espaço de soluções diferente do percorrido pela BLTO.

O Algoritmo 6 apresenta cada uma das etapas da BLTP. Comparando BLTP e BLTO, as diferenças podem ser observadas na linha 4, onde cada par de objetos é examinado a cada iteração e, conseqüentemente, nas linhas 5 e 6, já que os objetos  $i$  e  $j$  são removidos e o procedimento de reconstrução penaliza a reinserção destes objetos. Os demais passos são realizados da mesma forma como apresentado na BLTO.

---

**Algoritmo 6** *BLTP* ( $S$ )
 

---

```

1:  $S^* \leftarrow S$ ;
2: enquanto (melhorar a solução) faça
3:    $S'' \leftarrow S^*$ ;
4:   para todo par  $\{i, j\} \in S^*$  faça
5:      $S' \leftarrow S^* \setminus \{i, j\}$ 
6:      $S' \leftarrow$  Reconstrução_com_penalização ( $S', i, j$ );
7:     se ( $\text{custo}(S') < \text{custo}(S'')$ ) então
8:        $S'' \leftarrow S'$ ;
9:     fim se
10:  fim para
11:  se ( $\text{custo}(S'') < \text{custo}(S^*)$ ) então
12:     $S^* \leftarrow S''$ ;
13:  fim se
14: fim enquanto

```

---

### 3.3.3 Busca Local *Drop Add*

Como o ganho referente a inserção de um objeto de  $\mathcal{A}$  no conjunto solução  $S$  está associado ao estado atual de  $S$ , a ordem de inclusão dos objetos na solução é um fator que afeta diretamente a escolha do(s) próximo(s) objetos(s). Neste sentido, o procedimento Busca Local *Drop Add* (BLDA), apresentado no algoritmo 7, tem como objetivo provocar estados de  $S$  não observados durante a construção, para que outras soluções na vizinhança de  $S$  sejam analisadas. Esta heurística utiliza a estratégia *first-improving*.

O algoritmo recebe como parâmetro uma solução  $S$ , obtida por um dos algoritmos construtivos descritos na Seção 3.2, uma constante  $\gamma \in [0, 1]$  e um inteiro positivo  $k$  que indica o critério de parada do algoritmo. A primeira etapa do algoritmo consiste em retirar alguns dos objetos da solução corrente. Para isto, inicialmente são escolhidos aleatoriamente  $\gamma \times |S|$  objetos da solução corrente  $S$  formando o conjunto  $Q$  (linha 3). Os elementos que são removidos ficam impedidos de serem removidos nas próximas iterações, e, para isto, estes objetos são inseridos numa lista tabu  $lt$ . Após esta etapa, os elementos

**Algoritmo 7** *Busca Local Drop Add* ( $S, \gamma, k$ )

---

```

1:  $S^* \leftarrow S$ ;
2: enquanto (iterações sem melhoria  $< k$ ) faça
3:    $Q \leftarrow$  Seleção_objetos(  $S, \gamma, lt$  );
4:    $S \leftarrow S \setminus Q$ ;
5:    $S'' \leftarrow$  Reconstruir_solução ( $S$ );
6:   se ( custo( $S''$ )  $<$  custo( $S^*$ ) ) então
7:      $S^* \leftarrow S''$ ;
8:   fim se
9:    $S \leftarrow S^*$ ;
10: fim enquanto

```

---

de  $Q$  são retirados do conjunto solução (linha 4), tornando a mesma inviável, ou seja, deixando descobertos alguns dos elementos de  $\mathcal{U}$ . Para viabilizar a solução, gerando a solução  $S''$ , realiza-se a reconstrução considerando apenas os elementos descobertos em  $\mathcal{U}$  (linha 5). Na reconstrução é determinado o conjunto de objetos que deve ser incluído na solução de forma a cobrir estes elementos com um menor custo.

Caso a solução  $S''$  tenha um custo inferior ao custo da melhor solução encontrada até a iteração atual, esta é atualizada de forma a armazenar sempre a melhor solução (linha 7). Este processo se repete enquanto melhores soluções estiverem sendo encontradas, sendo interrompido quando o algoritmo realizar  $k$  iterações sem atualizar  $S^*$ .

## 3.4 Procedimentos Complementares

Devido às características do Problema de Cobertura de Conjunto por Pares, é possível se utilizarem técnicas auxiliares no processo de busca de soluções de melhor qualidade. Neste trabalho são sugeridos dois procedimentos: fixação de objetos, que deve ser realizado antes do início da construção; e remoção de objetos redundantes, para ser executado após a solução ser gerada.

### 3.4.1 Fixação de Objetos

Com a técnica de Fixação de Objetos é possível verificar se um objeto, ou conjunto de objetos, deve obrigatoriamente pertencer à solução. Esta situação pode acontecer com instâncias que apresentam uma das características ou ambas descritas a seguir:

- Um elemento possui apenas um par de objetos capaz de cobri-lo. Neste caso, o par de objetos obrigatoriamente pertencerá ao conjunto solução. Esta situação é

ilustrada na Figura 3.1, onde o elemento 7 possui apenas um par capaz de cobri-lo,  $\{1, 2\}$ . Neste caso, pode-se garantir que os objetos 1 e 2 farão parte da solução.

- Todos os pares que podem ser utilizados para cobrir um determinado elemento possuem um objeto em comum, situação ilustrada na Figura 3.2. O elemento 6 possui dois pares capazes de cobri-lo,  $\{2, 3\}$  e  $\{3, 4\}$ , assim, independente do par selecionado, o objeto 3 fará parte da solução.

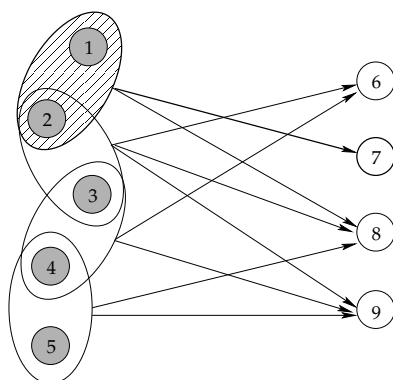


Figura 3.1: Instância que apresenta elemento com único par para cobertura.

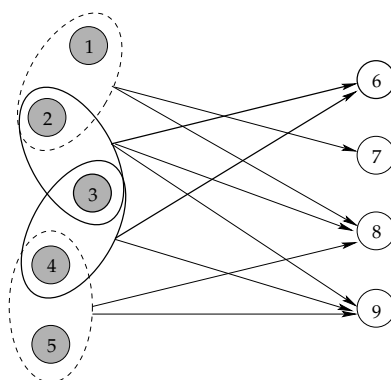


Figura 3.2: Instância onde elemento possui pares com objeto em comum.

### 3.4.2 Remoção de Objetos Redundantes

O procedimento de Remoção de Objetos Redundantes é utilizado na etapa pós-construção, possibilitando um refinamento na solução gerada. Quando um elemento possui mais de um par de objetos capaz de cobri-lo, talvez seja possível remover algum(ns) objeto(s) do conjunto solução. Considerando que os elementos de  $\mathcal{U}$  foram cobertos seguindo uma determinada ordem, por exemplo  $u_1, u_2, \dots, u_n$ , pode ser que os pares escolhidos para cobrir o elemento  $u_5$  cubram também  $u_1$ . Desta forma, o par selecionado para cobrir  $u_1$ , se não estiver sendo utilizado para cobrir nenhum outro elemento, pode ser removido do conjunto  $S$ .

Como o PCCP é um problema de minimização, na implementação deste procedimento os objetos são ordenados de forma não crescente de acordo com o custo. Depois disto, cada objeto é avaliado, de acordo com a ordem estabelecida, para verificar se é possível removê-lo do conjunto solução.

Na próxima seção são apresentados os resultados obtidos por cada uma das abordagens apresentadas neste capítulo.

## 3.5 Resultados Computacionais

As heurísticas propostas neste capítulo foram implementadas na linguagem C++, compiladas no g++ versão 4.4.1. Os testes com os algoritmos foram realizados em uma máquina Intel(R) Core(TM) i7 CPU 870 com 2.93GHz e 8GB de memória RAM.

Para uma comparação entre as abordagens heurísticas apresentadas neste trabalho quatro métricas são utilizadas.

- *#Melhor*

Dado um conjunto de algoritmos e um conjunto de instâncias, a métrica *#Melhor* indica o número de execuções em que cada algoritmo obteve uma solução de custo tão bom quanto o custo da melhor solução obtida para cada instância considerando-se este conjunto de abordagens.

- *MDif*

Seja *Dif* a diferença percentual entre a solução obtida por um dado algoritmo em uma execução e a melhor solução obtida pelos algoritmos considerados no experimento. A métrica *MDif* de um algoritmo indica, para o grupo de instâncias, o valor médio de *Dif* em todas as execuções.

- *Score*

Para um conjunto de algoritmos, a métrica *NScore* de um dado algoritmo  $\mathcal{A}$  para uma determinada instância indica o número de algoritmos que obtiveram solução melhor que a obtida por  $\mathcal{A}$  nesta instância. O *Score* deste algoritmo  $\mathcal{A}$ , como definido por Ribeiro et al. [54], é a soma do *NScore* obtido para cada uma das instâncias consideradas.

- *Tempo de CPU*

Tempo médio de CPU em segundos.

### 3.5.1 Heurísticas de Construção

Para uma comparação entre as heurísticas construtivas apresentados neste trabalho as instâncias dos grupos I e II são utilizadas.

Além das heurísticas propostas neste capítulo, nesta seção são apresentados também os resultados referente às heurísticas construtivas propostas por Hassin e Segev [35] e por Breslau et al. [9]. Na Tabela 3.1 são listadas as heurísticas de construção tratadas

nesta sessão. Testes preliminares foram realizados para verificar bons parâmetros para a heurística HMO, testes estes apresentados na primeira seção do Apêndice B. Assim, com base nestes resultados, para a heurística HMO foram considerados os parâmetros ( $p_1 = 1$  e  $p_2 = 0$ ) e ( $p_1 = 10$  e  $p_2 = 1$ ).

Tabela 3.1: Heurísticas Construtivas.

Identificador	Descrição
Hassin	Heurística proposta no trabalho de Hassin e Segev [35]
Breslau	Heurística proposta no trabalho de Breslau et al. [9]
HMO <sub>(1,0)</sub>	Heurística do Melhor Objeto - parâmetros $p_1 = 1$ e $p_2 = 0$
HMO <sub>(10,1)</sub>	Heurística do Melhor Objeto - parâmetros $p_1 = 10$ e $p_2 = 1$
HOMA	Heurística do Objeto com Melhor Alcance
HOMP	Heurística do Objeto com Melhor Participação
HMP	Heurística do Melhor Par
HAD	Heurística <i>Add Drop</i>

Em testes preliminares o procedimento complementar de fixação de objetos não se mostrou efetivo para as instâncias tratadas, visto que só foi possível executar fixação para pouca instâncias. Por este motivo, evitando onerar desnecessariamente os algoritmos, este procedimento não é realizado nos experimentos apresentados nesta seção.

Quanto ao procedimento de remoção de objetos redundantes, este mostrou-se eficiente quando aplicado após cada uma das heurísticas construtivas. Assim, estas heurísticas executam este procedimento complementar ao final de sua execução.

A heurística gulosa de Hassin considera, para cada elemento, a razão entre o número de novos elementos cobertos com a inserção deste elemento na solução e o custo deste objeto. Além de avaliar cada elemento individualmente, a inserção de pares de elementos também é considerada. O objeto, ou par de objetos, com maior valor associado a esta razão na iteração corrente é incluído na solução. O que difere a heurística de Breslau da heurística de Hassin é que na primeira a avaliação dos pares de objetos ocorre apenas quando não é possível cobrir nenhum elemento utilizando-se apenas um objeto.

Nas tabelas 3.2 e 3.3 são sintetizados os resultados obtidos pelas heurísticas construtivas. Para comparar cada uma destas abordagens foram realizados testes com as versões gulosas e randomizadas. Como a literatura apresenta apenas a versão gulosa das heurísticas de Hassin e Breslau, para estas abordagens não foram desenvolvidas versões randomizadas. Para a versão randomizada das heurísticas, o parâmetro  $\alpha$  foi fixado em 0,2 e cada abordagem foi executada dez vezes.

A Tabela 3.2 apresenta os resultados obtidos para as 560 instâncias do Grupo I. A

primeira coluna da tabela divide os resultados obtidos com a versão gulosa e com a versão randomizada das heurísticas.

Pelos resultados apresentados para as versões gulosas, observa-se que as heurísticas da literatura obtiveram bons resultados em relação às métricas *#Melhor*, *MDif* e *Score*. Entretanto, em relação ao tempo de CPU, observa-se uma diferença considerável em relação aos tempos requeridos pelas heurísticas propostas neste trabalho. Enquanto as heurísticas construtivas propostas consumiram, em média, menos de um segundo, o tempo médio de CPU da heurística de Hassin é superior a 100 segundos, ao passo que o tempo da heurística de Breslau foi superior a oito segundos.

Entre as oito abordagens gulosas apresentadas na Tabela 3.2, HMP se mostrou melhor que as demais em relação às métricas *#Melhor*, *MDif* e *Score*. Em relação ao tempo de CPU, entre as abordagens propostas neste trabalho, HMP é aquela que consome mais tempo.

Tabela 3.2: Heurísticas Construtivas - Instâncias do Grupo I.

	Métricas	Hassin	Breslau	HMO <sub>(1,0)</sub>	HMO <sub>(10,1)</sub>	HOMA	HOMP	HMP	HAD
Gulosa	<i>#Melhor</i>	512	512	113	113	5	521	<b>534</b>	283
	<i>MDif</i> (%)	0,31	0,31	5,62	5,62	66,76	0,89	<b>0,14</b>	3,62
	<i>Score</i>	81	81	1943	1943	3879	116	<b>25</b>	1418
	Tempo (s)	111,22	8,51	0,21	0,21	0,21	<b>0,09</b>	0,53	0,32
Rand.	<i>#Melhor</i>	-	-	512	488	51	1120	<b>5204</b>	2817
	<i>MDif</i> (%)	-	-	42,27	42,77	66,62	37,46	<b>0,20</b>	3,65
	<i>Score</i>	-	-	1485	1548	1904	1071	<b>669</b>	1139
	Tempo (s)	-	-	0,36	0,36	0,23	0,20	0,71	<b>0,16</b>

Em relação às versões randomizadas, a heurística HMP também se destaca das demais abordagens. Em 5204 execuções de um total de 5600, a solução obtida por HMP foi igual a melhor solução alcançada pelas heurísticas construtivas, o que equivale a 93% das execuções. A média da diferença percentual entre as soluções desta abordagem e a melhor solução foi de 0,20%. Além disso, quanto à métrica *Score*, a heurística HMP apresentou melhor resultado. Quanto ao tempo médio de CPU, os resultados são semelhantes aos obtidos nas versões gulosas, o tempo consumido por HMP é superior, mas não a ponto de inviabilizar a utilização desta heurística.

Para as instâncias do Grupo II, os resultados são apresentados na Tabela 3.3. Para este grupo de instâncias, a heurística HAD destaca-se com os melhores resultados para as quatro métricas consideradas. Vale ressaltar que para o cálculo da métrica *#Melhor*, considerou-se a melhor solução entre as soluções obtidas pelas heurísticas de construção nas versões gulosas e randomizadas. A heurística HAD, que obteve melhor resultado, na

sua versão gulosa conseguiu alcançar a melhor solução em 199 instâncias.

A heurística HMO com os dois conjuntos de parâmetros e a heurística HOMA não foram capazes de obter, para qualquer instância, uma solução de custo igual ao custo da melhor solução. Além disso, o valor da métrica *MDif* associado à versão gulosa da heurística HOMA foi muito superior ao obtido pelas demais abordagens. As duas heurísticas da literatura não obtiveram um bom desempenho para este segundo grupo de instâncias quando comparado as heurísticas HAD e HMP, mesmo executando em um tempo superior aos tempos destas duas heurísticas propostas.

Para as versões randomizadas das heurísticas construtivas, a heurística HAD também se destaca. O número de execuções em que a solução obtida foi tão boa quanto a melhor solução obtida é muito superior ao que se verifica para as demais abordagens. Além dos melhores resultados quanto a qualidade das soluções, HAD também possui o menor tempo médio de CPU. A métrica *Score*, que possibilita uma comparação entre as abordagens, mostra que a heurística HAD também tem o melhor resultado, seguida pelas heurísticas HMP e HOMP.

Tabela 3.3: Heurísticas Construtivas - Instâncias do Grupo II.

	Métricas	Hassin	Breslau	HMO <sub>(1,0)</sub>	HMO <sub>(10,1)</sub>	HOMA	HOMP	HMP	HAD
Gulosa	<i>#Melhor</i>	1	1	0	0	0	5	25	<b>199</b>
	<i>MDif</i> (%)	6,88	6,89	9,25	34,49	74,41	5,62	2,34	<b>0,89</b>
	<i>Score</i>	1091	1097	1981	2505	2940	1070	351	<b>65</b>
	Tempo (s)	562,76	29,7	1,03	1,7	0,73	0,38	1,81	<b>0,16</b>
Rand.	<i>#Melhor</i>	-	-	4	0	0	3	307	<b>730</b>
	<i>MDif</i> (%)	-	-	59,14	58,21	74,41	49,14	3,28	<b>2,57</b>
	<i>Score</i>	-	-	1505	1422	2100	852	284	<b>134</b>
	Tempo (s)	-	-	1,43	1,97	0,78	0,67	2,52	<b>0,16</b>

Ao se considerar estes dois grupos de instâncias, Grupo I e Grupo II, é possível verificar que a heurística HOMA obteve os piores resultados, seguido pela heurística HMO. Os melhores resultados foram apresentados pelas heurísticas HMP e HAD.

### 3.5.2 Heurísticas de Busca Local

Nesta seção são apresentados os resultados das heurísticas de busca local executadas a partir de soluções obtidas por cada um dos algoritmos de construção. Neste experimento foram utilizadas apenas as versões randomizadas das heurísticas construtivas. Para cada instância considerada, cada heurística construtiva foi executada dez vezes, duas vezes com cada valor de  $\alpha$  considerado ( $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ ) e, após a construção da solução,



a mesma foi submetida às diferentes heurísticas de busca local.

Inicialmente foram utilizadas as instâncias do Grupo I. Dado que, para cada instância, uma solução trivial pode ser obtida incluindo-se todos os objetos no conjunto solução, o custo desta solução trivial é considerado para efeito de comparação entre as abordagens. Na Figura 3.3(a) é apresentada a média, para cada uma das abordagens, da razão entre as soluções obtidas e o custo da solução trivial.

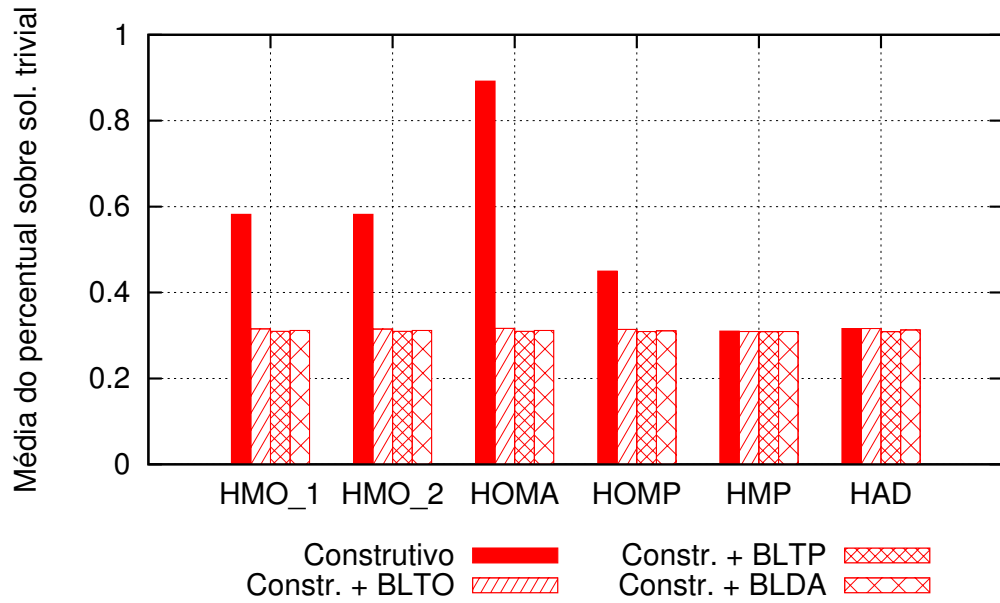
No gráfico da Figura 3.3(a) destaca-se a qualidade das soluções obtidas pelas heurísticas HMP e HAD, como foi verificado na seção anterior. Em média, o custo das soluções obtidas por estas abordagens corresponde a 30% do custo da solução trivial, ou seja, uma redução de 70%. Para este grupo de instâncias, a redução com a utilização dos algoritmos de busca local não alcançou 0,01%.

Para as demais heurísticas construtivas foi possível verificar a contribuição das heurísticas de busca local para o custo final das soluções. Mesmo partindo de soluções de qualidade ruim, as heurísticas de busca local foram capazes de reduzir os custos destas soluções para valores próximos aos obtidos por HMP e HAD.

