



## SELEÇÃO AUTOMÁTICA DE HEURÍSTICAS PARA ALGUNS PROBLEMAS DA OTIMIZAÇÃO COMBINATÓRIA

Luís Otávio Rigo Júnior

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

Orientador: Valmir Carneiro Barbosa

Rio de Janeiro

Julho de 2011

SELEÇÃO AUTOMÁTICA DE HEURÍSTICAS PARA ALGUNS PROBLEMAS  
DA OTIMIZAÇÃO COMBINATÓRIA

Luís Otávio Rigo Júnior

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

Examinada por:

---

Prof. Valmir Carneiro Barbosa, Ph.D.

---

Prof. Felipe Maia Galvão França, Ph.D.

---

Prof. Luís Alfredo Vidal de Carvalho, D.Sc.

---

Prof. Luiz Satoru Ochi, D.Sc.

---

Prof. Roseli Suzi Wedemann, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

JULHO DE 2011

Rigo Júnior, Luís Otávio

Seleção Automática de Heurísticas para Alguns Problemas da Otimização Combinatória/Luís Otávio Rigo Júnior. – Rio de Janeiro: UFRJ/COPPE, 2011.

XVII, 142 p.: il.; 29, 7cm.

Orientador: Valmir Carneiro Barbosa

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2011.

Referências Bibliográficas: p. 105 – 112.

1. Seleção Automática de Heurísticas. 2. Max-SAT. 3. Cobertura de Vértices Mínima. 4. Algoritmo Genético.  
I. Barbosa, Valmir Carneiro. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

*Com muito carinho, dedico este trabalho aos meus pais  
Luís Otávio Rigo e Maria Lídia Rigo  
à minha irmã Priscila R. Rigo*

# Agradecimentos

Primeiramente gostaria de agradecer ao Pai Celestial pela oportunidade de galgar mais um degrau nesta minha caminhada, a quem recorri nos momentos em que a distância dos familiares e o cansaço desta longa caminhada me faziam duvidar de minhas escolhas.

Aos meus pais, Luís Otávio e Maria Lídia, por aceitarem a minha escolha profissional, pelo incentivo que me oferecem e por toda a ajuda que me prestam em todos os momentos de minha vida. Sem esse apoio e o de minha irmã e amiga Priscila, o tempo de doutorado teria sido muito mais difícil e os momentos de dificuldades muito mais árduos.

Ao meu orientador, Valmir, pela confiança em mim depositada, pela paciência e pelas orientações, que sempre forneceram muito estímulo e segurança ao longo destes anos de estudo dedicado ao tema deste trabalho.

Aos meus grandes amigos Wilton, Ivairton e Fernanda pela amizade, companheirismo, ajuda e incentivo.

Aos colegas do LabIA, Land, AddLabs e Rio em geral, em especial ao André, Fábio, Guto, Kele e Luciana que ajudaram a sanar dúvidas e os momentos de descontração na hora do cafezinho e do almoço.

E por falar em almoço e cafezinho, não posso deixar de agradecer a Carolina, secretária do LAND, amiga e companheira de viagens (Rio-Niterói) que muito me ajudou com as correções do texto.

Aos colegas do CEUNES / UFES, que me incentivaram e tornaram o fim da minha jornada mais tranquila, dividindo comigo dúvidas, frustrações, conquistas e alegrias dos últimos momentos de minha caminhada. Em especial à Edilson, Flávio(s), Franklin, Laura, Maria, Roney, Renatinho, Rodolfo e Tatá, bons amigos e que estarão presentes nos próximos passos de minha jornada.

Aos meus alunos do CEUNES / UFES, que em alguns momentos de tensão tiveram que se contentar com aulas “meia boca”.

A todos os grandes amigos que eu fiz no rugby, em especial ao clube que eu trago no peito e defenderei sempre, UFF Rugby. Dentre todos eles, não poderia deixar de mencionar alguns dos quais convivi dentro e fora de campo, Bernardo, Brunão, Cleston, Glauber, Juanito, Leo, Lilian, Lú e Marcelinha, Mitrano, Moreno, Pardal, Raul, Serginho, Xereca e Xeroso, sempre estarão ao meu lado e em meus pensamentos.

Para finalizar, ao meu computador e aos fins de semana com chuva. Ao meu computador, por funcionar até este exato momento. Imagine: se a metade dos finais de semana que choveram aqui nesta cidade tivessem sido com sol, será que eu teria terminado este trabalho? Será? Sinceramente, não sei a resposta!!!

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

## SELEÇÃO AUTOMÁTICA DE HEURÍSTICAS PARA ALGUNS PROBLEMAS DA OTIMIZAÇÃO COMBINATÓRIA

Luís Otávio Rigo Júnior

Julho/2011

Orientador: Valmir Carneiro Barbosa

Programa: Engenharia de Sistemas e Computação

O problema de seleção de algoritmos consiste em selecionar o melhor algoritmo para uma dada instância de um problema computacional, de acordo com alguma característica da instância. Apresenta-se, nesta tese, a análise e o desenvolvimento de agentes de natureza híbrida na seleção automática de heurísticas para instâncias de problemas da otimização combinatória, especialmente os problemas Max-SAT e Cobertura de Vértices Mínima. Primeiramente, é realizada uma análise teórica e experimental de um método de seleção de algoritmos para o problema Max-SAT, que escolhe entre resolver uma instância exata ou aproximadamente, com ganho de complexidade. Algumas inconsistências na formulação original foram detectadas e corrigidas. Uma modificação do método foi proposta para resolver um problema de natureza experimental, detectado no mecanismo de seleção do método. Em sequência, foi desenvolvido um modelo para seleção automática de heurísticas para os problemas Max-SAT e Cobertura de Vértices Mínima. O seu principal componente, uma sequência de vértices a serem visitados e a ação tomada sobre eles, é chamado de heurística. A busca pela heurística que melhor resolve uma dada instância é realizada por meio de um Algoritmo Genético. Para um conjunto bem conhecido de heurísticas e instâncias de *benchmark*, demonstra-se que o modelo proposto apresenta bons resultados experimentais para os dois problemas, obtendo resultados melhores em uma fração significativa de casos.

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

AUTOMATIC SELECTION OF HEURISTICS FOR SOME COMBINATORIAL  
OPTIMIZATION PROBLEMS

Luís Otávio Rigo Júnior

July/2011

Advisor: Valmir Carneiro Barbosa

Department: Systems Engineering and Computer Science

The algorithm selection problem aims at selecting the best algorithm for a given instance of a computational problem, according/bounded to some special characteristics. The analysis and the development of hybrid agents for the automatic selection of algorithms for instance of combinatorial optimization problems, specially Max-SAT and Minimum Vertex Cover problems, is presented in this thesis. First, we provide a theoretical and an experimental analysis of a selection method of algorithms related to the problem Max-SAT, where a problem instance is solved exactly or approximately, with gain of complexity. Some inconsistencies in the original formulation were detected and corrected. A modification was proposed to solve a detected experimental problem in the selection mechanism. Thereafter, an experimental model was developed for automatic selection of heuristics for Max-SAT and Minimum Vertex Cover problems. Its main component, the heuristic, is a sequence of vertices to be visited and the action taken on them. The search for the best heuristic that solves a given instance is performed using a genetic algorithm. For a number of well established Max-SAT and Minimum Vertex Cover heuristics and benchmark instances, we demonstrate that the proposed model outperforms the related work in a significant fraction of the cases.



# Sumário

|  |            |
|--|------------|
| <b>Lista de Figuras</b>                                      | <b>xii</b> |
| <b>Lista de Tabelas</b>                                      | <b>xiv</b> |
| <b>1 Introdução</b>  | <b>1</b>   |
| 1.1 Motivação . . . . .                                      | 1          |
| 1.2 Alguns Trabalhos Relacionados . . . . .                  | 3          |
| 1.3 Organização . . . . .                                    | 6          |
| <b>2 Problemas Abordados</b>                                 | <b>8</b>   |
| 2.1 Problema Max-SAT . . . . .                               | 8          |
| 2.1.1 Definição do Problema . . . . .                        | 10         |
| 2.1.2 Maximização por Análise Local . . . . .                | 11         |
| 2.1.3 Trabalhos Relacionados . . . . .                       | 12         |
| 2.2 Problema da Cobertura de Vértices Mínima (CVM) . . . . . | 16         |
| 2.2.1 Definição do Problema . . . . .                        | 16         |
| 2.2.2 Minimização por Análise Local . . . . .                | 17         |
| 2.2.3 Trabalhos Relacionados . . . . .                       | 17         |
| <b>3 Seletor Exato vs. Aproximado para Max-E3-SAT</b>        | <b>22</b>  |
| 3.1 Descrição da Abordagem . . . . .                         | 23         |
| 3.1.1 Rotina de Seleção . . . . .                            | 25         |
| 3.1.2 Método Exato . . . . .                                 | 26         |
| 3.1.3 Método Aproximado . . . . .                            | 26         |
| 3.2 Experimentos Computacionais . . . . .                    | 29         |
| 3.2.1 Parâmetros e Dados . . . . .                           | 29         |

|          |  |            |
|----------|--|------------|
| 3.2.2    | Ambiente . . . . .   | 30         |
| 3.2.3    | Resultados Experimentais . . . . .                                   | 30         |
| 3.2.4    | Nova Formulação . . . . .  | 35         |
| 3.3      | Considerações Finais . . . . .                                       | 37         |
| <b>4</b> | <b>Seleção por Algoritmo Genético para Max-SAT (<i>SAHe-SAT</i>)</b> | <b>39</b>  |
| 4.1      | Descrição da Abordagem . . . . .                                     | 40         |
| 4.1.1    | Fuga de Máximos Locais de Baixa Qualidade . . . . .                  | 41         |
| 4.1.2    | Exploração do Espaço de Estados . . . . .                            | 43         |
| 4.2      | Experimentos Computacionais . . . . .                                | 49         |
| 4.2.1    | Heurísticas de Pós-Processamento . . . . .                           | 50         |
| 4.2.2    | Ambiente . . . . .   | 53         |
| 4.2.3    | Calibração do Modelo . . . . .                                       | 53         |
| 4.2.4    | Resultados Experimentais . . . . .                                   | 59         |
| 4.3      | Considerações Finais . . . . .                                       | 69         |
| <b>5</b> | <b>Seleção por Algoritmo Genético para CVM (<i>SAHe-CVM</i>)</b>     | <b>70</b>  |
| 5.1      | Descrição da Abordagem . . . . .                                     | 71         |
| 5.1.1    | Fuga de Máximos Locais de Baixa Qualidade . . . . .                  | 71         |
| 5.1.2    | Exploração do Espaço de Estados . . . . .                            | 72         |
| 5.2      | Experimentos Computacionais . . . . .                                | 74         |
| 5.2.1    | Heurísticas de Pós-Processamento Criadas . . . . .                   | 75         |
| 5.2.2    | Ambiente . . . . .   | 76         |
| 5.2.3    | Calibração do Modelo . . . . .                                       | 77         |
| 5.2.4    | Resultados Experimentais . . . . .                                   | 81         |
| 5.3      | Considerações Finais . . . . .                                       | 99         |
| <b>6</b> | <b>Considerações, Conclusões e Trabalhos Futuros</b>                 | <b>101</b> |
| 6.1      | Trabalhos Futuros . . . . .  | 103        |
|          | <b>Referências Bibliográficas</b>                                    | <b>105</b> |
| <b>A</b> | <b>Instâncias de <i>Benchmark</i> para o Problema <i>Max-SAT</i></b> | <b>113</b> |
| <b>B</b> | <b>Resultados Experimentais Completos do <i>SAHe-SAT</i></b>         | <b>118</b> |

|   |   |     |
|---|---|-----|
| C | Instâncias de <i>Benchmark</i> para o Problema <i>CVM</i> | 131 |
| D | Resultados Experimentais Completos do <i>SAHe-CVM</i>     | 134 |

# Lista de Figuras

|     |  |    |
|-----|--|----|
| 3.1 | Taxa de aproximação obtida nos experimentos computacionais com o seletor Exato/Aproximado, para valores de $\varepsilon$ e $n$ . . . . .   | 31 |
| 3.2 | Taxa de satisfatibilidade do conjunto de instâncias para Max-E3-SAT. . . . .   | 32 |
| 3.3 | Tempo de execução do algoritmo exato para Max-E3-SAT. . . . .  | 33 |
| 3.4 | Região de decisão do <i>seletor</i> entre os métodos exato e aproximado. Cada curva representa o número de instâncias em que o seletor decidiu efetuar o cálculo aproximado. . . . . | 34 |
| 3.5 | Região de decisão do <i>novo seletor</i> entre os métodos exato e aproximado. . . . .  | 37 |
| 4.1 | Exemplo de indivíduo para uma instância com 5 variáveis. . . . .   | 44 |
| 4.2 | Exemplo de aplicação do <i>crossover</i> de 2 pontos para $T_d = 3$ . . . . .  | 48 |
| 4.3 | Exemplo de mutação no indivíduo $P_1$ , gerando um novo indivíduo $F_1$ . . . . .  | 49 |
| 4.4 | Evolução dos resultados para uma execução do <i>SAHe-SAT+Hp</i> (curva em vermelho) e da <i>Hp-Pura</i> , com reinícios aleatórios (curva em preto). . . . .                         | 57 |
| 4.5 | Evolução dos resultados para uma execução do <i>SAHe-SAT+Hp</i> (curva em vermelho) e da <i>Hp-Pura</i> , com reinícios aleatórios (curva em preto). (continuação) . . . . .         | 58 |
| 4.6 | Comparação dos valores obtidos entre <i>SAHe-SAT+Hp's</i> e <i>Hp's-puras</i> , sobre as instâncias do conjunto de 2004 e de 2008, cujo valor ótimo é desconhecido. . . . .          | 65 |
| 4.7 | Comparação dos valores obtidos entre <i>SAHe-SAT+Hp's</i> e <i>Hp's-puras</i> , sobre todas as 100 instâncias do conjunto de 2004 e todas as 112 de 2008. . . . .                    | 67 |

|      |  |    |
|------|--|----|
| 4.8  | Distribuição acumulada da fração das instâncias (eixo vertical) com razão $\leq a X$ , onde $X$ é a fração entre o número de passos do melhor resultado do <i>SAHe-SAT+Hp</i> e o número de passos do melhor resultado <i>Hp-pura</i> (eixo horizontal). . . . .                 | 67 |
| 4.9  | Distribuição acumulada da fração das instâncias (eixo vertical) com razão $\leq a X$ , onde $X$ é a fração entre o número de passos do melhor resultado do <i>SAHe-SAT+Hp</i> e o número de passos do melhor resultado <i>Hp-pura</i> (eixo horizontal). (continuação) . . . . . | 68 |
| 5.1  | Comparação dos valores obtidos entre <i>SAHe-CVM+Hp's</i> e <i>Hp's-puras</i> , sobre as instâncias do conjunto de 1996, cujo valor ótimo é desconhecido. . . . .  | 85 |
| 5.2  | Comparação dos valores obtidos entre <i>SAHe-SAT+Hp's</i> e <i>Hp's-puras</i> , sobre todas as 80 instâncias do conjunto de 1996 e todas as 41 de 2010. . . . .  | 86 |
| 5.3  | Comparação dos valores obtidos entre <i>SAHe-SAT+Hp's</i> e <i>Hp's-puras</i> , sobre todas as 80 instâncias do conjunto de 1996 e todas as 41 de 2010. (continuação) . . . . .  | 87 |
| 5.4  | Comparação dos valores obtidos entre <i>Squeeze</i> e <i>SAHe-CVM</i> . . . . .  | 89 |
| 5.5  | Comparação dos valores obtidos entre <i>MIPO</i> e <i>SAHe-CVM</i> . . . . .   | 90 |
| 5.6  | Comparação dos valores obtidos entre <i>RB-clique</i> e <i>SAHe-CVM</i> . . . . .  | 90 |
| 5.7  | Comparação dos valores obtidos entre <i>XSD</i> e <i>SAHe-CVM</i> . . . . .  | 93 |
| 5.8  | Comparação dos valores obtidos entre <i>CBH</i> e <i>SAHe-CVM</i> . . . . .  | 93 |
| 5.9  | Comparação dos valores obtidos entre <i>ATA</i> e <i>SAHe-CVM</i> . . . . .  | 95 |
| 5.10 | Comparação dos valores obtidos entre <i>OCH</i> e <i>SAHe-CVM</i> . . . . .  | 97 |
| 5.11 | Comparação dos valores obtidos entre <i>WAO</i> e <i>SAHe-CVM</i> . . . . .  | 97 |
| 5.12 | Comparação dos valores obtidos entre <i>DLS-MC</i> e <i>SAHe-CVM</i> . . . . .   | 99 |

# Lista de Tabelas

|      |  |    |
|------|--|----|
| 4.1  | Instâncias utilizadas para calibração do <i>SAHe-SAT</i> . . . . .   | 53 |
| 4.2  | Resultados do <i>SAHe-SAT+Hp</i> 's para conjunto de calibração. . . . .   | 54 |
| 4.3  | Tempo (em minutos) do <i>SAHe-SAT+Hp</i> 's para conjunto de calibração. . . . .   | 55 |
| 4.4  | Resultados das <i>Hp</i> 's para conjunto de calibração. . . . .   | 56 |
| 4.5  | Tempo (em minutos) das <i>Hp</i> 's para conjunto de calibração. . . . .   | 56 |
| 4.6  | Resultados do <i>SAHe-SAT</i> e <i>Hp = Adapt novelty+</i> . . . . .   | 61 |
| 4.7  | Resultados do <i>SAHe-SAT</i> e <i>Hp = Novelty</i> . . . . .  | 62 |
| 4.8  | Resultados do <i>SAHe-SAT</i> e <i>Hp = Walksat-tabu</i> . . . . .   | 63 |
| 4.9  | Taxa de sucesso do <i>SAHe-SAT</i> , por tempo limite, sobre as instâncias do conjunto de 2004 e de 2008, cujo valor ótimo é desconhecido. . . . . | 64 |
| 4.10 | Taxa de sucesso do <i>SAHe-SAT</i> , por tempo limite, sobre todas as 100 instâncias do conjunto de 2004 e todas as 112 de 2008. . . . .           | 66 |
| 5.1  | Resultados do alg. <i>gcv</i> m para a instância <b>brock200_4</b> . . . . .   | 78 |
| 5.2  | Resultados do <i>SAHe-CVM+gcv</i> m para algumas combinações de parâmetros do AG <b>brock200_4</b> . . . . .                                       | 79 |
| 5.3  | Resultados do <i>SAHeCVM+gcv</i> m para <b>brock200_4</b> . . . . .  | 80 |
| 5.4  | Resultados do <i>SAHe-CVM</i> e <i>Hp = gcv</i> m. . . . .   | 82 |
| 5.5  | Resultados do <i>SAHe-CVM</i> e <i>Hp = gwcv</i> m. . . . .  | 83 |
| 5.6  | Resultados do <i>SAHe-CVM</i> e <i>Hp = hcvm</i> . . . . .   | 83 |
| 5.7  | Resultados do <i>SAHe-CVM</i> e <i>Hp = hwcv</i> m. . . . .  | 84 |
| 5.8  | Taxa de sucesso do <i>SAHe-CVM</i> por tempo limite, sobre as instâncias do conjunto de 1996 sem valor ótimo conhecido. . . . .                    | 85 |

|      |   |     |
|------|---|-----|
| 5.9  | Taxa de sucesso do <i>SAHe-CVM</i> por tempo limite, sobre todas as instâncias do conjunto de 1996 e todas as instâncias do conjunto de 2010. . . . . | 86  |
| 5.10 | Comparação de resultados entre <i>Squeeze</i> e <i>SAHe-CVM</i> . . . . .   | 88  |
| 5.11 | Comparação de resultados entre <i>MIPO</i> e <i>SAHe-CVM</i> . . . . .  | 90  |
| 5.12 | Comparação de resultados entre <i>RB-clique</i> e <i>SAHe-CVM</i> . . . . .   | 91  |
| 5.13 | Comparação de resultados entre <i>XSD</i> e <i>SAHe-CVM</i> . . . . .   | 92  |
| 5.14 | Comparação de resultados entre <i>CBH</i> e <i>SAHe-CVM</i> . . . . .   | 94  |
| 5.15 | Comparação de resultados entre <i>ATA</i> e <i>SAHe-CVM</i> . . . . .   | 95  |
| 5.16 | Comparação de resultados entre <i>OCH</i> e <i>SAHe-CVM</i> . . . . .   | 96  |
| 5.17 | Comparação de resultados entre <i>WAO</i> e <i>SAHe-CVM</i> . . . . .   | 98  |
| A.1  | Conjunto de instâncias de 2004. . . . .   | 114 |
| A.2  | Conjunto de instâncias de 2004. (continuação) . . . . .   | 115 |
| A.3  | Conjunto de instâncias de 2008. . . . .   | 116 |
| A.4  | Conjunto de instâncias de 2008. (continuação) . . . . .   | 117 |
| B.1  | Resultados completos do <i>SAHe-SAT</i> e $H_p = \textit{Adaptnovelty+}$ , para instâncias do conjunto de 2004. . . . .                               | 119 |
| B.2  | Resultados completos do <i>SAHe-SAT</i> e $H_p = \textit{Adaptnovelty+}$ , para instâncias do conjunto de 2004. (continuação) . . . . .               | 120 |
| B.3  | Resultados completos do <i>SAHe-SAT</i> e $H_p = \textit{Adaptnovelty+}$ , para instâncias do conjunto de 2008. . . . .                               | 121 |
| B.4  | Resultados completos do <i>SAHe-SAT</i> e $H_p = \textit{Adaptnovelty+}$ , para instâncias do conjunto de 2008. (continuação) . . . . .               | 122 |
| B.5  | Resultados completos do <i>SAHe-SAT</i> e $H_p = \textit{Novelty}$ , para instâncias do conjunto de 2004. . . . .                                     | 123 |
| B.6  | Resultados completos do <i>SAHe-SAT</i> e $H_p = \textit{Novelty}$ , para instâncias do conjunto de 2004. (continuação) . . . . .                     | 124 |
| B.7  | Resultados completos do <i>SAHe-SAT</i> e $H_p = \textit{Novelty}$ , para instâncias do conjunto de 2008. . . . .                                     | 125 |
| B.8  | Resultados completos do <i>SAHe-SAT</i> e $H_p = \textit{Novelty}$ , para instâncias do conjunto de 2008. (continuação) . . . . .                     | 126 |

|      |  |     |
|------|--|-----|
| B.9  | Resultados completos do <i>SAHe-SAT</i> e $Hp = Walksat-tabu$ , para instâncias do conjunto de 2004. . . . .                 | 127 |
| B.10 | Resultados completos do <i>SAHe-SAT</i> e $Hp = Walksat-tabu$ , para instâncias do conjunto de 2004. (continuação) . . . . . | 128 |
| B.11 | Resultados completos do <i>SAHe-SAT</i> e $Hp = Walksat-tabu$ , para instâncias do conjunto de 2008. . . . .                 | 129 |
| B.12 | Resultados completos do <i>SAHe-SAT</i> e $Hp = Walksat-tabu$ , para instâncias do conjunto de 2008. (continuação) . . . . . | 130 |
| C.1  | Conjunto de instâncias de 1996. . . . .  | 132 |
| C.2  | Conjunto de instâncias de 1996. (continuação) . . . . .  | 133 |
| C.3  | Conjunto de instâncias de 2010. . . . .  | 133 |
| D.1  | Resultados completos do <i>SAHe-CVM</i> e $Hp = gcvm$ , para instâncias do conjunto de 1996. . . . .                         | 135 |
| D.2  | Resultados completos do <i>SAHe-CVM</i> e $Hp = gcvm$ , para instâncias do conjunto de 1996. (continuação) . . . . .         | 136 |
| D.3  | Resultados completos do <i>SAHe-CVM</i> e $Hp = gcvm$ , para instâncias do conjunto de 2010. . . . .                         | 136 |
| D.4  | Resultados completos do <i>SAHe-CVM</i> e $Hp = gwcv$ , para instâncias do conjunto de 1996. . . . .                         | 137 |
| D.5  | Resultados completos do <i>SAHe-CVM</i> e $Hp = gwcv$ , para instâncias do conjunto de 1996. (continuação) . . . . .         | 138 |
| D.6  | Resultados completos do <i>SAHe-CVM</i> e $Hp = gwcv$ , para instâncias do conjunto de 2010. . . . .                         | 138 |
| D.7  | Resultados completos do <i>SAHe-CVM</i> e $Hp = hcvm$ , para instâncias do conjunto de 1996. . . . .                         | 139 |
| D.8  | Resultados completos do <i>SAHe-CVM</i> e $Hp = hcvm$ , para instâncias do conjunto de 1996. (continuação) . . . . .         | 140 |
| D.9  | Resultados completos do <i>SAHe-CVM</i> e $Hp = hcvm$ , para instâncias do conjunto de 2010. . . . .                         | 140 |
| D.10 | Resultados completos do <i>SAHe-CVM</i> e $Hp = hwcv$ , para instâncias do conjunto de 1996. . . . .                         | 141 |



|  |     |
|--|-----|
| D.11 Resultados completos do <i>SAHe-CVM</i> e $Hp = hwcv$ , para instâncias<br>do conjunto de 1996. (continuação) . . . . . | 142 |
| D.12 Resultados completos do <i>SAHe-CVM</i> e $Hp = hwcv$ , para instâncias<br>do conjunto de 2010. . . . .                 | 142 |

# Capítulo 1

## Introdução

Tipicamente, problemas de otimização combinatória são de difícil solução e estão relacionados a projetos de redes de computadores, inteligência artificial e outros campos da ciência. O uso de agentes híbridos é uma forma de utilizar as potencialidades dos métodos existentes para resolução destes problemas.

O tema desta tese é o estudo e o desenvolvimento de agentes de natureza híbrida na seleção automática de heurísticas (algoritmos) para instâncias de problemas da otimização combinatória, especialmente os problemas de *Maximização da Satisfatibilidade* (Max-SAT) de fórmulas Booleanas e da *Cobertura de Vértices Mínima* (CVM) em grafos. A partir do estudo inicial das técnicas de seleção de heurísticas existentes [1–4], concebeu-se um novo modelo capaz de selecionar automaticamente, por meio de um *algoritmo genético evolucionário*, a heurística mais adequada para a resolução de uma determinada instância do problema em questão. Essa *heurística* consiste de uma sequência de variáveis a serem visitadas e nas ações tomadas sobre elas, com o propósito de maximizar (ou minimizar) uma função local.

### 1.1 Motivação

Os problemas Max-SAT e CVM foram escolhidos por sua grande importância teórica e prática, e, com frequência, relacionam-se a problemas reais. Esta importância deve-se ao fato de serem problemas NP-difíceis clássicos, que possuem formulações simples e para os quais, ainda hoje, existem instâncias sem solução ótima conhecida. Além disso, estão estritamente relacionados a outros problemas,

de igual importância, tais como *satisfação de restrições* em inteligência artificial, *clique máxima* e *conjunto independente máximo* em grafos. Desta forma, a criação de novos métodos ou a melhora do desempenho alcançado pelos métodos existentes continua sendo um objetivo desafiador e de retorno promissor.

Ao longo dos anos, vários métodos exatos e heurísticos foram desenvolvidos para tratar das instâncias desses problemas. Em geral, cada método possui maneira própria de realizar a busca por uma solução, explorando diferentes características do problema e, conseqüentemente, obtendo desempenho diferente dos demais [5–7]. Além disso, o desempenho de cada algoritmo depende do aspecto sobre o qual ele está sendo avaliado.

A qualidade da solução encontrada ou a quantidade de recursos necessários para encontrá-la são, geralmente, utilizados na avaliação dos algoritmos, e funcionam como critérios objetivos em sua escolha para resolver um dado problema. Entretanto, apesar de alguns algoritmos apresentarem desempenho superior aos demais (em um ou mais aspectos), raramente existe um algoritmo melhor que todos os outros para todas as instâncias de um dado problema [1, 8, 9]. Ainda, habitualmente, cada algoritmo alcança melhor desempenho quando aplicado a um conjunto particular de instâncias do problema [10].

Este fenômeno é comum entre algoritmos para resolução dos problemas abordados nesta tese. Pois, para estes problemas, o tempo de execução dos algoritmos varia muito de instância para instância, e, pequenas alterações no tamanho da instância ou em sua estrutura interna, podem resultar em uma variação significativa no seu desempenho, causando, inclusive, uma divisão entre instâncias: aquelas que podem e as que não podem ser resolvidas por um determinado algoritmo, dado os recursos computacionais específicos e o tempo que se deseja utilizar [11].

Outro fator que influencia na escolha do melhor algoritmo é a dificuldade em saber, *a priori*, que tipo de instâncias serão geradas, como a computação irá proceder, ou, inclusive, quão eficientemente o hardware específico suportará as especificações dos diferentes algoritmos. Por vezes, um estudo cuidadoso pode revelar que a procura por um algoritmo com uma característica específica é mais produtiva do que a procura pelo melhor algoritmo para o problema [12].

De uma maneira geral, há um consenso sobre a necessidade de se encontrar o

algoritmo ideal para melhor resolver uma instância ou um conjunto de instâncias do problema. RICE [8] foi o primeiro pesquisador a apontar e descrever formalmente este problema como um problema computacional, definindo-o como *Problema de Seleção de Algoritmos* (PSA). Identificou, também, o objetivo da seleção: “*um mapeamento do espaço de instâncias para o espaço dos algoritmos, com objetivo de maximizar alguma medida de desempenho*” [8]. Em seu trabalho, RICE realizou a modelagem do problema e apontou as principais questões envolvidas no processo, não realizando, porém, experimentos computacionais. Apesar disso, as pesquisas subsequentes são derivadas do seu trabalho.

Dentre as questões levantadas por RICE [8], é possível destacar as questões que envolvem dois componentes implícitos no processo de resolução do PSA. O primeiro deles é a definição de quais características presentes nas instâncias do problema podem contribuir no processo de escolha do melhor algoritmo. O segundo componente, consiste na definição do conjunto de critérios utilizados para avaliar a solução proposta, tais como: *velocidade*, *exatidão*, *eficiência* e *facilidade de uso* dos algoritmos.

Ao visualizar o PSA como um *framework* de modelagem dos problemas sobre a teoria de aproximação, como propõe RICE [8], surgem outras questões que devem ser observadas, como *grau de convergência*, *complexidade* e *robustez* das soluções encontradas. Não menos importantes, estas questões também ajudam a compreender melhor os problemas, as instâncias e os algoritmos.

## 1.2 Alguns Trabalhos Relacionados

A abordagem mais comum para realizar a seleção de algoritmos é a medição do desempenho de diferentes métodos sobre um determinado conjunto de instâncias do problema, para então, utilizar-se somente o algoritmo com o menor tempo de execução médio. A esse processo dá-se o nome *vencedor-leva-tudo*. LEYTON-BROWN *et al.* [9] e LAGOUDAKIS *et al.* [12] são exemplos de trabalhos que utilizam este tipo de abordagem.