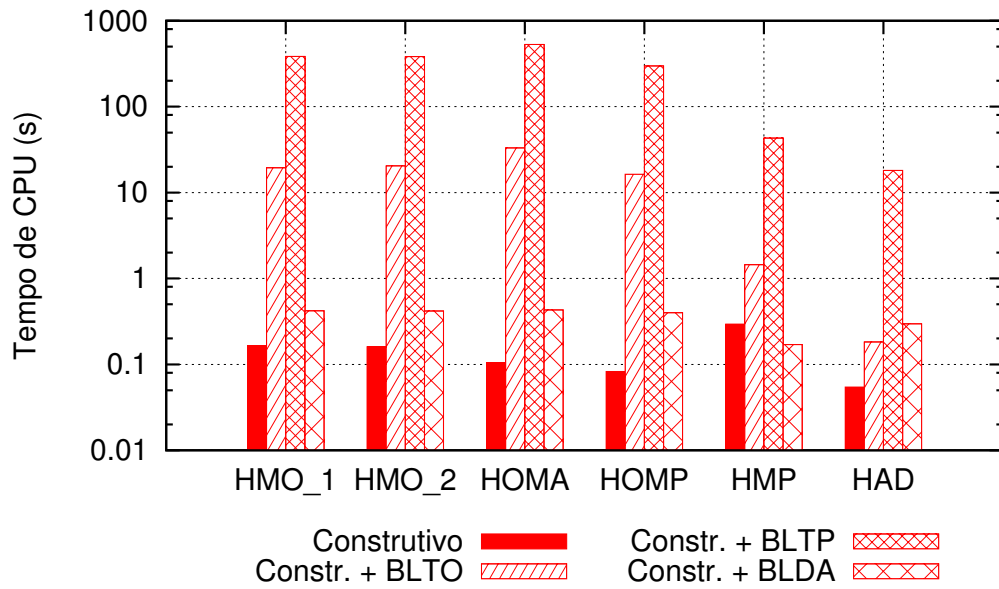
O gráfico da Figura 3.3(b) apresenta a média do tempo de CPU, em segundos, que cada abordagem consumiu. Para que fosse possível apresentar valores de diferentes ordens de grandeza, os resultados são apresentados em escala logarítmica. Neste gráfico, destaca-se o custo computacional da heurística de busca local BLTP. Esta abordagem gastou em média 275,81 segundos, chegando à média de 528,93 quando executada após HOMA.

Este mesmo experimento foi realizado para as instâncias do Grupo II, para o qual, os resultados são apresentados nas Figuras 3.4(a) e 3.4(b). Como pode ser observado, o mesmo padrão de comportamento apresentado para as instâncias do Grupo I pode ser verificado quanto ao custo das soluções. Comparando as Figuras 3.3(a) e 3.4(a), é possível verificar que mesmo atribuindo custos variados aos objetos, o comportamento dos algoritmos de construção não se alterou, e que os melhores resultados foram obtidos por HMP e HAD.

Como as instâncias do Grupo II são derivadas das instâncias do Grupo I com maior número de elementos e objetos, comparando-se as Figuras 3.3(b) e 3.4(b) é possível notar um aumento no tempo de execução tanto das heurísticas construtivas quanto nas heurísticas de busca local.

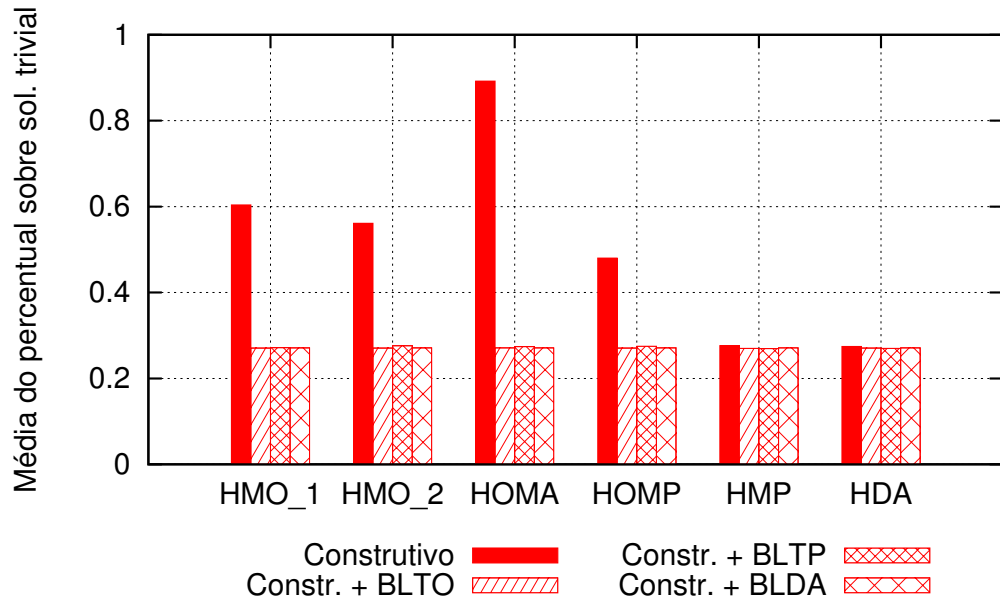


(a) Desvio em relação a solução trivial

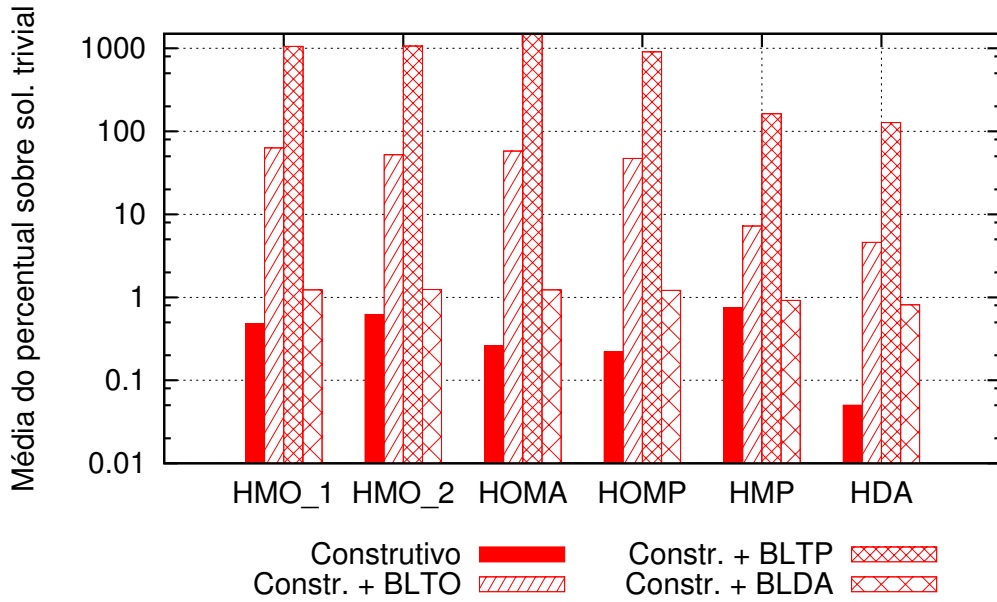


(b) Tempo médio de CPU (s)

Figura 3.3: Resultados comparativos para as instâncias do Grupo I em relação a qualidade da solução (a) e tempo de CPU (b).



(a) Desvio em relação a solução trivial



(b) Tempo médio de CPU (s)

Figura 3.4: Resultados comparativos para as instâncias do Grupo II em relação a qualidade da solução (a) e tempo de CPU (b).

# Capítulo 4

## Metaheurísticas

Para as instâncias propostas neste trabalho, combinaram-se os algoritmos apresentados no capítulo anterior com objetivo de melhorar a qualidade das soluções obtidas. Para tal, três abordagens foram avaliadas: as metaheurísticas *Greedy Randomized Adaptive Search Procedure* (GRASP) [52] e *Iterated local search* (ILS) [43], e uma abordagem híbrida que combina heurística e método exato.

### 4.1 *Greedy Randomized Adaptive Search Procedure* - GRASP

A metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP) é uma heurística iterativa do tipo *multistart* que alterna etapas de construção e busca local. Na etapa de construção, uma solução é determinada e sua vizinhança é investigada na etapa busca local.

As iterações GRASP são independentes, ou seja, na iteração atual não se considera nenhuma informação referente às iterações anteriores. O critério de parada normalmente utilizado é um número máximo de iterações. A melhor solução entre as soluções obtidas em cada uma das iterações é a solução final do GRASP.

O Algoritmo 8 ilustra a estrutura básica do GRASP para um problema de minimização, onde o parâmetro  $\alpha$  indica o grau de aleatoriedade do algoritmo de construção e *MaxIter* indica o número de iterações que são executadas.

Esta abordagem foi proposta no trabalho de Feo e Resende [18] e mostrou-se eficiente na solução de diversos problemas [20, 50, 45, 37, 51, 56, 19, 25].

Na implementação de GRASP proposta neste trabalho para o PCCP (GPCCP) foram

**Algoritmo 8** GRASP

---

```

1:  $S^* \leftarrow \infty$ ;
2: para  $k = 1, \dots, MaxIter$  faça
3:    $Sol \leftarrow$  Construção_gulosa_aleatória ( $\alpha$ );
4:    $Sol \leftarrow$  Busca_local ( $Sol$ );
5:   se ( $custo(Sol) < custo(S^*)$ ) então
6:      $S^* \leftarrow Sol$ ;
7:   fim se
8: fim para
9: Retornar ( $S^*$ );

```

---

utilizadas as heurísticas de construção e busca local apresentadas no Capítulo 3. No GPCCP optou-se por combinar duas extensões do modelo básico desta metaheurística, a utilização de várias heurísticas de construção e o GRASP Reativo [49].

Como uma característica desejável no GRASP é que, a cada iteração, soluções diferentes sejam construídas, neste trabalho optou-se por variar a heurística de construção. Desta forma, partindo de soluções com características diversas, é possível analisar regiões diferentes no espaço de soluções. Assim, um conjunto de algoritmos construtivos *SetC* será utilizado no GPCCP.

Além disto, como verificado no trabalho de Resende e Ribeiro [53], a escolha adequada do parâmetro  $\alpha$  é um fator relevante para determinar a qualidade das soluções obtidas pela heurística. Considerar um único valor para o parâmetro  $\alpha$  pode fazer com que o algoritmo deixe de encontrar uma boa solução, que poderia ser obtida se outro valor fosse associado a este parâmetro. Desta forma, neste trabalho é proposto um GRASP Reativo para o PCCP, onde o parâmetro  $\alpha$  não é fixo, mas selecionado entre um conjunto de valores pré-determinados.

O Algoritmo 9 apresenta o GPCCP, o algoritmo GRASP proposto neste trabalho para o Problema de Cobertura de Conjuntos por Pares.

Inicialmente, para cada algoritmo construtivo  $Alg \in SetC$ , uma iteração é executada (linhas 1 e 8). A solução obtida após a busca local terá seu custo incluído no vetor *Best*, onde cada posição está associada a um dos construtivos. Este vetor armazena o custo da melhor solução obtida com cada um dos construtivos e será utilizado, nas iterações posteriores, para determinar o construtivo que será utilizado. A melhor solução obtida durante as iterações iniciais é atribuída a  $S^*$ .

Após a etapa inicial, as demais iterações do GRASP são executadas (linhas 9 a 21). Na linha 10 é selecionado o algoritmo construtivo que será utilizado na iteração cor-

**Algoritmo 9** GSCPP (*MaxIter*)

---

```

1: para  $Alg \in SetC$  faça
2:    $S \leftarrow$  Construção_gulosa( $Alg$ );
3:    $S \leftarrow$  Busca_Local ( $S$ );
4:    $Best[Alg] \leftarrow$  custo( $S$ );
5:   se ( custo( $S$ ) < custo( $S^*$ ) ) então
6:      $S^* \leftarrow S$ ;
7:   fim se
8: fim para
9: para  $k = 1, \dots, MaxIter - |SetC|$  faça
10:   $Alg \leftarrow$  Selecionar_algoritmo_construtivo ( $SetC$ );
11:   $\alpha \leftarrow$  Escolher_Alfa ( $Alg$ );
12:   $S \leftarrow$  Construir_solução ( $Alg, \alpha$ );
13:   $S \leftarrow$  Busca_Local ( $S$ );
14:  Atualizar ( $Best[Alg]$ );
15:  se ( custo( $S$ ) < custo( $S^*$ ) ) então
16:     $S^* \leftarrow S$ ;
17:  fim se
18:  se (  $k$  for múltiplo de  $t$  ) então
19:    Ajustar_probabilidade ( $\alpha$ );
20:  fim se
21: fim para
22: Retornar ( $S^*$ );

```

---

rente. Para cada algoritmo construtivo  $i$  há uma chance de seleção proporcional igual à  $k_i = \sum_{j \in SetC} Best[j]/Best[i]$ . Para normalização, temos que as probabilidades de escolha de cada algoritmo  $i$ , denotada por  $PP_i$ , são dadas por  $PP_i = k_i / \sum_{j \in SetC} k_j$ . Assim, para seleção do algoritmo construtivo é realizada uma roleta onde cada algoritmo  $i$  tem chance de seleção igual a  $PP_i$ .

Após a seleção do algoritmo de construção será realizada a escolha do  $\alpha$  a ser utilizado (linha 11). Como descrito em [49], será considerado um conjunto  $T$  com  $m$  possíveis valores para o parâmetro  $\alpha$  onde, inicialmente, a probabilidade de selecionar cada valor é  $1/m$ .

Selecionados os parâmetros  $Alg$  e  $\alpha$ , na linha 12 uma solução é construída e sua viabilidade é analisada na linha 13. Caso esta solução seja melhor que a solução armazenada em  $Best[Alg]$ , o vetor é atualizado. Haverá atualização de  $S^*$  sempre que a solução  $S$  tiver custo inferior a  $S^*$ .

Periodicamente, a cada  $t$  iterações, as probabilidades associadas a cada um dos valores de  $\alpha$  são atualizadas de acordo com a qualidade das soluções obtidas até o momento (linhas 18 a 20). Considerando  $S^*$  a melhor solução conhecida até a iteração atual, a probabilidade é ajustada de acordo com a Equação (4.1), onde  $q_i = S^*/A_i$ , sendo  $A_i$  a

média das soluções obtidas utilizando  $\alpha = t_i$ .

$$p_i = q_i / \sum_{j=1}^m q_j \quad (4.1)$$

Na implementação avaliada neste trabalho utilizou-se  $m = 5$  e  $t = 10$ .

## 4.2 *Iterated Local Search* - ILS

O procedimento *Iterated Local Search* (ILS - Busca Local Iterada) é uma estratégia de busca onde a vizinhança de uma solução é analisada, aplicando-se perturbações sempre que a mesma se encontrar em um ótimo local.

Como descrito em [43], quatro componentes básicos são necessários para se implementar um ILS: (i) Procedimento para obtenção de uma solução inicial; (ii) Algoritmo de busca local; (iii) Uma estratégia de perturbação; e (iv) Critério de aceitação.

O Algoritmo 10 descreve a estrutura básica do ILS. Uma solução inicial é determinada pelo algoritmo construtivo (linha 1), e sua vizinhança é analisada na etapa seguinte (linha 2). Depois disto, enquanto o critério de terminação não for satisfeito, a cada iteração a solução  $S^*$  passa por uma perturbação (linha 4) e, em seguida, a vizinhança da nova solução é explorada pelo algoritmo de busca local (linhas 5). Caso o critério de aceitação seja satisfeito, a solução obtida após a etapa de busca local passa a ser a solução corrente (linha 6).

---

### Algoritmo 10 *Iterated Local Search*

---

- 1:  $S_0 = \text{Construtivo}(\ );$
  - 2:  $S^* = \text{Busca Local}(S_0);$
  - 3: **repita**
  - 4:  $S' \leftarrow \text{Perturbação}(S^*);$
  - 5:  $S'^* \leftarrow \text{Busca\_Local}(S');$
  - 6:  $S^* \leftarrow \text{Critério\_Aceitação}(S^*, S'^*);$
  - 7: **até** (critério de terminação ser satisfeito);
  - 8: Retornar ( $S^*$ );
- 

Uma característica fundamental para o bom funcionamento da ILS é a estratégia de perturbação. Caso a perturbação seja pequena, a busca local pode retornar a solução anterior a perturbação, fazendo com que poucas soluções sejam exploradas. Se a perturbação for muito grande, perde-se a relação com a solução anterior, e o algoritmo poderá se comportar como algoritmos do tipo *multistart*.

Na implementação avaliada neste trabalho, optou-se por utilizar a heurística de construção HAD (gulosa) para construção da solução inicial. Esta escolha levou em consideração a qualidade das soluções obtidas numa avaliação prévia dos algoritmos construtivos. Na etapa de busca local são utilizadas as heurísticas BLDA e BLTO, nesta ordem. Como critério de parada do algoritmo foi estabelecido um limite de iterações sem atualizar a melhor solução alcançada pelo ILS. Na estratégia de perturbação, a solução atual tem alguns dos seus elementos removidos, e a solução é reconstruída utilizando-se as heurísticas descritas no Capítulo 3 (exceto HAD).

### 4.3 Heurística Híbrida

Um algoritmo híbrido é aquele que combina ou integra duas ou mais abordagens. Desenvolver algoritmos híbridos envolvendo heurísticas é uma prática comum na busca de boas soluções para problemas de otimização [42, 57]. Recentemente, uma maior atenção vem sendo dada a integração entre métodos exatos e heurísticas [44], visto que esta integração vem apresentando resultados de boa qualidade. A combinação entre técnicas de programação matemática e metaheurísticas deu origem a uma nova classe de algoritmos, denominada *Matheuristics*.

Duas estratégias podem ser utilizadas para a integração de (meta)heurísticas com métodos exatos. A heurística pode funcionar como guia e controlar o procedimento exato, ou a abordagem exata aciona e controla a estratégia heurística. Neste trabalho, optou-se por tratar o método exato como uma intensificação realizada após o procedimento heurístico, fazendo-o funcionar como uma busca local. A estratégia de *Local Branching* [21, 22] pode ser vista como um exemplo de abordagem do tipo busca local.

Para o algoritmo proposto, considere *Pool* um conjunto formado por  $v$  soluções para o Problema de Cobertura de Conjuntos por Pares. Inicialmente as heurísticas serão utilizadas para preenchimento do *Pool* que, na etapa seguinte, será utilizado como base para indicar as variáveis que farão parte do modelo a ser resolvido pelo método exato.

Para determinar as soluções que irão compor o *Pool* serão utilizadas as heurísticas construtivas, de maneira semelhante à forma utilizada no GRASP. Ou seja, a etapa de busca local será removida, tornando o método exato responsável por esta etapa na segunda fase da Heurística Híbrida. O Algoritmo 11 ilustra o funcionamento desta abordagem.



**Algoritmo 11** Heurística Híbrida

---

```

1: para  $Alg \in SetC$  faça
2:    $S \leftarrow$  Construção_Gulosa ( $Alg$ );
3:   Best[ $Alg$ ]  $\leftarrow$  custo( $S$ );
4:   Atualizar ( $Pool$ );
5: fim para
6: para  $k = 1, \dots, MaxIter - |SetC|$  faça
7:    $Alg \leftarrow$  Selecionar_algoritmo_construtivo ( $SetC$ );
8:    $\alpha \leftarrow$  Escolher_Alfa ( $Alg$ );
9:    $S \leftarrow$  Construir_solução ( $Alg, \alpha$ );
10:  Atualizar_Best[ $Alg$ ];
11:  Atualizar ( $Pool$ );
12:  se ( $k$  for multiplo de  $t$ ) então
13:    Ajusta_probabilidade ( $\alpha$ );
14:  fim se
15: fim para
16: modelo  $\leftarrow$  Criar_modelo_simplificado ( $Pool$ );
17:  $S^* \leftarrow$  Resolvedor_exato (modelo);
18: Retornar ( $S^*$ );

```

---

A estrutura entre as linhas 1 e 15 é idêntica ao GRASP, sendo que a busca local foi removida e a atualização da melhor solução substituída pela atualização do *Pool* de soluções. Na atualização do conjunto de soluções são preservadas as  $v$  soluções de menor custo.

Na etapa seguinte, um modelo com número reduzido de variáveis é criado (linha 16). Nesta etapa optou-se por utilizar o modelo  $\mathcal{F}1^+$  que faz uso de variáveis estendidas (Seção 2.3), que obteve melhor desempenho em avaliações preliminares. No modelo reduzido, somente as variáveis que aparecem ativas em alguma das soluções do *Pool* serão inseridas no modelo. Ou seja, uma variável  $w_{i,j,e}$ , que apresenta o par de objetos  $i, j$  como uma opção para cobrir o elemento  $e$ , somente constará no modelo reduzido se os objetos  $i$  e  $j$  estiverem ambos presentes em alguma das soluções do *Pool*. O modelo reduzido não garante que a solução ótima será encontrada, mas determinará a melhor solução possível utilizando as colunas que aparecem nas soluções das heurísticas.

Na próxima seção são apresentados os resultados obtidos nos experimentos computacionais realizados com as metaheurísticas propostas.

## 4.4 Resultados Computacionais

As metaheurísticas propostas neste trabalho foram implementadas na linguagem C++, compiladas no g++ versão 4.4.1. Os algoritmos foram executados em uma máquina Intel(R) Core(TM) i7 CPU 870 com 2.93GHz e 8GB de memória RAM.

### 4.4.1 GRASP

Para a heurística GRASP proposta para o Problema de Cobertura de Conjuntos por Pares (GPCCP) é necessário definir o conjunto de heurísticas de construção ( $SetC$ ) e a estratégia de busca local utilizada. Nas versões avaliadas nesta tese um mesmo conjunto  $SetC$  será considerado e, quanto a estratégia de busca local, quatro diferentes configurações foram avaliadas.

A primeira configuração utiliza apenas BLDA na etapa de busca local, enquanto a segunda faz uso de BLTO. Com objetivo de verificar se BLTO consegue refinar a solução já aprimorada por BLDA, na terceira versão é utilizada BLDA seguida de BLTO. Uma vez que a heurística de busca local BLTP consome muito tempo de CPU, o que prejudica sua utilização em uma metaheurística como o GRASP, alguns testes foram realizados no sentido de avaliar o custo computacional de se utilizar tal abordagem. Para isso, uma quarta versão utilizando BLTP e BLTO é também testada. A Tabela 4.1 mostra as características de cada uma destas versões.

Tabela 4.1: Versões do GPCCP.