Uma alternativa à seleção de um único algoritmo é a *execução paralela*. Em alguns domínios, esta execução paralela pode ser uma técnica muito efetiva. Por exemplo, GOMES e SELMAN [13] propõem a execução paralela de um conjunto

de algoritmos para resolver o problema de satisfação de restrições. Esta abordagem será analisada mais detalhadamente na Subseção 2.1.3, onde serão discutidos alguns trabalhos sobre o problema Max-SAT.

Outra abordagem para o PSA é a *classificação*. Qualquer algoritmo de classificação padrão (uma árvore de decisão, por exemplo) pode ser utilizado para identificar, por meio de aprendizado, quais algoritmos escolher para a resolução de um problema, dadas as características da instância e os exemplos de treinamento. Os erros obtidos na classificação dos exemplos, geralmente, são usados como medida de desempenho de tais algoritmos. Todavia, para LEYTON-BROWN *et al.* [9], esta medida de desempenho é inadequada, uma vez que penaliza igualmente a classificação incorreta de exemplos, desconsiderando o seu custo. Quando este custo está relacionado ao tempo de execução do algoritmo, LEYTON-BROWN *et al.* afirmam que a penalização para uma classificação errada deve ser maior quando a diferença de tempo entre os algoritmos escolhidos for grande.

A escolha de um método apropriado para resolução de problemas, dentre os disponíveis, é uma habilidade crucial para especialistas em muitas áreas. Nesse sentido, FINK [14] descreve uma técnica para a seleção automática de algoritmos baseada na análise estatística do histórico de desempenho. Ele formaliza o problema estatístico envolvido na seleção de um algoritmo geral e eficiente para resolver todas as instâncias de um problema. Também, descreve um método de seleção de algoritmos capaz de escolher entre métodos disponíveis e decidir, através de uma técnica estatística, quando abandonar a escolha, caso fique provado que levará muito tempo para encontrar a solução. FINK [14] demonstra, assim, que determinar um limite apropriado para abandonar uma escolha mal feita é tão crucial para resolver eficientemente uma instância quanto escolher o método certo.

Em trabalho igualmente interessante, RUSSELL e SUBRAMANIAN [15] formalizam um problema geral de seleção de algoritmos e usam programação dinâmica para solucionar casos especiais. Noutro, MINTON [16] desenvolve um sistema de aprendizado indutivo que resolve instâncias do problema de satisfação de restrições pela seleção de uma dentre diversas estratégias de busca alternativas. Esta última abordagem será analisada mais detalhadamente na Subseção 2.1.3, onde serão tratados alguns trabalhos sobre o problema Max-SAT.

Em [1], VASSILEVSKA *et al.* apresentam um método de seleção de algoritmos com base na fundamentação teórica da complexidade de vários problemas NP-difíceis. Neste método, uma rotina de seleção, chamada de *seletor*, realiza a análise inicial de uma instância do problema. Através dessa análise, o seletor decide por resolver a instância de maneira exata ou por efetuar o cálculo aproximado para solucioná-la. O método de seleção adotado garante que, quando o seletor optar pela solução exata, o tempo gasto para sua resolução será sub-exponencial. Do contrário, o cálculo aproximado terá garantia de aproximação estritamente superior ao conhecido ou conjecturado para o problema, assumindo que  $P \neq NP$ .

GUO [2], em sua tese de doutorado, apresenta alguns resultados teóricos sobre o PSA, demonstrando que não existe um programa para selecionar automaticamente um algoritmo baseado somente na descrição da instância e dos algoritmos. Para tanto, GUO descreve um *framework* teórico abstrato sobre o desempenho de algoritmos e instâncias difíceis, baseado na complexidade de Kolmogorov, para mostrar que o problema geral de seleção de algoritmos é incomputável. Guiado por estes resultados teóricos, GUO propõe uma abordagem indutiva baseada em aprendizado de máquina para resolver o PSA.

É importante destacar o trabalho de ROPKE e PISINGER [3], que desenvolveram um *framework* para resolução de problemas de roteamento de veículos chamado ALNS (*Adaptive Large Neighborhood Search*). O ALNS é uma extensão do LNS proposto por SHAW [17], com camada adaptativa. O *framework* possui um conjunto de heurísticas de busca local responsável pela inserção e remoção de arestas no grafo (que representam rotas no problema de roteamento). Tais heurísticas atuam de forma competitiva e são usadas com frequência correspondente ao seu histórico de desempenho. Esta frequência de uso de uma heurística é definida pela camada adaptativa que atua como um intensificador ou diversificador da busca, através de penalização ou valorização das heurísticas. O ALNS tem gerado boas soluções e, atualmente é considerado o melhor *framework* para diversos problemas de roteamento de veículos [18].

## 1.3 Organização

O Capítulo 2 fornece uma breve definição de problemas de otimização combinatória para os quais serão desenvolvidas soluções inspiradas no PSA, bem como trata dos trabalhos relacionados a este tipo de problema.

A primeira abordagem, apresentada no Capítulo 3, consiste na análise teórica e experimental do método de seleção de algoritmos criado por VASSILEVSKA *et al.* [1], especificamente para o problema Max-SAT. Vassilevska desenvolveu uma rotina de seleção capaz de escolher, para uma dada instância, entre a execução de um algoritmo exato (com garantia de tempo sub-exponencial) e o cálculo aproximado (com garantia de aproximação estritamente superior ao conhecido para o problema). Pelo estudo desse método de seleção, verificou-se que existiam algumas inconsistências na análise de complexidade do método para o problema Max-SAT. Assim, a análise correta foi desenvolvida, realizando-se, conseqüentemente, uma série de experimentos computacionais.

Apesar de o método criado por VASSILEVSKA *et al.* [1] possuir interessantes propriedades teóricas, foi comprovado, através de experimentos, que o mesmo não possui um bom desempenho prático. Propôs-se, assim, uma modificação no modelo a fim de melhorar o desempenho prático do seletor.

Com base no estudo realizado sobre essa primeira abordagem, decidiu-se desenvolver um novo método para resolução desse e de outros problemas de otimização, também classificados como NP-difíceis. Com este objetivo, no Capítulo 4, foi desenvolvido um método de seleção automática de heurísticas para o problema Max-SAT, chamado de *SAHe-SAT*. O principal componente, chamado de heurística, consiste na definição da sequência de variáveis a serem visitadas e na definição das ações tomadas sobre elas. Com base nesta ação, e a partir da análise do estado da variável e da sua vizinhança, o resolvedor decide qual será o novo estado da variável. Após a execução da heurística, não existe garantia de que uma solução ótima para a instância tenha sido gerada e, deste modo, torna-se necessária a aplicação de uma rotina de pós-processamento para finalizar o processo de busca. Outro componente importante do método é o mecanismo de computação evolucionária, responsável pela exploração do espaço de heurísticas.

Este método, em conjunto com um pacote de heurísticas (UBCSAT [19]), foi

aplicado sobre instâncias industriais de competições, de análise de desempenho experimental sobre os problemas SAT [20] e Max-SAT [21]. Os resultados dos experimentos computacionais mostram que, em alguns casos, esta abordagem apresenta melhores resultados do que aqueles apresentados por uma heurística pura, tanto em relação ao tempo, quanto à solução encontrada.

Com o objetivo de testar a generalidade do método proposto, optou-se por aplicá-lo a um outro problema que apresenta características semelhantes ao Max-SAT e que também pode beneficiar-se do modelo criado. Para tanto, foi selecionado o problema da Cobertura de Vértices Mínima (CVM) em grafos. O Capítulo 5 descreve o modelo adaptado ao problema CVM, chamado de *SAHe-CVM*. Os resultados dos experimentos computacionais com o *SAHe-CVM* confirmam a melhora nos resultados obtidos por algumas heurísticas puras. É possível verificar também, que estes resultados são bons, quando comparados aos resultados obtidos por outros métodos heurísticos conhecidos (heurísticas descritas no *Second DIMACS Challenge* [22]), servindo, mesmo que de forma indireta, como base de referência para análise de desempenho [23].

Por fim, no Capítulo 6, são realizadas discussões sobre os resultados teóricos e experimentais encontrados, apresentando suas conclusões e proposta de trabalhos futuros.



# Capítulo 2

## Problemas Abordados

Existe uma série de problemas fáceis<sup>1</sup> que pode beneficiar-se com a aplicação de rotinas de seleção de algoritmos, obtendo melhores resultados (sob um ou mais aspectos) que aqueles encontrados pela aplicação de um algoritmo específico. Entretanto, a classe de problemas NP-difíceis compreende problemas de relevante importância teórica e prática, e, geralmente, está relacionada a muitos problemas reais. Ao longo dos anos, muitos métodos exatos e heurísticos foram desenvolvidos para tratar de instâncias difíceis<sup>2</sup> dos problemas dessa classe. A criação de novos mecanismos ou a melhora do desempenho alcançado por esses métodos continua sendo um objetivo desafiador e de retorno promissor.

Neste capítulo, serão apresentados dois problemas de otimização combinatória, comprovadamente NP-difíceis, para os quais serão desenvolvidas soluções inspiradas no PSA. São eles: *Maximização da Satisfatibilidade* (Max-SAT) e *Cobertura de Vértices Mínima* (CVM). Os trabalhos relacionados aos tópicos abordados serão tratados ainda neste capítulo.

### 2.1 Problema Max-SAT

Historicamente, um dos mais importantes problemas NP-Completo é determinar se as variáveis Booleanas de uma dada fórmula Booleana podem receber atribuições de valores-verdade (*verdadeiro* ou *falso*), de maneira que todas as suas restrições

---

<sup>1</sup>O termo fácil é aplicado aos problemas pertencentes à classe P (polinomial), para os quais se conhece pelo menos um algoritmo polinomial capaz de resolvê-lo.

<sup>2</sup>O termo difícil é aplicado, aqui, para instâncias cujo valor ótimo é desconhecido.

sejam satisfeitas – o problema da *Satisfatibilidade* (ou simplesmente SAT) –, onde: uma restrição, ou cláusula, é um conjunto de literais (variável ou sua negação); e uma cláusula Booleana é satisfeita quando existe pelo menos um literal *verdadeiro* nela. Em outras palavras, uma fórmula Booleana é satisfatível caso exista, ao menos, uma atribuição de valores-verdade para suas variáveis que a torne verdadeira.

Uma instância do problema SAT consiste em uma fórmula Booleana na Forma Normal Conjuntiva (FNC). Diz-se que uma fórmula está na FNC, se a mesma é uma conjunção de cláusulas, e, cada cláusula é uma disjunção de literais. Como exemplo de fórmula Booleana na FNC tem-se:

$$(x_1 \vee x_2) \wedge (x_3 \vee x_4 \vee \neg x_1) \wedge (x_4 \vee \neg x_1 \vee \neg x_2), \quad (2.1)$$

onde  $x_1, x_2, x_3$  e  $x_4$  são as variáveis da fórmula, o símbolo  $\wedge$  representa a *conjunção* de cláusulas (3 cláusulas neste exemplo) e  $\vee$  simboliza a *disjunção* de literais (variável  $x_i$  ou sua negação  $\neg x_i$ ).

Em sua versão de otimização, o problema de *Maximização da Satisfatibilidade* (ou simplesmente Max-SAT) consiste em encontrar o número máximo de restrições (ou cláusulas) que podem ser satisfeitas ao mesmo tempo por alguma associação de valores-verdade para suas variáveis. Este é um problema NP-difícil [24]. DANTSIN *et al.* [25] demonstram que este problema pode ser aproximado em tempo polinomial, com uma razão de aproximação de  $7/8 + \varepsilon$  do valor ótimo, para algum  $\varepsilon > 0$ .

Mais especificamente, Max- $Ek$ -SAT é o problema de Maximização da Satisfatibilidade para instâncias com exatamente  $k$  literais por cláusula, onde os literais são variáveis ou a sua negação. Com base nos valores de  $k$ , existem algumas propriedades para este problema:

- (1) para  $k < 3$ , o problema pode ser resolvido em tempo polinomial;
- (2) para  $k = 3$ , o problema é NP-difícil;
- (3) para  $k > 3$ , pode-se reduzir o problema ao caso em que  $k = 3$ , em tempo polinomial.

Além disso, cláusulas com mais de 3 literais podem ser transformadas em cláusulas com no máximo 3 literais, através da adição de variáveis e cláusulas auxiliares.

Algo semelhante pode ser feito para transformar uma cláusula com menos de 3 literais em uma cláusula com exatamente 3 literais, bastando adicionar uma constante *falsa* para cada literal ausente na cláusula em questão.

Uma formulação do problema Max-SAT, com exatamente 3 literais em cada cláusula, trata-se, portanto, de uma simplificação do caso geral. A formulação para este caso simplificado será apresentada nas próximas subseções. Vale ressaltar que o método desenvolvido nesta tese – e apresentado no Capítulo 4 – não possui esta limitação, podendo ser aplicado a instâncias com cláusulas de qualquer tamanho.

### 2.1.1 Definição do Problema

Com o objetivo de formular com maior precisão este problema, será usado o conjunto  $X$  com  $n$  variáveis que, neste caso, recebem valores de um conjunto  $D = \{V, F\}$ , onde  $V$  e  $F$  representam, respectivamente, *verdadeiro* e *falso*. Sobre o conjunto de variáveis  $X$ , são definidas  $m$  cláusulas disjuntivas  $c_1, \dots, c_m$ . Para  $1 \leq j \leq m$ , a cláusula  $c_j$  tem a forma:

$$c_j = l_{j,1} \vee l_{j,2} \vee l_{j,3} \quad (2.2)$$

ou seja, ela é uma disjunção lógica de três literais  $l_{j,1}$ ,  $l_{j,2}$  e  $l_{j,3}$ . Cada um deles é uma variável  $x_i$  (para algum  $1 \leq i \leq n$ ), sua negação ( $\neg x_i$ ) ou uma constante de valor  $F$ . O objetivo é encontrar o número máximo de cláusulas da fórmula:

$$c_1 \wedge \dots \wedge c_m \quad (2.3)$$

que podem ser satisfeitas, simultaneamente, em uma associação de valores para as variáveis em  $X$ .

Seja um ponto  $s \in \{V, F\}^n$ , cujas coordenadas representam os valores-verdade associados as variáveis em  $X$ . Considere a função:

$$f(x_1, \dots, x_n) = f(s) = \sum_{j=1}^m g(c_j), \quad (2.4)$$

onde  $g(c_j) = 1$ , caso a cláusula seja satisfeita (*verdadeira*) sobre a atribuição  $s$  e,  $g(c_j) = 0$ , caso a cláusula não seja satisfeita (*falsa*) sobre a atribuição  $s$ . Nota-se

que  $u$  cláusulas podem ser simultaneamente satisfeitas se, e somente se, existir um ponto  $s \in \{V, F\}^n$ , para o qual,  $f(s) = u$ , para  $0 \leq u \leq m$ . Assim, objetiva-se encontrar um ponto dito  $s^*$ , no qual,  $f$  seja máximo. Além disso, ao verificar que o valor de  $f(s^*)$  no ponto máximo é igual a  $m$ , tem-se também uma resposta para o problema de decisão (SAT). Ou seja, se  $f(s^*) = m$ , então a fórmula é *satisfatível*; caso contrário, ela é *insatisfatível*.

### 2.1.2 Maximização por Análise Local

Com base na formulação apresentada na Subseção 2.1.1, pode-se verificar a influência do valor de uma variável específica  $x_i$  no comportamento da função  $f(s)$  (Equação 2.4), onde  $s$  é uma atribuição de valores-verdade para as variáveis em  $X$ . Desta maneira, obtem-se um dos possíveis cenários para uma cláusula  $c_k$ :

1.  $x_i$  e  $\neg x_i$  não estão presentes em  $c_k$ ;
2.  $x_i$  (ou  $\neg x_i$ ) está presente e não é o único literal  $V$ ;
3.  $x_i$  (ou  $\neg x_i$ ) está presente e é o único literal  $V$ ;
4.  $c_k$  não está satisfeita e  $x_i$  (ou  $\neg x_i$ ) é um de seus literais.

O valor atribuído para a variável  $x_i$  não influencia no resultado da função local  $g(c_k)$  do cenário (1).  $x_i$  está presente no cenário (2). Entretanto, seu valor não influencia (atualmente) no resultado da função local  $g(c_k)$ , uma vez que existe pelo menos uma variável  $x_j$  (para  $j \neq i$ ), cujo valor torna a cláusula verdadeira (ou satisfeita), e a modificação do valor de  $x_i$  não alterará o resultado de  $f$ . Os cenários (3) e (4) são os mais interessantes e indicam que o valor de  $x_i$  (ou  $\neg x_i$ ) tornou a cláusula verdadeira e falsa, respectivamente. A alteração no seu valor tornará a cláusula falsa (para (3)) e verdadeira (para (4)).

Sejam  $q_{i3}$  e  $q_{i4}$ , respectivamente, os números de cláusulas em que os cenários (3) e (4) ocorrem na fórmula para uma variável  $x_i$ , para algum  $1 \leq i \leq n$ . É possível estabelecer um procedimento de maximização local, de maneira que, ao alterar o valor da variável  $x_i$ , o valor de  $f$  seja maximizado. Ou seja, se  $q_{i4} > q_{i3}$ , para uma variável  $x_i$ , então o valor dessa variável deverá ser alterado.

Como exemplo, considere a fórmula  $c_1 \wedge \dots \wedge c_5$ , dada por:

$$(x_1 \vee x_2) \wedge (x_3 \vee x_4 \vee \neg x_1) \wedge (x_2 \vee \neg x_5) \wedge (x_4 \vee \neg x_1 \vee \neg x_2) \wedge (\neg x_3 \vee x_2 \vee x_5). \quad (2.5)$$

Supondo que o ponto inicial escolhido para as variáveis do exemplo acima seja  $s = (V, F, V, V, F)$ . Neste caso,  $f(s)$  será igual a 4 e apenas a cláusula  $c_5$  não será satisfeita por  $s$ . Supondo que a variável  $x_3$  esteja sendo analisada e, ao calcular  $q_{33}$  e  $q_{34}$  obtenha-se 0 e 1, respectivamente. A cláusula  $c_5$  indica a ocorrência do cenário (4) para a variável  $x_3$  (representada pelo literal  $\neg x_3$ ) e, ao alterar o valor de  $x_3$  para  $F$  tem-se  $f(s') = 5$ . Como  $s' = m$ ,  $f(s')$  é máximo e a fórmula  $c_1 \wedge \dots \wedge c_5$  é satisfatível.

### 2.1.3 Trabalhos Relacionados

Ao longo dos anos, muitos métodos exatos e heurísticos foram desenvolvidos para tratar do problema Max-SAT. Nesta subseção, serão revistos alguns trabalhos diretamente relacionados à pesquisa de aplicação de abordagens para resolver o PSA aplicados ao Max-SAT. Será relacionado, também, um conjunto de algoritmos utilizados nos experimentos computacionais dos Capítulos 3 e 4.

Nos experimentos computacionais realizados nesses capítulos foram utilizados dois *softwares* exatos para resolução do problema Max-SAT. Sua aplicação objetivou obter resultados ótimos para as instâncias utilizadas nos experimentos, bem como para estabelecer quais delas não poderiam ser resolvidas por métodos exatos, caracterizando-as como instâncias difíceis. Estes *softwares* foram escolhidos por terem se destacado nas competições sobre o problema Max-SAT.

O primeiro *software* exato, utilizado nos experimentos do Capítulo 3, é o *MaxSATz*, proposto por LI *et al.* [26]. O *MaxSATz* obteve ótimos resultados na segunda competição sobre algoritmos exatos para Max-SAT [27] e consiste de uma implementação do conjunto de regras de inferência específicas para o problema Max-SAT e da unidade de resolução adaptada ao problema, criados por LI *et al.* [26]. Segundo os autores, ao serem aplicadas eficientemente, transformam instâncias do problema Max-SAT em instâncias equivalentes (do mesmo problema) que são fáceis de resolver.

O outro *software* exato, utilizado nos experimentos do Capítulo 4, é o *MSUnCore*, proposto por MANQUINHO *et al.* [28], que obteve o melhor desempenho na terceira competição sobre algoritmos exatos para Max-SAT [21]. O *MSUnCore* aproveita as características presentes nos resolvidores SAT atuais, essencialmente na identificação de subfórmulas não satisfeitas. Além disso, o método proposto fornece percepção adicional na relação entre subfórmulas não satisfeitas e o problema Max-SAT.

Urge destacar que, em paralelo ao esforço despendido no desenvolvimento de métodos exatos mais eficientes para resolver as instâncias do problema Max-SAT, pesquisas também foram realizadas com o objetivo de criar métodos probabilísticos mais eficientes para este problema.

Desde a introdução do algoritmo de busca gulosa para SAT, denominado *GSAT*, por SELMAN *et al.* [29], muita pesquisa tem sido conduzida com este objetivo. A maior parte do progresso nessa área deu-se pelo uso de estratégias de ruído [30] e pelo desenvolvimento do mecanismo *WalkSAT* [31]. Em cada passo de sua busca, o algoritmo *WalkSAT* escolhe a cláusula não satisfeita corrente e, então, troca o valor da variável desta cláusula. Novos experimentos resultaram na introdução de mecanismos sofisticados para a escolha da variável para troca, incluindo os conhecidos algoritmos *Novelty* e *R-Novelty* [32]. Ideias resultantes de propriedades teóricas e da natureza destes algoritmos motivaram mais recentes estudos, ocasionando, por exemplo, o desenvolvimento da variante *AdaptiveNovelty+* [33].

Em paralelo ao desenvolvimento de versões mais refinadas das estratégias de busca iterativa probabilística, outros métodos de busca local probabilística também foram desenvolvidos, como por exemplo, os métodos *Saps* [34] e *Sapsnr* [35] (uma variação do primeiro). O método *Saps*, desenvolvido por HUNTER *et al.* [34], é baseado na idéia de modificar a função de escolha das cláusulas e variáveis candidatas, com o objetivo de prevenir que a busca seja conduzida para um máximo local de baixa qualidade ou para áreas do espaço de busca com soluções indesejadas. Esta estratégia é chamada de *busca local dinâmica (DLS)* e, geralmente, associam pesos às cláusulas de uma dada fórmula, que são modificados durante o processo de busca. Estes algoritmos tentam, então, minimizar o total de pesos, especialmente, o número de cláusulas não satisfeitas.

É possível notar que todos os métodos probabilísticos, referenciados até o momento, fazem uso da característica de maximização por análise local, presente no problema Max-SAT, para efetuar a busca por uma solução ótima. Desta forma, estes métodos serão novamente trabalhados no Capítulo 4, onde serão utilizados como rotinas de pós-processamento do método proposto no mesmo capítulo.

Como mencionado anteriormente, GOMES e SELMAN [13] propõem a execução paralela de um conjunto de algoritmos para resolver problemas difíceis, apresentando alguns resultados experimentais para problemas de *satisfação de restrições* e de *programação inteira mista*. Em seu trabalho, GOMES e SELMAN definem algumas condições que tornam a execução paralela (ou intercalada, em um único processador) consideravelmente mais vantajosa, computacionalmente, que o uso dos melhores métodos tradicionais. Por exemplo, ao ser executada apenas uma vez, a estratégia de *busca gulosa* (*BG*) é mais robusta do que uma estratégia de *busca em profundidade* (*BP*). Entretanto, ao executar vários processos em paralelo, o conjunto de execuções da estratégia de *BP* obtém melhor desempenho que o conjunto de execuções da estratégia *BG*. Segundo os autores, isto acontece pois a *BP* é considerada uma estratégia mais “arriscada”, uma vez que possui uma variância bem maior que a *BG* e tem uma chance pequena, mas não desprezível, de encontrar boas soluções em poucos passos.

MINTON [16] desenvolve um sistema de aprendizado indutivo que resolve instâncias do problema de satisfação de restrições (PSR) pela seleção de uma estratégia de busca, dentre as diversas disponíveis. A ideia geral desse sistema é a escolha de um conjunto de regras heurísticas básicas que podem ser combinadas em um modelo de busca heurística genérico. Essa escolha é feita por dois métodos diferentes: um analítico e um indutivo. O método analítico seleciona as regras mais promissoras, pela análise das restrições do problema (no formato do PSR); o indutivo aplica um processo de busca *hill-climbing* paralela, com cada regra, sobre um conjunto de instâncias de treinamento. As regras que obtiveram o melhor desempenho e, portanto, mais adequadas ao problema e ao conjunto de instâncias, são compiladas em um modelo de busca heurística. Segundo o autor, o processo descrito é demorado e passa a ser vantajoso quando a solução compilada será utilizada várias vezes sobre instâncias do problema. Desta maneira, o custo de compilação será diluído entre as

execuções do método compilado.

Muitos problemas do mundo real são melhor representados por meio de *restrições prioritárias* (devem ser satisfeitas) and *desejáveis* (podem, idealmente, ser satisfeitas). THORNTON *et al.* [36] descrevem um método heurístico de busca local para o problema Max-SAT com restrições deste tipo. Nele, é apresentado um método heurístico, de dois níveis, que alterna a sua estratégia de busca ao detectar que a solução corrente possui restrições prioritárias insatisfeitas. As restrições prioritárias são destacadas pelo uso de um peso (custo) sobre cada restrição prioritária, que é maior que a soma dos pesos de todas as restrições desejáveis. Com isso, o procedimento de busca dará preferência às soluções que satisfaçam todas as restrições prioritárias, em detrimento das demais restrições. Um método de *busca tabu*, executado em paralelo (por um número fixo de iterações), procura por subclasses de restrições prioritárias que estejam totalmente satisfeitas ou totalmente insatisfeitas. Ao encontrar um destes cenários, os pesos das restrições da subclasse são ajustados, proporcionando que as violações das restrições desejáveis sejam minimizadas. Estas duas heurísticas são, então, combinadas com um mecanismo dinâmico que altera a importância relativa das restrições prioritárias durante o procedimento de busca. Resultados experimentais apresentados indicam que esse método heurístico de dois níveis oferece melhoras significativas de desempenho sobre as abordagens padrões.

Alguns dos métodos, supramencionados, podem ser considerados *algoritmos híbridos*, porquanto incorporem mecanismos heurísticos diferentes na busca de uma solução para uma instância. Tais métodos optam por aplicar um determinado mecanismo heurístico, em detrimento de outros, toda vez que o cenário de busca é mais vantajoso para ele. Assim, da mesma forma que outros trabalhos referenciados na literatura como exemplos de soluções para o PSA (veja os trabalhos descritos na Seção 1.2), os trabalhos aqui citados, também podem ser considerados como soluções para tal problema.



## 2.2 Problema da Cobertura de Vértices Mínima (CVM)

Nesta seção será descrito o problema da *Cobertura de Vértices Mínima* (ou simplesmente CVM), como também a semelhança que este problema possui com o problema Max-SAT, a saber: assim como é possível manipular localmente o estados das variáveis de uma instância do problema Max-SAT com o objetivo de encontrar uma solução para o problema, também é possível manipular um vértice do grafo do problema CVM para conduzir a uma solução global.

BARBOSA e GAFNI [37] descrevem uma formulação do problema CVM, que será utilizada como base para a formulação apresentada nas próximas subseções.

### 2.2.1 Definição do Problema

Para um grafo não direcionado  $G = (V, E)$ , onde  $V$  e  $E$  representam, respectivamente, o conjunto de *vértices* e *arestas* do grafo  $G$ , o problema da Cobertura de Vértices Mínima (CVM) é um problema de otimização que consiste em encontrar o menor subconjunto  $S$  de seus vértices, tal que cada aresta tenha, ao menos, uma extremidade em algum vértice de  $S$ . Em outras palavras, para cada aresta  $(i, j) \in E$ , ou  $i$  ou  $j$  deve ser um vértice de  $S$ . Formalmente [37]:

**Definição 2.1** *Problema de Cobertura de Vértices Mínima.* Encontrar um subconjunto  $S \subseteq V$  com estas propriedades:

- (1) não existe aresta  $(i, j) \in E$  tal que  $i \notin S$  e  $j \notin S$ ; e
- (2) nenhum outro subconjunto de  $V$ , para o qual a propriedade (1) se mantém, tem cardinalidade menor que  $|S|$ .

Seja  $x_i$  a variável Booleana associada ao vértice  $i$  ( $x_i = 1$  para  $i \in S$ ;  $x_i = 0$  para  $i \notin S$ ). Resolver o problema CVM equivale a minimizar a função  $f$ :

$$f(x_1, \dots, x_n) = \sum_i^n x_i + \sum_{(i,j) \in E} \begin{cases} 2, & \text{se } x_i = x_j = 0; \\ 0, & \text{caso contrário.} \end{cases} \quad (2.6)$$

Seja  $W_i$  o conjunto de todos os vértices  $j \in V$ , tal que  $(i, j) \in E$ . Localmente, cada vértice  $i$  possui a função de avaliação  $f_{\text{Local}}$ :

$$f_{\text{Local}}(x_i, W_i) = x_i + \sum_{j \in W_i} \begin{cases} 2, & \text{se } x_i = x_j = 0; \\ 0, & \text{caso contrário.} \end{cases} \quad (2.7)$$

### 2.2.2 Minimização por Análise Local

Portanto, para minimizar  $f_{\text{Local}}$ , basta manipular  $x_i$  de acordo com a seguinte regra:

$$x_i = \begin{cases} 0, & \text{se } x_j = 1, \forall j \in W_i; \\ 1, & \text{caso contrário.} \end{cases} \quad (2.8)$$

É possível notar que, ao manipular o estado do vértice  $i$  para minimizar  $f_{\text{Local}}$ ,  $f$  também será minimizada e, além disso, enquanto houver algum vértice desrespeitando a regra, a solução para o problema não terá sido atingida.

Um procedimento simples para encontrar uma Cobertura de Vértices é visitar os vértices do grafo em sequência, alterando o seu estado de acordo com a Regra 2.8. Isto deve ser repetido até que nenhuma alteração possa ser realizada. No entanto, não há garantia de que a solução encontrada, apesar de *viável*, seja mínima. O termo solução viável para uma cobertura de vértices indica uma solução *minimal* para o grafo em questão, e o termo minimal aplicado a um conjunto de vértices  $S$ , significa que não existe qualquer subconjunto  $S' \subset S$  com a mesma propriedade, ou seja, que também é uma cobertura de vértices.

Na Seção 5.1, é apresentado um método capaz de gerar heurísticas que utilizam a minimização local de  $f_{\text{Local}}$  para encontrar uma *solução viável ótima* (mínima) para o problema CVM.

### 2.2.3 Trabalhos Relacionados

O *Second DIMACS Challenge* [22] apresentou uma série de heurísticas para problemas em grafos, tais como: *clique máxima*, *conjunto independente máximo* e *cobertura de vértices mínima*. Após a sua realização, foram desenvolvidos diversos trabalhos que utilizam tais heurísticas como um referencial de comparação. Em virtude disso e por serem problemas equivalentes, algumas delas serão descritas aqui e

utilizadas na avaliação dos resultados apresentados no Capítulo 5.

MANNINO e SASSANO [38] apresentam a heurística *UPPROJ*, que consiste numa aproximação do método exato *branch-and-bound* para o problema do *Conjunto Independente Máximo*<sup>3</sup>. Os limites superiores são obtidos através de um procedimento derivado da operação de projeção de arestas, que é uma especialização da projeção de clique de LOVÁSZ e PLUMER [39], quando restrita sobre arestas. Segundo os autores, essa heurística parece ser bem eficiente para grafos esparsos, que geralmente correspondem à instâncias difíceis do problema.

Outro método heurístico que utiliza *branch-and-bound* para o problema de conjunto independente máximo é o *Squeeze*, proposto por BOURJOLLY *et al.* [40], cujos limites inferiores são obtidos por meio de uma redução ao problema de minimizar uma função quadrática 0-1 geral.

Já BALAS *et al.* [41] descreve uma heurística do método exato *branch-and-cut*, chamada de *MIPO*, que inicia com uma formulação de programação inteira do problema da *Clique Máxima*<sup>4</sup> e continua a partir do relaxamento das condições iniciais, dada pela geração de novos planos de corte e pela sua incorporação na programação linear corrente.

A heurística *Cliqmerge*, proposta por BALAS e NIEHAUS [42], é um procedimento de busca por uma clique máxima em subgrafos induzidos. A idéia principal desse método é a aplicação de um algoritmo de união sobre todos os pares de cliques (originais ou criadas a partir dessa união), que recebe duas cliques como entrada e encontra uma clique máxima contida no subgrafo induzido pelos vértices dessas cliques iniciais.

A heurística *RB-clique* de GOLDBERG e RIVENBURGH [43] é um método que utiliza *backtracking* restrito para efetuar a busca por uma clique máxima em um grafo. As coordenadas de *backtracking* estão sujeitas às restrições definidas como parte da entrada do problema.

SORIANO [44] propõe uma família de três heurísticas, duas determinísticas e uma probabilística, que são variações do método de *busca tabu* para resolução do

---

<sup>3</sup>O problema do *Conjunto Independente Máximo* de um grafo consiste em encontrar um conjunto de vértices máximo (maior cardinalidade) desse grafo que não possuam arestas em comum.

<sup>4</sup>O problema da *Clique Máxima* de um grafo consiste em encontrar o maior subgrafo *totalmente conexo* do grafo. O termo *totalmente conexo* significa que todos os vértices estão conectados uns aos outros.

problema da *Clique Máxima*.

No caso de JAGOTA e SANCHIS [45], foi proposta uma família de métodos heurísticos baseados em *redes neurais de Hopfield* para encontrar a clique máxima. Os métodos incluem aproximações descendentes determinísticas (heurísticas *SD* e *GSD*) e aleatórias (heurística *SSD*). Alguns desses métodos podem incorporar um passo de aprendizado por reforço entre as inicializações (a heurística *SSD* com aprendizado por reforço recebe o nome de *SSDRL*).

Outro trabalho, utilizado como referencial de comparação para problemas em grafos, é a heurística *CBH*, que é um método determinístico para resolução do problema da clique máxima. O mecanismo determinístico, desenvolvido por GIBBONS *et al.* [46], evita a necessidade de calibração dos parâmetros, enquanto mantém a competitividade de um procedimento guloso de busca aleatória que requer tal calibração.

Em outra linha está o caso da heurística *Adaptive t-Annealing (ATA)*, proposta por GROSSMAN [47], uma estratégia que utiliza *redes neurais recorrentes* para encontrar cliques máximas. O ponto mais importante desse método de aproximação é um procedimento adaptativo para a determinação do *threshold* apropriado e do estado inicial da rede neural.

AGGARWAL *et al.* [48] também propõe um método chamado *OCH*. Trata-se de um *algoritmo genético* (AG) aplicado ao problema de conjunto independente máximo, onde cada indivíduo é representado por um conjunto de  $n$ -dígitos binários. Cada dígito indica a pertinência ou não do vértice ao conjunto independente. O mecanismo de *crossover*, chamado de *crossover* otimizado, gera dois novos indivíduos, onde um é descendente ótimo, baseado no mesmo procedimento de união utilizado no método *Cliqmerge*, e um outro descendente de maneira aleatória.

Outro método heurístico, que também é um algoritmo genético, é chamado de *WAO*, e foi proposto por BARBOSA e CAMPOS [23] para resolver o problema do conjunto independente máximo. Este método baseia-se na relação existente entre o número independente e o conjunto de orientações acíclicas do grafo. A parte evolucionária da formulação consiste em um algoritmo genético, onde cada orientação acíclica do grafo é um indivíduo da população, representando uma solução candidata ao problema de conjunto independente máximo.

KOTECHA e GAMPHAHA [49] descrevem um *algoritmo genético híbrido* para a resolução do problema da cobertura de vértices mínima. No trabalho mencionado, é demonstrado que heurísticas de busca local podem ser adicionadas ao AG, com o objetivo de conduzi-lo, mais rapidamente, para solução ótima. Para tanto, foi criada uma heurística chamada de *HVX*, que atua diretamente sobre o operador de recombinação genética (*crossover*). Assim, quando o operador de *crossover* é aplicado sobre dois indivíduos selecionados, são gerados dois novos indivíduos: o primeiro resulta da aplicação de um operador de *crossover* clássico [50], e o segundo é gerado pela heurística *HVX*. Para demonstração deste método, o autor apresenta alguns resultados experimentais comparando-o ao uso dos operadores tradicionais no AG. Alguns resultados experimentais são demonstrados, indicando que o uso do método gera melhores soluções quando comparado ao uso de operadores de *crossover* clássicos.

Igualmente, PULLAN e HOOS [4] introduzem um método de busca estocástico para o problema da clique máxima, chamado de *DLS-MC*. O *DLS-MC* alterna entre fases de expansão e de alteração da clique candidata. Na fase de expansão, um mecanismo de construção gulosa é aplicado, inicializando-se com uma clique trivial (apenas um vértice) e expandindo-a pela adição de vértices adjacentes à ela. Quando tal expansão não pode mais ser realizada, podem existir vértices que são conectados a todos os vértices da clique, com exceção de apenas um. Neste caso, é possível fazer uma troca entre os vértices (fase de alteração), obtendo uma nova clique. Segundo os autores, após poucas trocas, a estratégia de expansão da clique corrente pode ser novamente uma opção. O dinamismo presente no *DLS-MC* advém do ajuste dinâmico das penalidades utilizadas no mecanismo de seleção dos vértices candidatos a entrar na clique. A análise empírica do método *DLS-MC* foi efetuada através de experimentos computacionais aplicados à todas as instâncias do *Second DIMACS Challenge* [22], obtendo-se bons resultados.

CAI *et al.* [51] também descrevem um novo método de busca local para o problema CVM, chamado *EWLS*. Este método baseia-se na idéia de extensão e manipulação de uma cobertura parcial de vértices, até que se encontre uma cobertura de vértices mínima. Com este propósito, o *EWLS* gera, por meio de um procedimento guloso, uma cobertura parcial de vértices que seja um limite superior próximo ao ta-

manho da cobertura de vértices mínima conhecida. Após este procedimento guloso, o *EWLS* atribui peso às arestas do grafo e aplica um procedimento de busca local iterativo, efetuando troca entre dois vértices do grafo: um vértice que está na cobertura parcial e possui pouca representatividade sobre a cobertura, e outro vértice que não está na cobertura, mas que representa o maior ganho sobre as arestas descobertas. Quando um ótimo local é atingido, os pesos das arestas são modificados e uma troca aleatória entre dois vértices (um vértice que está na cobertura parcial e outro que não está) é realizada.

Como destacado na Subseção 2.1.3, alguns dos métodos, supramencionados, também podem ser considerados soluções para o PSA.

Outro ponto a destacar é o fato de alguns desses métodos fazerem uso de mecanismos probabilísticos e de computação evolucionária. Estes mecanismos são úteis para a resolução de problemas NP-difíceis, dada a complexidade de algumas instâncias. Entretanto, ao passo que costumam produzir melhores resultados, a presença destes mecanismos, em geral, inviabiliza a comprovação teórica da eficiência do método. Mas, ainda assim, a eficiência destes métodos pode ser avaliada pela realização de experimentos empíricos.

Por fim, alguns desses métodos possuem resultados experimentais sobre as instâncias do *Second DIMACS Challenge* [22] e serão devidamente comparados aos resultados obtidos pelo método proposto no Capítulo 5.

## Capítulo 3

# Seletor Exato vs. Aproximado para Max-E3-SAT

Como apresentado anteriormente, o problema de *Seleção de Algoritmo* [8] pode ser empregado na escolha do algoritmo mais eficiente (para um ou mais aspectos) para a resolução de uma dada instância de um problema, de maneira que tempo (no caso de método exato) e razão de aproximação (para o método heurístico) sejam viáveis. Além disso, o tempo gasto na aplicação do processo de seleção deve ser polinomial em relação ao tamanho da instância.

VASSILEVSKA *et al.* [1] propõe um modelo de solução para o problema de seleção de algoritmos aplicável sobre alguns problemas, chamado em seu trabalho de *algoritmo híbrido*. Esse modelo é composto por uma coleção de heurísticas e uma rotina de seleção de tempo polinomial. Esta rotina de seleção, chamada de *seletor*, efetua uma análise preliminar sobre uma instância para decidir qual heurística será executada. O trabalho demonstra que para alguns problemas NP-difíceis, tal como Max-E $k$ -Lin- $p$ , Caminho Mais Longo e *Minimum Bandwidth*, pode-se criar um mecanismo de seleção capaz de decidir entre resolver uma dada instância com exatidão, em tempo “sub-exponencial” ( $2^{o(n)}$ ), ou aproximá-la em tempo polinomial, com uma razão de aproximação superior à inaproximabilidade conhecida ou conjecturada do problema, assumindo que  $P \neq NP$ .

Desta forma, o seletor proposto por VASSILEVSKA *et al.* [1] deve proporcionar uma melhora no desempenho dos métodos utilizados para a resolução do problema. Se o algoritmo exato aplicado a qualquer instância do problema for exponencial,

quando aplicado após a escolha pela rotina de seleção, haverá garantia de que o tempo utilizado será estritamente inferior a exponencial. E neste caso, haverá garantia de tempo “sub-exponencial”. Caso contrário, se ao executar o algoritmo aproximado para qualquer instância tem-se um fator de aproximação  $\delta$ <sup>1</sup>, a aplicação deste método, somente após a escolha realizada pelo seletor, garantirá um fator de aproximação estritamente superior a  $\delta$ . Ou seja, haverá uma melhora na aproximação da solução encontrada sobre a solução ótima.

Este capítulo expõe o estudo do método de seleção de algoritmos criado em [1]. A Seção 3.1 apresentará a formulação do método para o problema Max-E3-SAT, além de realizar uma análise de sua complexidade, mostrando algumas de suas inconsistências.

Na Seção 3.2 serão relatados alguns resultados gerados por testes experimentais, em que será verificado que é possível estender o uso do algoritmo exato na melhora do desempenho prático do seletor.

### 3.1 Descrição da Abordagem

Com base na definição do problema Max- $k$ -SAT, apresentada na Seção 2.1, considere o caso particular no qual toda instância tem exatamente 3 literais em cada cláusula, sem possuir *sobreposição* das mesmas. Existe sobreposição de cláusulas quando duas delas possuem exatamente as mesmas variáveis.

Em [1], um algoritmo para esta subclasse de instâncias do problema Max-SAT é apresentado. Nele, uma instância é submetida a uma rotina de seleção que decidirá qual método de resolução será aplicado, com base nas características dessa instância. Todavia, a análise de complexidade do algoritmo não foi detalhada no texto; e, após várias tentativas de realizar esta prova, foi verificado que o algoritmo não estava escrito corretamente. Mesmo após a troca de correspondência com WILLIAMS (co-autor do trabalho), constatou-se que a proposta para este problema não estava definida corretamente em [1], e, mesmo após a correção, não possuía a complexidade definida no Teorema 3.2 do trabalho.

O objetivo desta seção será mostrar o algoritmo para este problema específico e

---

<sup>1</sup> $0 \leq \delta \leq 1$  para problemas de maximização. Neste caso, quanto mais próxima de 1 está a razão de aproximação, mais perto do valor ótimo está a solução encontrada.



realizar a análise correta de sua complexidade.

A idéia do algoritmo, para uma instância  $I$ , é a execução de uma rotina de seleção que efetuará uma busca inicial por um conjunto  $S$  de cláusulas (definido posteriormente na Subseção 3.1.1) em  $I$ , que sendo pequeno, garantirá que a execução de um método exato sobre a instância realizar-se-á em tempo sub-exponencial; caso contrário, a execução de um método aproximado terá uma garantia de aproximação estritamente maior que  $(7/8 + \varepsilon)$ , para algum  $\varepsilon > 0$  e pequeno.

Os mecanismos descritos nas próximas subseções estão representados, resumidamente, no Algoritmo 1, onde: os passos 2 ao 4 correspondem ao processo de escolha dos conjuntos e do método de busca, realizado pelo *seletor*, descrito na Subseção 3.1.1; os passos 5 ao 10, representam o método *exato*, descrito na Subseção 3.1.2 e; os passos 12 ao 21, correspondem aos passos executados pelo método *aproximado*, descrito na Subseção 3.1.3.

---

**Algoritmo 1** Seletor Exato/Aproximado para *Max-E3-SAT*

---

- 1: Dada uma instância  $I$ , com  $n$  variáveis,  $m$  cláusulas de tamanho 3 e  $\varepsilon > 0$ ;
  - 2: Escolha sucessivamente dois conjuntos de cláusulas:  $S_3, S_2$ ;
  - 3: Faça  $S = S_3 \cup S_2$  e  $S_d = \text{maior}\{S_3, S_2\}$ ;
  - 4: **se**  $|S| \leq \varepsilon n/3$  **então**
  - 5:     **para** toda combinação de valores para as variáveis em  $S$  **faça**
  - 6:         Atribua os valores-verdade para estas variáveis;
  - 7:     **para** toda variável  $x_i \notin S$  **faça**
  - 8:         Escolha uma atribuição para  $x_i$ , tal que o maior número possível de cláusulas em  $I - S$  seja satisfeito;
  - 9:     **fim para**
  - 10:    **fim para**
  - 11: **senão**
  - 12:     Seja  $c_j(x_1, x_2, x_3)$  uma cláusula com as variáveis  $x_1, x_2$  e  $x_3$ ;
  - 13:     **para** cada cláusula  $c_j(x_1, x_2, x_3) \in S_d$  **faça**
  - 14:         Sejam  $q_{j,\text{ímpar}}$  e  $q_{j,\text{par}}$ , respectivamente, o número de atribuições satisfatórias para  $x_1, x_2$  e  $x_3$  com paridade ímpar e par;
  - 15:         **se**  $q_{j,\text{ímpar}} > q_{j,\text{par}}$  **então**
  - 16:             Selecione valores para  $x_1, x_2$  e  $x_3$ , tal que  $\{x_1, x_2, x_3\}$  possua paridade ímpar;
  - 17:         **senão**
  - 18:             Selecione valores para  $x_1, x_2$  e  $x_3$ , tal que  $\{x_1, x_2, x_3\}$  possua paridade par;
  - 19:         **fim se**
  - 20:     **fim para**
  - 21:     Atribua valor para as demais variáveis em  $I$ , com probabilidade  $1/2$ ;
  - 22: **fim se**
-

### 3.1.1 Rotina de Seleção

A rotina de seleção, ou simplesmente *seletor*, é um procedimento capaz de gerar, em tempo polinomial, dois conjuntos de cláusulas:  $S_3$  e  $S_2$ , onde  $S = S_3 \cup S_2$ , tendo as seguintes propriedades:

(*Assertiva*) para toda cláusula  $c_j \in I - S$ , ao menos duas variáveis de  $c_j$  aparecem em  $S$ .

(*Disjuntiva*) para  $S_d = \text{maior}\{S_3, S_2\}$ , toda  $c_j \in S_d$  tem ao menos duas variáveis que não aparecem em qualquer  $c_k \in S_d$ , desde que  $c_k \neq c_j$ .

Estas propriedades serão utilizadas pelos métodos exato e aproximado, em suas respectivas análises, sendo necessário construir um procedimento que mantenha a sua validade. Para tanto, o seletor efetuará o seguinte procedimento, organizado em passos:

1. escolha de um *conjunto disjunto maximal* de cláusulas  $S_3$  em  $I$ . Um conjunto é dito *disjunto* quando não possui nenhuma variável repetida em suas cláusulas;
2. criação de um conjunto  $I'$  com todas as cláusulas de  $I$ , retirando-se as variáveis que estão em  $S_3 \cap I$ ;
3. escolha de um novo *conjunto disjunto maximal* de cláusulas  $S_2'$  em  $I'$  sobre as cláusulas com duas variáveis.  $S_2$  será formado pelas cláusulas originais de  $I$  correspondentes às cláusulas de  $S_2'$ .

É possível notar que as propriedades *assertiva* e *disjuntiva* são mantidas para os conjuntos gerados neste procedimento, bem como este procedimento é polinomial sobre o número de cláusulas da instância.

Após a geração dos conjuntos, o seletor decide qual método será usado para a resolução da instância. Assim, se o conjunto  $S$  for pequeno<sup>2</sup>, o seletor escolherá o método exato. Caso contrário, escolherá o método aproximado. Vale ressaltar que a verificação do tamanho do conjunto é feita em tempo constante pelo seletor.

---

<sup>2</sup>O significado do termo *pequeno* será descrito no decorrer da Subseção 3.1.2.

### 3.1.2 Método Exato

O método exato, definido nos passos 5 a 10 do Algoritmo 1, efetua todas as possíveis atribuições para variáveis que aparecem em  $S$ . Pela propriedade *assertiva*, o conjunto de cláusulas em  $I - S$  é uma instância do problema MAX-1-SAT, podendo ser resolvida, exatamente, em tempo linear. Esta resolução é feita atribuindo-se para cada variável o valor-verdade que satisfaz o maior número de cláusulas em  $I - S$ .

O próximo passo é provar que  $S$  é pequeno para que a resolução da instância seja feita em tempo  $O(2^{\varepsilon n})$ , para algum  $\varepsilon > 0$  e pequeno, comprovando, desta maneira, a viabilidade do uso desse método.

Considere o seguinte raciocínio: para uma cláusula na Forma Normal Conjuntiva ser satisfeita, ela precisa ter ao menos um literal com atribuição do valor-verdade *verdadeiro* ( $V$ ). Portanto, para  $k = 3$  e o valor *falso* representado por  $F$ , existem sete atribuições que tornam uma cláusula satisfeita  $\{(FFV), (FVF), (FVV), (VFF), (VFV), (VVF), (VVV)\}$ , e um que não satisfaz a cláusula  $\{(FFF)\}$ . Então, para um conjunto  $S$  de cláusulas, verifica-se até  $8^{|S|}$  atribuições possíveis.

Assim, o procedimento de efetuar todas as atribuições possíveis para as variáveis do conjunto  $S$  (passos 5 e 6 do Algoritmo 1), somado às atribuições de valores para as variáveis que estão em  $I - S$  (passos 7 e 8 do Algoritmo 1), é realizado em tempo  $O(8^{|S|}) = O(2^{3|S|})$ . Portanto, para haver garantia de tempo “sub-exponencial”  $O(2^{\varepsilon n})$  o seletor deverá escolher o algoritmo exato somente quando encontrar um conjunto  $S$ , tal que  $|S| \leq \varepsilon n/3$ .

Logo, é possível concluir que o algoritmo exato, definido nos passos 5 a 10 do Algoritmo 1, retorna uma solução ótima para  $I$  em tempo  $O(2^{\varepsilon n})$ , quando aplicado por escolha do seletor.

### 3.1.3 Método Aproximado

Para a análise deste método, é necessário demonstrar que ele pode, simultaneamente, satisfazer todas as cláusulas de  $S_d$ , tão bem quanto  $7/8$  do número ótimo de cláusulas satisfatíveis no resto de  $I$ , desde que ele rode somente na hipótese de  $|S_d| > \varepsilon n/6$ . Como comprovar-se-á ao final desta análise, será produzido um incremento não-trivial na aproximação, se comparado ao demonstrado em [25].

Primeiramente, será feita a análise do que acontece quando uma atribuição alea-

tória é feita para as variáveis em  $I$ . Em sequência, verificar-se-á qual o valor esperado para uma atribuição aleatória de valores para variáveis em  $I$ . Para isto, considere  $c_j$  uma cláusula qualquer em  $I$  com as seguintes probabilidades:

$$\begin{cases} \text{probabilidade de } c_j \text{ não ser satisfeita} & \Rightarrow \frac{1}{2^k} \\ \text{probabilidade de } c_j \text{ ser satisfeita} & \Rightarrow 1 - \frac{1}{2^k} \end{cases} \quad (3.1)$$

onde  $k$  é o número de variáveis em  $c_j$ .

Considere também,  $E(I)$  o número esperado de cláusulas satisfeitas, tal que:

$$E(I) = \sum_{j=1}^m E(c_j) \quad (3.2)$$

onde  $m$  é o número de cláusulas em  $I$ , e  $E(c_j)$  a expectativa de satisfazer a cláusula  $c_j$ .

Substituindo 3.1 em 3.2, tem-se:

$$E(I) = \sum_{j=1}^m \left(1 - \frac{1}{2^{k_j}}\right) \quad (3.3)$$

Sabe-se que para uma instância do problema Max-E3-SAT, o número de variáveis em cada cláusula é sempre 3. Então, seja  $m^*$  o número máximo de cláusulas satisfeitas (solução ótima) para a instância  $I$ , tem-se que o valor esperado para uma atribuição aleatória para as variáveis de  $I$  será:

$$E(I) = \frac{7}{8}m \geq \frac{7}{8}m^* \quad (3.4)$$

Outra informação importante advém do tipo de atribuição realizada pelo algoritmo. Sabe-se que apenas uma atribuição de valores-verdade para uma cláusula de três literais na FNC poderá torná-la insatisfável. Assim, é possível atribuir valores-verdade para as variáveis das cláusula em  $S_d$ , de maneira que estas cláusulas sejam sempre satisfeitas.

É possível notar que os passos 12 ao 20 do Algoritmo 1 descrevem um procedimento que sempre satisfaz todas as cláusulas em  $S_d$ , pois, para cada cláusula  $c_j(x_1, x_2, x_3) \in S_d$ , os valores-verdade das variáveis  $x_1$ ,  $x_2$  e  $x_3$  são escolhidos com base na *paridade* que representa o maior número de atribuições sa-

tisfatíveis para essa cláusula. O termo *paridade*, usado neste contexto, indica se o número de variáveis, da cláusula, com valor-verdade  $V$  é par ou ímpar. Ou seja, para quaisquer 3 variáveis em uma cláusula, o conjunto de atribuições com paridade par é  $\{(FFF), (FVV), (VFV), (VVF)\}$  e com paridade ímpar é  $\{(FFV), (FVF), (VFF), (VVV)\}$ . Para exemplificar a aplicação deste procedimento, considere a cláusula  $c_j = (x_1 \vee x_2 \vee \neg x_3)$ . A única atribuição de valores-verdade para as variáveis  $x_1, x_2$  e  $x_3$  que torna esta cláusula insatisfeita é  $(FFV)$ . Como essa atribuição possui paridade ímpar,  $q_{j,\text{ímpar}}$  será menor que  $q_{j,\text{par}}$ , e, portanto, o passo 18 do algoritmo deverá ser executado, satisfazendo  $c_j$ .

Outra consequência da aplicação do procedimento, supramencionado, é que qualquer atribuição de valores para as variáveis de uma cláusula  $c_j(x_1, x_2, x_3) \in S_d$  terá distribuição de probabilidade dois a dois independente sobre  $\{x_1, x_2, x_3\}$ . Por conseguinte, e pela propriedade *disjuntiva* do conjunto  $S_d$ , tem-se uma probabilidade de  $7/8$  de satisfazer uma cláusula do conjunto  $I - S_d$ . Portanto, a atribuição de valores para as variáveis em  $S_d$  não altera a distribuição de probabilidade das cláusulas que não estão em  $S_d$ .

Assim, para  $m = |I|$  e  $m_d = |S_d|$ , tem-se:

$$\begin{aligned}
 E(I) &= m_d + \frac{7}{8}(m - m_d) \\
 &= \frac{m_d}{m} + \frac{7}{8}\left(\frac{m}{m} - \frac{m_d}{m}\right) \\
 &= \frac{m_d}{m} + \frac{7}{8} - \frac{7m_d}{8m} \\
 &= \frac{7}{8} + \frac{m_d}{8m}
 \end{aligned} \tag{3.5}$$

Como  $S_d$  é o maior dos conjuntos criados pelo procedimento de seleção, tem-se que  $m_d \geq |S|/2 \geq \varepsilon n/6$ . Substituindo em 3.5, tem-se:

$$E(I) \geq \frac{7}{8} + \frac{\varepsilon n}{48m} \tag{3.6}$$

Portanto, conclui-se que o algoritmo aproximado retorna, em tempo polinomial, uma solução  $\left(\frac{7}{8} + \frac{\varepsilon n}{48m}\right)$ -aproximada para o problema Max-E3-SAT.

## 3.2 Experimentos Computacionais

A comprovação teórica das propriedades deste método foi devidamente abordada e confirmada na Seção 3.1. Com o objetivo de comprovar na prática suas propriedades, vários experimentos foram realizados e seus resultados, descritos aqui.

### 3.2.1 Parâmetros e Dados

Inúmeros algoritmos exatos para Max-SAT foram desenvolvidos ao longo dos últimos anos. Em 2007 foi realizada a segunda avaliação de desempenho dos algoritmos exatos para Max-SAT [27], como um evento integrante da conhecida conferência sobre o problema de satisfatibilidade [52]. Dentre os vários algoritmos submetidos para avaliação nesta conferência, o algoritmo proposto por LI *et al.* [26], chamado de *MaxSATz*, obteve ótimos resultados.

Uma versão deste algoritmo foi disponibilizada pelo seu criador (em sua página pessoal) e utilizada para realização de testes preliminares, com o objetivo de definir o conjunto de parâmetros (número de variáveis e de cláusulas) para geração aleatória do conjunto de instâncias.

Com base nos resultados preliminares, obtidos pelo *MaxSATz*, um novo conjunto de instâncias (no formato padrão DIMACS [53]) com cláusulas na Forma Normal Conjuntiva (FNC) foi gerado aleatoriamente, utilizando os seguintes parâmetros:

- *Número de variáveis ( $n$ ):* foram escolhidos três valores 50, 100 e 150;
- *Número de cláusulas ( $m$ ):* a escolha do número de cláusulas foi feita baseada no número de variáveis do problema. Ou seja, para  $n = 50$ , o valor de  $m$  varia de 5 a 600; para  $n = 100$ ,  $m$  varia de 10 a 800; e, para  $n = 150$  o valor de  $m$  varia de 15 a 900;
- *Número de literais por cláusulas ( $k$ ):* todas as cláusulas foram geradas com exatamente três literais;
- *Sobreposição:* as instâncias geradas não possuem sobreposição de cláusulas.

Para cada combinação de valores de  $n$  e  $m$ , foram geradas 500 instâncias diferentes, as quais foram utilizadas na avaliação de desempenho do método descrito neste capítulo.

Nos experimentos computacionais, descritos a seguir, além dos parâmetros, acima listados, o valor da variável  $\varepsilon$  presente na Equação 3.6 também será manipulado, recebendo valores no intervalo  $(0.0, 1.0]$ .

### 3.2.2 Ambiente

Os algoritmos deste trabalho foram desenvolvidos na linguagem C, padrão *ANSI*, e os testes realizados em três configurações diferentes de máquinas:

- *1ª configuração*: cluster do Laboratório de Inteligência Artificial do PESC-COPPE/UFRJ, formado por 8 máquinas iguais, com processadores AMD Athlon XP 2400+, 2.0GHz de frequência, 1 GByte de memória, disco rígido de 40GBytes e uma rede Gigabit Ethernet. O sistema operacional é o *Scientific Linux* 4.6 com *kernel* 2.6.9-67 e compilador *gcc* de versão 3.2;
- *2ª configuração*: cluster do Laboratório de Inteligência Artificial do PESC-COPPE/UFRJ, formado por 8 máquinas iguais, com processadores Pentium 4 HT, 3.2GHz de frequência, 1 GByte de memória, disco rígido de 80GBytes e uma rede Gigabit Ethernet. O sistema operacional é o *Scientific Linux* 4.6 com *kernel* 2.6.9-67 e compilador *gcc* de versão 3.2;
- *3ª configuração*: uma máquina com processador Pentium Core Duo, 2.0GHz de frequência, 1 GByte de memória e disco rígido de 80GBytes. O sistema operacional é o *Ubuntu* 7.04 e compilador *gcc* de versão 4.0.

### 3.2.3 Resultados Experimentais

Primeiramente, os resultados atingidos pela aplicação do seletor no conjunto de instâncias deverão ser comparados aos valores esperados. Assim, a Figura 3.1 apresenta gráficos de desempenho do Algoritmo 1 para diversos intervalos de  $\varepsilon$  e variações de  $n$ . Vale ressaltar que, para cada valor de  $\varepsilon$  e  $n$ , o valor de  $m$  também foi manipulado e o Algoritmo 1 foi aplicado sobre 500 instâncias diferentes para cada combinação de valores de  $\varepsilon$ ,  $n$  e  $m$ . Nestes gráficos são apresentadas três curvas: a primeira delas representa o valor esperado para o modelo (Equação 3.6); a segunda informação representa o valor médio retornado pelo Algoritmo 1 em sua versão

aleatória; e a terceira, o resultado médio retornado por uma versão *desaleatorizada* deste algoritmo.