Versões	Conjunto de Construtivos	Estratégia de Busca Local
G-DA	$SetC = \{HOMP, HMP, HDA\}$	BLDA
G-TO		BLTO
G-DATO		BLDA + BLTO
G-TPTO		BLTP + BLTO

Testes iniciais foram realizados com objetivo de avaliar a qualidade das soluções e o tempo de CPU para cada uma das quatro versões. Para este experimento, um subconjunto de instâncias do Grupo II foi utilizado. Este subconjunto é composto por 15% das instâncias deste grupo, selecionadas aleatoriamente. Como critério de parada foi empiricamente definido número de iterações igual a 50.

Para cada instância, considera-se a média das soluções obtidas por cada versão do GRASP. Para uma dada instância, seja  $\bar{s}^*$  a menor média associada a uma dada versão. Para efeito de comparação entre as versões, na Tabela 4.2 é apresentada a média da

diferença percentual entre a média das soluções obtidas e o valor de  $\bar{s}^*$ , para todas as instâncias utilizadas no experimento. Além disso, nesta tabela também é apresentado o tempo médio de CPU gasto por cada uma das versões.

Tabela 4.2: Comparação entre quatro versões do GRASP para 15% das Instâncias do Grupo II.

Versões	Diferença para Melhor (%)	Tempo de CPU (s)
G-DA	0,14	108,42
G-TO	0,05	383,13
G-DATO	0,04	370,74
G-TPTO	0,00	12753,67

Nestes testes pode-se observar o alto custo computacional da busca local BLTP. Apesar da versão G-TPTO apresentar sempre as melhores médias, o tempo gasto é muito superior ao consumido pelas outras versões, que apresentam soluções onde a média dos custos fica a no máximo 0,14% da média das soluções obtidas por G-TPTO.

Em relação ao tempo de CPU, a abordagem que consumiu menos tempo foi a versão G-DA, mas que apresentou também a maior diferença em relação a melhor média.

Outra observação importante é a diferença entre o tempo consumido por G-TO e por G-DATO. Em G-DATO a busca local BLDA é executada antes de BLTO, melhorando o custo da solução inicial de BLTO. Desta forma, foi possível reduzir o tempo médio de CPU, mesmo incluindo um módulo ao algoritmo. Como pode ser verificado na Tabela 4.2, o tempo de CPU de G-TO é 3,27% superior ao consumido pela versão G-DATO. Os resultados individuais para cada uma das instâncias podem ser vistos nas Tabelas A.1 e A.2 (Apêndice A).

Como observado nos resultados apresentados, dado o elevado tempo computacional da busca local BLTP, nos próximos experimentos são avaliadas apenas as versões G-DA, G-TO e G-DATO.

Na Tabela 4.3 são apresentados os resultados para as métricas descritas no Capítulo 3 referentes a todas as instâncias dos Grupo II. Como pode ser observado, a versão G-DA possui maior valor associado a métrica *Score*, ou seja, é a versão que perde mais vezes para outras versões. Além disto, esta versão obteve menos vezes a melhor média (*#Melhor*) e possui maior valor associado à *MDif*.

Para este grupo de instâncias, as versões G-TO e G-DATO obtiveram resultados muito próximos em relação à qualidade média das soluções. Analisando as métricas *#Melhor*,

*MDif* e *Score* é possível verificar que estas duas versões pouco diferem. Apenas em relação ao tempo médio de CPU é possível notar uma diferença significativa. Em média, o tempo gasto pela G-TO é 17,36% superior ao consumido pela heurística G-DATO.

Tabela 4.3: Comparação entre três versões do GRASP - 420 Instâncias do Grupo II.

Versões GRASP	G-DA	G-TO	G-DATO
<i>#Melhor</i>	319	390	388
<i>MDif</i>	0.075%	0.012%	0.015%
<i>Score</i>	181	38	39
Tempo de CPU	155.94	643.03	547.90

Para uma análise do comportamento dos algoritmos em relação às características das instâncias, os resultados mostrados de forma condensada na Tabela 4.3 são apresentados na Tabela 4.4 separadamente para cada conjunto de instâncias que compõe o Grupo II. Cada grupo é composto por 60 instâncias. Nesta tabela é possível verificar que, para as instâncias com menor número de elementos (M8\_B8, M4\_B4 e M1\_B8), as três versões do GRASP tiveram desempenho muito semelhante, apresentando variação apenas quanto ao tempo de execução. Para os demais grupos, os resultados das versões G-TO e G-DATO se sobressaem em relação aos resultados obtidos pela versão G-DA.

Esta mesma análise é também realizada para as instâncias do Grupo III. Assim, na Tabela 4.5 são apresentados os resultados das três versões em relação a estas mesmas métricas. Para este grupo de instâncias é possível notar uma diferença maior entre os resultados verificados para as três versões analisadas. Para mais de 90% das 144 instâncias, a média das soluções obtidas pela metaheurística G-DATO foi igual a melhor média encontrada. Como o valor associado a métrica *MDif* para esta versão é muito baixo, se comparado com as demais versões, pode-se concluir que, para as instâncias onde a melhor média não é encontrada pela G-DATO, o valor de sua média é muito próximo ao valor da melhor média.

Quanto às outras duas versões, G-DA e G-TO, apesar destas abordagens alcançarem valores próximos em relação à métrica *#Melhor*, as métricas *Score* e *MDif* indicam que os resultados obtidos por G-TO são melhores que aqueles obtidos pela heurística G-DA. Entretanto, quanto ao tempo médio de CPU, G-TO chega a consumir em média quase quinze vezes mais tempo do que a versão G-DA. Assim, verifica-se que combinar as duas estratégias de busca local tornou possível a obtenção de soluções de melhor qualidade pela versão G-DATO em um tempo 50% inferior quando comparado ao tempo gasto por G-TO.

Na Tabela 4.6 são apresentados os resultados para cada um dos conjuntos de instâncias do Grupo III. A partir dos resultados apresentados nesta tabela é possível verificar que

Tabela 4.4: Comparação entre três versões do GRASP - 420 Instâncias do Grupo II por conjunto.

Grupo	Métricas	G-DA	G-TO	G-DATO
M1_B1	<i>#Melhor</i>	18	46	47
	<i>Score</i>	79	15	14
	<i>MDif</i>	0,284%	0,038%	0,031%
	Tempo de CPU	556,25	2683,90	2382,05
M1_B2	<i>#Melhor</i>	37	52	53
	<i>Score</i>	40	12	10
	<i>MDif</i>	0,119%	0,029%	0,021%
	Tempo de CPU	267,99	1018,72	817,65
M1_B4	<i>#Melhor</i>	48	59	56
	<i>Score</i>	21	1	5
	<i>MDif</i>	0,051%	0,000%	0,020%
	Tempo de CPU	134,74	392,37	280,13
M1_B8	<i>#Melhor</i>	58	60	59
	<i>Score</i>	3	0	1
	<i>MDif</i>	0,004%	0,000%	0,003%
	Tempo de CPU	59,35	144,72	93,03
M2_B2	<i>#Melhor</i>	43	56	54
	<i>Score</i>	29	6	7
	<i>MDif</i>	0,050%	0,010%	0,017%
	Tempo de CPU	64,38	237,82	236,62
M4_B4	<i>#Melhor</i>	58	60	59
	<i>Score</i>	3	0	2
	<i>MDif</i>	0,009%	0,000%	0,011%
	Tempo de CPU	7,75	22,02	23,78
M8_B8	<i>#Melhor</i>	57	57	60
	<i>Score</i>	6	4	0
	<i>MDif</i>	0,009%	0,005%	0,000%
	Tempo de CPU	1,10	1,66	2,02

Tabela 4.5: Comparação entre três versões do GRASP - Instâncias do Grupo III.

Versões GRASP	G-DA	G-TO	G-DATO
<i>#Melhor</i>	18	20	139
<i>Score</i>	250	139	10
<i>MDif</i>	3,529%	1,631%	0,192%
Tempo de CPU	235,53	3519,16	1683,08

para os três grupos, onde varia-se o número de pares nas instâncias, G-DATO obtém sempre os melhores resultados associados às métricas  $\#Melhor$ ,  $Score$  e  $MDif$ . Quanto ao tempo de CPU, novamente o tempo consumido por esta abordagem é superior ao consumido por G-DA e inferior ao tempo médio de G-TO. Para este conjunto de instâncias, as soluções obtidas por G-DATO apresentaram em média custos menores do que as obtidas por G-DA.

Tabela 4.6: Comparação entre três versões do GRASP - Instâncias do Grupo III por conjunto.

Grupo	Métricas	G-DA	G-TO	G-DATO
SCP_25	$\#Melhor$	7	4	45
	$Score$	76	50	3
	$MDif$	2,969%	1,945%	0,217%
	Tempo de CPU	300,08	4195,50	1700,72
SCP_50	$\#Melhor$	4	6	43
	$Score$	85	45	5
	$MDif$	3,289%	1,499%	0,096%
	Tempo de CPU	199,02	3323,18	1475,70
SCP_75	$\#Melhor$	7	10	51
	$Score$	89	44	2
	$MDif$	4,698%	1,618%	0,227%
	Tempo de CPU	232,03	3405,39	1555,28

Como o tempo de CPU consumido pelas versões do GRASP são muito diferentes, foi realizado um experimento onde o critério de parada foi a obtenção de uma solução com um custo menor ou igual a um valor previamente definido, denominado valor alvo. Esta análise de desempenho empírico foi proposta por Aiex et al. [2]. Para este teste, cada algoritmo foi executado 100 vezes para cada instância avaliada usando os mesmos parâmetros inicialmente adotados nas análises anteriores. Após as 100 execuções, os tempos foram tomados e dispostos em ordem crescente numa lista  $L$ . A cada tempo de CPU  $t_i$  obtido, foi associada a probabilidade  $p_i = (i - 0.5)/100$ , onde  $i$  é a ordem que  $t$  aparece na lista ordenada  $L$ . A plotagem foi feita então tomando-se cada ponto  $(t_i, p_i)$ . Um tempo limite foi estabelecido para cada instância analisada. Caso o alvo não fosse alcançado neste tempo, foi considerado que, nesta execução, o algoritmo não foi capaz de obter uma solução com custo igual ou inferior ao alvo.

Para este experimento foram selecionadas duas instâncias do Grupo III. Uma instância do conjunto SCP\_25 e uma de SCP\_75. Do conjunto SCP\_25 foi selecionada a instância scp410\_25, sendo definido como alvo a média obtida pela versão G-TO, maior que a média obtida pela versão G-DATO, mas menor que a média obtida por G-DA. A segunda instância selecionada foi scpclr10\_75. Para esta segunda instância, onde os métodos

exatos não conseguiram determinar o custo da solução ótima, o limite estabelecido foi o limite superior (*upper bound*) indicado pelo método exato obtido utilizando-se o Modelo de Gusfield Aprimorado após dez horas de processamento.

Para a instância `scp410_25` todas as versões conseguiram obter uma solução com qualidade igual ou melhor que o valor alvo estabelecido, em todas as execuções, em menos de 2000 segundos, como mostra na Figura 4.1. Para a abordagem G-DATO, a execução que demorou mais tempo para determinar a solução alvo gastou 98,4 segundos. Neste tempo, 45% das execuções de G-DA e apenas 2 execuções do G-TO alcançaram o valor alvo. Comparando-se G-TO e G-DATO, é possível verificar que, enquanto a primeira alcança o alvo em 100% das execuções em 736 segundos, a segunda só o faz para aproximadamente 55% das execuções neste mesmo tempo.

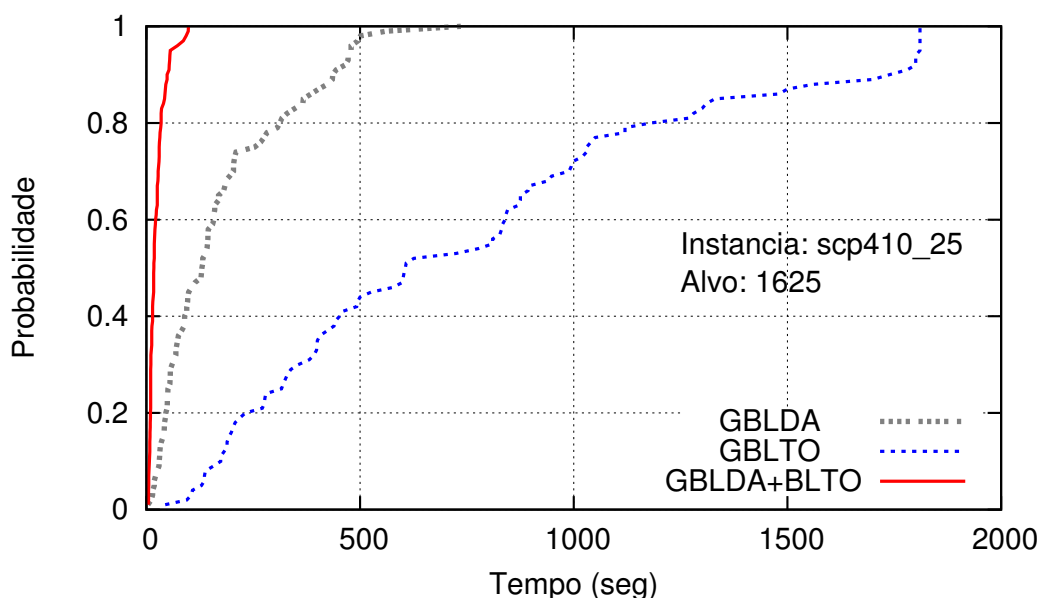


Figura 4.1: Análise probabilística para a instância `scp410_25`.

Para a instância `scplr10_75`, apenas a versão G-DATO foi capaz, em todas as execuções, de obter uma solução com custo igual ou menor que o valor do alvo no tempo limite estabelecido. A versão G-TO só conseguiu em 65% das execuções, enquanto G-DA em apenas 28%. Neste experimento, enquanto em 290 segundos há probabilidade de 90% da versão G-DATO alcançar o alvo, neste mesmo tempo a probabilidade para a versão G-TO é de 38% e para a G-DA é de 18%. Estes resultados confirmam a diferença de desempenho entre estas abordagens.

Vale ressaltar que em 24% das execuções realizadas para este experimento a versão que utiliza as duas estratégias de busca local foi capaz de obter soluções com custo inferior ao melhor limite alcançado pelo método exato quando executado por dez horas. Para as

outras versões, este percentual foi de 10% e 2%, para G-DA e G-TO, respectivamente.

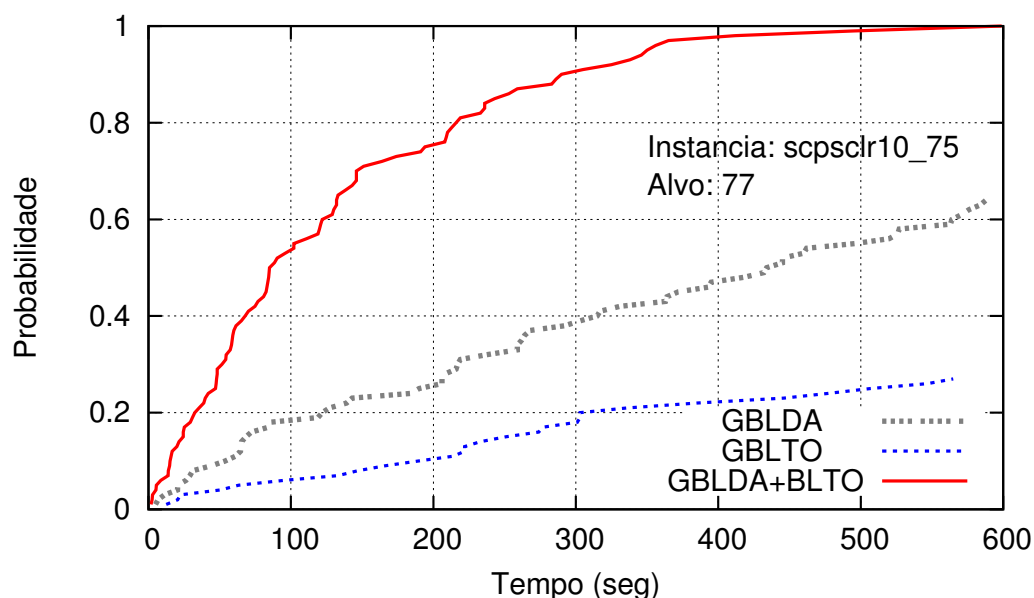


Figura 4.2: Análise probabilística para a instância scpsclr10\_75.

#### 4.4.2 ILS

O algoritmo ILS proposto neste trabalho para o PCCP foi implementado na mesma plataforma que o GRASP foi desenvolvido. Além disso, o ambiente de execução também permanece o mesmo. Como critério de parada, foi estabelecido o valor máximo de 10 iterações sem melhora e em cada perturbação 30% dos objetos da solução são removidos. O algoritmo foi executado 10 vezes para cada instância e os resultados são apresentados a seguir.

Nos testes realizados para a avaliação da heurística *Iterated Local Search* (ILS) foram utilizadas as instâncias dos Grupos II e III. Como entre as versões do GRASP testadas, G-DATO foi a que apresentou os melhores resultados, e pelo fato da ILS utilizar as mesmas estratégias de busca empregadas nesta abordagem (BLDA e BLTO), os resultados computacionais dos testes realizados com a ILS são comparados com aqueles obtidos por G-DATO. A Tabela 4.7 apresenta os resultados comparativos entre estas duas abordagens para as instâncias do Grupo II.

Pode-se observar pela métrica *Score* que as duas abordagens obtiveram resultados médios diferentes em 172 das 420 instâncias. Observa-se, entretanto, que o tempo médio de CPU do algoritmo GRASP é quase dez vezes maior que o tempo consumido pela heurística ILS. Em apenas uma instância, a média das soluções obtidas por ILS foi menor



Tabela 4.7: Comparação entre ILS e G-DATO - Instâncias do Grupo II.

Métricas	ILS	G-DATO
<i>#Melhor</i>	249	419
<i>Score</i>	171	1
<i>MDif</i>	0,308%	0,000%
Tempo de CPU	59,72	547,90

que a média apresentada por G-DATO. Para esta instância, o resultado médio do GRASP foi 0,15% superior ao resultado médio obtido pelo ILS, fazendo a média da diferença percentual em relação a melhor solução do GRASP (*MDif*) igual a 0,0004%.

Da mesma forma como apresentado para as diferentes versões do GRASP, na Tabela 4.8 são apresentados os resultados do ILS e do G-DATO para cada um dos conjuntos de instâncias do Grupo II. Nesta tabela é possível identificar que a instância onde o ILS possui média menor que a obtida pelo GRASP pertence ao conjunto das instâncias com maior número de elementos e objetos (M1\_B1).

Embora G-DATO tenha obtido maiores valores para a métrica *#Melhor*, o tempo médio apresentado pelo ILS quando comparado ao tempo consumido pelo GRASP indica que os valores obtidos para esta métrica pelo ILS são competitivos. Para vários conjuntos de instâncias, observa-se que o tempo médio que o GRASP demanda é pelo menos uma ordem de grandeza acima do tempo do ILS e que, ainda assim, o ILS foi capaz de obter uma solução igual à melhor apresentada pelo GRASP para mais da metade das instâncias do conjunto. Além disso, os valores apresentados pelo ILS para a métrica *MDif* mostram que, independente do conjunto de instâncias, os custos médios das soluções obtidas por este algoritmo estão a no máximo 0,6% das soluções obtidas pelo GRASP.

Para as instâncias do Grupo III, na Tabela 4.9 são apresentados os resultados para ILS e G-DATO. Pode-se observar pela métrica *Score* que o G-DATO não se apresenta como a melhor abordagem de forma tão expressiva como verificado nos testes com as instâncias do Grupo II. Enquanto o ILS foi capaz de encontrar solução média melhor que a apresentada pelo GRASP em apenas uma das 420 instâncias do Grupo II, observa-se que para 33 das 144 instâncias do Grupo III o custo médio das soluções obtidas pelo GRASP foi maior que o obtido pelo ILS. Mesmo não sendo tão expressiva, pode-se observar que há uma superioridade de G-DATO em relação à ILS no que se refere às métricas *#Melhor*, *Score* e *Mdif*.

Embora o tempo requerido pelo GRASP neste experimento, a exemplo do observado para as instâncias do Grupo II, tenha sido superior ao do ILS, pode-se verificar uma menor diferença entre os tempos das duas abordagens.

Tabela 4.8: Comparação entre ILS e G-DATO - Instâncias do Grupo II por conjunto.

Grupo	Métricas	ILS	G-DATO
M1_B1	<i>#Melhor</i>	16	59
	<i>Score</i>	44	1
	<i>MDif</i>	0,563%	0,003%
	Tempo de CPU	280,42	2382,05
M1_B2	<i>#Melhor</i>	23	60
	<i>Score</i>	37	0
	<i>MDif</i>	0,520%	0,000%
	Tempo de CPU	80,85	817,65
M1_B4	<i>#Melhor</i>	36	60
	<i>Score</i>	24	0
	<i>MDif</i>	0,291%	0,000%
	Tempo de CPU	22,47	280,13
M1_B8	<i>#Melhor</i>	53	60
	<i>Score</i>	7	0
	<i>MDif</i>	0,114%	0,000%
	Tempo de CPU	5,41	93,03
M2_B2	<i>#Melhor</i>	27	60
	<i>Score</i>	33	0
	<i>MDif</i>	0,366%	0,000%
	Tempo Médio	27,03	236,62
M4_B4	<i>#Melhor</i>	44	60
	<i>Score</i>	16	0
	<i>MDif</i>	0,167%	0,000%
	Tempo Médio	1,76	23,78
M8_B8	<i>#Melhor</i>	50	60
	<i>Score</i>	10	0
	<i>MDif</i>	0,137%	0,000%
	Tempo de CPU	0,07	2,02

Tabela 4.9: Comparação entre ILS e G-DATO - Instâncias do Grupo III.

Métricas	ILS	G-DATO
<i>#Melhor</i>	41	111
<i>Score</i>	103	33
<i>MDif</i>	1,563%	0,150%
Tempo de CPU	566,32	1618,49

Na comparação das abordagens conforme os conjuntos de instâncias que compõem o Grupo III, observa-se que o padrão de resultados observado para todo o grupo foi praticamente mantido para cada conjunto, o que demonstra que o desempenho das abordagens testadas não foi influenciado pelas características de cada conjunto.