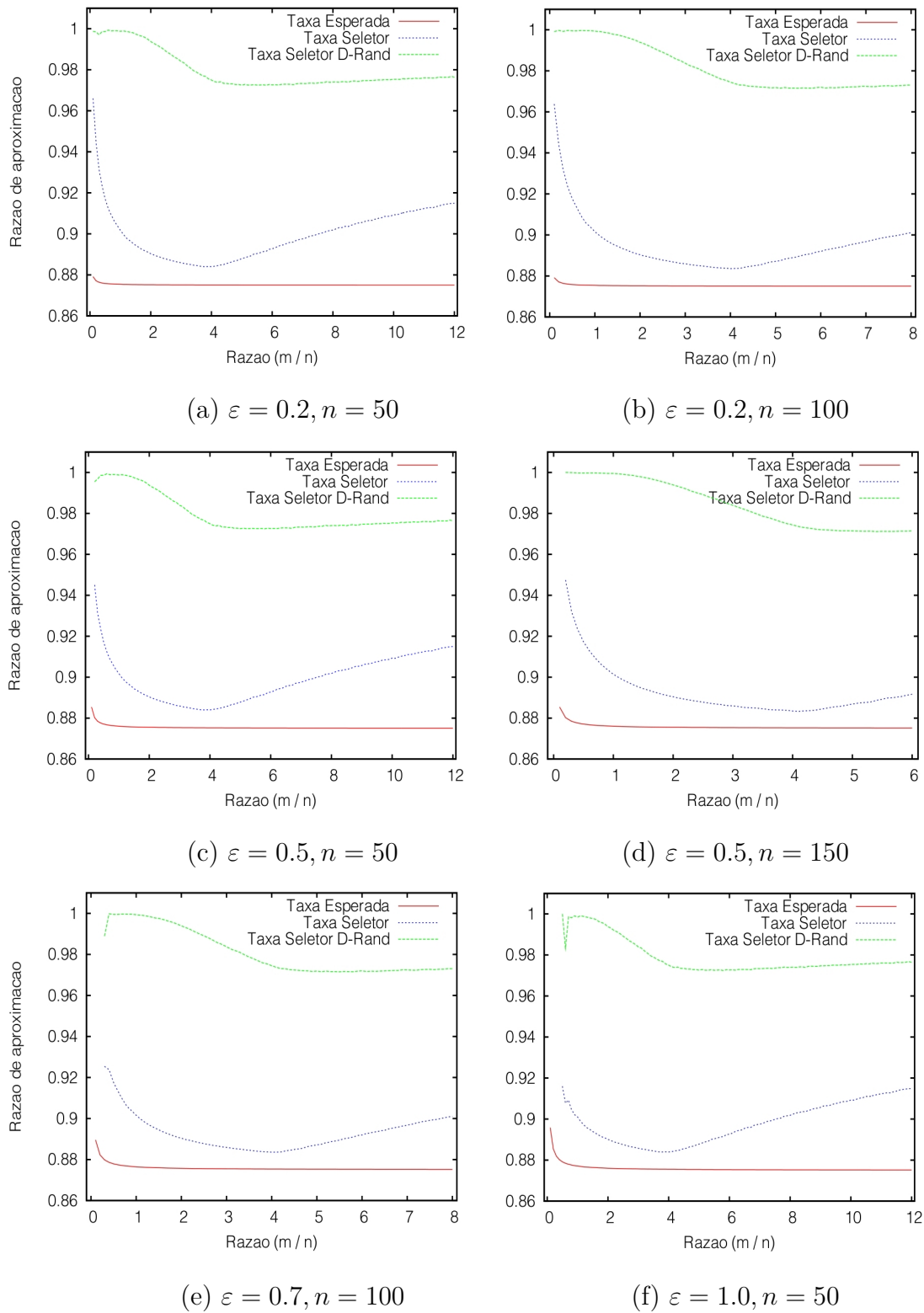


Figura 3.1: Taxa de aproximação obtida nos experimentos computacionais com o seletor Exato/Aproximado, para valores de  $\varepsilon$  e  $n$ .



A versão *desaleatorizada* do Algoritmo 1, mais precisamente da parte aproximada do algoritmo, é realizada em dois momentos. Primeiramente, são selecionadas as cláusulas no conjunto  $S_d$  (definido na Subseção 3.1.1). Para estas, faz-se a atribuição de valores-verdade, para as suas variáveis, que maximiza a probabilidade de satisfazer o maior número de cláusulas em  $I - S_d$ , levando em consideração apenas as atribuições da *paridade* que representa o maior número de atribuições satisfáveis para essa cláusula. Em sequência, para cada variável que não aparece em  $S_d$ , faz-se a atribuição de valor-verdade que maximiza a probabilidade de satisfazer o maior número de cláusulas em  $I - S_d$ .

Como é possível observar, os resultados ocorreram dentro das expectativas para o modelo, e pouca diferença se observa nos diferentes valores de  $\varepsilon$  e de  $n$ .

Outra informação passível de identificação é qual a *transição de fase* do problema Max-E3-SAT, quando aplicado ao conjunto de instâncias testadas. Um exemplo do estudo da transição de fase para problemas de satisfatibilidade é apresentado no trabalho de MONASSON *et al.* [54], onde, através de testes empíricos, observa-se que a transição de fase para problemas 3-SAT está no intervalo da fração de  $m/n \in [3.5, 5.0]$ . Em outro trabalho, SHEN e ZHAG [55] expõem um estudo empírico para Max-2-SAT.

As Figuras 3.2 e 3.3 mostram, respectivamente, a análise de satisfatibilidade e tempo (sobre alguns valores de  $n$ ), para o conjunto de instâncias testadas neste trabalho (500 para cada intervalo de razão  $m/n$ ).

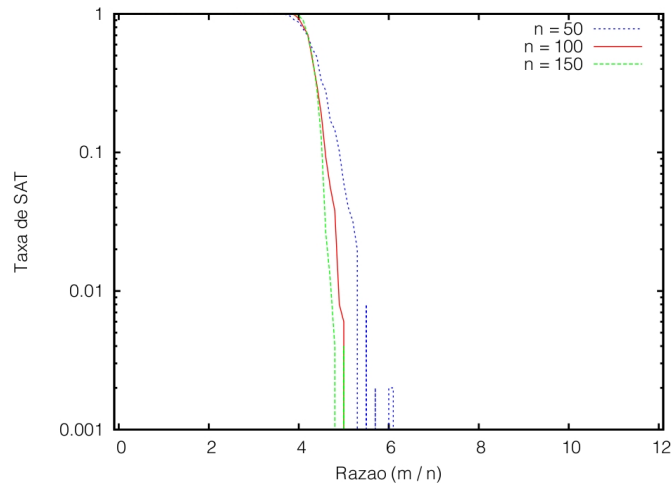


Figura 3.2: Taxa de satisfatibilidade do conjunto de instâncias para Max-E3-SAT.

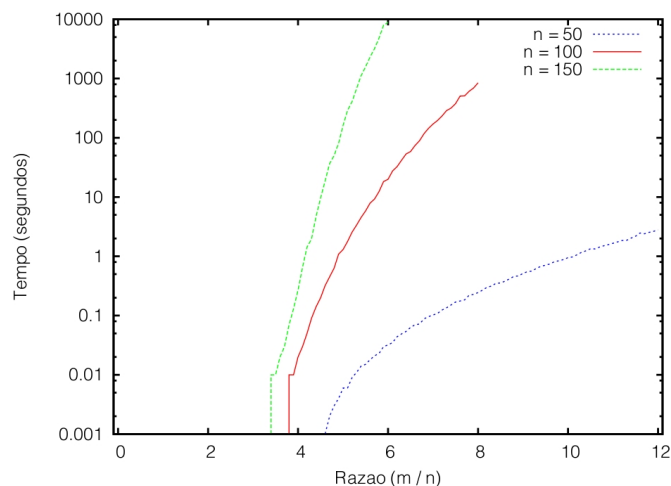


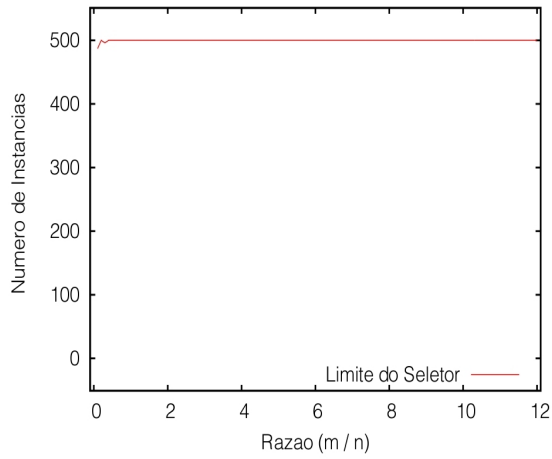
Figura 3.3: Tempo de execução do algoritmo exato para Max-E3-SAT.

A primeira informação, apresentada na Figura 3.2, demonstra que o conjunto de instâncias criado representa adequadamente o padrão apresentado nos dois estudos citados. Complementarmente, a informação presente no gráfico da Figura 3.3 indica o momento (transição de fase) em que as instâncias tornam-se mais difíceis (tempo necessário para encontrar a solução exata).

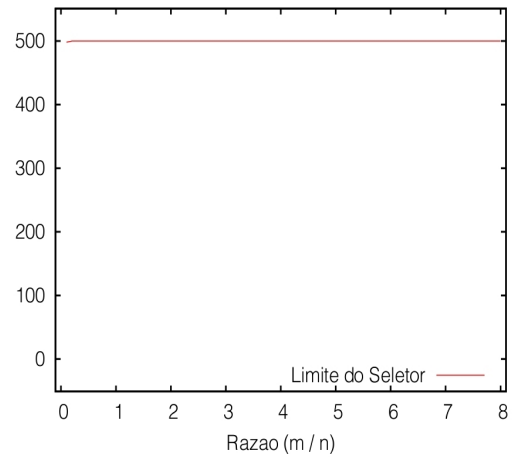
Como um método exato é aplicado no Algoritmo 1, sua execução é desejável a toda instância que está antes da transição de fase. Do contrário, soluções *fáceis* de resolver serão aproximadas, ao invés de serem resolvidas de maneira exata.

A Figura 3.4 mostra o limite das decisões tomadas pelo seletor para diversos valores de  $\varepsilon$ , novamente para  $n = 50$ ,  $n = 100$  e  $n = 150$ . Cada curva representa o número de instâncias em que o seletor decidiu efetuar o cálculo aproximado. Nota-se que ele abandona o método exato, muito antes de entrar na transição de fase. Ou seja, mesmo antes de encontrar uma instância difícil, o seletor abandona o uso do algoritmo exato e escolhe o algoritmo aproximado na totalidade dos casos, deixando de resolver várias instâncias fáceis.

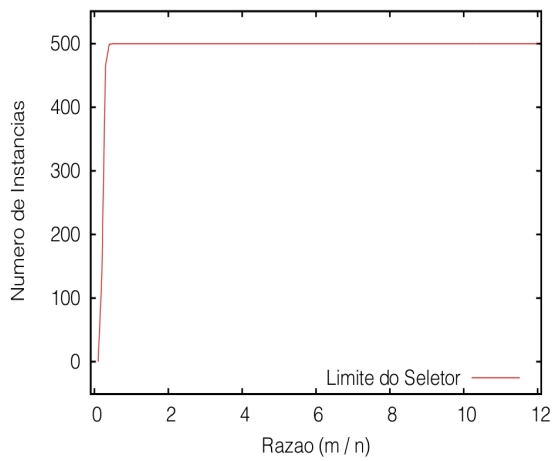
Levando em consideração estes resultados apontados, é possível identificar que o método, na prática, possui um *processo de seleção prematuro*. Ou seja, mesmo aqueles casos (localizados antes da região de transição) em que o *seletor* optou por resolver a instância pelo método aproximado, a solução exata (executada em paralelo pelo algoritmo *MaxSATz*) não apresentou alto custo de execução, e, por conseguinte, poderia ter sido escolhida.



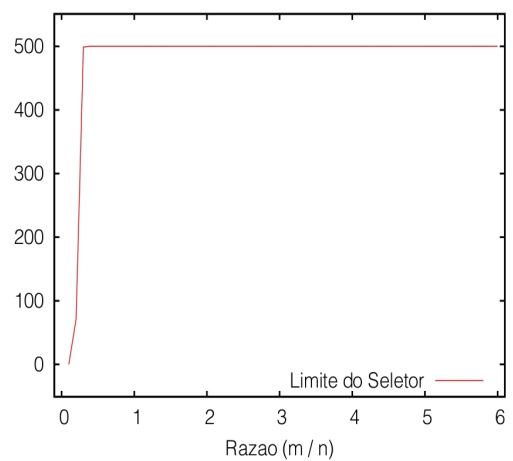
(a)  $\varepsilon = 0.2, n = 50$



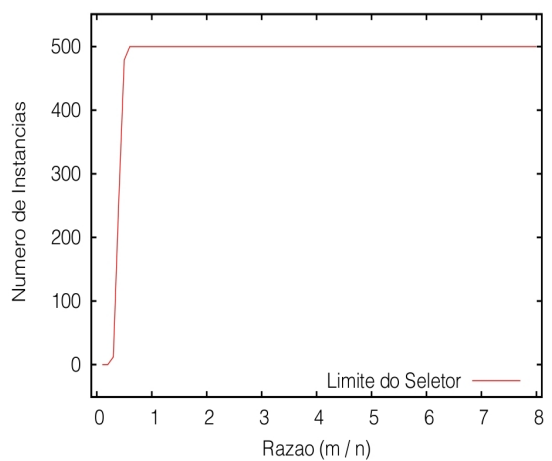
(b)  $\varepsilon = 0.2, n = 100$



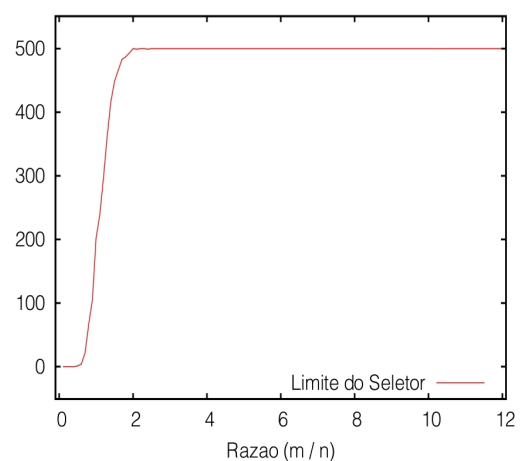
(c)  $\varepsilon = 0.5, n = 50$



(d)  $\varepsilon = 0.5, n = 150$



(e)  $\varepsilon = 0.7, n = 100$



(f)  $\varepsilon = 1.0, n = 50$

Figura 3.4: Região de decisão do *seletor* entre os métodos exato e aproximado. Cada curva representa o número de instâncias em que o seletor decidiu efetuar o cálculo aproximado.

Este fato não retira a importância dos resultados teóricos alcançados pelo método descrito em [1]. Inclusive, é possível efetuar pequenas alterações no método, capaz de melhorar o seu desempenho experimental, mas sem comprometer suas propriedades teóricas. Com este objetivo, uma nova formulação do mecanismo de seleção será descrita a seguir.

### 3.2.4 Nova Formulação

A modificação proposta no seletor é inspirada na análise de FINK [14], que descreve uma técnica para seleção de heurísticas, baseada na análise estatística do histórico do seu desempenho. Portanto, é possível incluir no mecanismo seletor (do Algoritmo 1) a análise dos parâmetros da instância com relação à região de transição do problema. Ressalta-se que esta análise deve ocorrer no momento em que o seletor fizer a escolha pelo método aproximado. Caso seja observado que a instância não se encontra na região de transição, a execução pelo método exato será realizada.

Outra característica crucial, demonstrada por FINK [14], para resolver eficientemente uma instância, é a determinação de um limite apropriado para abandonar uma escolha mal feita. Assim, é possível introduzir um mecanismo ao novo método para determinar este limite, caso o método exato seja selecionado. Para tanto, ao atingir um tempo-limite pré-estabelecido, se o algoritmo exato não tiver encontrado a solução ótima, o seletor deverá cancelar sua execução e efetuar o cálculo da solução pelo método aproximado. O Algoritmo 2 descreve um esquema resumido da nova formulação. Desta forma, caso o método exato retorne uma resposta em tempo inferior ao disposto a ser gasto com ele, haverá a garantia de que a solução encontrada seja ótima. Caso contrário, o seletor retornará uma solução  $\left(\frac{7}{8} + \frac{\varepsilon n}{48m}\right)$ -aproximada.

Através da realização de novos experimentos computacionais, constatou-se que, ao definir o tempo limite de execução do método exato em 1 minuto, todas as instâncias que apresentam a relação  $m/n < 3.5$  serão resolvidas pelo método exato e, após este valor, todas as instâncias serão resolvidas pelo método aproximado. Estes resultados indicam que o processo de seleção continua sendo prematuro. Para corrigir este problema, o mecanismo de verificação da região de transição foi retirado, gerando o Algoritmo 3.

---

**Algoritmo 2** Seletor Exato/Aproximado para *Max-E3-SAT* Modificado (versão 1)

---

```
1: Dada uma instância  $I$ , com  $n$  variáveis,  $m$  cláusulas de tamanho 3 e  $\varepsilon > 0$ .
2: Escolha sucessivamente dois conjuntos de cláusulas:  $S_3, S_2$ ;
3: Faça  $S = S_3 \cup S_2$  e  $S_d = \text{maior}\{S_3, S_2\}$ ;
4: se  $|S| \leq \varepsilon n/3$  então
5:   Executa o método exato
6: senão
7:   se  $m/n < 3.5$  então
8:     enquanto tempo limite não for atingido e a solução não for encontrada
9:       faça
10:        Executa o método exato
11:      fim enquanto
12:    se solução não foi encontrada então
13:      Executa o método aproximado
14:    fim se
15:  fim se
```

---

O Algoritmo 3 descreve um esquema alternativo, também resumido, da nova formulação. Nele, caso o método exato retorne uma resposta em tempo inferior ao disposto a ser gasto com ele, haverá a garantia de que a solução encontrada seja ótima. Caso contrário, o seletor retornará uma solução  $(\frac{7}{8} + \frac{\varepsilon n}{48m})$ -aproximada.

---

**Algoritmo 3** Seletor Exato/Aproximado para *Max-E3-SAT* Modificado (versão 2)

---

```
1: Dada uma instância  $I$ , com  $n$  variáveis,  $m$  cláusulas de tamanho 3 e  $\varepsilon > 0$ .
2: Escolha sucessivamente dois conjuntos de cláusulas:  $S_3, S_2$ ;
3: Faça  $S = S_3 \cup S_2$  e  $S_d = \text{maior}\{S_3, S_2\}$ ;
4: se  $|S| \leq \varepsilon n/3$  então
5:   Executa o método exato
6: senão
7:   enquanto tempo limite não for atingido e a solução não for encontrada
8:     faça
9:      Executa o método exato
10:    fim enquanto
11:   se solução não foi encontrada então
12:     Executa o método aproximado
13:   fim se
14: fim se
```

---

A Figura 3.5 demonstra o novo limite das decisões tomadas pelo seletor modificado (versão 2), respectivamente, para  $n = 100$  e  $n = 150$ . Cada curva contínua representa o número de instâncias em que o seletor original decidiu efetuar o cálculo aproximado. As demais curvas representam o número de instâncias em que o novo seletor decidiu efetuar o cálculo aproximado. Nota-se que, antes da região de

transição, apresentada na Figura 3.3, o algoritmo exato é escolhido com sucesso em 100% das instâncias e, mesmo nos outros casos, o algoritmo exato é executado com sucesso para uma porção considerável de instâncias. Os gráficos para  $n = 50$  foram omitidos, pois em 100% das instâncias, independente do limite de tempo e da razão  $m/n$ , o algoritmo exato foi aplicado com sucesso.

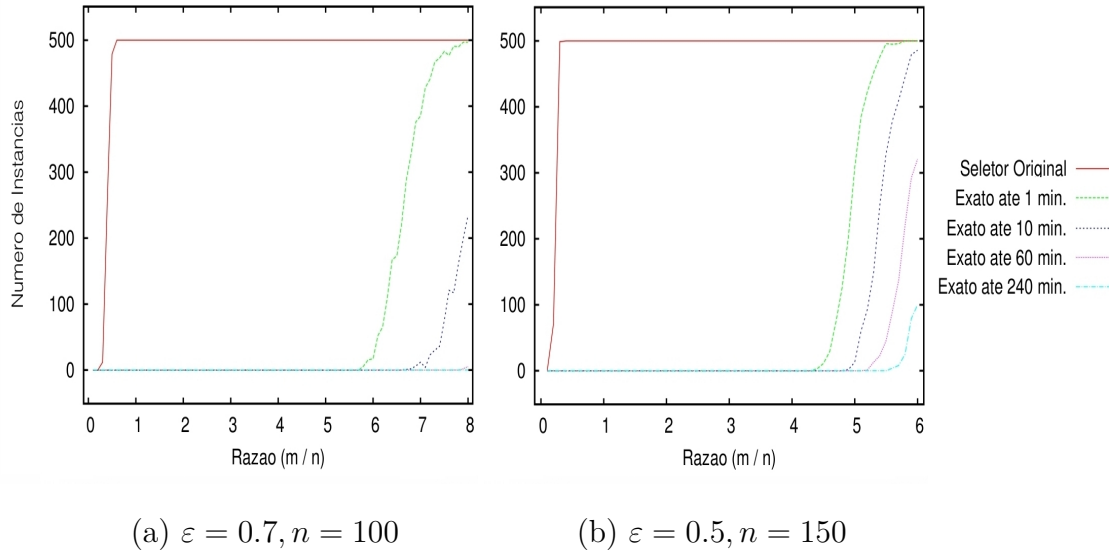


Figura 3.5: Região de decisão do *novo seletor* entre os métodos exato e aproximado.

### 3.3 Considerações Finais

Demonstrou-se, neste capítulo, a formulação e descrição detalhada do método teórico, criado por VASSILEVSKA *et al.* [1], para seleção automática de heurísticas para instâncias do problema Max-E3-SAT. Através da análise de sua complexidade, detectou-se algumas inconsistências no Teorema 3.2 do trabalho original, as quais foram devidamente corrigidas e apresentadas na Seção 3.1.

Após a análise teórica do método, foram realizados testes experimentais com o objetivo de comprovar, na prática, os resultados teóricos esperados. Constatou-se, entretanto, que o método apresenta resultados práticos insatisfatórios, pois abandonam instâncias fáceis de serem solucionadas pelo algoritmo exato, optando por resolvê-las de maneira aproximada.

Ao discorrer sobre o modelo proposto por VASSILEVSKA *et al.* [1] como uma solução para o PSA, verifica-se que a análise inicial, feita pelo *seletor*, identifica

características presentes na instância, que o ajudam a decidir entre a execução de um algoritmo exato ou o cálculo de uma solução aproximada. Com relação aos critérios de avaliação do seu desempenho, verifica-se que ele possui pontos positivos, como a comprovação teórica das propriedades do método, a redução no tempo de execução do método exato, a melhora na complexidade da busca por uma solução exata e a melhora na taxa de aproximação do método aproximado. O principal ponto negativo do método é a sua ineficiência prática.

Em vista dessa análise e dos resultados experimentais, foram propostas melhorias práticas no Algoritmo 1. É imprescindível ressaltar que estas melhorias não comprometem os resultados teóricos alcançados.

Em face do resultado dos testes apresentado neste capítulo, optou-se pelo desenvolvimento de novos métodos práticos para resolução desse e de outros problemas NP-Difíceis, propostos nos próximos capítulos.

# Capítulo 4

## Seleção por Algoritmo Genético para Max-SAT (*SAHe-SAT*)

Este capítulo propõe um novo método para resolução de instâncias do problema Max-SAT, utilizando a característica de manipulação local das variáveis na busca por uma solução ótima. Este novo método tenta maximizar os resultados alcançados por outras heurísticas, aqui utilizadas como rotinas de pós-processamento. Tal método denomina-se *seleção automática de heurísticas para o problema Max-SAT*, ou, simplesmente *SAHe-SAT*, oportunamente descrito na Seção 4.1.

Por tratar-se de um método de seleção automática de heurísticas, este mecanismo tem como objetivo encontrar automaticamente a heurística que conduz ao melhor resultado para uma dada instância. A ideia é encontrar a melhor sequência de *variáveis/ações*, que efetue a manipulação inicial dos estados das variáveis, para posterior emprego de uma rotina de pós-processamento, capaz de conduzir a uma solução viável ótima. Para efetuar a exploração do espaço de estados, utiliza-se um *Algoritmo Genético* [50, 56], definido na Seção 4.1.2.

Com o objetivo de realizar a análise experimental, utilizou-se um pacote de heurísticas chamado UBCSAT [19], com implementação eficiente para Max-SAT dentre os vários métodos heurísticos conhecidos e referenciados na literatura, tais como: *Walk-SAT*, *Tabu-search*, *Random Walk* e *Novelty*. Seus resultados são demonstrados na Seção 4.2 e referem-se à aplicação do método a um conjunto de instâncias da competição sobre o problema SAT [20] e outro conjunto de instâncias do problema Max-SAT [21]. Por fim, são feitas algumas considerações sobre os resultados



encontrados na Seção 4.3.

## 4.1 Descrição da Abordagem

Como visto na Seção 2.1, o problema Max-SAT possui a característica de maximização local. Ou seja, ao olhar para uma variável específica, é possível realizar alterações para melhorar o valor de sua função, resultando também na melhora da função global. Entretanto, para cada instância do problema, existe uma sequência diferente de alterações que conduzem à *solução ótima*.

Para melhor descrever algumas características do método heurístico proposto em 4.1.1, far-se-á uma comparação com métodos mais simples. Imaginando um procedimento simples de verificação das variáveis da instância, tem-se o seguinte procedimento:

*Atualização sequencial.* Com base na definição do problema Max-SAT, apresentada na Seção 2.1, e, principalmente, pela característica de *maximização por análise local* presente na formulação, pode-se criar um procedimento simples para efetuar uma análise e atualização sucessiva dos valores das variáveis com base na seguinte regra:

**Definição 4.1 Regra de maximização.** *O valor da variável  $x_i$  é alterado somente quando houver um acréscimo em  $f$  (Equação 2.4).*

Se esta regra for sempre respeitada, o procedimento de atualização sequencial maximizará o valor de  $f$  em um número finito de passos. Entretanto, esse é um procedimento lento, sem garantia de solução ótima e fortemente influenciado pela escolha inicial dos valores para as variáveis em  $X$  e pela sequência de variáveis visitadas. Alguns destes problemas podem ser resolvidos com o uso do método a seguir.

*Busca gulosa.* Aqui, o procedimento verifica qual é a variável cuja alteração de estado acarreta no maior ganho em  $f$ . Após esta pesquisa inicial, a variável com maior ganho é alterada. Esta alteração fará com que o valor de  $f$  seja alterado em direção ao ótimo. Ao repetir este procedimento até que nenhuma variável possa ser alterada, uma solução para a instância será encontrada.

A busca gulosa possui tempo de execução relativamente baixo e, em geral, produz boas soluções. No entanto, ele continua sendo sensível à configuração inicial das variáveis. E, por tratar-se de um método guloso, pode forçar alterações boas para o momento, mas que não necessariamente conduzirão a boas soluções.

Portanto, nenhum dos dois métodos, apresentados até o momento, é capaz de fugir de um ótimo local ainda distante de um ótimo global. Para fornecer uma alternativa a este tipo de situação, é possível efetuar uma pequena modificação nestes processos, permitindo que a regra seja desrespeitada em alguns momentos. Um exemplo mais elaborado deste tipo de abordagem é o método *Simulated Annealing* [37, 57], que realiza uma busca no espaço de soluções, desobedecendo a regra de aceitar uma nova solução com uma certa probabilidade, inicialmente alta, e que diminui gradativamente com o passar do tempo.

Inspirado nos métodos expostos e no mecanismo de busca presente no *Simulated Annealing*, optou-se por desenvolver uma heurística que, em um primeiro momento, parece aleatória e despropositada, mas, capaz de captar características da instância. E, mais ainda, capaz de efetuar a seleção desta heurística automaticamente para cada instância do problema.

Com base nestes objetivos, é possível selecionar duas características principais desejáveis em uma heurística:

- (1) **Escolha do roteiro.** A *seqüência de visita das variáveis* deve ser definida especificamente e automaticamente para a instância em questão;
- (2) **Escolha da ação.** Uma *ação* é definida para cada visita realizada. Ou seja, a ação indicará se a regra (Definição 4.1) será, ou não, respeitada em um determinado momento da execução da heurística.

#### 4.1.1 Fuga de Máximos Locais de Baixa Qualidade

Em síntese, uma *heurística H* consiste em uma seqüência de *variáveis/ações*, com tamanho máximo (número máximo de elementos na seqüência) e número mínimo de variáveis distintas pré-definidos.

O tamanho máximo será dado em função do tamanho da instância. Este valor será definido através de uma análise experimental inicial, descrita na Subseção 4.2.3.

Além disso, os experimentos também serão responsáveis por definir o *número mínimo de variáveis distintas* em uma heurística, o que é conhecido como *regra de validação* e tem a seguinte definição:

**Definição 4.2 Regra de validação de uma heurística.** *Uma heurística é considerada válida se, e somente se, apresentar pelo menos  $T_d$  variáveis distintas em sua sequência, para um  $T_d$  dado e fixo.*

Visando possibilitar a fuga de pontos de máximos locais de baixa qualidade, introduz-se um fator aleatório, através do qual para cada variável da sequência é realizado um sorteio com probabilidade igual sobre a decisão de:

0 - não respeitar a **regra de maximização**;

1 - respeitar a **regra de maximização**;

2 - não fazer nada.

Para exemplificar, suponha, que  $V = \{x_4, x_2, x_4, x_3, x_1, x_5, x_2\}$  e  $A = \{1, 0, 2, 1, 0, 2, 1\}$  sejam escolhidos, respectivamente, como sequência de variáveis e ações para a instância  $C_1 \wedge \dots \wedge C_5$  do problema Max-SAT, dada por:

$(x_1 \vee x_2) \wedge (x_3 \vee x_4 \vee \neg x_1) \wedge (x_2 \vee \neg x_5) \wedge (x_4 \vee \neg x_1 \vee \neg x_2) \wedge (\neg x_3 \vee x_2 \vee x_5)$ . Suponha ainda que o ponto inicial escolhido seja  $s = (V, F, V, V, F)$ , para o qual  $f(s) = 4$  e apenas a cláusula  $C_5$  não seja satisfeita por esta atribuição.

Analisando sucessivamente as variáveis em  $V$ , o primeiro elemento será a variável  $x_4$ , cujos componentes  $q_{43}$  e  $q_{44}$ <sup>1</sup> são ambos iguais a 0 e, neste caso, a regra diz que o valor de  $x_4$  deve permanecer inalterado. Como  $a_1 = 1$ , o procedimento respeitará a regra. No segundo elemento, a variável  $x_2$  possui  $q_{23} = 0$  e  $q_{24} = 1$ . Portanto, o valor de  $x_2$  deveria ser alterado (segundo a regra), mas como  $a_2 = 0$ , o procedimento não alterará o valor de  $x_2$ . Um cenário diferente ocorre para o terceiro elemento. Nele, independente dos valores de  $q_{43}$  e  $q_{44}$ , a ação  $a_3 = 2$  define que nada deverá ser feito com a variável  $x_4$ . Outro cenário pode ser observado no quarto elemento, onde a variável  $x_3$  é selecionada. Esta variável possui  $q_{33} = 0$  e  $q_{34} = 1$  e como  $a_4 = 1$ , o procedimento respeitará a regra, alterando o estado das variáveis para

<sup>1</sup>Cenários apresentados em 2.1.2:  $q_{43}$  e  $q_{44}$  indicam, respectivamente, número de ocorrências em que o literal  $x_4$  (ou  $\neg x_4$ ) está presente e é o único  $V$ , e número de cláusulas insatisfeitas que contém  $x_4$  como um dos seus literais.

$s' = (V, F, F, V, F)$ . O procedimento é repetido até que todas as variáveis em  $V$  sejam testadas.

Portanto, o procedimento descrito pode ser representado pelo Algoritmo 4, que executa uma série de análises e alterações locais em uma dada instância, seguindo uma determinada *heurística*  $H$ .

É possível notar que, após a aplicação da heurística, não existe garantia de que a configuração final das variáveis seja uma solução ótima ( $f$  máximo). Assim, após a execução do algoritmo, deve-se executar uma rotina de pós-processamento capaz de encontrar uma solução ótima para a instância.

---

**Algoritmo 4** Heurística para *SAHe-SAT*

---

- 1: Dada uma sequência de pares ordenados  $H = \langle (v_1, a_1), (v_2, a_2), \dots, (v_k, a_k) \rangle$ , tal que  $v_i \in X$  e  $a_i = \{0, 1, 2\}$ ;
  - 2: Efetue a inicialização aleatória do estado de cada variável  $x_i \in X$ ;
  - 3: **para todo**  $v_i$  **faça**
  - 4:     Calcule os valores dos cenários  $q_{v_i3}$  e  $q_{v_i4}$ ;
  - 5:     **se**  $q_{v_i3} < q_{v_i4}$  e  $a_i = 1$  **então**
  - 6:         Altere o valor de  $v_i$ ;
  - 7:     **senão se**  $q_{v_i3} > q_{v_i4}$  e  $a_i = 0$  **então**
  - 8:         Altere o valor de  $v_i$ ;
  - 9:     **fim se**
  - 10: **fim para**
- 

### 4.1.2 Exploração do Espaço de Estados

A escolha da *heurística*  $H$  pode ser vista como um problema de otimização sobre todas as combinações de *variáveis/ações* possíveis. Para  $n$  variáveis e uma sequência de tamanho fixo em  $t = |V|$ , tem-se  $3n^t$  combinações distintas.

Tal como um problema de otimização, o processo de escolha é extremamente desestruturado e não-diferenciável. Portanto, optou-se por construir um *algoritmo genético* que opera sobre indivíduos que representam as várias possibilidades para  $H$ , com o objetivo de encontrar o indivíduo que maximiza o valor de  $f$  (Equação 2.4 do Capítulo 2). Cada indivíduo é avaliado por uma *função objetivo*, que fornece uma medida de qualidade deste em relação à instância do problema Max-SAT. A função é denotada por:

$$\varphi(H, Hp) = f(s^*), \tag{4.1}$$

onde  $s^*$  corresponde ao estado das variáveis após a execução do Algoritmo 4 sobre a sequência  $H$  específica, e de alguma rotina de pós-processamento  $Hp$ , que é a mesma para todos os indivíduos.

Um exemplo de indivíduo para instâncias com até 5 variáveis é apresentado na Figura 4.1, onde o primeiro elemento da sequência é a variável  $x_4$ , e a ação sobre esta variável indica que a regra deverá ser respeitada. O segundo elemento da sequência é a variável  $x_2$ , e a ação sobre a variável indica que a regra deverá ser desrespeitada.

|                  |   |   |   |   |   |   |   |   |
|------------------|---|---|---|---|---|---|---|---|
| <i>variáveis</i> | 4 | 2 | 4 | 1 | 3 | 1 | 5 | 2 |
| <i>ações</i>     | 1 | 0 | 2 | 1 | 0 | 0 | 2 | 1 |

Figura 4.1: Exemplo de indivíduo para uma instância com 5 variáveis.

Com o objetivo de minimizar a possibilidade de uma má escolha da configuração inicial  $s$ , optou-se por reinicializar os valores das variáveis 10 vezes. Para cada escolha de  $s$ , o Algoritmo 4 deve ser executado, obtendo um valor  $s'$ . A rotina de pós-processamento  $Hp$  será executada apenas sobre o melhor valor de  $f(s')$ , obtendo uma configuração final  $s^*$  de valores para as variáveis e  $f(s^*)$  como medida de qualidade do indivíduo.

### Algoritmo Genético (AG) para a Formulação

O Algoritmo 5 descreve o algoritmo genético utilizado nos experimentos da Seção 4.2, que é uma variante do *algoritmo genético geracional* de MITCHELL [50]. Ele evolui até encontrar o valor ótimo conhecido ou até atingir um tempo máximo pré-definido. Cada geração compreende um número fixo de indivíduos e a saída é o melhor indivíduo encontrado no final da sua execução.

A população inicial é preenchida aleatoriamente, respeitando a *regra de validação de uma heurística*. Cada nova geração é obtida a partir da prévia, realizando primeiramente um passo elitista, onde os indivíduos mais aptos, dada uma fração fixa da população, são copiados diretamente para a nova geração. Em seguida, esta nova geração é preenchida por novos indivíduos, obtidos através da aplicação do *crossover* (como um par) ou *mutação* (individualmente) sobre indivíduos selecionados da geração anterior.

---

**Algoritmo 5** Algoritmo Genético para *SAHe-SAT*

---

```
1:  Seja  $P$  a população de cromossomos, e  $I$  a instância a ser resolvida.
2:  Inicializar  $P$  e  $I$ 
3:  enquanto o critério de parada não for satisfeito faça
4:    Avalie os indivíduos em  $P$  para a instância  $I$ 
5:    Ordene  $P$  pelo valor decrescente de aptidão
6:    Faça o elitismo dos  $e$  melhores indivíduos da população  $P$ 
7:    Insira a elite em uma nova população  $P'$ 
8:    repita
9:      Escolha aleatoriamente um número  $t$  no intervalo  $[0, 1]$ 
10:     se  $t < T_c$  então
11:       Selecione por rank linear 2 indivíduos em  $P$ 
12:       Aplique o operador de crossover
13:       Insira os novos indivíduos em  $P'$ 
14:     senão
15:       Selecione por rank linear 1 indivíduo em  $P$ 
16:       Aplique o operador de mutação
17:       Insira o novo indivíduo em  $P'$ 
18:     fim se
19:   até a população  $P'$  estar completa
20:    $P \leftarrow P'$ 
21: fim enquanto
```

---

## O Método de Seleção

Existem inúmeros esquemas de seleção e, após alguns experimentos iniciais, verificou-se que os indivíduos da população de heurísticas apresentavam valores de aptidão muito próximos, caracterizando uma pressão de seleção baixa e indicando que o valor da *função-objetivo* não deve ser usado diretamente como valor de *aptidão* de um indivíduo.

A fim de transformar a medida dada pela *função-objetivo* (Equação 4.1) e fornecer aos melhores indivíduos maior possibilidade de propagação de seus genes para as gerações futuras, escolheu-se o esquema de seleção por *rank linear* (para maiores informações sobre este esquema de seleção, consulte [58]). Neste esquema, os indivíduos são ordenados de maneira decrescente, de acordo com o valor da sua *função-objetivo* ( $\varphi(P_i, Hp)$ ) para um indivíduo  $P_i = [V, A]$ , e recebem uma probabilidade de seleção determinada por sua posição no *rank*.

A cada indivíduo  $P_k$  é atribuída uma probabilidade de seleção  $p_k$  ( $p_k > 0$ ) de

acordo com uma certa distribuição, desde que seja respeitada a seguinte restrição:

$$\sum_{i=1}^K p_i = 1. \quad (4.2)$$

Escolhida uma população de  $K$  indivíduos, representados por  $P$ , considere  $k$  como sendo a posição do  $k$ -ésimo indivíduo na população, onde  $k \in [1, K]$ . A *função de aptidão* será calculada com base na normalização linear da *função objetivo* e representada por  $\varphi_n(P_k)$ . Os valores serão distribuídos dentro do intervalo  $[\Phi, 1]$  através de uma reta decrescente. O parâmetro  $\Phi$  define a nova pressão de seleção, ou seja, quanto maior o valor de  $\Phi$ , maior a pressão. O valor de  $\varphi_n(P_k)$  depende da posição do indivíduo  $k$  dentro da população, e é fornecida pela seguinte equação:

$$\varphi_n(P_k) = \Phi - \left( \frac{\Phi - 1}{K - 1} \right) (k - 1) \quad (4.3)$$

Assim, a probabilidade do  $k$ -ésimo indivíduo ser selecionado,  $k \in [1, K]$ , representada por  $p_k$ , é dada pela equação:

$$p_k = \frac{\varphi_n(P_k)}{\sum_{j=1}^K \varphi_n(P_j)} \quad (4.4)$$

Como estas probabilidades são positivas e seu somatório total é igual a 1, a seleção de um indivíduo da população é feita pelo sorteio de um número aleatório  $a$ , no intervalo  $[0, 1]$ . Em seguida, soma-se as probabilidades dos indivíduos, iniciando em  $p_1$ , até que o seu resultado seja maior que o número  $a$  sorteado. Logo, o indivíduo selecionado, corresponde ao menor  $k$  que satisfaz a seguinte inequação:

$$\sum_{i=1}^k p_i \geq a \quad (4.5)$$

Como mencionado anteriormente, o AG implementado neste trabalho é geracional, com toda a população substituída por novos indivíduos. O artifício do elitismo é aplicado para que o material genético dos melhores não seja perdido no processo de reprodução.

A cada geração, faz-se o elitismo dos  $e$  melhores indivíduos da população e o restante da população é formada pela seleção e aplicação dos operadores genéticos.

## Operadores Genéticos

Os indivíduos da nova população são gerados pelo operador de recombinação (*crossover*) ou pelo operador de mutação. Esta escolha é feita por sorteio de um número aleatório no intervalo  $[0, 1]$ . Se o valor escolhido for menor que a taxa de probabilidade de *crossover*, representada por  $T_c$ , então, o *crossover* é aplicado sobre dois indivíduos selecionados. Caso contrário, o operador de mutação será aplicado sobre um indivíduo.

### Operador Genético de *Crossover*

Esta abordagem utiliza uma variação do *crossover* de dois pontos, tipificando um caso particular do *crossover* de  $n$  pontos, onde sorteia-se  $n$  pontos e o cromossomo dos indivíduos *pais* é dividido em  $n + 1$  segmentos. Os *filhos* receberão os segmentos concatenados de maneira alternada, ora de um pai, ora de outro. Observações sobre *crossover* de  $n$  pontos também são encontradas em [50, 56].

A modificação do modelo de *crossover* de dois pontos padrão é necessária, pois para esta formulação, a sequência gerada pelo operador deve respeitar a *regra de validade de uma heurística* (Definição 4.2). Para que esta regra de validade seja respeitada, deve-se escolher um segmento dentro de cada sequência selecionada que a respeite, dividindo o indivíduo em três segmentos de material genético:  $P_i[1, \dots, (a_i - 1)]$ ,  $P_i[a_i, \dots, b_i]$  e  $P_i[(b_i + 1), \dots, t_i]$ , onde  $t_i$  é o tamanho do indivíduo  $i$ .

O segmento  $P_i[a_i, \dots, b_i]$  deve ser um segmento minimal do indivíduo  $i$  que respeite a regra de validação. Para isto, dois pontos de corte,  $a_i$  e  $b_i$ , são escolhidos: inicialmente pelo sorteio de um ponto  $c$  ( $0 \leq c \leq t_i$ ), e pela definição de  $a_i = b_i = c$ . Após este sorteio, o segmento  $P_i[a_i, \dots, b_i]$  é expandido para ambos os lados até que satisfaça a regra de validação da heurística.

A Figura 4.2 ilustra o processo de busca pelo segmento minimal de dois indivíduos para um dado  $T_d = 3$ . Neste processo, realiza-se o sorteio de um ponto  $a_1 = b_1 = 5$  para o indivíduo  $P_1$ . O segmento é expandido, sucessivamente, até que compreenda os genes  $P_1[3, 4, 5, 6] = \{1, 2, 3, 3\}$ . Como este segmento não é um trecho minimal, faz-se uma compressão do mesmo, retirando-se o gene 6. O processo é repetido para  $P_2$ , sorteando um novo ponto ( $a_2 = b_2 = 6$ ), e gerando o segmento minimal



$P_2[3, 4, 5] = \{3, 2, 5\}$ ; a partir destes segmentos, os novos indivíduos  $F_1$  e  $F_2$  são gerados.

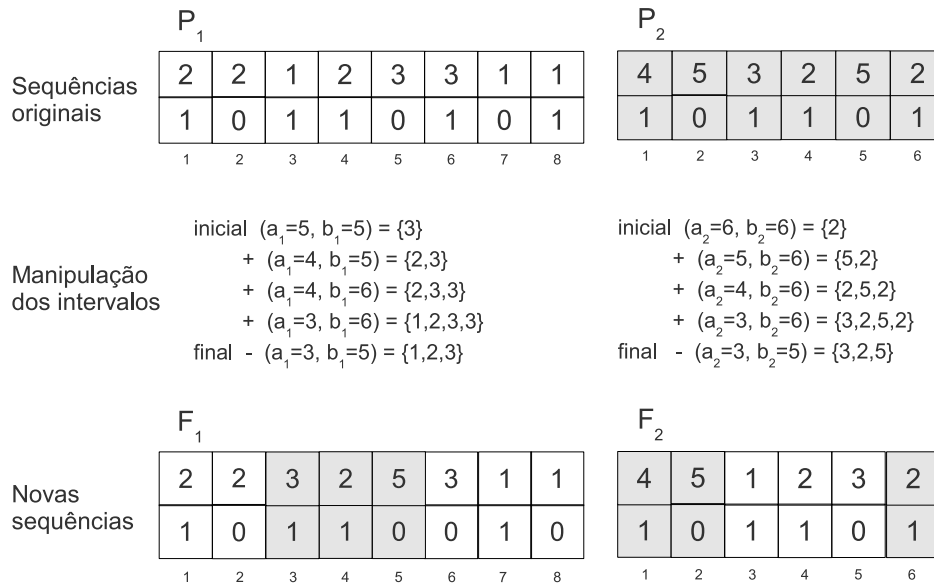


Figura 4.2: Exemplo de aplicação do *crossover* de 2 pontos para  $T_d = 3$ .

## Operador Genético de Mutação

O operador de mutação é aplicado somente quando o operador de *crossover* não o for (probabilidade  $1 - T_c$ ). Ao optar pelo uso do operador de mutação, um número fixo de genes do indivíduo, dado pela taxa  $T_m$ , será escolhido e alterado. A mutação sobre um gene escolhido aleatoriamente ocorre de duas maneiras:

- *mutação completa*: o gene sorteado é alterado completamente, tanto a variável como a ação;
- *mutação simples*: apenas a ação é alterada.

O primeiro tipo de mutação será aplicado sempre que o seu resultado gerar uma sequência válida (respeitando a Definição 4.2). Do contrário, a mutação simples será aplicada.

Para o caso da mutação completa, o gene só deverá ser alterado se o resultado for uma sequência válida. Desta forma, é necessário realizar um processamento inicial para verificar o número de ocorrências de cada variável e atualizar os contadores,

sempre que algum gene for alterado. Logo, o indivíduo gerado pelo processo de mutação também será válido.

A Figura 4.3 ilustra o uso do operador de mutação sobre um cromossomo. O indivíduo  $P_1$  dá origem ao indivíduo  $F_1$  após realizar uma sequência de mutações. Neste exemplo, os genes sorteados para alteração foram:  $g_4$ ,  $g_2$ ,  $g_8$ ,  $g_5$  e  $g_2$ . Nota-se que quando  $g_2$  é visitado pela primeira vez, sua alteração completa não pode ser realizada, pois o vértice 3 aparece apenas uma vez no cromossomo. Então, apenas a ação do gene é alterada (mutação simples). Quando uma nova tentativa de mutação em  $g_2$  é feita, o gene já pode ser alterado, pois 3 agora aparece duas vezes no indivíduo ( $g_2$  e  $g_5$ ). Finalizando este processo,  $F_1$  será inserido na nova população.

|       |  | indivíduo selecionado |   |   |   |   |   |   |   |   |    |
|-------|--|-----------------------|---|---|---|---|---|---|---|---|----|
| $P_1$ |  | 5                     | 3 | 2 | 4 | 2 | 4 | 1 | 4 | 1 | 5  |
|       |  | 1                     | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0  |
|       |  | 1                     | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Sequência de genes sorteados:  $\{g_4, g_2, g_8, g_5, g_2\}$   
 Novas variáveis/ações:  $\{5/0, 2/1, 2/0, 3/1, 4/0\}$

|       |  | indivíduo após mutação completa |   |   |   |   |   |   |   |   |    |
|-------|--|---------------------------------|---|---|---|---|---|---|---|---|----|
| $F_1$ |  | 5                               | 4 | 2 | 5 | 3 | 4 | 1 | 2 | 1 | 5  |
|       |  | 1                               | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0  |
|       |  | 1                               | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Figura 4.3: Exemplo de mutação no indivíduo  $P_1$ , gerando um novo indivíduo  $F_1$ .

## 4.2 Experimentos Computacionais

Com o objetivo de realizar a análise experimental do método *SAHe-SAT*, descrito na Seção 4.1, foram efetuados uma série de experimentos computacionais. Os resultados presentes nesta seção referem-se a dois conjuntos de instâncias extraídos de competições para métodos exatos de resolução dos problemas SAT e Max-SAT. O primeiro deles é um conjunto de instâncias extraído do repositório da competição sobre o problema SAT [20], chamado, a partir deste momento, de *conjunto de 2004*,

o qual possui 100 (cem) instâncias industriais. O segundo conjunto, contendo 112 (cento e doze) instâncias industriais, foi extraído da terceira avaliação de métodos para resolução de instâncias do problema Max-SAT [21], referenciado como *conjunto de 2008*.

Apesar de utilizar instâncias provenientes de uma competição sobre o problema SAT, o objetivo deste método é a resolução do problema em sua versão de otimização. Ou seja, encontrar o número máximo de cláusulas satisfeitas para uma dada instância. Além disso, nas duas competições referenciadas foram utilizados métodos exatos, diferentemente do que se propõe aqui. Desta forma, os resultados alcançados (valor ótimo conhecido) em [21] foram utilizados como critério de parada do *SAHe-SAT* e a implementação exata que obteve o melhor desempenho na competição sobre Max-SAT, o *MSUnCore* [28], foi selecionada com o objetivo de encontrar o valor ótimo para as instâncias das competições que permaneciam sem solução. Mesmo após várias tentativas de execução do *MSUnCore*, algumas instâncias das duas competições permaneceram sem solução.

O conjunto completo de instâncias de *benchmark*, utilizadas nos experimentos computacionais, está descrito nas Tabelas A.1 a A.4, do Anexo A, que também apresenta o número máximo de cláusulas satisfeitas e o tempo em que a solução foi encontrada pelo algoritmo exato *MSUnCore*. As entradas omitidas, nessas tabelas, indicam os casos que o algoritmo exato não encontrou uma solução.

### 4.2.1 Heurísticas de Pós-Processamento

Para avaliar o desempenho do método proposto, foi selecionado um pacote de heurísticas chamado *UBCSAT* [19], com implementação eficiente e padronizada de vários métodos heurísticos conhecidos e referenciados na literatura. Destes métodos, após a realização de experimentos iniciais, foram selecionados dez métodos para os experimentos de calibração do modelo proposto. São eles:

*gsat*: busca gulosa para SAT [29]. Para cada reinicialização aleatória dos valores das variáveis (respeitando um máximo definido) realiza-se um conjunto máximo de trocas (respeitando um máximo definido), até que a instância seja satisfeita. Cada troca é realizada, escolhendo-se a variável que, ao ser modificada, gerará o maior ganho para a função  $f(x)$ ;

*gwsat*: *gsat* com *random walk* [30]. Também conhecido como *gsat-walk*, *rws-gsat*, *grsat* ou *gsatrw*. Com probabilidade  $p$ , uma variável deve ser escolhida de uma cláusula insatisfeita e trocada a sua atribuição. Caso contrário ( $1 - p$ ), o procedimento padrão do *gsat* deve ser efetuado;

*gsat-tabu*: *gsat* com *busca tabu* [59]. Também conhecido como *tsat*, efetua o mesmo processo do *gsat*, mas mantém uma lista de variáveis alteradas (lista tabu), não permitindo que elas sejam modificadas por um certo período;

*hsat*: *gsat* com informação sobre histórico [60]. O algoritmo escolhe dentre as variáveis selecionadas para troca, aquela que foi alterada há mais tempo. Se dentre as variáveis selecionadas existirem variáveis que nunca foram alteradas, a escolha deverá ser feita de acordo com uma ordem fixa e pré-estabelecida;

*hwsat*: *hsat* com *random walk* [61]. Com probabilidade  $p$ , uma variável deve ser escolhida de uma cláusula insatisfeita e trocada a sua atribuição. Caso contrário ( $1 - p$ ), o procedimento padrão do *hsat* deve ser efetuado;

*novelty*: seleciona variáveis de uma cláusula insatisfeita, como em *gwsat*, mas prioriza a variável alterada há mais tempo [32]. Considerando a primeira e a segunda melhor variáveis desta seleção, se a primeira melhor não é a variável trocada mais recentemente na cláusula, então ela deve ser selecionada. Caso contrário, com probabilidade  $p$ , seleciona a segunda melhor variável, e com probabilidade  $1 - p$  a melhor variável;

*adaptnovelty+*: *novelty* adaptativo com *random walk* [33]. Em cada passo de busca, com probabilidade  $wp$ , a variável a ser trocada é selecionada aleatoriamente na cláusula escolhida, enquanto nos demais casos a variável é selecionada de acordo com a heurística *novelty*. Além disso, o termo *adaptativo* representa a adição de uma variável dinâmica utilizada para aumentar ou diminuir a probabilidade da escolha gulosa da heurística *novelty*, fazendo com que a busca fuja mais rapidamente de um estado de estagnação ou diminua a possibilidade de movimentos aleatórios;

*saps*: procedimento de busca local dinâmica (DLS). Este método associa pesos às cláusulas (inicialmente 1) e executa dois passos, sucessivamente, até satisfazer

a instância ou atingir um número máximo de iterações [34]. No primeiro, uma busca com pesos é realizada, onde as variáveis que aparecem em cláusulas insatisfeitas são randomicamente escolhidas e trocadas, enquanto houver melhora no total de pesos das cláusulas insatisfeitas. Quando não houver possibilidade de melhora (mínimo local), a busca continuará, com probabilidade  $wp$ , pela escolha aleatória de variáveis das cláusulas insatisfeitas. O segundo passo efetua o ajuste dos pesos apenas das cláusulas insatisfeitas (multiplicando por  $\alpha$ ) e, logo após esse processo, com probabilidade  $p_{smooth}$ , efetua um passo de suavização, onde o peso de todas as cláusulas é ajustado com base na média dos pesos após aquele primeiro ajuste;

*sapsnr*: versão derandomizada do *saps* [35]. Difere do *saps* nos seguintes aspectos: na escolha entre duas variáveis com mesmo ganho, o algoritmo escolhe deterministicamente a variável com menor índice, fixando o parâmetro  $wp$  em zero (desta forma, o passo aleatório nunca será executado); e o estágio de suavização probabilístico é substituído por uma suavização determinística e periódica, realizada a cada  $1/p_{smooth}$  ocorrências do mínimo local;

*walksat-tabu*: Seleciona uma variável que minimiza o número de cláusulas insatisfeitas [32]. Entretanto, a cada passo, a troca de qualquer variável, que tenha sido alterada nos últimos  $t$  passos (lista tabu) é recusada; se todas as variáveis da cláusula escolhida estiverem na lista tabu, é necessário optar por uma variável de outra cláusula insatisfeita e, caso estejam na lista tabu, devem ser ignoradas.

A inclusão de novas heurísticas e de módulos às heurísticas existentes é feita de modo simples e direto, pois o pacote *UBCSAT* utiliza um sistema de gatilhos, no qual as novas funcionalidades podem ser incluídas.

O pacote foi modificado para executar o *SAHe-SAT* e controlar a execução de uma determinada heurística de pós-processamento  $Hp$ . Desta forma, o AG realiza a manipulação dos seus operadores genéticos e, como parte do cálculo da aptidão de cada indivíduo de sua população (Passo 4 do Algoritmo 5), efetua a execução da heurística  $Hp$ . Para tanto, quando um indivíduo da população é selecionado, a heurística que ele representa (sequência *variáveis/ações*) é executada gerando um

novo conjunto de valores para as variáveis, utilizados como entrada da *Hp*. Após a sua execução, obtém-se uma solução viável para a instância e, por consequência, é obtida a aptidão do indivíduo.

## 4.2.2 Ambiente

Os algoritmos deste trabalho foram desenvolvidos na linguagem C padrão *ANSI*, e os testes realizados em duas configurações diferentes de máquinas:

- *1a configuração*: cluster do Laboratório de Inteligência Artificial do PESC-COPPE/UFRJ, formado por 15 máquinas iguais, com processadores Intel Dual Core, 3.0GHz de frequência, 4 GBytes de memória, disco rígido de 80GBytes e uma rede Gigabit Ethernet. O sistema operacional é o *Ubuntu 9.04* com *kernel 2.6.24-28* e compilador *gcc* de versão 4.2;
- *2a configuração*: cluster do Laboratório de Inteligência Artificial do PESC-COPPE/UFRJ, formado por 10 máquinas iguais, com processadores Intel Quad Core, 3.0GHz de frequência, 4 GBytes de memória, disco rígido de 80GBytes e uma rede Gigabit Ethernet. O sistema operacional é o *Ubuntu 9.04* com *kernel 2.6.24-28* e compilador *gcc* de versão 4.2.

## 4.2.3 Calibração do Modelo

O conjunto de treinamento usado para calibração do *SAHe-SAT* foi selecionado a partir do sub-conjunto de instâncias de tamanho médio do conjunto de 2004. Estas instâncias estão listadas na Tabela 4.1, incluindo-se o valor ótimo para algumas delas, e o tempo (em minutos) que o método exato *MSUnCore* [28] levou para encontrá-los.

Tabela 4.1: Instâncias utilizadas para calibração do *SAHe-SAT*.

| Instância | Variáveis | Cláusulas | <i>MSUnCore</i> |          |
|-----------|-----------|-----------|-----------------|----------|
|           |           |           | Ótimo           | Tempo(m) |
| C880mul   | 1612      | 9373      | 9372            | 26.570   |
| term1mul  | 3504      | 22229     | 22228           | 826.832  |
| c3540mul  | 5248      | 33199     | -               | -        |
| vdamul    | 5444      | 34509     | 34508           | 99.402   |
| am_8_8    | 7361      | 25538     | -               | -        |

As instâncias, onde o valor ótimo não está definido, indicam situações que o método exato não gerou resultado. O problema deriva do modo como o método exato funciona. O *MSUnCore* é baseado na identificação de sub-fórmulas não satisfeitas, e o resolvidor SAT, utilizado por ele, não consegue resolver a instância, significando que o *MSUnCore*, eventualmente, abortará. A razão para abortar se deve à alguma limitação de recursos, como por exemplo, o tempo de CPU ou a memória.

Mesmo após a realização de algumas experiências, com os melhores métodos exatos, publicamente disponíveis, não foi possível resolver essas instâncias. Desta forma, verificou-se que os resolvidores exatos atuais para SAT não são capazes de resolver essas instâncias, tornando-os mais atraentes para o método proposto.

### Execução do *SAHe-SAT* + *Hp*'s

Os testes preliminares com o *SAHe-SAT* foram realizados com o objetivo de encontrar a melhor configuração de parâmetros utilizada no processo evolucionário desempenhado pelo AG.

A Tabela 4.2 descreve os melhores resultados encontrados pelo *SAHe-SAT* com algumas heurísticas de pós-processamento (*Hp*'s), após a definição desses parâmetros. Os valores tabulados referem-se ao número máximo de cláusulas satisfeitas após a execução de cada *SAHe-SAT* + *Hp*. Algumas entradas na tabela estão destacadas (em negrito) para indicar que o *SAHe-SAT*+*Hp* encontrou a melhor solução, dentre as variantes do método *SAHe-SAT*, para a instância em questão.

Tabela 4.2: Resultados do *SAHe-SAT*+*Hp*'s para conjunto de calibração.

| Heurísticas               | Instâncias  |              |              |              |              |
|---------------------------|-------------|--------------|--------------|--------------|--------------|
|                           | C880mul     | term1mul     | c3540mul     | vdamul       | am_8_8       |
| <i>SAHe+adaptnovelty+</i> | <b>9372</b> | 22226        | 33157        | 34460        | <b>25525</b> |
| <i>SAHe+gsat</i>          | 8386        | 19741        | 29369        | 30523        | 21621        |
| <i>SAHe+gsat-tabu</i>     | 8413        | 19763        | 29367        | 30552        | 21615        |
| <i>SAHe+gwsat</i>         | 8399        | 19772        | 29357        | 30560        | 21640        |
| <i>SAHe+hsat</i>          | 8413        | 19700        | 29366        | 30532        | 21614        |
| <i>SAHe+hwsat</i>         | 8393        | 19773        | 29401        | 30558        | 21628        |
| <i>SAHe+novelty</i>       | <b>9372</b> | <b>22228</b> | <b>33177</b> | 34491        | 25325        |
| <i>SAHe+saps</i>          | 8359        | 19721        | 29342        | 30540        | 21613        |
| <i>SAHe+sapsnr</i>        | 8357        | 19720        | 29338        | 30545        | 21581        |
| <i>SAHe+walksat-tabu</i>  | <b>9372</b> | 22214        | 33164        | <b>34492</b> | 25466        |

É possível notar que, das dez heurísticas do pacote testadas, apenas três delas

(*adaptnovelty+*, *novelty* e *walksat-tabu*) tiveram resultados iguais ou próximos do valor ótimo encontrado pelo algoritmo exato, conforme a Tabela 4.2.

Para complementar estas informações, a Tabela 4.3 descreve o tempo (em minutos) que os melhores resultados foram encontrados. É possível notar que as heurísticas que obtiveram melhores resultados, também foram as mais rápidas, inclusive quando comparadas ao resultado do algoritmo exato. Algumas entradas na tabela estão destacadas (em negrito) para indicar o menor tempo para a instância em questão.

Tabela 4.3: Tempo (em minutos) do *SAHe-SAT+Hp's* para conjunto de calibração.

| Heurísticas               | Instâncias   |               |               |               |               |
|---------------------------|--------------|---------------|---------------|---------------|---------------|
|                           | C880mul      | term1mul      | c3540mul      | vdamul        | am__8         |
| <i>SAHe+adaptnovelty+</i> | <b>2.689</b> | 52.222        | 123.489       | 132.070       | 104.636       |
| <i>SAHe+gsat</i>          | 142.191      | 78.582        | 223.611       | 166.711       | 183.455       |
| <i>SAHe+gsat-tabu</i>     | 175.226      | 137.922       | 135.572       | 122.811       | 158.082       |
| <i>SAHe+gwsat</i>         | 198.069      | 127.819       | 197.362       | 85.046        | 217.237       |
| <i>SAHe+hsat</i>          | 228.612      | 68.295        | 70.923        | 139.944       | 96.029        |
| <i>SAHe+hwsat</i>         | 133.892      | 64.727        | 101.101       | 138.591       | 130.133       |
| <i>SAHe+novelty</i>       | 3.709        | <b>30.500</b> | 100.397       | 103.686       | 157.474       |
| <i>SAHe+saps</i>          | 130.242      | 109.864       | 122.748       | 222.224       | 158.510       |
| <i>SAHe+sapsnr</i>        | 111.179      | 130.619       | 113.469       | 226.950       | 103.022       |
| <i>SAHe+walksat-tabu</i>  | 3.306        | 79.280        | <b>51.120</b> | <b>67.233</b> | <b>75.139</b> |

### Execução das Heurísticas Puras (*Hp's*)

Com o objetivo de comparar o resultado do modelo proposto com o uso da *heurística de pós-processamento pura*, as heurísticas, descritas na Subseção 4.2.1, foram executadas várias vezes com reinicialização aleatória das variáveis da instância, até que o valor ótimo fosse encontrado ou até que o tempo máximo (4 horas) fosse atingido, armazenando-se o melhor resultado encontrado.