Tabela 4.10: Comparação entre ILS e G-DATO - Instâncias do Grupo III por conjunto.

Grupo	Métricas	ILS	G-DATO
SCP_25	<i>#Melhor</i>	13	39
	<i>Score</i>	35	9
	<i>MDif</i>	1,483%	0,119%
	Tempo de CPU	913,36	1877,87
SCP_50	<i>#Melhor</i>	14	35
	<i>Score</i>	34	13
	<i>MDif</i>	1,920%	0,109%
	Tempo de CPU	523,99	1629,42
SCP_75	<i>#Melhor</i>	14	37
	<i>Score</i>	34	11
	<i>MDif</i>	1,287%	0,222%
	Tempo de CPU	261,60	1348,19

Como foi possível verificar a partir das Tabelas 4.8 e 4.10, os valores apresentados pelas duas abordagens comparadas, quanto à diferença percentual média em relação à melhor solução, indicada pela métrica *MDif*, mostraram uma superioridade de G-DATO em relação a ILS. Entretanto, pelos tempos demandados pelos dois algoritmos, experimentos complementares foram realizados para uma melhor comparação entre os algoritmos.

Com o objetivo de avaliar o tempo necessário para as abordagens convergirem para uma boa solução, no próximo experimento os algoritmos foram executados tendo um dado valor para a solução alvo como critério de parada. Como apresentado nos testes comparativos das três versões de GRASP apresentadas na Subseção 4.4.1, os gráficos apresentados nas Figuras 4.3 e 4.4 mostram o comportamento do G-DATO e da ILS quando se estabelece um valor alvo para as instâncias *scp410\_25* e *scpsclr10\_75*, respectivamente.

Observa-se que, para a instância *scp410\_25*, a probabilidade de que ILS obtenha uma solução igual ao alvo em 20 segundos de processamento é cerca de 90%, enquanto neste mesmo tempo, a chance de G-DATO encontrar uma solução com este custo é de apenas 60%. Com menos de 40 segundos de processamento, pode-se garantir que ILS obteve uma solução com custo igual ou menor ao valor alvo. A abordagem GRASP requer praticamente duas vezes e meia o tempo do ILS para que se garanta, em termos probabilísticos, que uma solução com custo máximo igual ao alvo seja obtida.

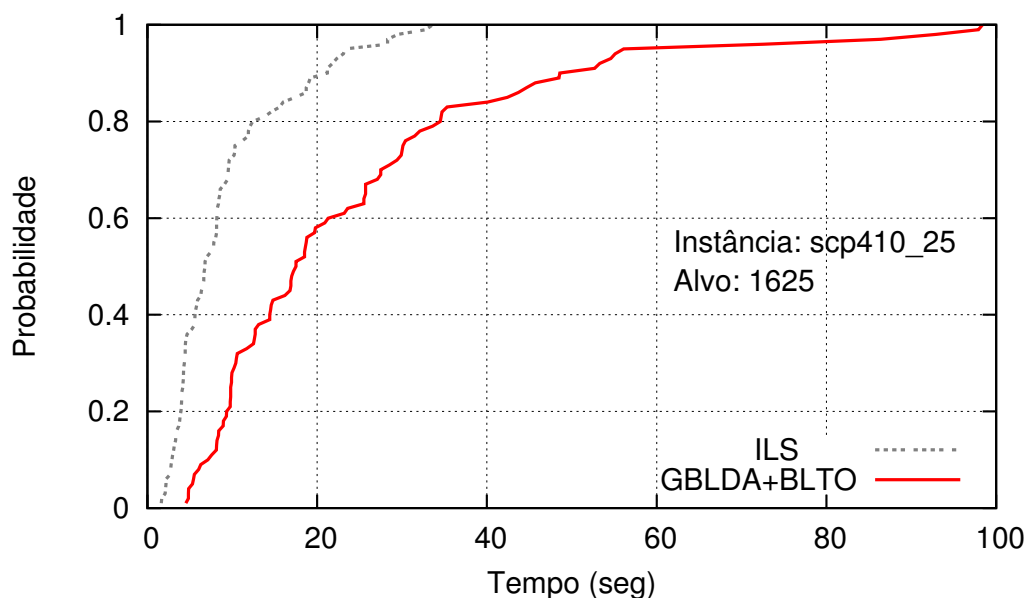


Figura 4.3: Análise probabilística para a instância scp410\_25.

Já para a instância scpsclr10\_75, para a qual o alvo estabelecido foi definido pelo custo obtido pela abordagem exata com tempo limite de CPU de dez horas, observa-se a probabilidade de que uma solução com custo igual a este alvo seja obtida em menos de 50 segundos é de 80% para o ILS e, neste mesmo tempo, a probabilidade é de apenas 20% para o GRASP. Enquanto o GRASP requer até 600 segundos para que se verifique probabilidade de 100% da solução alvo ser encontrada, o ILS requer menos de um terço deste tempo.

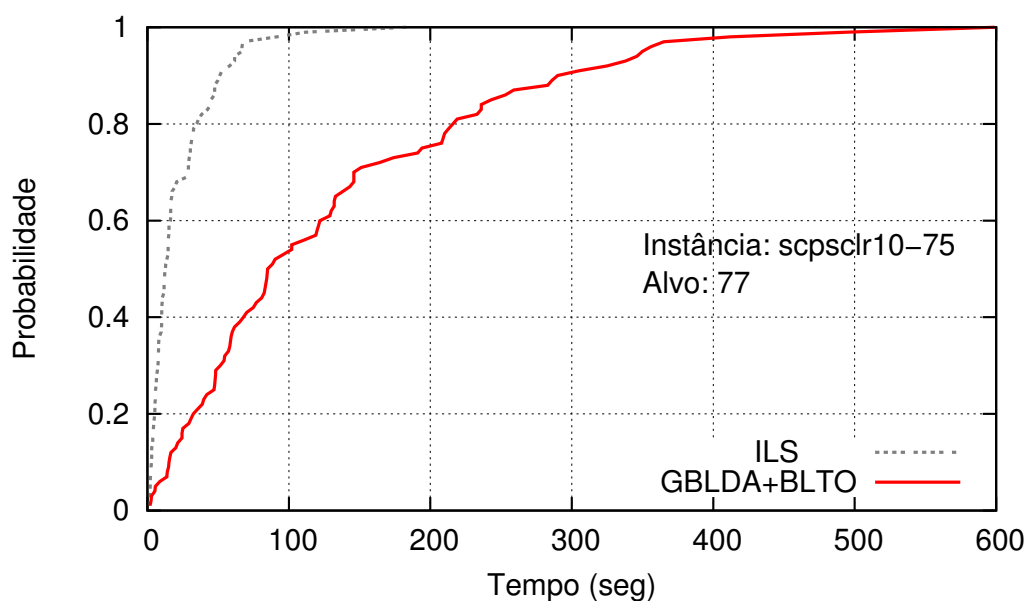


Figura 4.4: Análise probabilística para a instância scpsclr10\_75.

Dado o comportamento destas abordagens nesta análise por alvo, um novo experimento foi realizado. Nestes testes, o algoritmo ILS foi executado tendo como critério de parada o tempo limite dado pelo tempo médio consumido por G-DATO para cada instância. Neste teste são utilizadas as instâncias do Grupo III e os resultados são apresentados na Tabela 4.11.

Tendo como critério de parada o tempo de execução do GRASP, os resultados apresentados na tabela mostram que o ILS foi capaz de continuar aprimorando as soluções de forma que a média dos custos das suas soluções foram menores que as médias do GRASP para a maioria das instâncias. Nas três métricas relacionadas à qualidade das soluções é possível observar que o ILS obteve resultados melhores que os obtidos pelo GRASP.

Tabela 4.11: Comparação entre ILS por tempo e G-DATO - Instâncias do Grupo III.

Métricas	G-DATO	ILS por tempo
<i>#Melhor</i>	22	133
<i>Score</i>	122	11
<i>MDif</i>	1,561%	0,054%
Tempo de CPU	1618,49	1630,78

Na Tabela 4.12 este mesmo conjunto de resultados é apresentado para cada conjunto de instâncias do Grupo III. Observa-se na tabela que os resultados são homogêneos para os três conjuntos de instâncias. Em todos os conjuntos, o ILS obteve a melhor média dos custos das soluções para um maior número de instâncias, sendo que o GRASP ficou em média a 1,5% acima da média obtida pelo ILS.

Tabela 4.12: Comparação entre ILS por tempo e G-DATO - Instâncias do Grupo III por conjunto.

Conjunto	Métricas	G-DATO	ILS por tempo
SCP_25	<i>#Melhor</i>	5	44
	<i>Score</i>	43	4
	<i>MDif</i>	1,612%	0,071%
	Tempo de CPU	1877,87	1896,77
SCP_50	<i>#Melhor</i>	7	44
	<i>Score</i>	41	4
	<i>MDif</i>	1,435%	0,021%
	Tempo de CPU	1629,42	1638,34
SCP_75	<i>#Melhor</i>	10	45
	<i>Score</i>	38	3
	<i>MDif</i>	1,637%	0,071%
	Tempo de CPU	1348,19	1357,23

Para verificar a diferença no desempenho do ILS quando executado com cada um dos critérios de parada, na Tabela 4.13 são apresentados os resultados para o ILS utilizando

como critério de parada o número de iterações sem melhora e por tempo limite de execução. A diferença entre as métricas  $MDif$  indicam que a heurística ILS, quando executada por mais tempo, foi capaz de reduzir o custo das soluções em 2,95%, em média.

Tabela 4.13: Comparação ILS com diferentes critérios de parada - Instâncias do Grupo III.

Métricas	ILS	ILS por tempo
$\#Melhor$	13	137
$Score$	131	7
$MDif$	2,976%	0,026%
Tempo Médio	566,32	1630,78

### 4.4.3 Heurística Híbrida

Mesmo destacando a viabilidade na obtenção da solução ótima através da utilização do modelo  $\mathcal{F}1^+$  para as instâncias do Grupo II, na comparação da Heurística Híbrida com os demais métodos, além das instâncias do Grupo III, foram utilizadas as instâncias do Grupo II. Nos testes realizados foi estabelecido um tempo limite de execução para o resolvidor de PLIM, o mesmo limite de dez horas utilizado na avaliação dos modelos exatos. O conjunto de heurísticas construtivas utilizado no algoritmo foi  $SetC = \{HOMP, HMP, HDA\}$ . O número máximo de iterações das heurísticas construtivas e o conjunto de valores de  $\alpha$  são os mesmos utilizados nas versões do GRASP avaliadas anteriormente.

Na Tabela 4.14 são apresentados os resultados computacionais para os testes realizados utilizando-se as 420 instâncias do Grupo II. Nesta tabela, além das métricas já descritas, é apresentada também a métrica  $MDif^*$ , que indica a média da diferença percentual entre a solução ótima de cada instância e a média obtida pelos algoritmos nas diversas execuções. Nesta tabela considera-se a execução do ILS tendo como critério de parada o número máximo de 10 iterações sem melhora.

Tabela 4.14: Comparação entre ILS, G-DATO e Híbrido - Instâncias do Grupo II.

Métricas	ILS	G-DATO	Híbrido
$\#Melhor$	247	402	329
$Score$	338	18	101
$MDif$	0,314%	0,006%	0,063%
$MDif^*$	0,345%	0,037%	0,094%
Tempo de CPU	59,72	547,90	14,51

Para este grupo de instâncias, o algoritmo híbrido não alcançou o limite de tempo de execução em nenhum caso, ou seja, o resolvidor conseguiu obter a solução ótima para os modelos heurísticos (modelo reduzido) em todas as execuções. Entretanto, como a

solução do modelo depende das soluções obtidas pelas heurísticas, para as instâncias do Grupo II, o método híbrido foi capaz de obter a solução ótima conhecida (considerando o modelo  $\mathcal{F}1^+$  completo) para 307 instâncias. No geral, como apresentado pela métrica  $MDif^*$ , a média das soluções obtidas pela heurística híbrida ficou 0,094% acima do valor ótimo conhecido. Neste caso, destaca-se a diferença em relação do tempo médio de CPU entre as duas abordagens, uma vez que o GRASP consome, em média, 37 vezes mais tempo que a abordagem híbrida.

Enquanto o modelo  $\mathcal{F}1^+$ , quando submetido ao resolvidor CPLEX, demorou em média 44,25 segundos para apontar a solução ótima das instâncias do Grupo II, como apresentado na Tabela 2.5 (página 24), a heurística híbrida conseguiu obter uma solução com custo em média 0,094% acima do custo da solução ótima em 14,51 segundos. A abordagem GRASP precisou de 547,9 segundos para obter soluções com custo médio 0,037% acima da média das soluções ótimas neste grupo de instâncias.

Na Tabela 4.15 são apresentados os resultados para cada subconjunto de instâncias do Grupo II. Observando o tempo médio de CPU consumido para as instâncias dos conjuntos M1\_B8, M4\_B4 e M8\_B8, é possível verificar que, para as instâncias com menor número de objetos e elementos, a obtenção da solução ótima através do modelo completo requer menos tempo do que a execução da heurística híbrida. Isso se deve ao tempo gasto na etapa onde o *Pool* de soluções é preenchido, executada antes da resolução do modelo reduzido. Para os conjuntos formados por instâncias maiores (M1\_B1, M1\_B2 e M1\_B4) é possível notar uma redução significativa no tempo de execução da heurística híbrida, se comparado com o tempo necessário quando utilizado o modelo completo.

Para as instâncias do Grupo II, o tempo gasto para determinar a solução ótima utilizando o modelo matemático é pequeno. Como esperado, o tempo gasto na solução do modelo heurístico (modelo reduzido) é ainda menor. O que onera a execução desta abordagem para as instâncias menores é a execução das heurísticas de construção. Ainda assim, apesar do grande número de instâncias deste conjunto onde as abordagens obtêm, em média, os mesmos resultados, como pode ser observado através da métrica  $\#Melhor$ , a métrica  $MDif$  indica que, em média, os custos das soluções obtidas pela heurística híbrida são inferiores aos custos das soluções do ILS.

Para as instâncias do Grupo III, o resumo dos resultados é apresentado na Tabela 4.16. Em 11 das 144 instâncias deste grupo, a execução da heurística híbrida foi interrompida devido ao limite de tempo de execução estabelecido. Por este motivo, a média do tempo de CPU desta abordagem foi de 2786,24. Vale ressaltar que nestas 11 instâncias, a

Tabela 4.15: Comparação entre ILS, G-DATO e Híbrido - Instâncias do Grupo II por conjunto.

Conjuntos	Métricas	ILS	G-DATO	Híbrido
M1_B1	<i>#Melhor</i>	15	52	23
	<i>Score</i>	87	8	43
	<i>MDif</i>	0,576%	0,016%	0,191%
	Tempo de CPU	280,42	2382,05	54,57
M1_B2	<i>#Melhor</i>	23	56	39
	<i>Score</i>	73	4	24
	<i>MDif</i>	0,533%	0,013%	0,076%
	Tempo de CPU	80,85	817,65	23,72
M1_B4	<i>#Melhor</i>	36	58	52
	<i>Score</i>	46	2	9
	<i>MDif</i>	0,296%	0,004%	0,080%
	Tempo de CPU	22,47	280,13	11,20
M1_B8	<i>#Melhor</i>	53	60	57
	<i>Score</i>	14	0	3
	<i>MDif</i>	0,114%	0,000%	0,010%
	Tempo Médio	5,41	93,03	4,92
M2_B2	<i>#Melhor</i>	26	57	45
	<i>Score</i>	66	3	15
	<i>MDif</i>	0,371%	0,005%	0,053%
	Tempo de CPU	27,03	236,62	6,09
M4_B4	<i>#Melhor</i>	44	59	59
	<i>Score</i>	32	1	1
	<i>MDif</i>	0,171%	0,004%	0,004%
	Tempo de CPU	1,76	23,78	0,90
M8_B8	<i>#Melhor</i>	50	60	54
	<i>Score</i>	20	0	6
	<i>MDif</i>	0,137%	0,000%	0,025%
	Tempo de CPU	0,07	2,02	0,15



solução ótima não é conhecida. Além disso, para estas instâncias as soluções obtidas pela heurística híbrida possuem custo inferior ao limite superior (*upper bound*) obtido pela utilização da formulação  $\mathcal{F}1^+$  no mesmo limite de tempo. Se forem consideradas apenas as instâncias onde o limite de tempo foi suficiente para o resolvidor indicar a solução ótima, o tempo médio de CPU da abordagem híbrida reduz para 28,06 segundos, enquanto os tempos médios de ILS e GRASP passam a ser 378,15 e 1618,11 segundos, respectivamente.

Na Tabela 4.16 pode-se verificar que o GRASP obtém em média os melhores resultados, se comparados com as outras duas abordagens.

Tabela 4.16: Comparação entre ILS, G-DATO e Híbrido - Instâncias do Grupo III.

Métricas	ILS	G-DATO	Híbrido
<i>#Melhor</i>	35	94	34
<i>Score</i>	172	60	175
<i>MDif</i>	1,723%	0,309%	2,214%
Tempo de CPU	566,32	1618,49	2786,24

A Tabela 4.17 apresenta os resultados dos três algoritmos para cada conjunto de instâncias do Grupo III. Pode-se observar nos três conjuntos de instâncias que compõem o grupo que, para as três métricas relacionadas à qualidade das soluções, a abordagem GRASP apresentou os melhores resultados, embora o tempo médio requerido por esta abordagem seja muito superior ao do ILS. Como já indicado, o tempo médio apresentado pela abordagem híbrida foi influenciado pelos tempos das execuções onde não foi possível provar a otimalidade das soluções.

Comparando-se o comportamento da abordagem ILS com a híbrida para os três conjuntos de instâncias, os valores de *MDif* apresentados por ILS mostram que, mesmo com um tempo médio menor, ILS apresenta soluções com menor custo médio que aquelas apresentadas pela abordagem híbrida, embora ILS tenha apresentado valores de *Score* próximos aos da abordagem híbrida.

Para este conjunto de instâncias, os algoritmos GRASP e Híbrido apresentaram tempos médios de CPU maiores que os obtidos por ILS. Para uma comparação mais justa, na Tabela 4.18 os resultados obtidos pelo GRASP e pelo algoritmo Híbrido são comparados com ILS utilizando como critério de parada o tempo do GRASP. Na Tabela 4.19 são apresentados os resultados apenas para as instâncias onde o tempo limite não foi extrapolado pela heurística híbrida.

Tabela 4.17: Comparação entre ILS, G-DATO e Híbrido - Instâncias do Grupo III por conjunto.

Grupo	Métricas	ILS	G-DATO	Híbrido
SCP_25	<i>#Melhor</i>	10	31	15
	<i>Score</i>	57	21	56
	<i>MDif</i>	1,773%	0,409%	2,499%
	Tempo de CPU	913,36	1877,87	5298,42
SCP_50	<i>#Melhor</i>	11	30	9
	<i>Score</i>	58	21	62
	<i>MDif</i>	2,028%	0,217%	2,486%
	Tempo de CPU	523,99	1629,42	1528,24
SCP_75	<i>#Melhor</i>	14	33	10
	<i>Score</i>	57	18	57
	<i>MDif</i>	1,366%	0,301%	1,657%
	Tempo de CPU	261,60	1348,19	1532,07

Tabela 4.18: Comparação entre ILS por tempo, G-DATO e Híbrido - 144 Instâncias do Grupo III.

Métricas	ILS por tempo	G-DATO	Híbrido
<i>#Melhor</i>	122	22	19
<i>Score</i>	23	149	229
<i>MDif</i>	0,108%	1,616%	3,545%
Tempo de CPU	1630,78	1618,49	2786,24

Tabela 4.19: Comparação entre ILS por tempo, G-DATO e Híbrido - 133 Instâncias do Grupo III.

Métricas	ILS por tempo	G-DATO	Híbrido
<i>#Melhor</i>	115	20	16
<i>Score</i>	19	135	216
<i>MDif</i>	0.084%	1.606%	3.157%
Tempo de CPU	1629.39	1618.11	28.06

Nestas tabelas é possível verificar que o algoritmo ILS obteve, em ambos os casos, a maior quantidade de melhores soluções. Comparando-se o tempo médio de execução do algoritmo híbrido nas duas tabelas, observa-se o impacto na média devido às instâncias onde o tempo limite é esgotado pelo resolvidor exato.

Na Tabela A.8 do Apêndice A são apresentados os resultados das quatro abordagens para cada uma das instâncias do Grupo III, comparadas como o melhor limite superior obtido pelo modelo  $\mathcal{F}1^+$ .

Para uma comparação entre as abordagens exatas e heurísticas consideradas neste trabalho, foi realizado um experimento onde foi registrado o custo da melhor solução obtida por cada método durante a sua execução. Nos gráficos das Figuras 4.5 e 4.6 são apresentados os resultados obtidos para as instâncias `scp410_25` e `scpsclr10_75`, respectivamente. Os dois modelos analisados neste trabalho ( $\mathcal{F}1^+$  e  $\mathcal{F}2$ ) e a heurística híbrida foram executados com tempo limite de 10 horas e as metaheurísticas GRASP e ILS por uma hora.

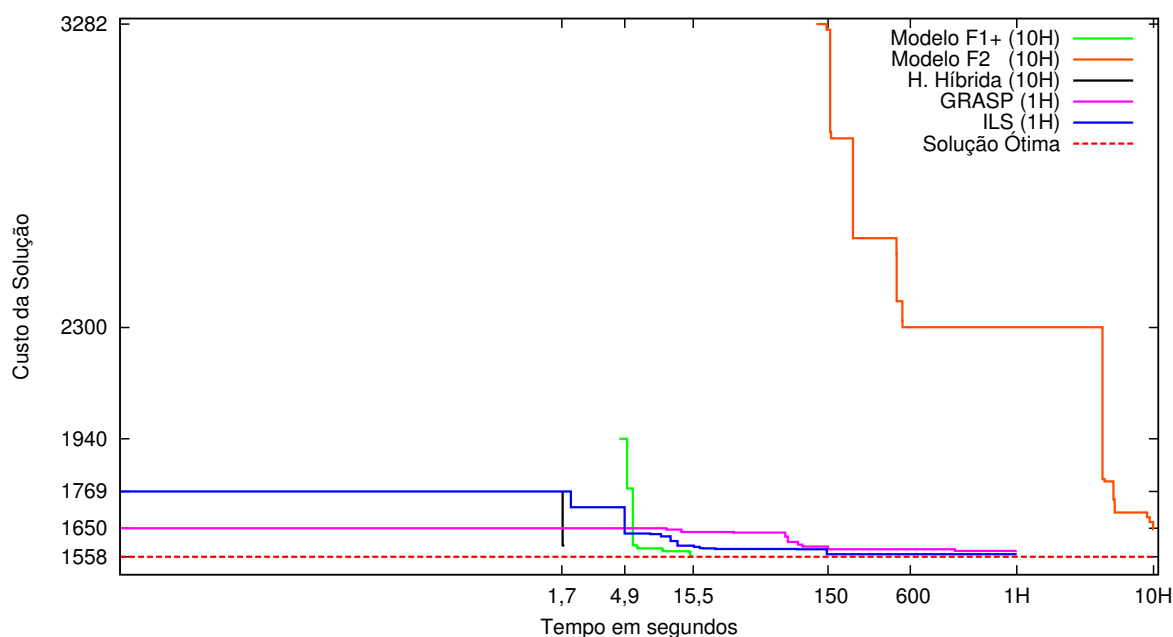


Figura 4.5: Comparação entre todas as abordagens para a instância `scp410_25`

Para a instância `scp410_25`, a solução ótima é conhecida e possui valor igual a 1558, como destacado no gráfico da Figura 4.5. Esta solução foi obtida após 15,5 segundos de processamento a partir do modelo  $\mathcal{F}1^+$  e nenhuma outra abordagem conseguiu obter uma solução com este mesmo custo neste experimento. Após 10 horas de processamento a melhor solução alcançada a partir do modelo  $\mathcal{F}2$  ainda estava a 5,71% do custo da solução ótima. A abordagem híbrida, utilizando o modelo reduzido, terminou sua execução em

1,7 segundos, mas a solução final ficou com custo 2,25% acima da solução ótima. A abordagem cujo custo da solução mais se aproximou do custo da solução ótima foi a ILS, que após 147 segundos conseguiu obter uma solução com custo de 1566, ou seja, 0,5% acima do valor ótimo. A melhor solução obtida com a heurística GRASP possui custo igual a 1577 e foi alcançada após 1200 segundos de processamento. Em uma comparação entre GRASP e ILS é possível verificar neste experimento que até 4,9 segundos a solução obtida pelo GRASP possui custo inferior ao custo da solução do ILS. Após este instante a solução do ILS possui custo inferior, o que permanece até o final da uma hora de execução.

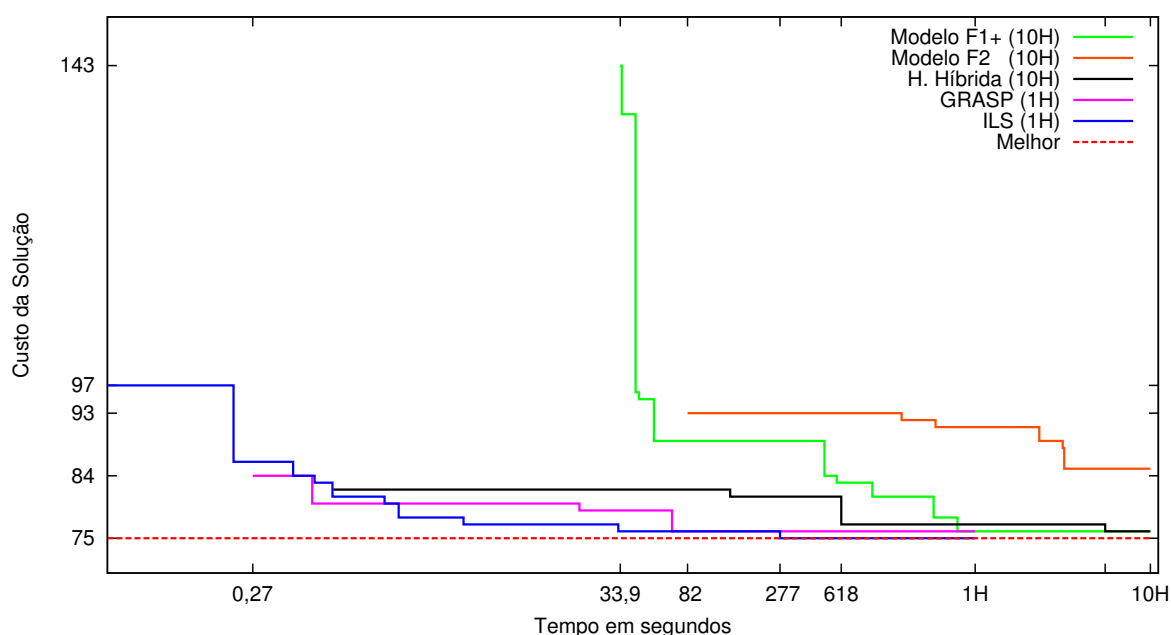


Figura 4.6: Comparação entre todas as abordagens para a instância scpsclr10\_75

Para a instância scpsclr10\_75, os modelos  $\mathcal{F}1^+$  e  $\mathcal{F}2$  não terminaram a execução dentro do tempo limite de 10 horas. Para esta instância a solução ótima não foi determinada, sendo portanto destacado no gráfico da Figura 4.6 o custo da melhor solução conhecida. A melhor solução para esta instância possui custo igual a 75, solução esta obtida a partir do ILS em 277 segundos. No tempo limite estabelecido, a partir do Modelo de Lancia ( $\mathcal{F}2$ ) foi possível obter uma solução com custo igual a 85, enquanto a solução obtida por  $\mathcal{F}1^+$  possui custo igual a 76, mesmo custo obtido pelas heurísticas híbrida e GRASP.

# Capítulo 5

## Conclusões

Nesta tese foi abordado o Problema de Cobertura de Conjuntos por Pares, uma variante do clássico Problema de Cobertura de Conjuntos. No modelo tratado, diferente do PCC, a cobertura dos elementos está relacionada a pares de objetos.

O propósito inicial deste trabalho foi determinar boas soluções para o Problema do Conjunto Mínimo de Monitoramento (PCMM), uma aplicação que pode ser modelada como o PCCP. Durante o desenvolvimento deste trabalho outros problemas que também podem ser tratados por este modelo foram identificados.

Dois modelos matemáticos da literatura para o PCCP foram apresentados. Além destes, a partir da combinação das restrições de um dos modelos da literatura, um novo modelo ( $\mathcal{F}1^+$ ) para o problema foi proposto. A partir da comparação entre os modelos, foi verificado que o modelo proposto é mais forte que os modelos da literatura. Os testes computacionais realizados com estes modelos confirmaram este resultado.

Utilizando-se o modelo proposto foi possível obter a solução ótima para todas as instâncias apresentadas no trabalho de Breslau et al. [9]. Como nestas instâncias todos os objetos possuem custo unitário, um segundo conjunto de instâncias contendo objetos com custos variados foi proposto. Este conjunto foi baseado nas instâncias apresentadas em [9]. Para este segundo grupo de instâncias, também foi possível através do modelo  $\mathcal{F}1^+$ , obter a solução ótima para todas as instâncias.

Além das instâncias baseadas no PCMM, para um terceiro conjunto de instâncias proposto, derivadas de instâncias da literatura para o PCC, não foi possível obter a solução ótima para todas as instâncias, sendo que em média o *gap* final do modelo  $\mathcal{F}1^+$  foi de 8,68% com limite de dez horas de processamento.

Com o objetivo de obter soluções de boa qualidade, principalmente para as instâncias

do terceiro grupo, abordagens heurísticas foram desenvolvidas. Cinco heurísticas de construção e três heurísticas de busca local foram propostas. Combinando estas heurísticas, foram propostas também abordagens baseadas na metaheurística GRASP, na metaheurística ILS e uma abordagem híbrida, que combina as heurísticas de construção com o modelo matemático proposto neste trabalho.

Utilizando-se as heurísticas de construção seguidas das heurísticas de busca local foi possível determinar soluções com mesmo custo da solução ótima para todas as instâncias da literatura. Este resultado foi observado também para cerca de 50% das instâncias derivadas das instâncias do PCMM com custo variado.

Considerando-se o valor das soluções ótimas e do limite superior obtido a partir dos modelos matemáticos, para aquelas instâncias onde o valor da solução ótima não é conhecido, as soluções das metaheurísticas ficaram a menos de 4% da melhor solução. Utilizando as abordagens heurísticas foi possível determinar soluções com custo inferior ao limite obtido pelo modelo proposto nesta tese para 31 das 144 instâncias derivadas das instâncias do PCC.

Como trabalhos futuros, sugere-se o estudo de outras formas de hibridização utilizando o modelo proposto neste trabalho. Uma possibilidade seria a utilização de busca local durante a resolução pelo resolvedor exato, sempre que uma nova solução inteira fosse obtida. Como o modelo é restrito a um subconjunto de colunas, a busca local poderia ser utilizada para verificar a vizinhança completa da solução, ou seja, considerando todas as colunas.

Outra sugestão de trabalhos futuros refere-se às estruturas de busca local. A busca local BLTP, apesar de requerer muito tempo computacional, obtem soluções de boa qualidade. A elaboração de um estratégia de busca local baseada na BLTP, que não seja exaustiva, pode vir a obter bons resultados em um tempo médio menor. Para a estratégia BLDA, um ajuste dinâmico do parâmetro  $\gamma$ , que indica o número de elementos a serem removidos da solução corrente, pode ser avaliado. De forma semelhante, o ajuste na perturbação da heurística ILS, fazendo modificações maiores quando o algoritmo deixa de obter boas soluções, também pode ser explorado.

# Referências

- [1] ABRAYAYA, S.; SEGAL, M. Low complexity algorithms for optimal consumer push-pull partial covering in the plane. *European Journal of Operational Research* 197, 2 (2009), 456 – 464.
- [2] AIEX, R. M.; RESENDE, M. G. C.; RIBEIRO, C. C. Probability distribution of solution time in GRASP: An experimental investigation. *Journal of Heuristics* 8 (2002), 343–3732.
- [3] BEAN, J. C. Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing* 6, 2 (1994), 154–160.
- [4] BEASLEY, J. Or-library: Distributing test problems by electronic mail. *Journal of the Operational Research Society* 41, 11 (1990), 1069–1072.
- [5] BERMAN, P.; DASGUPTA, B.; SONTAG, E. Randomized approximation algorithms for set multicover problems with applications to reverse engineering of protein and gene networks. In *Proc. 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems* (Cambridge, MA, USA, 2004), Springer, pp. 39–50.
- [6] BERMAN, P.; DASGUPTA, B.; SONTAG, E. Randomized approximation algorithms for set multicover problems with applications to reverse engineering of protein and gene networks. *Discrete Appl. Math.* 155, 6-7 (2007), 733–749.
- [7] BERTOLAZZI, P.; GODI, A.; LABBÉ, M.; TINININI, L. Solving haplotyping inference parsimony problem using a new basic polynomial formulation. *Comput. Math. Appl.* 55, 5 (2008), 900–911.
- [8] BIENKOWSKI, M.; BYRKA, J. Bucket game with applications to set multicover and dynamic page migration. In *ESA* (Palma de Mallorca, Spain, 2005), G. S. Brodal and S. Leonardi, Eds., vol. 3669 of *Lecture Notes in Computer Science*, Springer, pp. 815–826.
- [9] BRESLAU, L.; DIAKONIKOLAS, I.; DUFFIELD, N. G.; GU, Y.; HAJIAGHAYI, M.; JOHNSON, D. S.; KARLOFF, H.; RESENDE, M. G. C.; SEN, S.; TOWSLEY, D. Optimal node placement for path disjoint network monitoring. Tech. rep., AT&T Labs Research, NJ, USA, 2007.
- [10] BROWN, D. G.; HARROWER, I. M. Integer programming approaches to haplotype inference by pure parsimony. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 3 (2006), 141–154.

- 
- [11] CAMPELLO, R. E.; MACULAN, N. *Algoritmos e Heurísticas, Desenvolvimento e Avaliação de Performance*, vol. 1. EDUFF, Niterói, RJ, 1994.
- [12] CATALANO, M. S. F.; MALUCELLI, F. Parallel randomized heuristics for the set covering problem. 113–132.
- [13] CATANZARO, D.; LABBÉ, M. The pure parsimony haplotyping problem: Overview and computational advances. *International Transactions in Operational Research* 16, 5 (September 2009), 561–584.
- [14] CHEKURI, C.; CLARKSON, K. L.; HAR-PELED, S. On the set multi-cover problem in geometric settings. In *SCG '09: Proceedings of the 25th annual symposium on computational geometry* (New York, NY, USA, 2009), ACM, pp. 341–350.
- [15] CHVÁTAL, V. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research* 4 (1979), 233–235.
- [16] DESROCHERS, M.; SOUMIS, F. A column generation approach to the urban transit crew scheduling problem. *Transportation Science* 23 (1989), 1–13.
- [17] FEO, T.; RESENDE, M. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters* 8 (1989), 67–71.
- [18] FEO, T.; RESENDE, M. Greedy randomized adaptive search procedures. *Journal of Global Optimization* 6 (1995), 109–133.
- [19] FERREIRA, C. M. S.; OCHI, L. S. Metaheurísticas GRASP com memória adaptativa para a solução do problema da Árvore geradora mínima generalizado. In *Anais do VI Encontro Nacional de Inteligência Artificial (ENIA)* (Rio de Janeiro, Julho 2007), I. de Castro Dutra, C. M. G. M. de Oliveira, and R. Choren, Eds., SBC, pp. 1202–1211.
- [20] FESTA, P.; PARDALOS, P.; PITSOULIS, L.; RESENDE, M. GRASP with path-relinking for the weighted maximum satisfiability problem. In *Proceedings of IV Workshop on Efficient and Experimental Algorithms (WEA2005)*, S. Nikolettseas, Ed., vol. 3503. Springer, Santorini Island, Greece, 2005, pp. 367–379.
- [21] FISCHETTI, M.; LODI, A. Local branching. *Mathematical Programming B*, 98 (2003), 23–47.
- [22] FISCHETTI, M.; LODI, A. Repairing mip infeasibility through local branching. *Computers and Operations Research*, 35 (2008), 1436–1445.
- [23] GANDHI, R.; KHULLER, S.; SRINIVASAN, A. Approximation algorithms for partial covering problems. *J. Algorithms* 53 (2004), 55–84.
- [24] GAREY, M. R.; JOHNSON, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
- [25] GONÇALVES, L. B.; MARTINS, S. L.; OCHI, L. S. A GRASP with adaptive memory for a period vehicle routing problem. *Computational Intelligence for Modelling, Control and Automation* 1 (2005), 721–727.



- [26] GONÇALVES, L. B.; MARTINS, S. L.; OCHI, L. S. Effective heuristics for the set covering with pairs problem. *International Transactions in Operational Research* (2010), 1–13.
- [27] GONÇALVES, L. B.; MARTINS, S. L.; OCHI, L. S.; RESENDE, M. G. C. Effective heuristics for the set cover by pairs problem. In *VI ALIO - EURO Conference on Applied Combinatorial Optimization* (2008).
- [28] GONÇALVES, L. B.; MARTINS, S. L.; OCHI, L. S.; RESENDE, M. G. C. Uma heurística eficiente para o problema de cobertura de conjunto por pares. In *Anais do XL Simpósio Brasileiro de Pesquisa Operacional* (2008), pp. 1916–1923.
- [29] GREENBERG, H. J.; HART, W. E.; LANCIA, G. Opportunities for combinatorial optimization in computational biology. *INFORMS Journal on Computing* 16, 3 (2004), 211–231.
- [30] GU, Y.; BRESLAU, L.; DUFFIELD, N.; SEN, S. GRE encapsulated multicast probing: A scalable technique for measuring one-way loss. In *INFOCOM 2008. The 27th Conference on Computer Communications* (Phoenix, Arizona, USA, 2008), IEEE, pp. 1651–1659.
- [31] GU, Y.; BRESLAU, L.; DUFFIELD, N. G.; SEN, S. GRE encapsulated multicast probing: A scalable technique for measuring one-way loss. In *SIGMETRICS '07: Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems* (New York, NY, USA, 2007), ACM, pp. 355–356.
- [32] GUSFIELD, D. Haplotype inference by pure parsimony. In *Combinatorial Pattern Matching, 14th Annual Symposium, CPM 2003* (Morelia, Michocán, Mexico, 2003), R. A. Baeza-Yates, E. Chávez, and M. Crochemore, Eds., vol. 2676 of *Lecture Notes in Computer Science*, Springer.
- [33] GUSFIELD, D.; ORZACK, S. H. Haplotype inference. In *Handbook of Computational Molecular Biology*, S. Aluru, Ed. Chapman & Hall/CRC Press, Boca Raton, FL, USA, 2006, pp. 18.1–18.28.
- [34] HALL, N. G.; HOCHBAUM, D. S. The multicovering problem. *European Journal of Operational Research* 62, 3 (November 1992), 323–339.
- [35] HASSIN, R.; SEGEV, D. The set cover with pairs problem. In *FSTTCS* (Hyderabad, India, 2005), R. Ramanujam and S. Sen, Eds., vol. 3821 of *Lecture Notes in Computer Science*, Springer, pp. 164–176.
- [36] HERMELIN, D.; RAWITZ, D.; RIZZI, R.; VIALETTE, S. The minimum substring cover problem. *Information and Computation* 206 (2008), 1303–1312.
- [37] HIRSCH, M. J.; DE MENESES, C. N.; PARDALOS, P. M.; RESENDE, M. G. C. Global optimization by continuous GRASP. *Optimization Letters* 1, 2 (2007), 201–212.
- [38] HUA, Q.-S.; WANG, Y.; YU, D.; LAU, F. C. M. Set multi-covering via inclusion-exclusion. *Theor. Comput. Sci.* 410, 38-40 (2009), 3882–3892.

- [39] ILOG, INC. Solver cplex 11.2, ilog concert technolog, 2009. (acessado em 15 Maio 2009).
- [40] LANCIA, G.; PINOTTI, M. C.; RIZZI, R. Haplotyping populations by pure parsimony: Complexity of exact and approximation algorithms. *INFORMS Journal on Computing* 16, 4 (2004), 348–359.
- [41] LANCIA, G.; SERAFINI, P. A set-covering approach with column generation for parsimony haplotyping. *INFORMS Journal on Computing* 21, 1 (2009), 151–166.
- [42] LIN, S.-W.; LEE, Z.-J.; YING, K.-C.; LEE, C.-Y. Applying hybrid meta-heuristics for capacitated vehicle routing problem. *Expert Systems with Applications* 36, 2 (2009), 1505–1512.
- [43] LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search. In *Handbook of Metaheuristics*, F. GLOVER and G. KOCHENBERGER, Eds., vol. 57. Springer - Kluwer Academic Publishers, New York, 2003, pp. 320–353.
- [44] MANIEZZO, V.; STÜTZLE, T.; VOSS, S., Eds. *Matheuristics: Hybridizing Metaheuristics and Mathematical Programming*, vol. 10 of *Annals of Information Systems*. Springer US, New York, 2009.
- [45] MARTINS, S.; PARDALOS, P.; RESENDE, M.; RIBEIRO, C. Greedy randomized adaptive search procedures for the Steiner problem in graphs. In *Randomization methods in algorithm design*, P. Pardalos and S. Rajasekaran, Eds., vol. 43 of *DI-MACS Series on Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, Providence, Rhode Island, USA, 1999, pp. 133–145.
- [46] MENDES, J. J. M.; GONÇALVES, J. F.; RESENDE, M. G. C. A random key based genetic algorithm for the resource constrained project scheduling problem. *Comput. Oper. Res.* 36, 1 (2009), 92–109.
- [47] PESSÔA, L. *Heurísticas para o problema de k-cobertura de conjuntos*. Tese de Doutorado, Universidade Federal Fluminense, Niterói - RJ, 2009.
- [48] PESSÔA, L.; RESENDE, M.; RIBEIRO, C. A hybrid lagrangean heuristic with GRASP and path-relinking for set k-covering. In *Third International Workshop on Model-Based Metaheuristics* (Vienna, Austria, 2010).
- [49] PRAIS, M.; RIBEIRO, C. C. Reactive grasp: An application to a matrix decomposition problem in tdma traffic assignment. *INFORMS Journal on Computing* 12, 3 (2000), 164–176.
- [50] RESENDE, M.; RIBEIRO, C. A GRASP for graph planarization. *Networks* 29 (1997), 173–189.
- [51] RESENDE, M.; RIBEIRO, C. GRASP with path-relinking: Recent advances and applications. In *Metaheuristics: Progress as Real Problem Solvers*, T. Ibaraki, K. Nobe, and M. Yagiura, Eds. Springer, USA, 2005, pp. 29–63.
- [52] RESENDE, M.; RIBEIRO, C. Greedy randomized adaptive search procedures: Advances and applications. In *Handbook of Metaheuristics*, M. G. e J.-Y. Potvin, Ed., 2a edição ed. Springer, 2009.