Seus resultados (número máximo de cláusulas satisfeitas) e o tempo são apresentados, respectivamente, nas Tabelas 4.4 e 4.5, onde é possível constatar que o bom desempenho foi mantido pelo mesmo grupo de heurísticas (*adaptnovelty+*, *novelty* e *walksat-tabu*). Analogamente, algumas entradas das Tabelas 4.4 estão destacadas (em negrito) para indicar melhores valores (solução e tempo) para uma dada instância.



Tabela 4.4: Resultados das  $Hp$ 's para conjunto de calibração.

| Heurísticas          | Instâncias  |              |              |              |              |
|----------------------|-------------|--------------|--------------|--------------|--------------|
|                      | C880mul     | term1mul     | c3540mul     | vdamul       | am_8_8       |
| <i>adaptnovelty+</i> | <b>9372</b> | 22226        | 33153        | 34472        | <b>25521</b> |
| <i>gsat</i>          | 8344        | 19681        | 29310        | 30523        | 21600        |
| <i>gsat-tabu</i>     | 8354        | 19691        | 29345        | 30527        | 21611        |
| <i>gwsat</i>         | 8351        | 19710        | 29334        | 30564        | 21608        |
| <i>hsat</i>          | 8347        | 19691        | 29341        | 30545        | 21612        |
| <i>hwsat</i>         | 8359        | 19709        | 29332        | 30531        | 21610        |
| <i>novelty</i>       | <b>9372</b> | <b>22228</b> | <b>33176</b> | 34486        | 25332        |
| <i>saps</i>          | 8340        | 19674        | 29373        | 30535        | 21590        |
| <i>sapsnr</i>        | 8352        | 19718        | 29326        | 30494        | 21600        |
| <i>walksat-tabu</i>  | <b>9372</b> | 22211        | 33164        | <b>34497</b> | 25468        |

Tabela 4.5: Tempo (em minutos) das  $Hp$ 's para conjunto de calibração.

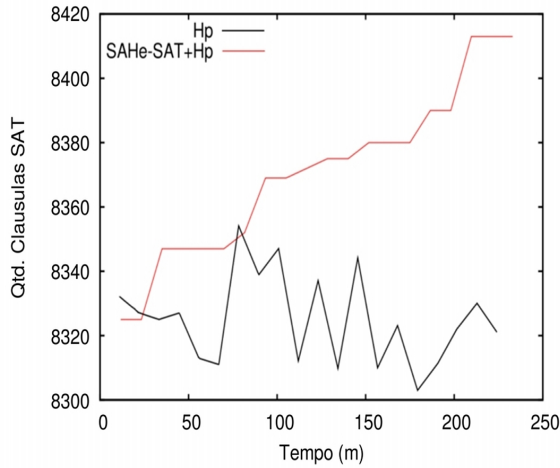
| Heurísticas          | Instâncias   |              |              |               |               |
|----------------------|--------------|--------------|--------------|---------------|---------------|
|                      | C880mul      | term1mul     | c3540mul     | vdamul        | am_8_8        |
| <i>adaptnovelty+</i> | 2.010        | <b>8.810</b> | <b>7.062</b> | 217.900       | 86.960        |
| <i>gsat</i>          | 209.163      | 128.680      | 183.106      | 196.210       | 192.810       |
| <i>gsat-tabu</i>     | 78.410       | 220.566      | 189.270      | 153.605       | 123.808       |
| <i>gwsat</i>         | 26.538       | 84.724       | 184.695      | 183.780       | 211.850       |
| <i>hsat</i>          | 228.076      | 109.495      | 186.119      | 164.130       | 169.267       |
| <i>hwsat</i>         | 192.790      | 99.285       | 166.380      | 188.770       | 229.023       |
| <i>novelty</i>       | <b>1.996</b> | 23.810       | 233.815      | <b>68.361</b> | 139.240       |
| <i>saps</i>          | 205.240      | 83.531       | 227.279      | 162.850       | 179.780       |
| <i>sapsnr</i>        | 228.327      | 183.790      | 148.720      | 189.960       | 133.060       |
| <i>walksat-tabu</i>  | 6.340        | 177.760      | 82.870       | 100.090       | <b>57.330</b> |

## Evolução dos Resultados através do uso do Algoritmo Genético

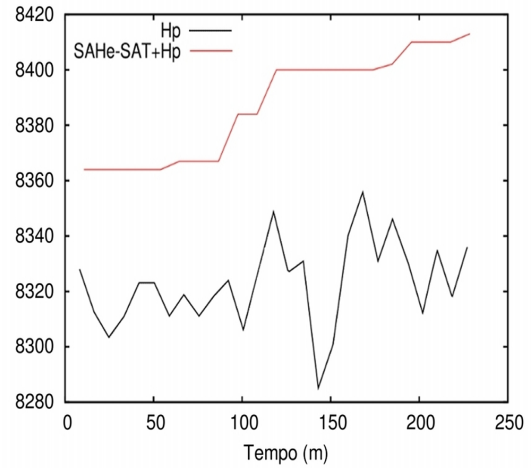
Além do desempenho individual das  $Hp$ 's, ao comparar as Tabelas 4.2 e 4.4, é possível verificar que o processo de manipulação das instâncias pelo *SAHe-SAT* gerou melhores resultados. Isto acontece até mesmo para heurísticas fracas.

As Figuras 4.4 e 4.5 mostram o conjunto de gráficos de evolução dos resultados das técnicas testadas. Em cada gráfico, estão representadas duas curvas, que indicam o resultado da aplicação de uma  $Hp$ -Pura (curva pontilhada) e do *SAHe-SAT*, em conjunto com esta mesma  $Hp$  (curva contínua), sobre uma instância do problema.

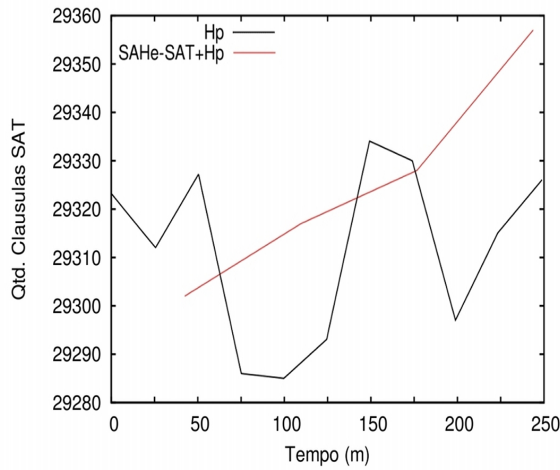
É possível notar que o uso do processo evolucionário conduziu a melhores resultados, inclusive para as heurísticas mais “fracas”. Sobre estas heurísticas, vale destacar, que a melhora apresentada no uso do método foi bem maior que nos outros casos.



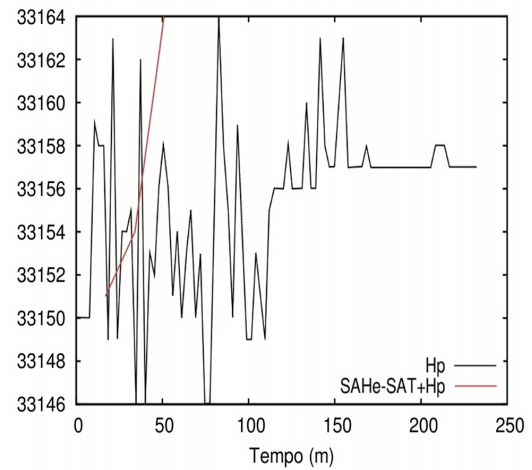
(a) Instância C880mul / Hp *gsat-tabu*



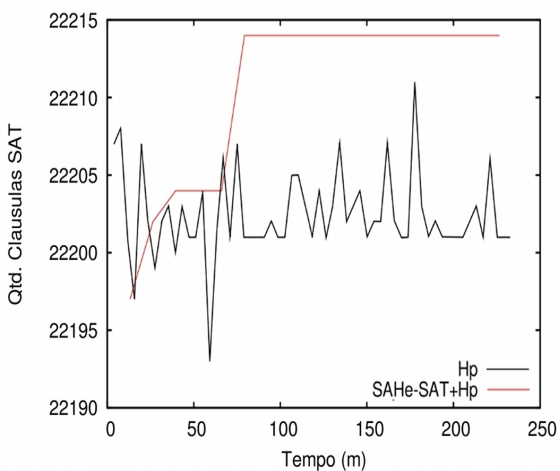
(b) Instância C880mul / Hp *hsat*



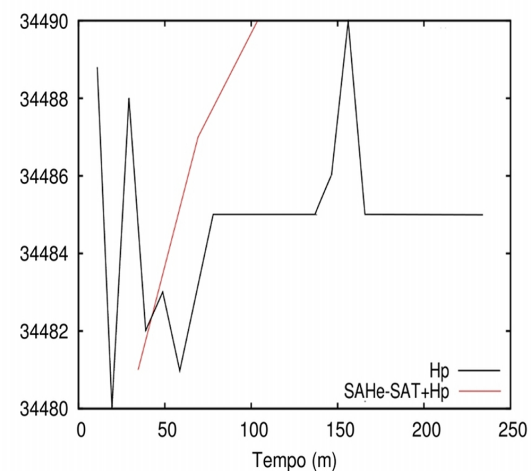
(c) Instância c3540mul / Hp *gwsat*



(d) Instância c3540mul / Hp *walksat-tabu*

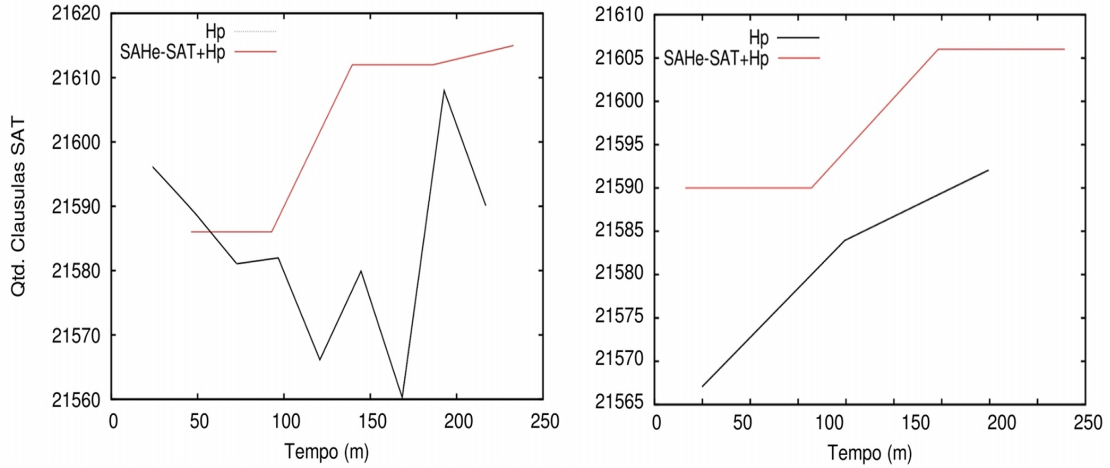


(e) Instância term1mul / Hp *walksat-tabu*



(f) Instância vdamul / Hp *novelty*

Figura 4.4: Evolução dos resultados para uma execução do *SAHe-SAT+Hp* (curva em vermelho) e da *Hp-Pura*, com reinícios aleatórios (curva em preto).



(g) Instância am-8-8 / Hp *gsat*

(h) Instância am-8-8 / Hp *saps*

Figura 4.5: Evolução dos resultados para uma execução do *SAHe-SAT+Hp* (curva em vermelho) e da *Hp-Pura*, com reinícios aleatórios (curva em preto). (continuação)

Como destacado anteriormente, a resolução de instâncias difíceis, sobre as quais o método exato não consegue ter êxito, é um objetivo desejável. Obter bons resultados com o uso de heurísticas que já são boas, quando aplicadas isoladamente sobre instâncias difíceis, torna-se um objetivo ainda mais interessante. O gráfico (e) da Figura 4.4 mostra exatamente este acontecimento. Além disso, os gráficos (d), (e) e (f) demonstram que é possível obter desempenho superior com o uso do *SAHe-SAT* em conjunto com as melhores heurísticas, indicando um futuro promissor e, para sua comprovação, serão realizados novos experimentos sobre as três melhores heurísticas (*adapt novelty+*, *novelty* e *walksat-tabu*).

### Conjunto de Parâmetros do AG

Após a finalização dos testes preliminares de calibração do *SAHe-SAT*, definiu-se o formato e os parâmetros do AG com os melhores resultados que serão utilizados no conjunto total de instâncias. São eles:

- o AG será executado sobre cada instância, até que o valor ótimo seja encontrado ou o tempo máximo (quatro horas) seja alcançado;
- cada geração contém  $K = 50$  indivíduos;
- cada indivíduo contém até  $1.5n$  genes (para  $n = |X|$ );

- a taxa mínima de variáveis distintas em cada indivíduo é 40% do número total de variáveis da instância;
- o elitismo da população é 20%;
- a decisão entre *crossover* e mutação possui taxa de 25% para a escolha do primeiro;
- a mutação ocorre em até 50% dos genes;
- escolha de  $\Phi = 20$  como probabilidade de seleção (Equação 4.5).

A seguir, serão descritos os resultados dos experimentos computacionais finais.

#### 4.2.4 Resultados Experimentais

Como apontado anteriormente, a fase de calibração produziu três heurísticas de pós-processamento *Hp* “campeãs”: *adaptnovelty+*, *novelty* e *walksat-tabu*. Os resultados apresentados nesta subseção referem-se exclusivamente a elas, utilizadas em conjunto com o *SAHe-SAT*, como rotinas de pós-processamento, ou sozinhas, como *heurísticas puras*. Neste último caso, cada heurística é executada repetidamente, cada vez com uma nova associação aleatória de valores verdade para as variáveis, até satisfazer o critério de parada: valor ótimo encontrado ou tempo máximo atingido.

O resultado reportado pelo *SAHe-SAT* refere-se ao melhor indivíduo da última população que foi preenchida. Ao encontrar o valor ótimo, ou atingir um tempo máximo previamente estipulado, o algoritmo recebe um sinal, sendo encerrado no momento que a próxima população de indivíduos for preenchida. Este critério de parada do AG foi definido por limitações de interferência no ambiente de teste utilizado (pacote *UBCSAT* [19]) e, principalmente, pela complexidade de algumas instâncias, para as quais o AG não havia completado nenhuma população de indivíduos ao atingir o tempo máximo definido. Desta forma, em alguns casos, os valores demonstrados nesta seção contêm um tempo superior ao máximo definido.

Com o objetivo de comparar o desempenho do *SAHe-SAT*, tanto quanto possível, o resultado reportado para a *heurística pura* corresponde ao melhor resultado obtido a cada 50 (cinquenta) repetições. Vale ressaltar que, assim como aconteceu com o

método proposto, em alguns casos de aplicação da *heurística pura*, não foi possível fechar nenhum ciclo de repetições no tempo máximo definido, adotando-se o critério de fechamento de ciclo para finalização do experimento.

Todos os experimentos foram realizados no ambiente *UBCSAT* [19]. O ótimo, quando conhecido, foi encontrado separadamente, via execução do software exato *MSUnCore* para Max-SAT [28], de atualização 2010.

Nos experimentos foram utilizadas todas as 100 instâncias do conjunto de 2004 [20] e todas as 112 de 2008 [21]. O tempo limite alocado para cada instância do *SAHe-SAT* e para cada uma das três *heurísticas puras* foi de 240 minutos (4 horas), sempre com configuração de hardware e software idênticas e com acesso exclusivo ao sistema.

Como apontado anteriormente, o contexto mais interessante para análise do *SAHe-SAT* é o de instâncias difíceis dos dois conjuntos, que corresponde aos casos em que o software exato *MSUnCore* não encontrou uma resposta para a instância como resultado de sua computação, ocasionando o travamento da máquina e posterior cancelamento de sua execução por falta de memória. São 52 instâncias do conjunto de 2004 e 11 no de 2008, totalizando 62 instâncias.

Os resultados, nesse contexto, são apresentados nas Tabelas 4.6 a 4.8, respectivamente, para  $Hp = \textit{adaptnovelty+}$ ,  $Hp = \textit{novelty}$  e  $Hp = \textit{walksat-tabu}$ . Cada tabela contém uma linha para cada uma das 62 instâncias com uma linha horizontal, separando as instâncias do conjunto de 2004 das do conjunto de 2008. Para cada instância, estão descritos o número  $n$  de variáveis, bem como o número  $m$  de cláusulas e os resultados para as estratégias *SAHe-SAT+Hp* e *Hp-pura* em questão. Estes resultados são relativos ao número de cláusulas satisfeitas e ao tempo em que esta solução foi encontrada, pela primeira vez, durante o tempo limite de 240 minutos.

Algumas entradas nas tabelas estão destacadas (em negrito) para indicar que o *SAHe-SAT+Hp* encontrou uma solução estritamente melhor que a encontrada pela *heurística pura* ou, ainda, uma solução satisfazendo o mesmo número de cláusulas, mas que fora encontrada em um tempo menor pelo *SAHe-SAT+Hp*. No primeiro caso, destacou-se apenas o número de cláusulas satisfeitas; no segundo, o tempo também foi destacado.

Tabela 4.6: Resultados do *SAHe-SAT* e  $Hp = \text{Adapt novelty+}$ .

| Instância                    | $n$     | $m$     | <i>SAHe-SAT+Adapt novelty+</i> |              | <i>Adapt novelty+</i> |          |
|------------------------------|---------|---------|--------------------------------|--------------|-----------------------|----------|
|                              |         |         | Num. cls. sat.                 | Tempo(m)     | Num. cls. sat.        | Tempo(m) |
| c3540mul                     | 5248    | 33199   | <b>33158</b>                   | 230.755      | 33156                 | 80.479   |
| c6288mul                     | 9540    | 61421   | <b>61385</b>                   | 1.100        | 61382                 | 36.361   |
| dalumul                      | 9426    | 59991   | <b>59924</b>                   | 50.854       | 59920                 | 44.117   |
| frg1mul                      | 3230    | 20575   | <b>20574</b>                   | <b>0.495</b> | 20574                 | 1.290    |
| k2mul                        | 11680   | 74581   | <b>74418</b>                   | 112.313      | 74407                 | 16.932   |
| x1mul                        | 8760    | 55571   | <b>55570</b>                   | <b>1.145</b> | 55570                 | 2.477    |
| am_6_6                       | 2269    | 7814    | <b>7813</b>                    | <b>0.103</b> | 7813                  | 0.267    |
| am_7_7                       | 4264    | 14751   | <b>14750</b>                   | <b>0.363</b> | 14750                 | 0.558    |
| am_8_8                       | 7361    | 25538   | <b>25525</b>                   | 237.715      | 25523                 | 180.424  |
| am_9_9                       | 11908   | 41393   | <b>41298</b>                   | 81.680       | 41293                 | 53.522   |
| li-exam-61                   | 28147   | 108436  | 108027                         | 36.642       | 108045                | 85.697   |
| li-exam-62                   | 28147   | 108436  | 108031                         | 45.118       | 108032                | 17.074   |
| li-exam-63                   | 28147   | 108436  | <b>108033</b>                  | 17.467       | 108026                | 49.837   |
| li-exam-64                   | 28147   | 108436  | <b>108041</b>                  | 80.617       | 108037                | 49.076   |
| li-test4-100                 | 36809   | 142491  | 141896                         | 60.697       | 141902                | 35.010   |
| li-test4-101                 | 36809   | 142491  | 141894                         | 44.930       | 141904                | 105.123  |
| li-test4-94                  | 36809   | 142491  | 141894                         | 36.740       | 141908                | 37.992   |
| li-test4-95                  | 36809   | 142491  | <b>141923</b>                  | 91.495       | 141904                | 95.294   |
| li-test4-96                  | 36809   | 142491  | 141902                         | 100.277      | 141908                | 64.973   |
| li-test4-97                  | 36809   | 142491  | 141908                         | 62.687       | 141920                | 33.317   |
| li-test4-98                  | 36809   | 142491  | 141892                         | 79.786       | 141907                | 52.417   |
| li-test4-99                  | 36809   | 142491  | <b>141911</b>                  | 41.042       | 141906                | 12.445   |
| gripper10u                   | 2312    | 18666   | <b>18665</b>                   | <b>1.468</b> | 18665                 | 5.913    |
| gripper11u                   | 3084    | 26019   | 26018                          | 27.940       | 26018                 | 15.914   |
| gripper12u                   | 3352    | 29412   | 29411                          | 71.192       | 29411                 | 56.930   |
| gripper13u                   | 4268    | 38965   | 38963                          | 1.809        | 38963                 | 0.454    |
| gripper14u                   | 4584    | 43390   | <b>43389</b>                   | 59.523       | 43388                 | 0.456    |
| bc56-sensors-1-k391          | 561371  | 1778987 | 1623366                        | 85.786       | 1623430               | 36.854   |
| bc56-sensors-2-k592          | 850398  | 2694319 | <b>2394072</b>                 | 122.112      | 2393894               | 6.102    |
| bc57-sensors-1-k303          | 435701  | 1379987 | <b>1282461</b>                 | 100.646      | 1282338               | 34.813   |
| dme-03-1-k247                | 261352  | 773077  | <b>747079</b>                  | 102.536      | 746975                | 75.608   |
| motors-stuck-1-k407          | 654766  | 2068742 | <b>1867495</b>                 | 70.249       | 1867266               | 56.079   |
| motors-stuck-2-k314          | 505536  | 1596837 | <b>1467190</b>                 | 94.221       | 1467153               | 104.632  |
| valves-gates-1-k617          | 985042  | 3113540 | <b>2742563</b>                 | 71.558       | 2742310               | 18.078   |
| 6pipe                        | 15800   | 394739  | 393814                         | 82.111       | 393829                | 81.745   |
| 7pipe                        | 23910   | 751118  | 749667                         | 125.703      | 749670                | 145.208  |
| comb1                        | 5910    | 16804   | <b>16761</b>                   | 51.819       | 16759                 | 31.832   |
| dp12u11                      | 11137   | 30792   | <b>30723</b>                   | 95.372       | 30722                 | 12.257   |
| f2clk_50                     | 34678   | 101319  | 100430                         | 42.333       | 100435                | 3.063    |
| fifo8_300                    | 194762  | 530713  | <b>516445</b>                  | 104.923      | 516380                | 35.441   |
| homer17                      | 286     | 1742    | <b>1738</b>                    | <b>0.134</b> | 1738                  | 0.330    |
| homer18                      | 308     | 2030    | <b>2024</b>                    | <b>0.138</b> | 2024                  | 0.345    |
| homer19                      | 330     | 2340    | <b>2332</b>                    | <b>0.146</b> | 2332                  | 0.367    |
| homer20                      | 440     | 4220    | <b>4202</b>                    | <b>0.175</b> | 4202                  | 0.440    |
| k2fix_gr_2pinvar_w8          | 3771    | 270136  | <b>269951</b>                  | 118.768      | 269930                | 39.335   |
| k2fix_gr_2pinvar_w9          | 5028    | 307674  | <b>307562</b>                  | 101.736      | 307559                | 28.284   |
| k2fix_gr_2pin_w8             | 9882    | 295998  | 295637                         | 21.429       | 295645                | 11.987   |
| k2fix_gr_2pin_w9             | 13176   | 345426  | 345118                         | 11.300       | 345150                | 51.661   |
| k2fix_gr_rcs_w8              | 10056   | 271393  | 271287                         | 87.388       | 271290                | 39.640   |
| sha1                         | 61377   | 255417  | <b>252631</b>                  | 99.706       | 252617                | 120.679  |
| sha2                         | 61377   | 255417  | <b>252633</b>                  | 7.391        | 252630                | 88.739   |
| rsdecoder1_blackbox_KESblock | 707330  | 1106376 | <b>1042885</b>                 | 167.766      | 1042849               | 35.251   |
| rsdecoder4                   | 237783  | 933978  | <b>905052</b>                  | 133.385      | 905038                | 152.771  |
| rsdecoder-problem.dimacs_38  | 1198012 | 3865513 | <b>3374563</b>                 | 220.802      | 3374454               | 99.393   |
| rsdecoder-problem.dimacs_41  | 1186710 | 3829036 | <b>3343967</b>                 | 55.766       | 3343848               | 229.876  |
| SM_MAIN_MEM_buggy1           | 870975  | 3812147 | <b>3431928</b>                 | 15.243       | 3431544               | 100.193  |
| wb_4m8s1                     | 463080  | 1759150 | 1637425                        | 6.646        | 1637511               | 209.569  |
| wb_4m8s4                     | 463080  | 1759150 | <b>1636787</b>                 | 15.974       | 1636765               | 189.417  |
| wb_4m8s-problem.dimacs_47    | 2691648 | 8517027 | 7189162                        | 95.320       | 7189244               | 19.575   |
| wb_4m8s-problem.dimacs_49    | 2785108 | 8812799 | <b>7432769</b>                 | 20.241       | 7432645               | 216.399  |
| wb_conmax1                   | 277950  | 1221020 | 1175797                        | 96.978       | 1175846               | 104.276  |
| wb_conmax3                   | 277950  | 1221020 | 1175833                        | 12.656       | 1175942               | 14.228   |

Tabela 4.7: Resultados do *SAHe-SAT* e  $H_p = Novelty$ .

| Instância                    | $n$     | $m$     | <i>SAHe-SAT+Novelty</i> |               | <i>Novelty</i> |          |
|------------------------------|---------|---------|-------------------------|---------------|----------------|----------|
|                              |         |         | Num. cls. sat.          | Tempo(m)      | Num. cls. sat. | Tempo(m) |
| c3540mul                     | 5248    | 33199   | 33176                   | 17.965        | 33183          | 187.185  |
| c6288mul                     | 9540    | 61421   | 61379                   | 81.437        | 61384          | 103.835  |
| dalumul                      | 9426    | 59991   | 59975                   | 62.621        | 59976          | 77.171   |
| frg1mul                      | 3230    | 20575   | <b>20574</b>            | <b>0.543</b>  | 20574          | 1.390    |
| k2mul                        | 11680   | 74581   | <b>74524</b>            | <b>1.729</b>  | 74524          | 96.870   |
| x1mul                        | 8760    | 55571   | <b>55570</b>            | <b>1.550</b>  | 55570          | 5.383    |
| am_6_6                       | 2269    | 7814    | <b>7813</b>             | <b>0.263</b>  | 7813           | 0.267    |
| am_7_7                       | 4264    | 14751   | <b>14744</b>            | 55.268        | 14739          | 120.361  |
| am_8_8                       | 7361    | 25538   | <b>25337</b>            | 69.707        | 25334          | 203.760  |
| am_9_9                       | 11908   | 41393   | <b>40982</b>            | 1.827         | 40970          | 36.322   |
| li-exam-61                   | 28147   | 108436  | 108011                  | 47.997        | 108041         | 22.993   |
| li-exam-62                   | 28147   | 108436  | <b>108018</b>           | 112.380       | 108008         | 100.084  |
| li-exam-63                   | 28147   | 108436  | <b>108011</b>           | 101.528       | 108003         | 19.108   |
| li-exam-64                   | 28147   | 108436  | 107996                  | 70.593        | 108013         | 34.106   |
| li-test4-100                 | 36809   | 142491  | 141844                  | 34.564        | 141870         | 88.830   |
| li-test4-101                 | 36809   | 142491  | 141858                  | 40.733        | 141865         | 31.374   |
| li-test4-94                  | 36809   | 142491  | <b>141863</b>           | 2.556         | 141859         | 74.018   |
| li-test4-95                  | 36809   | 142491  | 141865                  | 63.408        | 141868         | 16.500   |
| li-test4-96                  | 36809   | 142491  | <b>141865</b>           | 101.795       | 141851         | 112.326  |
| li-test4-97                  | 36809   | 142491  | 141855                  | 29.113        | 141861         | 91.341   |
| li-test4-98                  | 36809   | 142491  | <b>141862</b>           | 40.632        | 141852         | 90.647   |
| li-test4-99                  | 36809   | 142491  | 141859                  | 12.537        | 141870         | 102.007  |
| gripper10u                   | 2312    | 18666   | 18663                   | 0.877         | 18664          | 76.493   |
| gripper11u                   | 3084    | 26019   | <b>26017</b>            | 15.145        | 26016          | 6.459    |
| gripper12u                   | 3352    | 29412   | <b>29409</b>            | <b>14.485</b> | 29409          | 29.184   |
| gripper13u                   | 4268    | 38965   | <b>38962</b>            | <b>87.752</b> | 38962          | 118.531  |
| gripper14u                   | 4584    | 43390   | 43386                   | 23.381        | 43387          | 97.279   |
| bc56-sensors-1-k391          | 561371  | 1778987 | <b>1600354</b>          | 88.186        | 1600149        | 86.918   |
| bc56-sensors-2-k592          | 850398  | 2694319 | 2366874                 | 48.252        | 2366923        | 13.745   |
| bc57-sensors-1-k303          | 435701  | 1379987 | <b>1262046</b>          | 79.846        | 1262043        | 14.386   |
| dme-03-1-k247                | 261352  | 773077  | <b>736229</b>           | 49.434        | 736147         | 106.616  |
| motors-stuck-1-k407          | 654766  | 2068742 | <b>1842645</b>          | 112.121       | 1842299        | 7.294    |
| motors-stuck-2-k314          | 505536  | 1596837 | 1445274                 | 53.357        | 1445323        | 57.764   |
| valves-gates-1-k617          | 985042  | 3113540 | 2714448                 | 58.357        | 2714681        | 30.268   |
| 6pipe                        | 15800   | 394739  | 394722                  | 112.168       | 394723         | 241.421  |
| 7pipe                        | 23910   | 751118  | 751077                  | 187.551       | 751088         | 1860.378 |
| comb1                        | 5910    | 16804   | <b>16756</b>            | 89.910        | 16755          | 101.368  |
| dp12u11                      | 11137   | 30792   | 30785                   | 11.239        | 30789          | 17.881   |
| f2clk_50                     | 34678   | 101319  | 100629                  | 18.233        | 100668         | 29.506   |
| fifo8_300                    | 194762  | 530713  | 506281                  | 72.155        | 506329         | 49.814   |
| homer17                      | 286     | 1742    | <b>1738</b>             | <b>0.126</b>  | 1738           | 0.317    |
| homer18                      | 308     | 2030    | <b>2024</b>             | <b>0.131</b>  | 2024           | 0.331    |
| homer19                      | 330     | 2340    | <b>2332</b>             | <b>0.142</b>  | 2332           | 0.349    |
| homer20                      | 440     | 4220    | <b>4202</b>             | <b>0.170</b>  | 4202           | 0.429    |
| k2fix_gr_2pinvar_w8          | 3771    | 270136  | <b>269918</b>           | 47.861        | 269916         | 116.270  |
| k2fix_gr_2pinvar_w9          | 5028    | 307674  | 307563                  | 43.461        | 307565         | 42.008   |
| k2fix_gr_2pin_w8             | 9882    | 295998  | 295684                  | 69.188        | 295691         | 5.697    |
| k2fix_gr_2pin_w9             | 13176   | 345426  | 345243                  | 87.169        | 345255         | 81.536   |
| k2fix_gr_rcs_w8              | 10056   | 271393  | <b>271296</b>           | 17.291        | 271292         | 58.947   |
| sha1                         | 61377   | 255417  | 251884                  | 126.938       | 251927         | 48.348   |
| sha2                         | 61377   | 255417  | <b>251938</b>           | 87.569        | 251918         | 72.757   |
| rsdecoder1_blackbox_KESblock | 707330  | 1106376 | 1030117                 | 203.708       | 1030149        | 97.698   |
| rsdecoder4                   | 237783  | 933978  | 896371                  | 140.864       | 896441         | 51.221   |
| rsdecoder-problem.dimacs_38  | 1198012 | 3865513 | 3350860                 | 203.150       | 3351073        | 46.902   |
| rsdecoder-problem.dimacs_41  | 1186710 | 3829036 | <b>3320522</b>          | 182.019       | 3320305        | 224.181  |
| SM_MAIN_MEM_buggy1           | 870975  | 3812147 | <b>3416883</b>          | 137.656       | 3416837        | 37.047   |
| wb_4m8s1                     | 463080  | 1759150 | <b>1624751</b>          | 17.008        | 1624585        | 207.639  |
| wb_4m8s4                     | 463080  | 1759150 | <b>1624017</b>          | 4.333         | 1623808        | 164.746  |
| wb_4m8s-problem.dimacs_47    | 2691648 | 8517027 | <b>7159756</b>          | 56.460        | 7159458        | 53.487   |
| wb_4m8s-problem.dimacs_49    | 2785108 | 8812799 | <b>7402599</b>          | 144.220       | 7402249        | 183.336  |
| wb_conmax1                   | 277950  | 1221020 | 1168273                 | 23.774        | 1168351        | 200.654  |
| wb_conmax3                   | 277950  | 1221020 | <b>1168628</b>          | 212.218       | 1168466        | 147.748  |