- 
- [53] RESENDE, M.; RIBEIRO, C. Greedy randomized adaptive search procedures: Advances and applications. In *Handbook of Metaheuristics*, M. Gendreau and J.-Y. Potvin, Eds., 2 ed., vol. (a ser publicado). Springer, 2009.
- [54] RIBEIRO, C. C.; UCHOA, E.; WERNECK, R. F. A hybrid grasp with perturbations for the steiner problem in graphs. *INFORMS Journal on Computing* 14 (2002), 228–246.
- [55] SALVESON, M. E. The assembly line balancing problem. *Journal of Industrial Engineering* 6 (1995), 18–25.
- [56] SILVA, A. R. V.; OCHI, L. S. Effective GRASP for the dynamic resource task scheduling problem. In *Proceedings of the INOC 2007* (Spa, Belgium, Apr. 2007), International Network Optimization Conference.
- [57] SOUZA, M. J. F.; MACULAN, N.; OCHI, L. S. A GRASP-tabu search algorithm for solving school timetabling problems. 659–672.
- [58] TOREGAS, C.; SWAIN, R.; REVELLE, C.; BERGMAN, L. The location of emergency service facilities. *Operations Research* 19 (1971), 1363–1373.
- [59] TORREGAS, C.; REVELLE, C.; BERGMAN, L. The location of emergency service facilities. *Operations Research* 19 (1970), 1363–1373.
- [60] VAZIRANI, V. V. *Approximation algorithms*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.
- [61] WANG, R.-S.; ZHANG, X.-S.; SHENG, L. Haplotype inference by pure parsimony via genetic algorithm. In *5th International Symposium on Operations Research and Its Applications* (Tibet, China, Agosto 2005), X.-S. Zhang, D.-G. Liu, and L.-Y. Wu, Eds., vol. 5 of *Lecture Notes in Operations Research*, Beijing World Publishing Corporation, pp. 308–318.

# APÊNDICE A

## A.1 Gerador de Instâncias

Para avaliação empírica das abordagens desenvolvidas neste trabalho foram propostos dois novos conjuntos de instâncias para o PCCP. O primeiro é baseado nas heurísticas da literatura, onde custos variados foram atribuídos às instâncias propostas por Breslau et al. [9]. O segundo conjunto é composto por 144 instâncias derivadas das instâncias da literatura para o Problema de Cobertura de Conjuntos.

O Algoritmo 12 descreve o procedimento para obtenção de uma instância do PCCP a partir de instâncias do PCC.

---

**Algoritmo 12** *Geração das Instâncias do Grupo III ( $p$ )*

---

```

1: para cada linha  $l$  faça
2:    $nCol \leftarrow nCobertura(l)$ ;
3:   para  $k \leftarrow 1 \dots nCol$  faça
4:     para  $i \leftarrow 1 \dots k - 1$  faça
5:        $sort \leftarrow \text{rand}() \bmod 100$ ;
6:       se ( $sort > p$ ) então
7:          $instanciaSCPP \leftarrow \text{tupla}(i, k, l)$ ;
8:       fim se
9:     fim para
10:  fim para
11: fim para

```

---

O parâmetro  $p \in [0, 100]$  indica, para cada possível par de colunas da instância do PCC, a probabilidade  $(1-p)$  deste par estar presente na instância do PCCP cobrindo uma dada linha. Para cada linha a ser coberta  $l$ , a variável  $nCol$  receberá o número de colunas que a cobre na instância original do Problema de Cobertura de Conjuntos (linha 2). Em seguida, as colunas são tomadas duas a duas. Para cada par de colunas é sorteado um valor entre  $[0, 100]$ , que é armazenado na variável  $sort$ . Se o valor sorteado for menor que o valor de  $p$ , o par considerado é inserido na nova instância, como pode ser visto na

linha 7 do algoritmo. Na inserção é indicado o par que está sendo inserido e a linha que o mesmo poderá cobrir.

## A.2 Tabelas Adicionais

Nesta seção são apresentadas tabelas adicionais que visam apresentar os resultados obtidos para cada instância individualmente.

A Tabela A.1 apresenta, para um subconjunto das instâncias do Grupo II, a média dos resultados obtidos para cada uma das versões do GRASP apresentadas no Capítulo 4. De maneira complementar, a Tabela A.2 apresenta o tempo médio de CPU gasto por cada abordagem em cada uma das instâncias.

Tabela A.1: Média das soluções obtidas pelas versões GRASP.

Instância	G-DA	G-TO	G-DATO	G-TPTO	
1	n300-i0-m300-b150-c10.cst	194,0	194,0	194,0	194,0
2	n300-i0-m300-b150-c1000.cst	15433,0	15433,0	15433,0	15433,0
3	n300-i0-m300-b300-c10.cst	286,0	286,0	286,0	286,0
4	n300-i0-m300-b300-c100.cst	2538,0	2536,0	2534,0	2534,0
5	n300-i0-m300-b300-c1000.cst	25021,0	24998,0	24998,0	24998,0
6	n300-i0-m300-b75-c100.cst	1224,0	1222,0	1222,0	1222,0
7	n300-i0-m300-b75-c1000.cst	8591,0	8586,3	8591,0	8585,0
8	n300-i1-m300-b150-c10.cst	178,0	178,0	178,0	178,0
9	n300-i1-m300-b150-c1000.cst	14918,0	14918,0	14918,0	14918,0
10	n300-i1-m300-b300-c100.cst	2889,0	2885,4	2885,0	2885,0
11	n300-i1-m300-b75-c1000.cst	10202,0	10202,0	10202,0	10202,0
12	n300-i2-m150-b150-c10.cst	233,0	233,4	233,0	233,0
13	n300-i2-m150-b150-c100.cst	2371,0	2371,0	2371,0	2371,0
14	n300-i2-m150-b150-c1000.cst	23245,0	23214,0	23214,0	23214,0
15	n300-i2-m300-b150-c100.cst	1991,0	1990,1	1990,0	1990,0
16	n300-i2-m300-b150-c1000.cst	20594,0	20568,0	20568,0	20568,0
17	n300-i2-m300-b300-c100.cst	3301,0	3301,0	3301,0	3301,0
18	n300-i3-m150-b150-c10.cst	233,0	233,0	233,0	233,0
19	n300-i3-m300-b150-c10.cst	176,0	174,9	174,0	174,0
20	n300-i3-m300-b150-c1000.cst	18351,0	18356,2	18351,0	18351,0
21	n300-i3-m300-b300-c10.cst	274,0	274,1	274,0	274,0
22	n300-i3-m300-b300-c100.cst	2403,0	2400,0	2400,0	2400,0
23	n300-i3-m300-b300-c1000.cst	21978,0	21940,8	21963,0	21938,0
24	n300-i3-m300-b75-c100.cst	1218,0	1218,0	1218,0	1218,0
25	n300-i3-m300-b75-c1000.cst	9178,0	9178,0	9178,0	9178,0
26	n300-i4-m150-b150-c10.cst	224,0	224,0	224,0	224,0
27	n300-i4-m300-b150-c10.cst	186,0	185,4	186,0	185,0
28	n300-i4-m300-b150-c100.cst	1887,0	1887,0	1887,0	1887,0
29	n300-i4-m300-b150-c1000.cst	15944,0	15925,0	15925,0	15925,0
30	n300-i4-m300-b300-c10.cst	306,0	304,6	304,0	304,0
31	n300-i4-m300-b300-c100.cst	2925,0	2922,7	2922,0	2922,0
32	n300-i4-m300-b300-c1000.cst	27521,0	27521,0	27521,0	27521,0
33	n300-i5-m150-b150-c1000.cst	18031,0	18031,0	18031,0	18031,0
34	n300-i5-m300-b150-c10.cst	171,0	171,0	171,0	171,0
35	n300-i5-m300-b150-c100.cst	1648,0	1648,0	1648,0	1648,0
36	n300-i6-m300-b150-c100.cst	1822,0	1822,0	1822,0	1822,0
37	n300-i6-m300-b150-c1000.cst	18182,0	18178,0	18178,0	18178,0
38	n300-i6-m300-b300-c10.cst	310,0	310,0	310,0	310,0
39	n300-i6-m300-b300-c1000.cst	28941,0	28820,0	28767,0	28766,0
40	n300-i6-m300-b75-c1000.cst	11137,0	11137,0	11137,0	11137,0
41	n300-i7-m150-b150-c1000.cst	21637,0	21637,0	21637,0	21637,0
42	n300-i7-m300-b150-c100.cst	1710,0	1710,0	1710,0	1710,0
43	n300-i7-m300-b300-c10.cst	265,0	263,4	263,0	263,0
44	n300-i7-m300-b300-c100.cst	2812,0	2799,7	2790,0	2780,0
45	n300-i7-m300-b300-c1000.cst	25886,0	25886,0	25886,0	25886,0
46	n300-i7-m300-b75-c100.cst	1131,0	1132,3	1130,0	1130,0
47	n300-i8-m300-b150-c100.cst	1702,0	1702,0	1702,0	1702,0
48	n300-i8-m300-b300-c1000.cst	26076,0	26056,2	26066,0	26027,0
49	n300-i8-m300-b75-c100.cst	1148,0	1148,0	1148,0	1148,0
50	n300-i9-m300-b150-c100.cst	1783,0	1761,3	1761,0	1761,0
51	n300-i9-m300-b300-c10.cst	262,0	262,0	262,0	262,0
52	n300-i9-m300-b300-c100.cst	2711,0	2707,6	2711,0	2699,0
53	n300-i9-m300-b75-c100.cst	1060,0	1060,0	1060,0	1060,0
54	n558-i0-m140-b140-c100.cst	3123,0	3123,0	3123,0	3123,0
55	n558-i0-m279-b279-c10.cst	363,0	363,0	363,0	363,0
56	n558-i0-m279-b279-c100.cst	3409,0	3409,0	3409,0	3409,0
57	n558-i0-m279-b279-c1000.cst	34141,0	34141,0	34141,0	34141,0
58	n558-i0-m558-b140-c10.cst	145,0	145,0	145,0	145,0
59	n558-i0-m558-b140-c100.cst	1772,0	1772,0	1772,0	1772,0
60	n558-i0-m558-b140-c1000.cst	12387,0	12387,0	12437,0	12387,0
61	n558-i0-m558-b279-c10.cst	270,0	269,2	270,0	269,0
62	n558-i0-m558-b279-c100.cst	2771,0	2771,0	2771,0	2771,0
63	n558-i0-m558-b558-c1000.cst	41890,0	41890,0	41890,0	41890,0
	Média	8011,3	8005,9	8006,2	8003,9

Tabela A.2: Tempo médio de CPU, em segundos, consumido pelas versões GRASP.

	Instância	G-DA	G-TO	G-DATO	G-TPTO
1	n300-i0-m300-b150-c10.cst	64,70	210,11	222,00	3971,11
2	n300-i0-m300-b150-c1000.cst	61,70	210,22	213,00	4960,00
3	n300-i0-m300-b300-c10.cst	115,00	449,89	435,00	11677,78
4	n300-i0-m300-b300-c100.cst	133,00	489,89	518,00	17200,00
5	n300-i0-m300-b300-c1000.cst	208,00	516,78	366,00	16850,00
6	n300-i0-m300-b75-c100.cst	28,60	100,69	95,30	1322,22
7	n300-i0-m300-b75-c1000.cst	26,20	74,63	59,40	750,56
8	n300-i1-m300-b150-c10.cst	52,45	133,56	146,00	3308,33
9	n300-i1-m300-b150-c1000.cst	48,70	191,11	126,00	3454,44
10	n300-i1-m300-b300-c100.cst	144,50	582,78	644,00	18333,33
11	n300-i1-m300-b75-c1000.cst	36,90	89,31	59,40	1377,78
12	n300-i2-m150-b150-c10.cst	15,50	37,16	46,60	1603,33
13	n300-i2-m150-b150-c100.cst	16,80	48,16	53,00	1413,33
14	n300-i2-m150-b150-c1000.cst	15,50	54,80	53,50	1912,22
15	n300-i2-m300-b150-c100.cst	49,50	200,89	194,00	5671,67
16	n300-i2-m300-b150-c1000.cst	53,80	239,11	176,00	5125,00
17	n300-i2-m300-b300-c100.cst	129,00	637,78	812,00	19477,78
18	n300-i3-m150-b150-c10.cst	16,00	42,91	43,90	1160,00
19	n300-i3-m300-b150-c10.cst	69,50	167,11	204,00	3255,56
20	n300-i3-m300-b150-c1000.cst	80,30	233,78	207,00	5038,33
21	n300-i3-m300-b300-c10.cst	106,00	368,89	397,00	10180,00
22	n300-i3-m300-b300-c100.cst	107,00	470,11	517,00	9030,00
23	n300-i3-m300-b300-c1000.cst	136,00	579,56	639,00	12655,56
24	n300-i3-m300-b75-c100.cst	33,90	97,53	79,50	1793,33
25	n300-i3-m300-b75-c1000.cst	33,00	100,51	81,80	1213,33
26	n300-i4-m150-b150-c10.cst	18,30	60,72	53,90	1498,89
27	n300-i4-m300-b150-c10.cst	55,90	202,56	143,00	6448,89
28	n300-i4-m300-b150-c100.cst	69,90	179,44	179,00	3865,56
29	n300-i4-m300-b150-c1000.cst	70,40	198,33	227,00	6915,00
30	n300-i4-m300-b300-c10.cst	158,00	572,89	615,00	22666,67
31	n300-i4-m300-b300-c100.cst	164,00	530,56	592,00	21011,11
32	n300-i4-m300-b300-c1000.cst	165,00	683,67	794,00	18116,67
33	n300-i5-m150-b150-c1000.cst	14,50	42,46	34,50	946,44
34	n300-i5-m300-b150-c10.cst	54,70	156,33	167,00	2628,89
35	n300-i5-m300-b150-c100.cst	46,00	145,33	148,00	3626,67
36	n300-i6-m300-b150-c100.cst	60,10	176,67	134,00	4503,33
37	n300-i6-m300-b150-c1000.cst	69,15	166,22	172,00	4965,00
38	n300-i6-m300-b300-c10.cst	109,50	388,56	381,00	13755,56
39	n300-i6-m300-b300-c1000.cst	121,50	592,56	577,00	20433,33
40	n300-i6-m300-b75-c1000.cst	23,20	71,29	51,40	869,22
41	n300-i7-m150-b150-c1000.cst	13,80	39,99	56,10	1166,67
42	n300-i7-m300-b150-c100.cst	62,40	167,78	209,00	4114,44
43	n300-i7-m300-b300-c10.cst	104,00	438,89	449,00	11533,33
44	n300-i7-m300-b300-c100.cst	121,00	402,00	429,00	18116,67
45	n300-i7-m300-b300-c1000.cst	115,00	728,56	679,00	20033,33
46	n300-i7-m300-b75-c100.cst	24,60	59,13	54,40	1310,00
47	n300-i8-m300-b150-c100.cst	64,90	208,78	191,00	3621,67
48	n300-i8-m300-b300-c1000.cst	118,00	423,56	426,00	14244,44
49	n300-i8-m300-b75-c100.cst	44,00	92,27	68,10	1286,67
50	n300-i9-m300-b150-c100.cst	67,80	206,11	207,00	5385,00
51	n300-i9-m300-b300-c10.cst	116,00	328,44	344,00	10408,33
52	n300-i9-m300-b300-c100.cst	110,00	329,78	387,00	24733,33
53	n300-i9-m300-b75-c100.cst	30,70	71,18	58,80	1180,00
54	n558-i0-m140-b140-c100.cst	12,20	29,83	32,10	900,22
55	n558-i0-m279-b279-c10.cst	107,00	302,44	287,00	15277,78
56	n558-i0-m279-b279-c100.cst	95,20	349,89	331,00	15900,00
57	n558-i0-m279-b279-c1000.cst	109,00	315,56	358,00	17011,11
58	n558-i0-m558-b140-c10.cst	174,00	385,44	452,00	9202,22
59	n558-i0-m558-b140-c100.cst	254,00	757,22	623,00	15066,67
60	n558-i0-m558-b140-c1000.cst	246,00	459,00	328,00	11225,56
61	n558-i0-m558-b279-c10.cst	408,00	1348,89	1480,00	39333,33
62	n558-i0-m558-b279-c100.cst	441,00	1462,22	1390,00	45944,44
63	n558-i0-m558-b558-c1000.cst	1010,00	4735,56	3860,00	221500,00
	Média	108,42	383,13	370,74	12753,67

A Tabela A.3 apresenta o resumo dos resultados obtidos pelo metaheurística ILS quando utiliza apenas a busca local BLTO. O critério de parada adotado foi o número de iterações sem melhora. Nesta bateria de teste foi adotado  $k_{max} = 6$ ; Os resultados são apresentados para um subconjunto de 65 instâncias derivadas das instâncias de Breslau. Estas instâncias foram selecionadas para uma comparação inicial entre as metaheurísticas. O grupo é composto por instâncias onde  $n = 300$  de  $n = 558$ , dos diferentes conjuntos.

Para este mesmo subconjunto de instâncias a Tabela A.4 apresenta os resultados do GRASP utilizando a busca local BLTO, onde o critério de parada adotado foi a realização de 50 iterações. De maneira semelhante, a Tabela A.5 apresenta os resultados obtidos pelo GRASP utilizando as estruturas de busca local BLTP e BLTO, nesta ordem. Por último, a Tabela A.6 apresenta os resultados da abordagem Híbrida para este mesmo conjunto de instâncias.



Tabela A.3: ILS utilizando BLTO.

Instância	Melhor Solução	Custo Médio	Dif. Percentual	Tempo Médio (s)
n300-i0-m150-b150-c100.cst	1995	1997,78	0,14%	3,9
n300-i0-m300-b150-c10.cst	195	195,00	0,00%	7,4
n300-i0-m300-b150-c1000.cst	15433	15746,78	2,03%	14,6
n300-i0-m300-b300-c10.cst	288	288,00	0,00%	22,8
n300-i0-m300-b300-c100.cst	2618	2662,33	1,69%	30,3
n300-i0-m300-b300-c1000.cst	25021	25021,00	0,00%	24,8
n300-i0-m300-b75-c100.cst	1224	1224,00	0,00%	4,8
n300-i0-m300-b75-c1000.cst	8591	8591,00	0,00%	2,6
n300-i1-m300-b150-c10.cst	180	182,00	1,11%	12,3
n300-i1-m300-b150-c1000.cst	14921	14921,00	0,00%	9,8
n300-i1-m300-b300-c100.cst	2929	2961,00	1,09%	61,9
n300-i1-m300-b75-c1000.cst	10413	10591,67	1,72%	2,6
n300-i1-m75-b75-c1000.cst	17196	17486,11	1,69%	0,3
n300-i2-m150-b150-c10.cst	234	237,11	1,33%	3,4
n300-i2-m150-b150-c100.cst	2416	2439,11	0,96%	2,4
n300-i2-m150-b150-c1000.cst	23352	24025,33	2,88%	6,7
n300-i2-m300-b150-c100.cst	1991	1991,00	0,00%	13,5
n300-i2-m300-b150-c1000.cst	20594	20594,00	0,00%	20,4
n300-i2-m300-b300-c100.cst	3329	3372,11	1,30%	83,4
n300-i3-m150-b150-c10.cst	233	233,89	0,38%	2,5
n300-i3-m300-b150-c10.cst	179	181,56	1,43%	4,7
n300-i3-m300-b150-c1000.cst	18533	18574,78	0,23%	14,9
n300-i3-m300-b300-c10.cst	275	276,11	0,40%	30,2
n300-i3-m300-b300-c100.cst	2403	2403,00	0,00%	44,7
n300-i3-m300-b300-c1000.cst	21971	21990,44	0,09%	66,0
n300-i3-m300-b75-c100.cst	1222	1245,11	1,89%	4,4
n300-i3-m300-b75-c1000.cst	9178	9454,89	3,02%	6,8
n300-i4-m150-b150-c10.cst	225	225,00	0,00%	1,6
n300-i4-m300-b150-c10.cst	188	188,00	0,00%	9,5
n300-i4-m300-b150-c100.cst	1887	1921,44	1,83%	22,2
n300-i4-m300-b150-c1000.cst	16450	16450,00	0,00%	20,9
n300-i4-m300-b300-c10.cst	306	307,11	0,36%	58,3
n300-i4-m300-b300-c100.cst	2974	2996,22	0,75%	52,1
n300-i4-m300-b300-c1000.cst	27535	27895,44	1,31%	102,8
n300-i5-m150-b150-c1000.cst	18106	18106,00	0,00%	3,9
n300-i5-m300-b150-c10.cst	171	171,89	0,52%	8,9
n300-i5-m300-b150-c100.cst	1648	1669,00	1,27%	18,6
n300-i6-m300-b150-c100.cst	1822	1853,78	1,74%	19,3
n300-i6-m300-b150-c1000.cst	18178	18206,44	0,16%	13,4
n300-i6-m300-b300-c10.cst	313	315,00	0,64%	32,4
n300-i6-m300-b300-c1000.cst	29145	29145,00	0,00%	63,5
n300-i6-m300-b75-c1000.cst	11296	11296,00	0,00%	2,2
n300-i7-m150-b150-c1000.cst	22298	22298,00	0,00%	4,7
n300-i7-m300-b150-c100.cst	1739	1766,56	1,58%	15,3
n300-i7-m300-b300-c10.cst	267	267,00	0,00%	35,3
n300-i7-m300-b300-c100.cst	2864	2884,22	0,71%	36,1
n300-i7-m300-b300-c1000.cst	26111	26567,67	1,75%	89,3
n300-i7-m300-b75-c100.cst	1166	1182,89	1,45%	0,9
n300-i8-m300-b150-c100.cst	1714	1735,78	1,27%	11,9
n300-i8-m300-b300-c1000.cst	26177	26345,11	0,64%	30,3
n300-i8-m300-b75-c100.cst	1150	1150,00	0,00%	1,9
n300-i9-m300-b150-c100.cst	1808	1808,00	0,00%	10,7
n300-i9-m300-b300-c10.cst	263	264,56	0,59%	27,6
n300-i9-m300-b300-c100.cst	2719	2739,56	0,76%	31,0
n300-i9-m300-b75-c100.cst	1116	1116,00	0,00%	3,0
n558-i0-m140-b140-c100.cst	3123	3137,00	0,45%	1,0
n558-i0-m279-b279-c10.cst	363	365,11	0,58%	59,5
n558-i0-m279-b279-c100.cst	3440	3490,89	1,48%	44,8
n558-i0-m279-b279-c1000.cst	34832	34832,00	0,00%	19,3
n558-i0-m558-b140-c10.cst	145	145,78	0,54%	26,9
n558-i0-m558-b140-c100.cst	1772	1789,11	0,97%	42,6
n558-i0-m558-b140-c1000.cst	12437	12496,56	0,48%	21,7
n558-i0-m558-b279-c10.cst	270	271,11	0,41%	148,4
n558-i0-m558-b279-c100.cst	2788	2788,00	0,00%	108,2
n558-i0-m558-b558-c1000.cst	41890	42029,78	0,33%	397,8
<b>Média</b>	<b>8117,08</b>	<b>8170,82</b>	<b>0,71%</b>	<b>32,3</b>

Tabela A.4: GRASP utilizando BLTO.