Tabela 4.8: Resultados do SAHe-SAT e  $H_p = Walksat-tabu$ .

| Instância                    | $n$     | $m$     | SAHe-SAT+Walksat-tabu |                | Walksat-tabu   |         |
|------------------------------|---------|---------|-----------------------|----------------|----------------|---------|
|                              |         |         | Num. cls. sat.        | Tempo          | Num. cls. sat. | Tempo   |
| c3540mul                     | 5248    | 33199   | 33170                 | 147.363        | 33170          | 129.171 |
| c6288mul                     | 9540    | 61421   | <b>61392</b>          | 3.884          | 61389          | 23.451  |
| dalumul                      | 9426    | 59991   | 59896                 | 2.730          | 59910          | 2.546   |
| frg1mul                      | 3230    | 20575   | <b>20574</b>          | 141.607        | 20573          | 106.643 |
| k2mul                        | 11680   | 74581   | <b>74351</b>          | <b>114.570</b> | 74351          | 119.464 |
| x1mul                        | 8760    | 55571   | 55563                 | 88.259         | 55564          | 67.999  |
| am_6_6                       | 2269    | 7814    | 7812                  | 233.708        | 7812           | 121.373 |
| am_7_7                       | 4264    | 14751   | <b>14734</b>          | 162.201        | 14731          | 130.805 |
| am_8_8                       | 7361    | 25538   | 25466                 | 19.596         | 25477          | 137.402 |
| am_9_9                       | 11908   | 41393   | <b>41197</b>          | 62.596         | 41190          | 89.407  |
| li-exam-61                   | 28147   | 108436  | <b>107996</b>         | 69.841         | 107982         | 31.107  |
| li-exam-62                   | 28147   | 108436  | 107986                | 74.585         | 107992         | 53.727  |
| li-exam-63                   | 28147   | 108436  | 107980                | 48.127         | 107983         | 100.824 |
| li-exam-64                   | 28147   | 108436  | 107985                | 3.946          | 107991         | 79.596  |
| li-test4-100                 | 36809   | 142491  | 141790                | 39.717         | 141793         | 66.380  |
| li-test4-101                 | 36809   | 142491  | <b>141804</b>         | 19.039         | 141800         | 11.141  |
| li-test4-94                  | 36809   | 142491  | <b>141806</b>         | 2.853          | 141780         | 44.018  |
| li-test4-95                  | 36809   | 142491  | <b>141796</b>         | 67.216         | 141790         | 72.285  |
| li-test4-96                  | 36809   | 142491  | <b>141804</b>         | 24.995         | 141791         | 33.641  |
| li-test4-97                  | 36809   | 142491  | 141791                | 26.641         | 141805         | 22.784  |
| li-test4-98                  | 36809   | 142491  | 141787                | 92.233         | 141791         | 54.528  |
| li-test4-99                  | 36809   | 142491  | 141780                | 96.418         | 141795         | 111.411 |
| gripper10u                   | 2312    | 18666   | <b>18662</b>          | <b>42.287</b>  | 18662          | 58.256  |
| gripper11u                   | 3084    | 26019   | <b>26014</b>          | <b>2.383</b>   | 26014          | 10.450  |
| gripper12u                   | 3352    | 29412   | <b>29406</b>          | <b>2.104</b>   | 29406          | 2.788   |
| gripper13u                   | 4268    | 38965   | <b>38960</b>          | 88.907         | 38959          | 50.868  |
| gripper14u                   | 4584    | 43390   | <b>43383</b>          | <b>2.521</b>   | 43383          | 93.167  |
| bc56-sensors-1-k391          | 561371  | 1778987 | <b>1600245</b>        | 36.329         | 1599963        | 43.683  |
| bc56-sensors-2-k592          | 850398  | 2694319 | <b>2361372</b>        | 33.625         | 2361225        | 39.390  |
| bc57-sensors-1-k303          | 435701  | 1379987 | <b>1264444</b>        | 33.863         | 1264065        | 73.125  |
| dme-03-1-k247                | 261352  | 773077  | <b>740068</b>         | 15.573         | 739924         | 53.964  |
| motors-stuck-1-k407          | 654766  | 2068742 | <b>1840520</b>        | 79.129         | 1840216        | 68.053  |
| motors-stuck-2-k314          | 505536  | 1596837 | <b>1446222</b>        | 85.452         | 1445841        | 80.114  |
| valves-gates-1-k617          | 985042  | 3113540 | <b>2706084</b>        | 77.354         | 2705761        | 53.865  |
| 6pipe                        | 15800   | 394739  | 394727                | 48.824         | 394730         | 156.274 |
| 7pipe                        | 23910   | 751118  | <b>751108</b>         | 135.140        | 751099         | 878.490 |
| comb1                        | 5910    | 16804   | 16716                 | 116.380        | 16717          | 9.986   |
| dp12u11                      | 11137   | 30792   | <b>30775</b>          | 22.338         | 30773          | 27.777  |
| f2clk_50                     | 34678   | 101319  | <b>100087</b>         | 49.688         | 100077         | 96.540  |
| fifo8_300                    | 194762  | 530713  | 509196                | 2.029          | 509252         | 56.291  |
| homer17                      | 286     | 1742    | <b>1738</b>           | <b>0.105</b>   | 1738           | 0.268   |
| homer18                      | 308     | 2030    | <b>2024</b>           | <b>0.112</b>   | 2024           | 0.285   |
| homer19                      | 330     | 2340    | <b>2332</b>           | <b>0.119</b>   | 2332           | 0.309   |
| homer20                      | 440     | 4220    | <b>4202</b>           | <b>0.152</b>   | 4202           | 0.374   |
| k2fix_gr_2pinvar_w8          | 3771    | 270136  | <b>269860</b>         | 73.909         | 269859         | 83.152  |
| k2fix_gr_2pinvar_w9          | 5028    | 307674  | <b>307490</b>         | 31.671         | 307488         | 117.265 |
| k2fix_gr_2pin_w8             | 9882    | 295998  | 295544                | 12.155         | 295554         | 43.311  |
| k2fix_gr_2pin_w9             | 13176   | 345426  | <b>345065</b>         | 3.299          | 345044         | 47.063  |
| k2fix_gr_rcs_w8              | 10056   | 271393  | 271302                | 81.870         | 271302         | 64.336  |
| sha1                         | 61377   | 255417  | 251374                | 30.878         | 251376         | 90.648  |
| sha2                         | 61377   | 255417  | 251390                | 7.669          | 251392         | 16.520  |
| rsdecoder1_blackbox_KESblock | 707330  | 1106376 | 1028280               | 55.013         | 1028328        | 33.828  |
| rsdecoder4                   | 237783  | 933978  | <b>896382</b>         | 100.396        | 896184         | 137.977 |
| rsdecoder-problem.dimacs_38  | 1198012 | 3865513 | 3334250               | 34.227         | 3334644        | 145.764 |
| rsdecoder-problem.dimacs_41  | 1186710 | 3829036 | 3304320               | 65.006         | 3304476        | 4.832   |
| SM_MAIN_MEM_buggy1           | 870975  | 3812147 | 3386126               | 37.134         | 3386263        | 58.493  |
| wb_4m8s1                     | 463080  | 1759150 | <b>1612444</b>        | 64.671         | 1611909        | 95.642  |
| wb_4m8s4                     | 463080  | 1759150 | 1611219               | 113.248        | 1611361        | 56.378  |
| wb_4m8s-problem.dimacs_47    | 2691648 | 8517027 | 7133234               | 33.361         | 7133580        | 44.962  |
| wb_4m8s-problem.dimacs_49    | 2785108 | 8812799 | 7374846               | 3.899          | 7374923        | 181.258 |
| wb_conmax1                   | 277950  | 1221020 | <b>1157042</b>        | 60.081         | 1156892        | 201.127 |
| wb_conmax3                   | 277950  | 1221020 | <b>1157179</b>        | 64.238         | 1157029        | 186.785 |



O número de instâncias destacadas gera a razão apresentada na linha “Final” da Tabela 4.9. As demais linhas apresentam a mesma informação para diversas fatias de tempo. Cada linha contém o tempo limite em minutos e a razão de ganho do *SAHe-SAT+Hp*, que estão agrupadas por conjuntos de instâncias: conjunto de 2004, de 2008 e os dois conjuntos combinados. Cada coluna contém a razão de ganho do *SAHe-SAT+Hp*, combinada com alguma *Hp* ( $Hp_1 = adaptnovelty+$ ,  $Hp_2 = novelty$  e  $Hp_3 = walksat-tabu$ ) e aplicada sobre um conjunto de instâncias.

Tabela 4.9: Taxa de sucesso do *SAHe-SAT*, por tempo limite, sobre as instâncias do conjunto de 2004 e de 2008, cujo valor ótimo é desconhecido.

| Tempo | Conjunto de 2004 |              |              | Conjunto de 2008 |              |        | Conjuntos de 2004 e 2008 combinados |              |              |
|-------|------------------|--------------|--------------|------------------|--------------|--------|-------------------------------------|--------------|--------------|
|       | $Hp_1$           | $Hp_2$       | $Hp_3$       | $Hp_1$           | $Hp_2$       | $Hp_3$ | $Hp_1$                              | $Hp_2$       | $Hp_3$       |
| 10    | 0.431            | <b>0.510</b> | <b>0.529</b> | <b>0.636</b>     | <b>0.545</b> | 0.455  | 0.468                               | <b>0.516</b> | <b>0.516</b> |
| 20    | 0.490            | <b>0.529</b> | <b>0.569</b> | <b>0.545</b>     | <b>0.727</b> | 0.273  | 0.500                               | <b>0.565</b> | <b>0.516</b> |
| 30    | <b>0.510</b>     | <b>0.529</b> | <b>0.569</b> | 0.455            | <b>0.727</b> | 0.273  | 0.500                               | <b>0.565</b> | <b>0.516</b> |
| 40    | 0.471            | <b>0.569</b> | <b>0.667</b> | 0.455            | <b>0.636</b> | 0.273  | 0.468                               | <b>0.581</b> | <b>0.597</b> |
| 50    | <b>0.510</b>     | <b>0.549</b> | <b>0.725</b> | 0.455            | <b>0.636</b> | 0.091  | 0.500                               | <b>0.565</b> | <b>0.613</b> |
| 60    | <b>0.588</b>     | <b>0.540</b> | <b>0.667</b> | <b>0.545</b>     | <b>0.545</b> | 0.000  | <b>0.581</b>                        | <b>0.541</b> | <b>0.548</b> |
| 70    | <b>0.510</b>     | <b>0.549</b> | <b>0.647</b> | <b>0.545</b>     | 0.455        | 0.364  | <b>0.516</b>                        | <b>0.532</b> | <b>0.597</b> |
| 80    | <b>0.549</b>     | <b>0.510</b> | <b>0.647</b> | <b>0.545</b>     | 0.455        | 0.364  | <b>0.548</b>                        | 0.500        | <b>0.597</b> |
| 90    | <b>0.549</b>     | <b>0.588</b> | <b>0.647</b> | <b>0.545</b>     | 0.455        | 0.364  | <b>0.548</b>                        | <b>0.565</b> | <b>0.597</b> |
| 100   | <b>0.588</b>     | <b>0.549</b> | <b>0.627</b> | 0.455            | 0.364        | 0.364  | <b>0.565</b>                        | <b>0.516</b> | <b>0.581</b> |
| 110   | <b>0.627</b>     | <b>0.529</b> | <b>0.588</b> | 0.455            | 0.455        | 0.364  | <b>0.597</b>                        | <b>0.516</b> | <b>0.548</b> |
| 120   | <b>0.627</b>     | <b>0.549</b> | <b>0.608</b> | 0.455            | <b>0.545</b> | 0.364  | <b>0.597</b>                        | <b>0.548</b> | <b>0.565</b> |
| 130   | <b>0.647</b>     | <b>0.549</b> | <b>0.608</b> | <b>0.545</b>     | <b>0.545</b> | 0.364  | <b>0.629</b>                        | <b>0.548</b> | <b>0.565</b> |
| 140   | <b>0.647</b>     | <b>0.549</b> | <b>0.608</b> | <b>0.636</b>     | <b>0.545</b> | 0.364  | <b>0.645</b>                        | <b>0.548</b> | <b>0.565</b> |
| 150   | <b>0.627</b>     | <b>0.549</b> | <b>0.627</b> | <b>0.636</b>     | <b>0.545</b> | 0.364  | <b>0.629</b>                        | <b>0.548</b> | <b>0.581</b> |
| 160   | <b>0.627</b>     | <b>0.549</b> | <b>0.627</b> | <b>0.545</b>     | <b>0.545</b> | 0.364  | <b>0.613</b>                        | <b>0.548</b> | <b>0.581</b> |
| 170   | <b>0.627</b>     | <b>0.549</b> | <b>0.627</b> | <b>0.545</b>     | <b>0.545</b> | 0.364  | <b>0.613</b>                        | <b>0.548</b> | <b>0.581</b> |
| 180   | <b>0.627</b>     | <b>0.549</b> | <b>0.627</b> | <b>0.545</b>     | <b>0.545</b> | 0.364  | <b>0.613</b>                        | <b>0.548</b> | <b>0.581</b> |
| 190   | <b>0.627</b>     | <b>0.569</b> | <b>0.627</b> | <b>0.545</b>     | <b>0.545</b> | 0.364  | <b>0.613</b>                        | <b>0.565</b> | <b>0.581</b> |
| 200   | <b>0.627</b>     | <b>0.569</b> | <b>0.627</b> | <b>0.545</b>     | <b>0.545</b> | 0.364  | <b>0.613</b>                        | <b>0.565</b> | <b>0.581</b> |
| 210   | <b>0.627</b>     | <b>0.569</b> | <b>0.627</b> | <b>0.545</b>     | <b>0.545</b> | 0.364  | <b>0.613</b>                        | <b>0.565</b> | <b>0.581</b> |
| 220   | <b>0.627</b>     | <b>0.569</b> | <b>0.627</b> | <b>0.545</b>     | <b>0.636</b> | 0.364  | <b>0.613</b>                        | <b>0.581</b> | <b>0.581</b> |
| 230   | <b>0.627</b>     | <b>0.569</b> | <b>0.627</b> | <b>0.636</b>     | <b>0.636</b> | 0.364  | <b>0.629</b>                        | <b>0.581</b> | <b>0.581</b> |
| 240   | <b>0.647</b>     | <b>0.569</b> | <b>0.627</b> | <b>0.636</b>     | <b>0.636</b> | 0.364  | <b>0.645</b>                        | <b>0.581</b> | <b>0.581</b> |
| Final | <b>0.647</b>     | <b>0.529</b> | <b>0.627</b> | <b>0.636</b>     | <b>0.636</b> | 0.364  | <b>0.645</b>                        | <b>0.548</b> | <b>0.581</b> |

Algumas entradas nas tabelas estão destacadas (em negrito) para indicar que o *SAHe-SAT+Hp* obteve um desempenho superior à *heurística pura* em mais da metade dos casos. Claramente, com a notável excessão da  $Hp = walksat-tabu$  sobre o conjunto de 2008 (no qual o uso da *Hp-pura* obteve um desempenho superior ao *SAHe-SAT+Hp* em todos os instantes de tempo), o *SAHe-SAT+Hp* obteve um bom desempenho sobre uma fração significativa das instâncias. Uma comparação mais detalhada dos valores presentes na linha “Final” da Tabela 4.9 é apresentada através dos gráficos da Figura 4.6, onde é possível notar que menos de 6% das instâncias o resultado encontrado pela aplicação da *Hp-pura* foi superior ao resultado da sua aplicação em conjunto com o método *SAHe-SAT*.

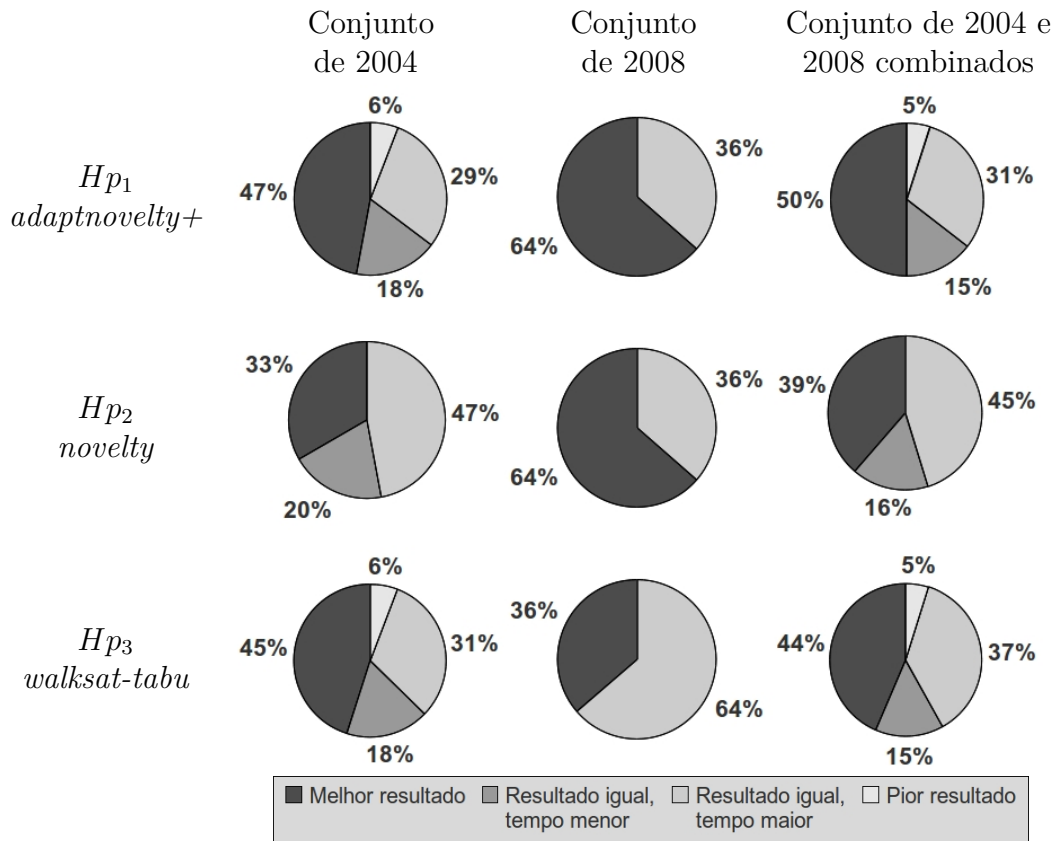


Figura 4.6: Comparação dos valores obtidos entre *SAHe-SAT+Hp*'s e *Hp*'s-puras, sobre as instâncias do conjunto de 2004 e de 2008, cujo valor ótimo é desconhecido.

Ao efetuar uma análise semelhante sobre todas as instâncias de ambos os conjuntos (incluindo resultados omitidos nas Tabelas 4.6 até 4.8), tem-se as razões presentes na Tabela 4.10 e os gráficos da Figura 4.7, demonstrando que o *SAHe-SAT+Hp* obteve um desempenho ainda melhor sobre todas as 100 instâncias do conjunto de 2004. Os resultados sobre o conjunto total de instâncias, que deram origem as razões da Tabela 4.10, estão descritos nas Tabelas B.1 a B.12, do Anexo B.

A Tabela 4.10 e os gráficos da Figura 4.7 também demonstram um suave decréscimo na razão do *SAHe-SAT+Hp*, para  $H_p = novelty$ , e um decréscimo superior na razão do *SAHe-SAT+Hp*, para  $H_p = adaptnovelty+$ , sobre as 112 instâncias do conjunto de 2008. Entretanto, para  $H_p = walksat-tabu$ , é possível notar um ótimo incremento no desempenho do *SAHe-SAT+Hp*. Para esta  $H_p$ , em particular, sobre o conjunto completo de 2008, o *SAHe-SAT+Hp* foi melhor que a *Hp-pura*. Isto indica que o *SAHe-SAT+Hp* obteve desempenho superior para uma fração significativa das instâncias mais fáceis.

Tabela 4.10: Taxa de sucesso do *SAHe-SAT*, por tempo limite, sobre todas as 100 instâncias do conjunto de 2004 e todas as 112 de 2008.

| Tempo | Conjunto de 2004       |                        |                        | Conjunto de 2008       |                        |                        | Conjuntos de 2004 e 2008 combinados |                        |                        |
|-------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|-------------------------------------|------------------------|------------------------|
|       | <i>Hp</i> <sub>1</sub> | <i>Hp</i> <sub>2</sub> | <i>Hp</i> <sub>3</sub> | <i>Hp</i> <sub>1</sub> | <i>Hp</i> <sub>2</sub> | <i>Hp</i> <sub>3</sub> | <i>Hp</i> <sub>1</sub>              | <i>Hp</i> <sub>2</sub> | <i>Hp</i> <sub>3</sub> |
| 10    | 0.480                  | <b>0.560</b>           | <b>0.580</b>           | <b>0.554</b>           | <b>0.509</b>           | <b>0.562</b>           | <b>0.519</b>                        | <b>0.533</b>           | <b>0.571</b>           |
| 20    | <b>0.510</b>           | <b>0.570</b>           | <b>0.620</b>           | 0.464                  | <b>0.562</b>           | <b>0.545</b>           | 0.486                               | <b>0.566</b>           | <b>0.580</b>           |
| 30    | <b>0.560</b>           | <b>0.580</b>           | <b>0.640</b>           | 0.491                  | <b>0.580</b>           | <b>0.580</b>           | <b>0.524</b>                        | <b>0.580</b>           | <b>0.608</b>           |
| 40    | <b>0.550</b>           | <b>0.620</b>           | <b>0.690</b>           | <b>0.509</b>           | <b>0.571</b>           | <b>0.536</b>           | <b>0.528</b>                        | <b>0.594</b>           | <b>0.608</b>           |
| 50    | <b>0.570</b>           | <b>0.620</b>           | <b>0.710</b>           | <b>0.527</b>           | <b>0.527</b>           | 0.500                  | <b>0.547</b>                        | <b>0.571</b>           | <b>0.599</b>           |
| 60    | <b>0.630</b>           | <b>0.590</b>           | <b>0.700</b>           | <b>0.509</b>           | <b>0.518</b>           | <b>0.518</b>           | <b>0.566</b>                        | <b>0.552</b>           | <b>0.604</b>           |
| 70    | <b>0.590</b>           | <b>0.610</b>           | <b>0.690</b>           | 0.482                  | 0.491                  | <b>0.545</b>           | <b>0.533</b>                        | <b>0.547</b>           | <b>0.613</b>           |
| 80    | <b>0.600</b>           | <b>0.590</b>           | <b>0.690</b>           | <b>0.509</b>           | 0.482                  | <b>0.536</b>           | <b>0.552</b>                        | <b>0.533</b>           | <b>0.608</b>           |
| 90    | <b>0.590</b>           | <b>0.620</b>           | <b>0.700</b>           | <b>0.509</b>           | <b>0.536</b>           | <b>0.536</b>           | <b>0.547</b>                        | <b>0.575</b>           | <b>0.613</b>           |
| 100   | <b>0.610</b>           | <b>0.590</b>           | <b>0.670</b>           | <b>0.518</b>           | <b>0.509</b>           | <b>0.545</b>           | <b>0.561</b>                        | <b>0.547</b>           | <b>0.604</b>           |
| 110   | <b>0.640</b>           | <b>0.580</b>           | <b>0.650</b>           | 0.500                  | 0.500                  | <b>0.545</b>           | <b>0.566</b>                        | <b>0.538</b>           | <b>0.594</b>           |
| 120   | <b>0.630</b>           | <b>0.590</b>           | <b>0.670</b>           | 0.464                  | <b>0.518</b>           | <b>0.545</b>           | <b>0.542</b>                        | <b>0.552</b>           | <b>0.604</b>           |
| 130   | <b>0.640</b>           | <b>0.590</b>           | <b>0.670</b>           | 0.455                  | <b>0.545</b>           | <b>0.545</b>           | <b>0.542</b>                        | <b>0.566</b>           | <b>0.604</b>           |
| 140   | <b>0.640</b>           | <b>0.590</b>           | <b>0.670</b>           | 0.464                  | <b>0.545</b>           | <b>0.554</b>           | <b>0.547</b>                        | <b>0.566</b>           | <b>0.608</b>           |
| 150   | <b>0.630</b>           | <b>0.590</b>           | <b>0.680</b>           | 0.464                  | <b>0.545</b>           | <b>0.562</b>           | <b>0.542</b>                        | <b>0.566</b>           | <b>0.618</b>           |
| 160   | <b>0.630</b>           | <b>0.590</b>           | <b>0.680</b>           | 0.464                  | <b>0.536</b>           | <b>0.580</b>           | <b>0.542</b>                        | <b>0.561</b>           | <b>0.627</b>           |
| 170   | <b>0.630</b>           | <b>0.590</b>           | <b>0.680</b>           | 0.446                  | <b>0.536</b>           | <b>0.589</b>           | <b>0.533</b>                        | <b>0.561</b>           | <b>0.632</b>           |
| 180   | <b>0.630</b>           | <b>0.590</b>           | <b>0.680</b>           | 0.446                  | <b>0.536</b>           | <b>0.589</b>           | <b>0.533</b>                        | <b>0.561</b>           | <b>0.632</b>           |
| 190   | <b>0.630</b>           | <b>0.600</b>           | <b>0.680</b>           | 0.446                  | <b>0.562</b>           | <b>0.589</b>           | <b>0.533</b>                        | <b>0.580</b>           | <b>0.632</b>           |
| 200   | <b>0.630</b>           | <b>0.600</b>           | <b>0.680</b>           | 0.464                  | <b>0.589</b>           | <b>0.589</b>           | <b>0.542</b>                        | <b>0.594</b>           | <b>0.632</b>           |
| 210   | <b>0.630</b>           | <b>0.590</b>           | <b>0.680</b>           | 0.473                  | <b>0.598</b>           | <b>0.571</b>           | <b>0.547</b>                        | <b>0.594</b>           | <b>0.623</b>           |
| 220   | <b>0.630</b>           | <b>0.590</b>           | <b>0.680</b>           | 0.464                  | <b>0.598</b>           | <b>0.571</b>           | <b>0.542</b>                        | <b>0.594</b>           | <b>0.623</b>           |
| 230   | <b>0.630</b>           | <b>0.590</b>           | <b>0.680</b>           | 0.455                  | <b>0.580</b>           | <b>0.562</b>           | <b>0.538</b>                        | <b>0.585</b>           | <b>0.618</b>           |
| 240   | <b>0.640</b>           | <b>0.590</b>           | <b>0.680</b>           | 0.455                  | <b>0.598</b>           | <b>0.562</b>           | <b>0.542</b>                        | <b>0.594</b>           | <b>0.618</b>           |
| Final | <b>0.640</b>           | <b>0.570</b>           | <b>0.680</b>           | 0.446                  | <b>0.589</b>           | <b>0.562</b>           | <b>0.538</b>                        | <b>0.580</b>           | <b>0.618</b>           |

Outra informação importante, obtida a partir dos resultados experimentais, é referente ao número de passos, utilizados pelos métodos heurísticos, até atingir a melhor solução encontrada.

Os gráficos das Figuras 4.8 e 4.9 demonstram a relação entre os dois métodos (*SAHe-SAT+Hp* e *Hp-pura*), ao considerar-se apenas os passos utilizados pelas heurísticas de pós-processamento. Cada gráfico contém três curvas: uma para cada *Hp*. As curvas representam uma distribuição acumulada da fração das instâncias (eixo vertical) com razão  $\leq$  a  $X$ , onde  $X$  é o resultado da divisão do número de passos executados pela *Hp* que gerou o melhor resultado do *SAHe-SAT+Hp*, pelo número de passos executados pela mesma *Hp* que gerou o melhor resultado *Hp-pura* (eixo horizontal). Ou seja, as curvas representam a fração de ocorrências (instâncias) em que o número de passos da *SAHe-SAT+Hp* dividido pelo número de passos da *Hp-pura* foi menor ou igual que um dado valor. Os gráficos (a) e (b) correspondem às relações para todas as instâncias do conjunto de 2004 e para as instâncias do mesmo conjunto sem ótimo conhecido, respectivamente; (c) e (d) possuem as mesmas informações sobre o conjunto de 2008 e; (e) e (f) apresentam as relações para os dois conjuntos combinados.