Instância	Melhor Solução	Custo Médio	Dif. Percentual	Tempo Médio (s)
n300-i0-m150-b150-c100.cst	1995	1995,00	0,00%	48,9
n300-i0-m300-b150-c10.cst	194	194,00	0,00%	210,1
n300-i0-m300-b150-c1000.cst	15433	15433,00	0,00%	210,2
n300-i0-m300-b300-c10.cst	286	286,00	0,00%	449,9
n300-i0-m300-b300-c100.cst	2534	2536,00	0,08%	489,9
n300-i0-m300-b300-c1000.cst	24998	24998,00	0,00%	516,8
n300-i0-m300-b75-c100.cst	1222	1222,00	0,00%	100,7
n300-i0-m300-b75-c1000.cst	8585	8586,33	0,02%	74,6
n300-i1-m300-b150-c10.cst	178	178,00	0,00%	133,6
n300-i1-m300-b150-c1000.cst	14918	14918,00	0,00%	191,1
n300-i1-m300-b300-c100.cst	2885	2885,44	0,02%	582,8
n300-i1-m300-b75-c1000.cst	10202	10202,00	0,00%	89,3
n300-i1-m75-b75-c1000.cst	17186	17186,00	0,00%	3,4
n300-i2-m150-b150-c10.cst	233	233,44	0,19%	37,2
n300-i2-m150-b150-c100.cst	2371	2371,00	0,00%	48,2
n300-i2-m150-b150-c1000.cst	23214	23214,00	0,00%	54,8
n300-i2-m300-b150-c100.cst	1990	1990,11	0,01%	200,9
n300-i2-m300-b150-c1000.cst	20568	20568,00	0,00%	239,1
n300-i2-m300-b300-c100.cst	3301	3301,00	0,00%	637,8
n300-i3-m150-b150-c10.cst	233	233,00	0,00%	42,9
n300-i3-m300-b150-c10.cst	174	174,89	0,51%	167,1
n300-i3-m300-b150-c1000.cst	18351	18356,22	0,03%	233,8
n300-i3-m300-b300-c10.cst	274	274,11	0,04%	368,9
n300-i3-m300-b300-c100.cst	2400	2400,00	0,00%	470,1
n300-i3-m300-b300-c1000.cst	21938	21940,78	0,01%	579,6
n300-i3-m300-b75-c100.cst	1218	1218,00	0,00%	97,5
n300-i3-m300-b75-c1000.cst	9178	9178,00	0,00%	100,5
n300-i4-m150-b150-c10.cst	224	224,00	0,00%	60,7
n300-i4-m300-b150-c10.cst	185	185,44	0,24%	202,6
n300-i4-m300-b150-c100.cst	1887	1887,00	0,00%	179,4
n300-i4-m300-b150-c1000.cst	15925	15925,00	0,00%	198,3
n300-i4-m300-b300-c10.cst	304	304,56	0,18%	572,9
n300-i4-m300-b300-c100.cst	2922	2922,67	0,02%	530,6
n300-i4-m300-b300-c1000.cst	27521	27521,00	0,00%	683,7
n300-i5-m150-b150-c1000.cst	18031	18031,00	0,00%	42,5
n300-i5-m300-b150-c10.cst	171	171,00	0,00%	156,3
n300-i5-m300-b150-c100.cst	1648	1648,00	0,00%	145,3
n300-i6-m300-b150-c100.cst	1822	1822,00	0,00%	176,7
n300-i6-m300-b150-c1000.cst	18178	18178,00	0,00%	166,2
n300-i6-m300-b300-c10.cst	310	310,00	0,00%	388,6
n300-i6-m300-b300-c1000.cst	28766	28820,00	0,19%	592,6
n300-i6-m300-b75-c1000.cst	11137	11137,00	0,00%	71,3
n300-i7-m150-b150-c1000.cst	21637	21637,00	0,00%	40,0
n300-i7-m300-b150-c100.cst	1710	1710,00	0,00%	167,8
n300-i7-m300-b300-c10.cst	263	263,44	0,17%	438,9
n300-i7-m300-b300-c100.cst	2788	2799,67	0,42%	402,0
n300-i7-m300-b300-c1000.cst	25886	25886,00	0,00%	728,6
n300-i7-m300-b75-c100.cst	1130	1132,33	0,21%	59,1
n300-i8-m300-b150-c100.cst	1702	1702,00	0,00%	208,8
n300-i8-m300-b300-c1000.cst	26027	26056,22	0,11%	423,6
n300-i8-m300-b75-c100.cst	1148	1148,00	0,00%	92,3
n300-i9-m300-b150-c100.cst	1761	1761,33	0,02%	206,1
n300-i9-m300-b300-c10.cst	262	262,00	0,00%	328,4
n300-i9-m300-b300-c100.cst	2702	2707,56	0,21%	329,8
n300-i9-m300-b75-c100.cst	1060	1060,00	0,00%	71,2
n558-i0-m140-b140-c100.cst	3123	3123,00	0,00%	29,8
n558-i0-m279-b279-c10.cst	363	363,00	0,00%	302,4
n558-i0-m279-b279-c100.cst	3409	3409,00	0,00%	349,9
n558-i0-m279-b279-c1000.cst	34141	34141,00	0,00%	315,6
n558-i0-m558-b140-c10.cst	145	145,00	0,00%	385,4
n558-i0-m558-b140-c100.cst	1772	1772,00	0,00%	757,2
n558-i0-m558-b140-c1000.cst	12387	12387,00	0,00%	459,0
n558-i0-m558-b279-c10.cst	269	269,22	0,08%	1348,9
n558-i0-m558-b279-c100.cst	2771	2771,00	0,00%	1462,2
n558-i0-m558-b558-c1000.cst	41890	41890,00	0,00%	4735,6
<b>Média</b>	<b>8052,86</b>	<b>8054,69</b>	<b>0,04%</b>	<b>372,15</b>

Tabela A.5: GRASP utilizando BLTP e BLTO.

Instância	Melhor Solução	Custo Médio	Dif. Percentual	Tempo Médio (s)
n300-i0-m150-b150-c100.cst	1995	1995,00	0,00%	903,9
n300-i0-m300-b150-c10.cst	194	194,00	0,00%	3971,1
n300-i0-m300-b150-c1000.cst	15433	15433,00	0,00%	4960,0
n300-i0-m300-b300-c10.cst	286	286,00	0,00%	11677,8
n300-i0-m300-b300-c100.cst	2534	2534,00	0,00%	17200,0
n300-i0-m300-b300-c1000.cst	24998	24998,00	0,00%	16850,0
n300-i0-m300-b75-c100.cst	1222	1222,00	0,00%	1322,2
n300-i0-m300-b75-c1000.cst	8585	8585,00	0,00%	750,6
n300-i1-m300-b150-c10.cst	178	178,00	0,00%	3308,3
n300-i1-m300-b150-c1000.cst	14918	14918,00	0,00%	3454,4
n300-i1-m300-b300-c100.cst	2885	2885,00	0,00%	18333,3
n300-i1-m300-b75-c1000.cst	10202	10202,00	0,00%	1377,8
n300-i1-m75-b75-c1000.cst	17186	17186,00	0,00%	88,8
n300-i2-m150-b150-c10.cst	233	233,00	0,00%	1603,3
n300-i2-m150-b150-c100.cst	2371	2371,00	0,00%	1413,3
n300-i2-m150-b150-c1000.cst	23214	23214,00	0,00%	1912,2
n300-i2-m300-b150-c100.cst	1990	1990,00	0,00%	5671,7
n300-i2-m300-b150-c1000.cst	20568	20568,00	0,00%	5125,0
n300-i2-m300-b300-c100.cst	3301	3301,00	0,00%	19477,8
n300-i3-m150-b150-c10.cst	233	233,00	0,00%	1160,0
n300-i3-m300-b150-c10.cst	174	174,00	0,00%	3255,6
n300-i3-m300-b150-c1000.cst	18351	18351,00	0,00%	5038,3
n300-i3-m300-b300-c10.cst	274	274,00	0,00%	10180,0
n300-i3-m300-b300-c100.cst	2400	2400,00	0,00%	9030,0
n300-i3-m300-b300-c1000.cst	21938	21938,00	0,00%	12655,6
n300-i3-m300-b75-c100.cst	1218	1218,00	0,00%	1793,3
n300-i3-m300-b75-c1000.cst	9178	9178,00	0,00%	1213,3
n300-i4-m150-b150-c10.cst	224	224,00	0,00%	1498,9
n300-i4-m300-b150-c10.cst	185	185,00	0,00%	6448,9
n300-i4-m300-b150-c100.cst	1887	1887,00	0,00%	3865,6
n300-i4-m300-b150-c1000.cst	15925	15925,00	0,00%	6915,0
n300-i4-m300-b300-c10.cst	304	304,00	0,00%	22666,7
n300-i4-m300-b300-c100.cst	2922	2922,00	0,00%	21011,1
n300-i4-m300-b300-c1000.cst	27521	27521,00	0,00%	18116,7
n300-i5-m150-b150-c1000.cst	18031	18031,00	0,00%	946,4
n300-i5-m300-b150-c10.cst	171	171,00	0,00%	2628,9
n300-i5-m300-b150-c100.cst	1648	1648,00	0,00%	3626,7
n300-i6-m300-b150-c100.cst	1822	1822,00	0,00%	4503,3
n300-i6-m300-b150-c1000.cst	18178	18178,00	0,00%	4965,0
n300-i6-m300-b300-c10.cst	310	310,00	0,00%	13755,6
n300-i6-m300-b300-c1000.cst	28766	28766,00	0,00%	20433,3
n300-i6-m300-b75-c1000.cst	11137	11137,00	0,00%	869,2
n300-i7-m150-b150-c1000.cst	21637	21637,00	0,00%	1166,7
n300-i7-m300-b150-c100.cst	1710	1710,00	0,00%	4114,4
n300-i7-m300-b300-c10.cst	263	263,00	0,00%	11533,3
n300-i7-m300-b300-c100.cst	2780	2780,00	0,00%	18116,7
n300-i7-m300-b300-c1000.cst	25886	25886,00	0,00%	20033,3
n300-i7-m300-b75-c100.cst	1130	1130,00	0,00%	1310,0
n300-i8-m300-b150-c100.cst	1702	1702,00	0,00%	3621,7
n300-i8-m300-b300-c1000.cst	26027	26027,00	0,00%	14244,4
n300-i8-m300-b75-c100.cst	1148	1148,00	0,00%	1286,7
n300-i9-m300-b150-c100.cst	1761	1761,00	0,00%	5385,0
n300-i9-m300-b300-c10.cst	262	262,00	0,00%	10408,3
n300-i9-m300-b300-c100.cst	2699	2699,00	0,00%	24733,3
n300-i9-m300-b75-c100.cst	1060	1060,00	0,00%	1180,0
n558-i0-m140-b140-c100.cst	3123	3123,00	0,00%	900,2
n558-i0-m279-b279-c10.cst	363	363,00	0,00%	15277,8
n558-i0-m279-b279-c100.cst	3409	3409,00	0,00%	15900,0
n558-i0-m279-b279-c1000.cst	34141	34141,00	0,00%	17011,1
n558-i0-m558-b140-c10.cst	145	145,00	0,00%	9202,2
n558-i0-m558-b140-c100.cst	1772	1772,00	0,00%	15066,7
n558-i0-m558-b140-c1000.cst	12387	12387,00	0,00%	11225,6
n558-i0-m558-b279-c10.cst	269	269,00	0,00%	39333,3
n558-i0-m558-b279-c100.cst	2771	2771,00	0,00%	45944,4
n558-i0-m558-b558-c1000.cst	41890	41890,00	0,00%	221500,0
<b>Média</b>	<b>8052,69</b>	<b>8052,69</b>	<b>0,00%</b>	<b>12376,52</b>

Tabela A.6: Heurística Híbrida.

Instância	Melhor	Custo Médio	Dif. Percentual	Tempo Total Médio (s)	Tempo Exato Médio (s)
n300-i0-m150-b150-c100.cst	1995	1995,00	0,00%	2,1	0,2
n300-i0-m300-b150-c10.cst	194	194,44	0,23%	6,5	0,3
n300-i0-m300-b150-c1000.cst	15433	15433,00	0,00%	7,6	0,2
n300-i0-m300-b300-c10.cst	286	286,67	0,23%	14,5	1,0
n300-i0-m300-b300-c100.cst	2534	2544,22	0,40%	17,5	1,6
n300-i0-m300-b300-c1000.cst	24998	25008,22	0,04%	13,5	0,9
n300-i0-m300-b75-c100.cst	1222	1222,89	0,07%	3,1	0,1
n300-i0-m300-b75-c1000.cst	8585	8590,33	0,06%	3,1	0,1
n300-i1-m300-b150-c10.cst	178	178,22	0,12%	6,3	0,2
n300-i1-m300-b150-c1000.cst	14918	14918,33	0,00%	6,1	0,2
n300-i1-m300-b300-c100.cst	2885	2886,78	0,06%	16,2	0,9
n300-i1-m300-b75-c1000.cst	10202	10202,11	0,00%	3,5	0,1
n300-i1-m75-b75-c1000.cst	17186	17188,22	0,01%	0,3	0,1
n300-i2-m150-b150-c10.cst	233	233,11	0,05%	1,9	0,2
n300-i2-m150-b150-c100.cst	2371	2371,00	0,00%	2,4	0,6
n300-i2-m150-b150-c1000.cst	23214	23248,11	0,15%	1,9	0,3
n300-i2-m300-b150-c100.cst	1990	1990,67	0,03%	5,1	0,2
n300-i2-m300-b150-c1000.cst	20568	20582,44	0,07%	5,2	0,2
n300-i2-m300-b300-c100.cst	3301	3301,00	0,00%	15,9	0,9
n300-i3-m150-b150-c10.cst	233	233,00	0,00%	2,2	0,3
n300-i3-m300-b150-c10.cst	174	174,00	0,00%	7,9	0,3
n300-i3-m300-b150-c1000.cst	18351	18371,89	0,11%	9,0	0,2
n300-i3-m300-b300-c10.cst	274	274,67	0,24%	14,2	1,6
n300-i3-m300-b300-c100.cst	2403	2403,00	0,00%	13,1	0,8
n300-i3-m300-b300-c1000.cst	21938	21950,00	0,05%	16,0	1,0
n300-i3-m300-b75-c100.cst	1218	1218,00	0,00%	3,2	0,1
n300-i3-m300-b75-c1000.cst	9178	9178,00	0,00%	3,7	0,1
n300-i4-m150-b150-c10.cst	224	224,00	0,00%	2,0	0,3
n300-i4-m300-b150-c10.cst	186	187,22	0,66%	7,1	0,3
n300-i4-m300-b150-c100.cst	1887	1887,00	0,00%	6,3	0,3
n300-i4-m300-b150-c1000.cst	15925	15927,11	0,01%	8,3	0,3
n300-i4-m300-b300-c10.cst	305	306,33	0,44%	20,0	1,3
n300-i4-m300-b300-c100.cst	2922	2924,67	0,09%	15,7	1,3
n300-i4-m300-b300-c1000.cst	27521	27528,78	0,03%	16,7	1,4
n300-i5-m150-b150-c1000.cst	18031	18039,33	0,05%	2,0	0,3
n300-i5-m300-b150-c10.cst	171	171,11	0,06%	7,3	0,3
n300-i5-m300-b150-c100.cst	1648	1648,00	0,00%	7,0	0,4
n300-i6-m300-b150-c100.cst	1822	1822,00	0,00%	6,4	0,3
n300-i6-m300-b150-c1000.cst	18178	18192,22	0,08%	6,6	0,3
n300-i6-m300-b300-c10.cst	310	310,33	0,11%	14,1	1,2
n300-i6-m300-b300-c1000.cst	28766	28913,67	0,51%	14,2	1,1
n300-i6-m300-b75-c1000.cst	11137	11137,00	0,00%	2,3	0,1
n300-i7-m150-b150-c1000.cst	21637	21660,67	0,11%	1,7	0,2
n300-i7-m300-b150-c100.cst	1710	1710,00	0,00%	6,3	0,2
n300-i7-m300-b300-c10.cst	263	265,11	0,80%	13,8	0,9
n300-i7-m300-b300-c100.cst	2780	2796,22	0,58%	16,9	1,5
n300-i7-m300-b300-c1000.cst	25886	25898,11	0,05%	14,7	1,0
n300-i7-m300-b75-c100.cst	1145	1150,11	0,45%	2,4	0,1
n300-i8-m300-b150-c100.cst	1702	1702,00	0,00%	6,7	0,2
n300-i8-m300-b300-c1000.cst	26027	26102,44	0,29%	13,4	1,5
n300-i8-m300-b75-c100.cst	1148	1148,00	0,00%	3,8	0,1
n300-i9-m300-b150-c100.cst	1761	1773,22	0,69%	7,6	0,3
n300-i9-m300-b300-c10.cst	262	262,00	0,00%	14,8	1,1
n300-i9-m300-b300-c100.cst	2708	2710,78	0,10%	13,1	0,9
n300-i9-m300-b75-c100.cst	1060	1060,00	0,00%	2,7	0,1
n558-i0-m140-b140-c100.cst	3123	3123,00	0,00%	1,7	0,3
n558-i0-m279-b279-c10.cst	363	363,00	0,00%	12,6	1,0
n558-i0-m279-b279-c100.cst	3409	3409,00	0,00%	10,8	1,2
n558-i0-m279-b279-c1000.cst	34141	34171,56	0,09%	11,5	1,1
n558-i0-m558-b140-c10.cst	145	145,00	0,00%	21,7	0,4
n558-i0-m558-b140-c100.cst	1772	1772,00	0,00%	20,8	0,2
n558-i0-m558-b140-c1000.cst	12387	12455,67	0,55%	20,2	0,2
n558-i0-m558-b279-c10.cst	269	269,44	0,17%	44,8	1,3
n558-i0-m558-b279-c100.cst	2771	2771,00	0,00%	50,6	0,9
n558-i0-m558-b558-c1000.cst	41890	41901,56	0,03%	103,5	3,4
<b>Média</b>	<b>8053,14</b>	<b>8061,78</b>	<b>0,12%</b>	<b>11,6</b>	<b>0,6</b>

A Tabela A.7 apresenta um resumo das tabelas A.3, A.4, A.5, A.6. A linha *#Opt* apresenta o número de soluções ótimas encontradas por cada abordagem. Na linha *Score Best* é possível observar uma comparação feita entre as abordagens heurísticas em relação ao número de vezes que a melhor solução de cada heurística foi superada pelas melhores soluções das demais heurísticas, uma comparação similar a realizada em relação a solução média das abordagens (linha *Score*). A linha *MDif\** mostra a diferença percentual média entre a melhor solução da heurística e a solução ótima, enquanto *MDif* apresenta esta diferença entre a solução média de cada abordagem e a solução ótima. Por fim, a linha rotulada com *Tempo de CPU* apresenta o tempo médio em segundos gasto por cada heurística. Cada instância foi executada 9 vezes para cada um dos algoritmos.

Tabela A.7: Resumo para 65 instâncias do Grupo II.

Algoritmos	GRASP BLTO	GRASP BLTP-BLTO	HÍBRIDO	ILS BLTO
<i>#Opt</i>	63	65	60	14
<i>Score Best</i>	3	0	10	152
<i>MDif*</i>	0,01%	0,00%	0,04%	0,90%
<i>Score</i>	23	0	80	194
<i>MDif</i>	0,05%	0,00%	0,16%	1,61%
Tempo de CPU	372,15	12376,52	11,60	32,31

Na Tabela A.8 são apresentados os resultados das instâncias do Grupo III em relação ao limite superior obtido pelo modelo de Gusfield Aprimorado ( $\mathcal{F}1^+$ ). A coluna LB refere-se ao limite inferior obtido quando o modelo  $\mathcal{F}1^+$  é submetido ao resolvidor por um tempo limite de dez horas. De maneira correspondente, UB refere-se ao limite superior. Para as instâncias onde a célula da tabela correspondente ao rótulo LB está preenchida com \*, o valor UB apresentado corresponde ao custo da solução ótima para a referida instância. Para cada uma das abordagens consideradas nesta tabela, a coluna *solução* apresenta a diferença entre a média dos custos das soluções obtidas pelo método para a instância, em relação ao valor apresentado na coluna UB. Na coluna *tempo*, é apresentado o tempo médio em segundos consumido pelos algoritmos.

Tabela A.8: Comparação das heurísticas para as instâncias do Grupo III.

Instância	$\mathcal{F}1^+$		ILS		G-DATO		Híbrido		ILS tempo	
	UB	LB	solução	tempo	solução	tempo	solução	tempo	solução	tempo
scp41_25	1373	*	3,2%	55,36	2,4%	238,33	5,0%	1,80	1,3%	239,00
scp41_50	1697	*	4,0%	36,48	3,9%	237,33	6,3%	1,69	1,9%	238,00
scp41_75	2415	*	8,0%	38,74	5,2%	219,67	6,7%	1,23	3,4%	221,00
scp410_25	1558	*	3,5%	20,08	1,5%	216,00	4,4%	1,76	0,7%	217,00
scp410_50	1851	*	3,6%	30,72	4,1%	191,33	4,4%	2,00	1,9%	192,00
scp410_75	2655	*	3,8%	38,12	4,5%	195,33	5,4%	1,43	2,4%	196,00
scp42_25	1449	*	4,2%	33,76	2,2%	225,33	5,5%	1,57	1,3%	227,00
scp42_50	1722	*	4,4%	26,16	2,7%	224,00	3,8%	1,78	2,6%	224,00
scp42_75	2380	*	4,1%	36,94	4,5%	192,00	4,8%	1,15	2,7%	194,00
scp43_25	1398	1396,72	2,2%	30,62	2,2%	205,67	1,2%	1,87	2,1%	207,00
scp43_50	1709	*	1,6%	47,46	2,1%	224,67	2,5%	1,85	1,7%	227,67
scp43_75	2483	*	4,5%	32,92	5,0%	185,33	4,9%	1,55	2,1%	186,00
scp44_25	1364	*	3,7%	19,82	2,2%	200,67	3,1%	1,59	1,3%	202,00
scp44_50	1623	*	6,0%	28,30	2,4%	205,00	4,3%	1,50	1,5%	206,00
scp44_75	2398	*	3,3%	29,74	2,4%	182,33	3,0%	0,83	1,3%	184,00
scp45_25	1470	*	3,7%	32,02	2,9%	213,00	4,1%	1,67	1,6%	214,00
scp45_50	1792	*	2,8%	42,56	2,8%	221,00	7,3%	1,62	1,8%	222,00
scp45_75	2440	*	5,0%	24,70	3,3%	201,33	5,4%	1,06	1,5%	205,00
scp46_25	1503	*	3,6%	28,70	2,6%	227,00	4,1%	1,87	1,5%	228,00
scp46_50	1741	*	3,1%	32,22	3,4%	194,00	4,0%	1,57	2,5%	195,00
scp46_75	2403	*	1,8%	41,06	1,8%	192,00	5,6%	1,31	1,2%	192,00
scp47_25	1320	*	3,6%	51,56	2,8%	205,00	4,3%	1,57	2,8%	206,33
scp47_50	1620	*	4,6%	30,22	4,2%	197,67	7,0%	1,71	1,9%	200,00
scp47_75	2266	*	4,9%	55,88	5,6%	232,00	8,3%	1,47	2,2%	232,00
scp48_25	1360	*	2,2%	26,80	1,6%	189,67	3,4%	1,90	0,4%	190,00
scp48_50	1645	*	3,0%	17,72	2,5%	179,33	5,3%	1,62	1,3%	180,00
scp48_75	2416	*	4,2%	18,50	2,3%	167,67	2,8%	1,83	1,2%	170,00
scp49_25	1701	*	4,1%	33,98	2,7%	268,67	3,7%	1,92	2,5%	270,00
scp49_50	1996	*	3,9%	28,94	3,8%	227,33	4,7%	1,56	2,1%	228,00
scp49_75	2849	*	4,6%	38,22	5,5%	274,33	6,4%	1,96	2,3%	275,00
scp51_25	670	*	4,8%	83,65	3,0%	638,67	3,8%	8,87	1,8%	639,00
scp51_50	830	*	5,8%	87,85	4,0%	602,00	6,3%	5,17	3,1%	605,00
scp51_75	1118	*	5,7%	100,40	2,8%	602,67	5,5%	4,93	1,7%	609,00
scp510_25	760	*	4,4%	89,88	3,2%	717,33	5,4%	8,06	1,7%	721,00
scp510_50	929	*	1,7%	157,50	2,0%	694,00	4,4%	5,31	1,7%	703,00
scp510_75	1283	*	2,9%	133,25	2,3%	612,00	4,3%	5,33	1,6%	623,00
scp52_25	795	*	4,7%	198,50	2,6%	781,00	8,6%	6,98	2,1%	792,00
scp52_50	917	*	6,3%	95,25	2,4%	761,00	4,7%	5,60	2,5%	762,00
scp52_75	1314	*	5,2%	130,50	3,2%	550,67	6,0%	6,56	1,1%	554,00
scp53_25	683	*	6,7%	131,85	2,8%	860,33	1,9%	7,67	2,3%	862,00
scp53_50	812	*	6,4%	125,35	4,1%	705,00	5,7%	6,54	1,6%	705,00
scp53_75	1081	*	3,6%	72,05	3,1%	515,67	6,6%	5,76	2,9%	517,00
scp54_25	662	*	2,2%	177,00	2,4%	813,00	5,9%	6,33	1,7%	819,00
scp54_50	834	*	1,9%	190,00	2,2%	765,00	5,2%	6,28	1,6%	768,00
scp54_75	1174	*	4,3%	101,25	4,2%	638,33	6,3%	3,62	1,0%	646,00
scp55_25	653	651,67	2,0%	93,70	0,8%	831,00	2,9%	7,41	0,3%	843,00
scp55_50	821	*	2,1%	200,00	2,6%	684,00	4,5%	6,17	2,1%	689,00
scp55_75	1249	*	5,0%	126,00	5,6%	524,33	6,0%	6,78	4,9%	528,00
scp56_25	646	*	4,2%	151,00	1,3%	723,00	5,1%	6,65	0,6%	729,00
scp56_50	787	*	4,0%	152,50	2,6%	674,25	4,2%	6,38	2,8%	678,33

Tabela A.8: Comparação das heurísticas para as instâncias do Grupo III (continuação).

Instância	$\mathcal{F}1^+$		ILS		G-DATO		Híbrido		ILS tempo	
	UB	LB	solução	tempo	solução	tempo	solução	tempo	solução	tempo
scp56_75	1152	*	2,4%	86,85	2,7%	563,25	5,6%	3,17	1,6%	564,00
scp57_25	815	*	3,5%	122,50	1,7%	732,67	3,8%	7,12	1,3%	739,33
scp57_50	994	*	5,8%	51,90	2,8%	652,00	5,8%	5,49	3,2%	655,00
scp57_75	1461	*	6,5%	114,25	4,2%	641,00	5,7%	5,31	2,4%	648,00
scp58_25	766	*	6,9%	82,75	2,8%	721,00	4,4%	7,29	2,3%	721,00
scp58_50	913	*	4,9%	75,55	3,4%	750,75	3,3%	6,34	1,6%	751,00
scp58_75	1264	*	2,9%	119,00	2,7%	652,33	3,2%	4,70	1,3%	654,00
scp59_25	810	*	7,2%	70,50	3,5%	748,25	4,6%	9,46	1,9%	755,00
scp59_50	954	*	2,9%	107,90	2,5%	635,00	3,2%	9,24	1,2%	640,00
scp59_75	1344	*	5,4%	127,70	4,4%	577,50	4,2%	5,39	3,4%	578,00
scp61_25	339	*	4,9%	46,10	1,4%	389,25	5,3%	9,42	0,6%	391,00
scp61_50	416	*	6,0%	62,05	2,9%	332,67	6,4%	6,21	1,4%	336,00
scp61_75	609	579,01	4,7%	43,95	1,4%	263,25	1,6%	2,54	2,3%	265,00
scp62_25	356	*	4,6%	78,70	1,2%	399,67	5,6%	5,17	0,7%	404,00
scp62_50	431	*	6,7%	32,05	1,7%	327,75	4,5%	3,28	0,9%	328,00
scp62_75	638	627,31	1,3%	64,65	3,6%	294,50	4,1%	5,18	0,8%	297,00
scp63_25	366	*	9,4%	46,85	3,8%	358,00	6,3%	6,45	2,3%	359,33
scp63_50	442	*	4,5%	38,40	3,8%	359,25	4,4%	6,04	2,9%	360,00
scp63_75	605	591,50	4,2%	25,50	1,6%	253,67	3,5%	5,67	0,8%	255,00
scp64_25	347	*	2,7%	51,50	3,5%	379,50	6,7%	3,92	2,0%	385,00
scp64_50	426	*	6,0%	79,70	2,9%	358,33	5,2%	7,21	3,3%	360,00
scp64_75	581	561,72	4,4%	54,45	3,4%	274,00	3,8%	5,00	1,4%	274,00
scp65_25	409	*	2,6%	69,35	2,6%	392,00	3,9%	8,37	3,7%	397,00
scp65_50	505	490,55	5,5%	45,20	4,0%	378,33	4,4%	5,30	3,8%	386,00
scp65_75	724	687,38	3,3%	69,80	2,8%	285,33	2,2%	5,14	1,7%	287,00
scpa1_25	661	644,91	4,1%	510,50	3,6%	3023,33	3,2%	20,79	1,8%	3040,00
scpa1_50	795	775,95	3,3%	482,00	3,1%	3233,33	5,2%	26,71	1,3%	3250,00
scpa1_75	1173	1057,94	2,6%	461,50	2,6%	2193,33	4,4%	17,41	2,3%	2210,00
scpa2_25	648	*	2,3%	648,00	2,9%	2863,33	3,9%	28,19	0,5%	2870,00
scpa2_50	766	752,69	3,1%	570,50	3,5%	2213,33	7,8%	15,84	2,6%	2220,00
scpa2_75	1086	1041,56	5,6%	378,00	4,6%	2393,33	5,6%	13,99	2,4%	2430,00
scpa3_25	617	609,96	4,2%	483,00	4,0%	2923,33	3,8%	29,24	2,4%	2950,00
scpa3_50	709	695,83	5,9%	460,00	3,8%	2863,33	4,8%	18,48	0,6%	2890,00
scpa3_75	1033	983,33	6,4%	473,50	3,4%	2170,00	4,9%	18,57	3,9%	2180,00
scpa4_25	616	*	3,8%	566,00	4,2%	2910,00	5,4%	26,29	3,7%	2940,00
scpa4_50	734	719,44	3,6%	535,00	4,0%	2616,67	4,1%	19,92	2,9%	2640,00
scpa4_75	1058	987,16	5,2%	455,50	4,1%	2170,00	6,6%	10,09	2,9%	2180,00
scpa5_25	644	*	4,3%	465,00	3,5%	3250,00	3,9%	28,16	2,2%	3250,00
scpa5_50	772	761,24	4,3%	570,50	3,5%	2646,67	3,7%	21,58	2,6%	2670,00
scpa5_75	1093	1023,96	2,4%	635,50	3,8%	2183,33	5,1%	18,04	2,2%	2190,00
scpc1_25	594	569,25	3,6%	1075,00	1,9%	6790,00	1,2%	64,48	0,3%	6880,00
scpc1_50	798	681,82	0,0%	1595,00	-1,9%	8980,00	-1,6%	141,45	-4,5%	9030,00
scpc1_75	1122	940,93	-4,4%	2055,00	-2,4%	6910,00	-3,9%	46,64	-3,7%	6940,00
scpc2_25	613	555,61	-0,2%	2785,00	-0,4%	10123,33	-1,0%	63,78	-1,1%	10300,00
scpc2_50	751	642,40	-1,3%	1755,00	-1,0%	8073,33	-0,3%	99,26	-3,1%	8070,00
scpc2_75	1036	863,61	-2,4%	1385,00	-1,4%	6660,00	-2,0%	366,01	-2,7%	6730,00
scpc3_25	662	600,74	4,3%	1224,50	-0,3%	10110,00	0,2%	124,61	0,0%	10200,00
scpc3_50	772	692,92	5,6%	1200,50	0,9%	7853,33	4,3%	42,00	-0,9%	7860,00
scpc3_75	1138	929,00	-2,2%	1490,00	-2,4%	5623,33	-3,2%	147,48	-4,0%	5620,00

Tabela A.8: Comparação das heurísticas para as instâncias do Grupo III (continuação).

Instância	$\mathcal{F}1^+$		ILS		G-DATO		Híbrido		ILS tempo	
	UB	LB	solução	tempo	solução	tempo	solução	tempo	solução	tempo
scpc4_25	594	556,81	1,2%	1560,00	0,5%	10012,50	2,9%	81,42	0,2%	10100,00
scpc4_50	748	636,44	-3,1%	2580,00	-2,0%	7835,00	-0,4%	61,31	-3,1%	7910,00
scpc4_75	1071	875,47	-0,7%	932,00	-2,9%	6285,00	-1,0%	57,26	-4,4%	6350,00
scpc5_25	579	545,26	2,6%	1455,00	1,3%	8540,00	0,8%	96,29	-0,2%	8570,00
scpc5_50	751	656,24	0,6%	1615,00	-0,1%	7547,50	-0,2%	93,33	-0,8%	7570,00
scpc5_75	1053	891,60	0,8%	1160,40	0,0%	6312,50	0,5%	70,71	-1,9%	6370,00
scplr10_25	57	42,00	1,1%	2,34	-2,2%	22,30	0,6%	35988,17	-3,5%	22,40
scplr10_50	66	42,00	-3,3%	2,61	-4,5%	18,00	-4,5%	35988,93	-6,1%	18,20
scplr10_75	77	55,13	2,9%	2,79	2,3%	14,95	0,0%	35986,07	-1,3%	15,10
scplr11_25	71	33,06	-18,6%	14,60	-21,5%	157,00	-3,3%	36011,77	-21,1%	159,00
scplr11_50	105	33,18	-36,6%	14,98	-38,3%	108,75	-24,4%	36001,87	-38,7%	109,00
scplr11_75	99	33,98	-16,0%	14,50	-17,2%	83,08	-6,4%	35989,97	-17,2%	83,20
scpcyc06_25	119	*	3,5%	0,61	4,6%	2,62	0,6%	74,22	0,8%	2,63
scpcyc06_50	142	*	1,5%	0,43	0,9%	1,83	0,7%	0,17	0,0%	1,85
scpcyc06_75	170	*	0,0%	0,13	0,0%	1,49	0,0%	0,20	0,0%	1,51
scpcyc07_25	291	269,92	2,6%	6,91	2,8%	22,10	0,0%	35980,13	1,4%	22,30
scpcyc07_50	345	*	1,4%	3,35	1,3%	14,30	0,3%	0,52	0,3%	14,40
scpcyc07_75	376	*	0,0%	0,63	0,0%	8,66	0,0%	0,39	0,0%	8,73
scpcyc08_25	701	633,63	2,8%	100,45	2,4%	133,75	-0,7%	35983,77	0,7%	134,00
scpcyc08_50	825	*	1,3%	27,53	1,3%	90,08	0,1%	2,95	0,6%	90,60
scpcyc08_75	915	*	0,1%	3,54	0,0%	51,78	0,0%	3,05	0,0%	52,60
scpcyc09_25	1679	1459,80	-1,0%	1647,00	0,7%	414,75	-2,0%	36005,10	-0,9%	420,00
scpcyc09_50	1886	*	1,0%	348,13	1,3%	497,50	0,4%	22,19	0,9%	500,00
scpcyc09_75	2117	*	0,0%	41,28	0,0%	300,50	0,0%	13,31	0,0%	305,00
scpcyc10_25	4037	3262,67	-5,9%	14900,00	-5,0%	2330,00	-5,1%	36167,67	-6,4%	2340,00
scpcyc10_50	4272	*	0,9%	7556,00	1,3%	2442,50	0,2%	152,60	0,7%	2460,00
scpcyc10_75	4783	*	0,0%	181,80	0,0%	1715,00	0,0%	136,52	0,0%	1720,00
scpcyc11_25	10816	6830,56	-20,7%	14550,00	-20,7%	14550,00	-4,3%	37383,07	-22,0%	14800,00
scpcyc11_50	9596	*	1,5%	4000,00	1,5%	10200,00	0,2%	508,84	0,7%	10300,00
scpcyc11_75	10728	*	0,0%	1152,67	0,0%	10240,00	0,0%	530,54	0,0%	10300,00
scpe1_25	10	7,50	5,0%	2,54	5,0%	52,23	16,7%	9,84	6,7%	52,70
scpe1_50	12	7,23	12,5%	2,32	-8,3%	34,60	0,0%	5,59	-8,3%	35,00
scpe1_75	14	7,82	0,0%	2,58	-1,8%	19,45	2,4%	5,00	-7,1%	19,50
scpe2_25	10	7,26	5,0%	6,17	0,0%	57,68	0,0%	7,23	-10,0%	58,30
scpe2_50	11	7,26	12,7%	3,22	4,5%	40,10	9,1%	6,55	-9,1%	40,10
scpe2_75	13	7,14	12,3%	1,35	3,8%	20,60	7,7%	5,34	0,0%	20,60
scpe3_25	10	6,88	0,0%	3,78	0,0%	55,40	0,0%	7,44	-10,0%	55,70
scpe3_50	11	6,80	0,0%	2,03	-2,3%	36,40	0,0%	5,66	-9,1%	36,90
scpe3_75	13	7,00	7,7%	1,48	-1,9%	20,25	0,0%	4,71	0,0%	20,40
scpe4_25	10	7,24	0,0%	4,99	0,0%	61,28	0,0%	9,06	0,0%	61,50
scpe4_50	11	7,18	10,9%	2,43	0,0%	42,03	9,1%	6,01	0,0%	42,40
scpe4_75	13	7,27	10,8%	1,83	5,8%	22,53	7,7%	4,32	0,0%	22,60
scpe5_25	10	7,06	4,0%	3,40	0,0%	59,03	0,0%	6,59	-10,0%	59,50
scpe5_50	10	7,28	10,0%	3,17	10,0%	42,45	20,0%	6,36	10,0%	43,10
scpe5_75	13	7,12	10,8%	1,52	7,7%	23,40	10,3%	4,98	0,0%	23,60
Médias			2,8%	566,32	1,4%	1618,49	3,2%	2786,24	-0,1%	1630,78



## APÊNDICE B - Testes para calibração de parâmetros

### B.1 Parâmetro $p_1$ e $p_2$ da Heurística do Melhor Objeto

Para ajustar os parâmetros  $p_1$  e  $p_2$  da Heurística do Melhor Objeto (HMO) foram realizados testes com as instâncias dos Grupos I e II. Para cada uma das instâncias foram realizados testes com o algoritmo guloso e o randomizado. Quatro pares de valores são avaliados nestes testes iniciais.

Na comparação das abordagens propostas, para cada instância é verificada a qualidade da melhor solução conhecida, *MelhorSolução*. Então, para cada método, calcula-se o desvio percentual entre o valor médio obtido pelo método nesta instância e a *MelhorSolução*. A métrica *Desv.* representa a média dos desvios percentuais obtidos considerando todas as instâncias. Além disto, para cada método avaliado, é apresentado *#Melhor*, o número de instâncias onde a média das qualidades das soluções obtida com este método corresponde à menor média entre as abordagens comparadas. O *Score* foi calculado como definido por Ribeiro et al. [54]. Para cada instância, o *NScore* de um método  $M$  é definido como o número de métodos que obtiveram melhor solução média que a solução média de  $M$ . O *Score* de um método é a soma do *NScore* para um conjunto de instâncias. Assim, quanto menor o valor de *Score*, melhor é o algoritmo.

- Resultados para as Instâncias do Grupo I

Na Tabela B.1 são apresentados os resultados da heurística construtiva HMO para as 560 instâncias do Grupo I. O algoritmo guloso foi executado uma vez e o randomizado 10 vezes, com diferentes sementes.

Considerando as 560 instâncias, a versão gulosa da HMO foi capaz de obter 113 soluções com mesmo custo que a solução ótima, resultado que independe dos parâmetros. Já na versão randomizada, como cada instância foi executada 10 vezes, temos 5600 execuções. O percentual de soluções ótimas encontradas foi bem menor, inferior a 0,1%

Tabela B.1: Calibrando parâmetros da heurística HMO - Grupo I.

Abordagem	Métricas	$p_1 = 1$	$p_1 = 2$	$p_1 = 5$	$p_1 = 10$
		$p_2 = 0$	$p_2 = 1$	$p_2 = 1$	$p_2 = 1$
Gulosa	<i>#Opt</i>	113	113	113	113
	<i>MDif</i>	5,62	5,77	5,77	5,62
	Tempo de CPU	0,21	0,21	0,21	0,21
	<i>Score</i>	0	2	2	0
Randomizada	<i>#Opt</i>	512	512	533	488
	<i>MDif</i>	42,27	42,30	42,47	42,77
	Tempo de CPU	0,36	0,36	0,36	0,36
	<i>Score</i>	458	461	809	796

(512 = 0.091%, 533 = 0.095% e 488 = 0.087%) para todas as configurações. O comportamento quanto a diferença em relação a solução ótima também se manteve constante. A métrica *Score* também mostra o comportamento homogêneo da abordagem gulosa, mas destaca a diferença entre os resultados médios encontrados pelas heurísticas randomizadas. Isto indica que usando parâmetros (1,5) e (1,10) os resultados são inferiores.

- Resultados para as Instâncias do Grupo II

Para as 420 instâncias do Grupo II, os resultados são apresentados na Tabela B.2. Para este segundo grupo de instâncias, as heurísticas apresentaram um desempenho ruim em relação à obtenção de soluções ótimas. Considerando a abordagem gulosa, utilizando-se os parâmetros (1,0) os melhores resultados foram alcançados. Para as abordagens randomizadas, utilizando-se os parâmetros (1,0) e (10,1) obtiveram-se melhores resultados em relação à métricas *MDif* e *Score*.

Tabela B.2: Calibrando parâmetros da heurística HMO - Grupo II.

Abordagem	Métricas	$p_1 = 1$	$p_1 = 2$	$p_1 = 5$	$p_1 = 10$
		$p_2 = 0$	$p_2 = 1$	$p_2 = 1$	$p_2 = 1$
Gulosa	<i>#Opt</i>	0	0	0	0
	<i>MDif</i>	9,25	63,03	48,03	34,49
	Tempo de CPU	1,03	2,00	1,85	1,70
	<i>Score</i>	19	1260	825	413
Randomizada	<i>#Opt</i>	4	0	0	0
	<i>MDif</i>	59,14	67,03	61,58	58,21
	Tempo de CPU	1,43	2,15	2,06	1,97
	<i>Score</i>	412	1239	672	205

Para os testes apresentados no Capítulo 3 dois pares de parâmetros são utilizados (1,0) e (10,1). Estes parâmetros foram selecionados pois o primeiro obteve bons resultados

---

para os dois grupos de instâncias e o segundo se destacou nos resultados das instâncias do Grupo II na versão randomizada da heurística.



# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)