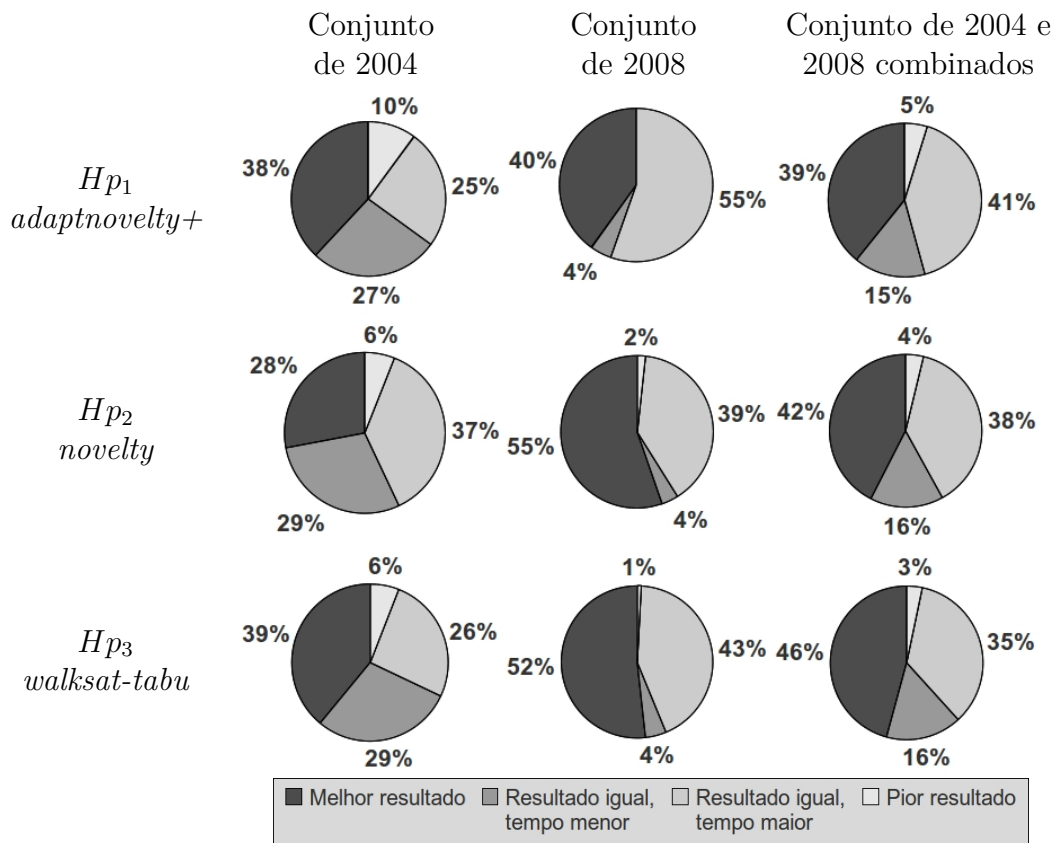


Figura 4.7: Comparação dos valores obtidos entre  $SAHe-SAT+Hp$ 's e  $Hp$ 's-puras, sobre todas as 100 instâncias do conjunto de 2004 e todas as 112 de 2008.

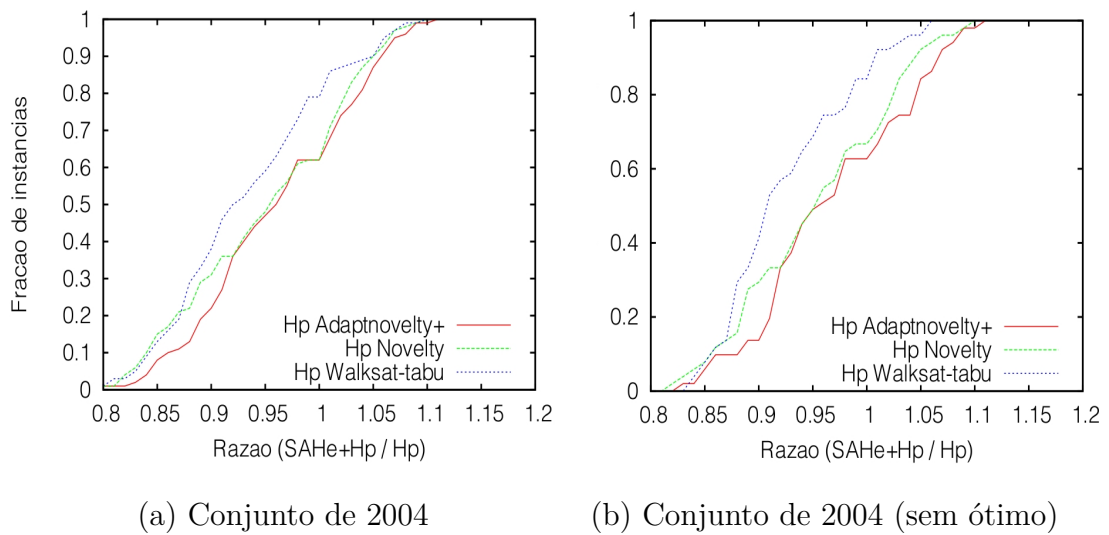
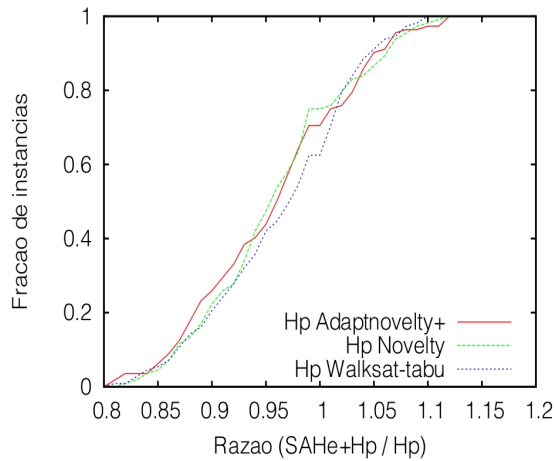
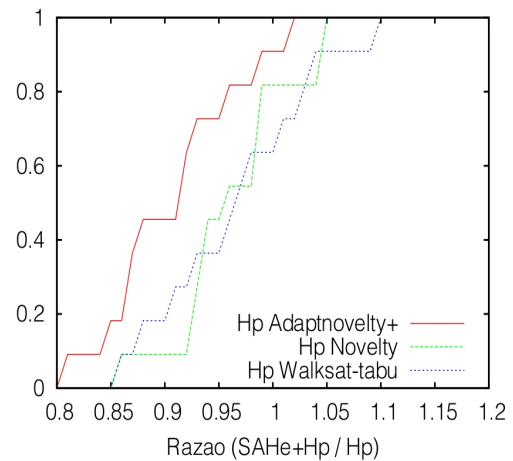


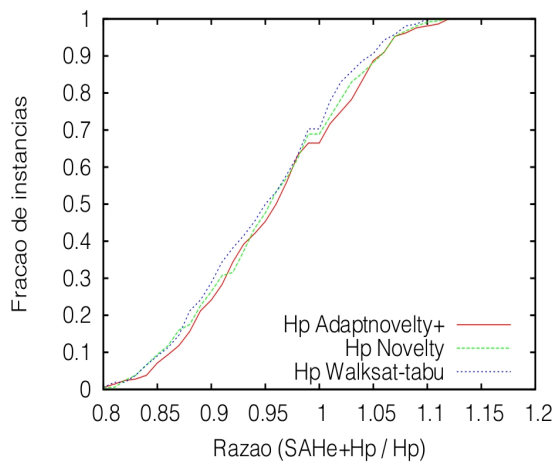
Figura 4.8: Distribuição acumulada da fração das instâncias (eixo vertical) com razão  $\leq X$ , onde  $X$  é a fração entre o número de passos do melhor resultado do  $SAHe-SAT+Hp$  e o número de passos do melhor resultado  $Hp-pura$  (eixo horizontal).



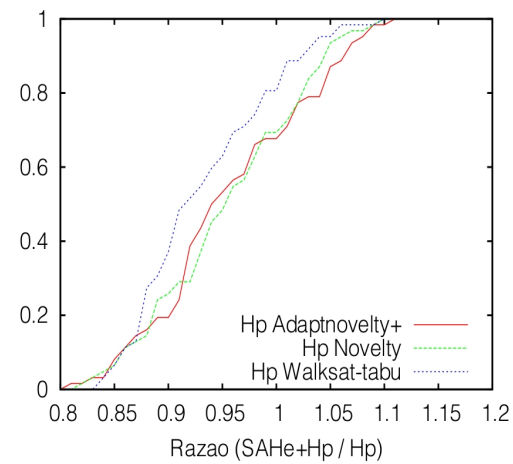
(c) Conjunto de 2008



(d) Conjunto de 2008 (sem ótimo)



(e) Conjuntos combinados



(f) Conjuntos combinados (sem ótimo)

Figura 4.9: Distribuição acumulada da fração das instâncias (eixo vertical) com razão  $\leq X$ , onde  $X$  é a fração entre o número de passos do melhor resultado do *SAHe-SAT+Hp* e o número de passos do melhor resultado *Hp-pura* (eixo horizontal). (continuação)

É possível notar que em todas as curvas dos gráficos das Figuras 4.8 e 4.9, entre 60% e 70% das instâncias, possuem razão inferior à 1.0, significando que, para esta fração de instâncias, o *SAHe-SAT* contribuiu para a diminuição do número de passos utilizados pela *Hp*, até alcançar o seu melhor resultado.

### 4.3 Considerações Finais

Foram apresentadas neste capítulo a formulação e descrição de um novo método para resolução de instâncias do problema Max-SAT, denominado *SAHe-SAT*, que maximiza os resultados alcançados por outras heurísticas, utilizadas aqui como rotinas de pós-processamento (*Hp*).

Por tratar-se de um problema largamente estudado e que possui heurísticas eficientes, a criação ou modificação de métodos para que se obtenham resultados superiores é uma tarefa árdua e muitas vezes improdutiva. Assim, os resultados experimentais alcançados não apresentaram diferenças tão marcantes e facilmente identificáveis. Entretanto, mesmo neste contexto desfavorável, o uso do *SAHe-SAT* conduziu à melhores soluções em mais de 50% das instâncias testadas.

Outra característica importante, apontada pelos experimentos, foi a diminuição dos passos executados pelas heurísticas de pós-processamento até atingir-se a solução ótima. Fato que ocorreu entre 60 e 70% das instâncias testadas, demonstrando que o mecanismo criado retirou o caráter aleatório das condições iniciais de busca das heurísticas, fazendo-as operar em condições iniciais mais favoráveis.

Por acreditar que soluções similares a este método podem ser aplicados a problemas com características também similares, resolveu-se testá-lo no problema da Cobertura de Vértices Mínima (CVM). Sua formulação e resultados experimentais serão apresentados no próximo capítulo.

Esta abordagem será novamente trabalhada no Capítulo 6, onde algumas conjecturas sobre novos testes e possíveis modificações nesta abordagem serão elencadas.

# Capítulo 5

## Seleção por Algoritmo Genético para CVM (*SAHe-CVM*)

Este capítulo apresenta um método de geração automática de heurísticas para instâncias do problema de Cobertura de Vértices Mínima (CVM). O método proposto é uma variação do método, descrito no Capítulo 4, denominado *seleção automática de heurísticas para o problema CVM*, ou, simplesmente, *SAHe-CVM*, e será oportunamente descrito na Seção 5.1.

Diferentemente do que ocorreu com os experimentos realizados com o problema Max-SAT, não foram encontrados pacotes com implementações dos melhores métodos heurísticos disponíveis para o problema CVM e que também pudessem ser utilizados como rotinas de pós-processamento. Com o objetivo de realizar a análise experimental, foram implementados alguns métodos heurísticos simples, que também realizam a manipulação local das variáveis na busca por uma solução viável ótima. São eles: *Guloso*, *Guloso com Random-Walk*, *Guloso com Histórico* e *Guloso com Histórico e com Random-Walk*. De modo complementar, os resultados experimentais obtidos com o *SAHe-CVM* serão comparados com algumas heurísticas, descritas na Subseção 2.2.3, que também possuem resultados experimentais sobre o mesmo conjunto de instâncias. Por fim, serão feitas considerações sobre os resultados encontrados na Seção 5.3.

## 5.1 Descrição da Abordagem

Como visto na Seção 2.2, o problema CVM possui a característica de minimização local. Ou seja, ao olhar para um vértice específico e sua vizinhança, é possível realizar alterações no estado do vértice com objetivo de melhorar o valor de sua função de minimização local, resultando, também, na melhora da função de minimização global. Para cada instância do problema, entretanto, existe uma sequência diferente de alterações que conduzem à *solução ótima*.

Neste capítulo, semelhante ao apresentado no Capítulo 4, foi elaborado um método para seleção automática de heurísticas aplicáveis à instâncias do problema CVM. A ideia é encontrar a melhor sequência de *vértices/ações*, que efetue a manipulação inicial dos estados dos vértices, para posterior emprego de uma rotina de pós-processamento, capaz de conduzir à uma solução viável ótima. Para efetuar a exploração do espaço de estados, utiliza-se um Algoritmo Genético definido na Seção 5.1.2.

Considerando que o método descrito a seguir é uma variante do método apresentado no Capítulo 4, cuja motivação e mecanismo já foi descrito, torna-se desnecessário tecer novos comentários. Assim, partir-se-á, desde logo, para a sua definição.

### 5.1.1 Fuga de Máximos Locais de Baixa Qualidade

Dada uma instância  $I = (V, E)$ , onde  $V$  é o conjunto de vértices,  $E$  o conjunto de arestas e  $W_i$  o conjunto de todos os vértices  $j \in V$ , tal que  $(i, j) \in E$ . Uma heurística pode ser definida como uma sequência de pares ordenados  $H = \langle (v_1, a_1), (v_2, a_2), \dots, (v_k, a_k) \rangle$ , onde  $v_i$  é o vértice visitado no passo  $i$  e  $a_i$  a ação efetuada sobre este vértice. Para cada passo  $i$  da heurística,  $v_i \cup W_i$  é consultado e o estado de  $v_i$  é alterado respeitando a ação  $a_i$ .

Considerando o procedimento de minimização por análise local, descrito na Subseção 2.2.2, e a definição de heurística, é possível chegar ao Algoritmo 6, que executa uma série de análises e alterações locais em uma dada instância, seguindo uma certa *heurística*.

Verifica-se que, após efetuar todos os passos da heurística, não existe garantia de que o estado final dos vértices será uma solução viável para a instância. Portanto,



torna-se imprescindível a aplicação de outra rotina para a geração de uma solução viável (ótima ou não). Qualquer método, que retorne uma solução viável, pode ser aplicado a partir do estado final dos vértices manipulados pela heurística.

---

**Algoritmo 6** Heurística para *SAHe-CVM*

---

```

1: Dada uma sequência de pares ordenados  $H = \langle (v_1, a_1), (v_2, a_2), \dots, (v_k, a_k) \rangle$ ,
   tal que  $v_i \in V$  e  $a_i = \{0, 1, 2\}$ ;
2: Seja  $x_i$  uma variável Booleana associada ao vértice  $i \in V$ ;
3: Seja  $W_i$  o conjunto de todos os vértices  $j \in V$ , tal que  $(i, j) \in E$ ;
4: Efetue a inicialização aleatória do estado de cada  $x_i$ ;
5: para todo  $v_i$  faça
6:   se  $x_j = 1, \forall j \in W_i$  então
7:     se  $a_i = 1$  então
8:        $x_i \leftarrow 0$ ;
9:     senão se  $a_i = 0$  então
10:       $x_i \leftarrow 1$ ;
11:     fim se
12:   senão
13:     se  $a_i = 1$  então
14:        $x_i \leftarrow 1$ ;
15:     senão se  $a_i = 0$  então
16:        $x_i \leftarrow 0$ ;
17:     fim se
18:   fim se
19: fim para

```

---

### 5.1.2 Exploração do Espaço de Estados

Como acontece com a formulação proposta para o problema Max-SAT, a escolha da heurística  $H$  pode ser vista como um problema de otimização sobre todas as suas combinações possíveis.

Optou-se por replicar o mesmo mecanismo evolucionário, apresentado na Seção 5.1.2, cujos indivíduos representam as várias possibilidades para  $H$ , com o objetivo de encontrar o indivíduo que minimiza o valor de  $f$  (Equação 2.6). Cada indivíduo é avaliado por uma *função objetivo*, que fornece uma medida de qualidade em relação à instância do problema CVM, denotada por:

$$\varphi(H, Hp) = f(s^*) \tag{5.1}$$

onde  $s^*$  corresponde ao estado dos vértices após a execução do Algoritmo 6 sobre

a sequência  $H$  específica, e de alguma rotina de pós-processamento  $Hp$ , que é a mesma para todos os indivíduos.

Com o objetivo de minimizar a possibilidade de uma má escolha da configuração inicial  $s$ , optou-se por reinicializar os estados dos vértices 10 vezes. Para cada escolha, o Algoritmo 6 deve ser executado, obtendo um valor  $s'$ . A rotina de pós-processamento  $Hp$  será executada apenas sobre o melhor valor de  $f(s')$ , obtendo uma configuração final  $s^*$  de valores para os vértices e  $f(s^*)$ , como medida de qualidade do indivíduo.

### **Algoritmo Genético (AG) para a Formulação**

O algoritmo genético usado nos experimentos da Seção 5.2 é uma variação do algoritmo utilizado na formulação para o problema Max-SAT (Algoritmo 5). Na formulação original não havia reinicialização da população durante o processo evolucionário. Entretanto, durante a realização dos experimentos de calibração do modelo para o problema CVM, percebeu-se que, após algumas gerações, havia uma estagnação da população em torno de um ótimo local. Para reduzir a ocorrência deste fenômeno indesejado, optou-se por inserir periodicamente o mecanismo de reinicialização da população.

O algoritmo da formulação original foi alterado e a nova formulação está descrita no Algoritmo 7, cujos passos 5 a 8 são responsáveis por verificar e realizar a reinicialização da população, gerando, aleatoriamente, uma população totalmente nova. Assim como em sua versão original, este algoritmo evolui sobre um número fixo de gerações, cada uma compreendendo um número fixo de indivíduos, sendo a saída o melhor indivíduo encontrado no final da sua execução. Dois critérios de parada podem determinar o fim da execução: encontrar a solução ótima (quando conhecida) ou atingir um tempo limite estipulado. A população inicial e aquela obtida por reinicialização são preenchidas aleatoriamente, dados  $n$  (número de vértices da instância) e um  $t$  máximo (tamanho máximo da sequência). A partir delas, cada nova geração é obtida a partir da geração prévia, realizando primeiramente um passo elitista, onde os indivíduos mais aptos, dada uma fração fixa da população, são copiados diretamente para a nova geração. Em sequência, os demais indivíduos da população são obtidos através da aplicação do *crossover* (como um par) ou mutação

(individualmente) sobre indivíduos selecionados da geração anterior. Os indivíduos são selecionados aleatoriamente para o *crossover* e para a mutação, com probabilidade proporcional à normalização linear de sua posição na população, como descrito na Seção 4.1.2.

---

**Algoritmo 7** Algoritmo Genético para *SAHe-CVM*

---

```

1:  Seja  $P$  a população de cromossomos, e  $I$  a instância a ser resolvida.
2:  Inicializar  $P$  e  $I$ 
3:  enquanto o critério de parada não for satisfeito faça
4:    Avalie os indivíduos em  $P$  para a instância  $I$ 
5:    se o passo atual é um passo de reinicialização então
6:      Reinicialize a população  $P$  com novos indivíduos
7:      Avalie os indivíduos em  $P$  para a instância  $I$ 
8:    fim se
9:    Ordene  $P$  pelo valor decrescente de aptidão
10:   Faça o elitismo dos  $e$  melhores indivíduos da população  $P$ 
11:   Insira a elite em uma nova população  $P'$ 
12:   repita
13:     Escolha aleatoriamente um número  $t$  no intervalo  $[0, 1]$ 
14:     se  $t < T_c$  então
15:       Selecione por rank linear 2 indivíduos em  $P$ 
16:       Aplique o operador de crossover
17:       Insira os novos indivíduos em  $P'$ 
18:     senão
19:       Selecione por rank linear 1 indivíduo em  $P$ 
20:       Aplique o operador de mutação
21:       Insira o novo indivíduo em  $P'$ 
22:     fim se
23:   até a população  $P'$  estar completa
24:    $P \leftarrow P'$ 
25: fim enquanto

```

---

## 5.2 Experimentos Computacionais

Com o objetivo de realizar a análise experimental do método *SAHe-CVM*, descrito na Seção 5.1, foram efetuados uma série de experimentos computacionais. Os resultados apresentados nesta seção referem-se a dois conjuntos de instâncias – conhecidos e referenciados na literatura [4, 23, 51, 62] –, extraídos de repositórios de instâncias para problemas em grafos, tais como: *clique máxima*, *conjunto independente máximo*, *cobertura de vértices mínima* e *coloração de vértices*. O primeiro deles é um conjunto instâncias extraído do repositório do desafio *Second DIMACS*

*Challenge* [22], chamado, a partir deste momento, de *conjunto de 1996*, o qual possui 80 (oitenta) grafos, distribuídos em 9 (nove) famílias distintas. O segundo conjunto, contendo 41 (quarenta e um) grafos, foi extraído do repositório criado por XU [62], referenciado como *conjunto de 2010*. Os grafos deste segundo conjunto são gerados aleatoriamente, segundo o *modelo RB* [63].

A Subseção 5.2.1 descreve um conjunto de métodos heurísticos, criados e utilizados como rotinas de pós-processamento do método *SAHe-CVM*. Em seguida, na Subseção 5.2.2, é apresentado o ambiente experimental onde os testes foram realizados; na Subseção 5.2.3 constam os resultados dos testes de calibração do modelo, que apontaram para a definição de um conjunto de parâmetros utilizados no restante dos experimentos.

Os resultados experimentais gerais são descritos na Subseção 5.2.4, onde, primeiramente, são apresentados os resultados do método *SAHe-CVM* sobre todas as instâncias dos conjuntos de 1996 e de 2010, comparado às heurísticas de pós-processamento (descritas na Subseção 5.2.1) e utilizadas de maneira isolada. Em sequência, são feitas comparações entre os resultados obtidos pelo *SAHe-CVM* e os obtidos por outras heurísticas (descritas na Seção 2.2.3), quando aplicados as instâncias do conjunto de 1996.

O conjunto completo de instâncias de *benchmark*, utilizadas nos experimentos computacionais, está descrito nas Tabelas C.1 a C.3, do Anexo C, que também apresenta o valor ótimo da cobertura de vértices do grafo. As entradas omitidas, nessas tabelas, indicam os casos que o valor ótimo não é conhecido.

### 5.2.1 Heurísticas de Pós-Processamento Criadas

Como afirmado anteriormente, não existem implementações disponíveis dos melhores métodos heurísticos para o problema CVM. Com o objetivo de realizar a análise experimental, foram implementadas algumas variações de métodos heurísticos gulosos. Estas variações são bem conhecidas e comumente utilizadas, como parte dos métodos heurísticos aplicados para resolução de problemas em grafos e otimização combinatória. Além disso, eles também realizam a manipulação local das variáveis na busca por uma solução viável ótima. Os métodos são:

*gcv*: busca gulosa para CVM. Para cada reinicialização aleatória dos valores dos

vértices (respeitando um máximo definido) realiza-se um conjunto máximo de trocas (respeitando um máximo definido), até que o número mínimo de vértices faça parte da cobertura. Cada troca é realizada escolhendo-se o vértice que, ao ser modificado, gerará o maior ganho para a função  $f(x)$ ;

*gwcvm*: *gcv*m com *random walk*. Com probabilidade  $p$ , deve-se escolher aleatoriamente um vértice cuja alteração produza um ganho (minimização) em  $f(x)$ , devendo-se trocar seu estado. Caso contrário ( $1 - p$ ), deve-se efetuar o procedimento padrão do *gcv*m;

*hcv*m: *gcv*m com informação sobre histórico. O algoritmo escolhe, dentre os vértices selecionados para troca, aquele que foi alterado há mais tempo. Se dentre os vértices selecionados existirem os que nunca foram alterados, a escolha deverá ser feita de acordo com uma ordem fixa e pré-estabelecida;

*hwcvm*: *hcv*m com *random walk*. Com probabilidade  $p$ , deve-se escolher aleatoriamente um vértice cuja alteração produza um ganho (minimização) em  $f(x)$ , devendo-se trocar seu estado. Caso contrário ( $1 - p$ ), deve-se efetuar o procedimento padrão do *hcv*m.

Estas heurísticas foram incluídas como rotinas de pós-processamento  $Hp$  do método *SAHe-CVM*. Da mesma forma que acontece no modelo, apresentado no Capítulo 4, aqui o AG também realiza a manipulação dos seus operadores genéticos e, como parte do cálculo da aptidão de cada indivíduo de sua população (Passo 4 do Algoritmo 7), efetua a execução da heurística  $Hp$ . Para tanto, quando um indivíduo da população é selecionado, a heurística que ele representa (sequência *vértices/ações*) é executada, gerando um novo conjunto de valores para os vértices, utilizados como entrada da  $Hp$ . Após a execução da  $Hp$ , obtém-se uma solução viável para a instância e, por consequência, a aptidão do indivíduo.

## 5.2.2 Ambiente

Os algoritmos deste trabalho foram desenvolvidos na linguagem C padrão *ANSI*, e os testes realizados em duas configurações diferentes de máquinas:

- *1a configuração*: cluster do Laboratório de Inteligência Artificial do PESC-COPPE/UFRJ, formado por 15 máquinas iguais, com processadores Intel Dual Core, 3.0GHz de frequência, 4 GBytes de memória, disco rígido de 80GBytes e uma rede Gigabit Ethernet. O sistema operacional é o *Ubuntu 9.04* com *kernel 2.6.24-28* e compilador *gcc* de versão 4.2;
- *2a configuração*: cluster do Laboratório de Inteligência Artificial do PESC-COPPE/UFRJ, formado por 10 máquinas iguais, com processadores Intel Quad Core, 3.0GHz de frequência, 4 GBytes de memória, disco rígido de 80GBytes e uma rede Gigabit Ethernet. O sistema operacional é o *Ubuntu 9.04* com *kernel 2.6.24-28* e compilador *gcc* de versão 4.2.

### 5.2.3 Calibração do Modelo

Dos métodos heurísticos descritos na Subseção 5.2.1, o método guloso (*gcv*) foi selecionado para os experimentos iniciais e calibração do método proposto. Como conjunto de instâncias para esta primeira etapa de experimentos, foi selecionado o conjunto de 1996 <sup>1</sup>.

Em um primeiro momento, foram realizados alguns testes com o objetivo de verificar o comportamento da heurística *gcv* e do método *SAHe-CVM*, aplicado a um subconjunto de instâncias com até 200 vértices e com valor ótimo conhecido. Para cada instância, a metodologia de testes empregada foi a seguinte: uma configuração inicial de estados para os vértices foi gerada e a heurística *gcv* foi executada até que uma solução viável fosse encontrada; caso esta solução fosse ótima, o algoritmo era finalizado; do contrário, o processo era repetido com uma nova configuração inicial. Este processo foi repetido algumas vezes para cada instância. Complementarmente, testes com o *SAHe-CVM+gcv* também foram realizados para o mesmo subconjunto de instâncias.

Ao fim dessa primeira etapa, a instância `brock200_4` (da família `brock $n$ _ $x$`  [64] <sup>2</sup>) foi escolhida para realizar experimentos mais refinados com os parâmetros do AG.

<sup>1</sup>[4, 23, 41, 43, 46, 47] são exemplos de trabalhos que também utilizam este conjunto de instâncias como *benchmark*.

<sup>2</sup>Esta família é composta por grafos aleatórios com  $n$  nós, que são gerados para conter uma clique máxima muito maior do que se espera para instâncias com o mesmo grau. O número  $x$  diferencia uma instância da outra.

## Execução da Heurística *gcv*m Pura

Novos experimentos com a heurística *gcv*m foram realizados sobre a instância `brock200_4` (totalizando 20 execuções), e seus resultados constam na Tabela 5.1. Nesta tabela, a coluna “Sementes” contém o número de configurações iniciais geradas até o ótimo ser atingido; a coluna “Trocas” indica o número de trocas efetuadas no estado dos vértices durante a execução que conduziu ao valor ótimo e; a coluna “Tempo(m)” da tabela contém o tempo médio dos experimentos com o *gcv*m (expresso em minutos). Ainda, a linha “Mínimo” da tabela contém os dados referentes à execução do *gcv*m que encontrou a solução ótima mais rapidamente; a linha “Máximo” contém os dados referentes à execução do *gcv*m que demorou mais tempo para encontrar a solução ótima e a média dos dados é informada na linha “Média” e está destacada (em negrito).

Tabela 5.1: Resultados do alg. *gcv*m para a instância `brock200_4` [64].

|              | Sementes       | Trocas      | Tempo(m)       |
|--------------|----------------|-------------|----------------|
| Mínimo       | 650967         | 79          | 21.376         |
| Máximo       | 21568421       | 97          | 708.787        |
| <b>Média</b> | <b>6297871</b> | <b>89.5</b> | <b>238.982</b> |

## Execução do *SAHe-CVM + gcv*m

O refinamento dos parâmetros do *SAHe-CVM+gcv*m foi realizado manipulando-se todos os parâmetros do AG. A metodologia empregada para estes testes foi a seguinte: realizou-se um total de 20 (vinte) execuções do AG; para cada execução uma nova população inicial de heurísticas foi gerada aleatoriamente e reinicializada de tempos em tempos, também de forma aleatória. Seu critério de parada foi a geração de uma heurística que conduz à solução ótima.

Alguns resultados do *SAHe-CVM+gcv*m, para diferentes configurações, estão dispostos na Tabela 5.2. Nesta tabela, a primeira coluna contém o número de gerações para reinício da população, onde “(a)” indica reinicialização após  $x$  gerações, e “(b)” reinicialização após  $x$  gerações sem a melhora na aptidão<sup>3</sup>. Já, as colunas “Pop.” e “Ind.” indicam, respectivamente, o tamanho da população e do indivíduo.

<sup>3</sup>Este tipo de reinicialização foi testado com o objetivo de reduzir o fenômeno de estagnação da população, detectado nos testes de calibração do modelo.

Os parâmetros utilizados no processo de seleção e no cálculo de aptidão dos indivíduos estão listados nas colunas de 4 a 8. Por fim, a última coluna da tabela tem o tempo médio dos testes (expresso em minutos).

Tabela 5.2: Resultados do *SAHe-CVM+gcv*m para algumas combinações de parâmetros do AG `brock200_4` [64].

| Reinício     | Pop.       | Ind.       | $e$       | $T_c$      | $T_m$      | $\Phi$     | $t$       | Tempo(m)      |
|--------------|------------|------------|-----------|------------|------------|------------|-----------|---------------|
| 20(a)        | 500        | 2000       | 5%        | 25%        | 50%        | 5.0        | 10        | 201.615       |
| 50(a)        | 100        | 300        | 5%        | 25%        | 75%        | 2.0        | 10        | 192.575       |
| <b>50(a)</b> | <b>100</b> | <b>300</b> | <b>5%</b> | <b>25%</b> | <b>75%</b> | <b>5.0</b> | <b>10</b> | <b>93.545</b> |
| 50(a)        | 100        | 2000       | 5%        | 25%        | 75%        | 2.0        | 10        | 144.994       |
| 50(a)        | 500        | 2000       | 5%        | 25%        | 50%        | 5.0        | 10        | 180.594       |
| 50(a)        | 500        | 2000       | 5%        | 25%        | 75%        | 2.0        | 20        | 324.389       |
| 100(a)       | 500        | 300        | 5%        | 25%        | 50%        | 5.0        | 10        | 174.830       |
| 100(a)       | 500        | 2000       | 5%        | 25%        | 50%        | 5.0        | 20        | 170.261       |
| 100(a)       | 500        | 2000       | 5%        | 50%        | 50%        | 2.0        | 10        | 158.366       |
| 30(b)        | 500        | 300        | 5%        | 25%        | 75%        | 5.0        | 10        | 263.746       |
| 30(b)        | 500        | 500        | 5%        | 25%        | 75%        | 2.0        | 10        | 194.547       |
| 50(b)        | 500        | 2000       | 5%        | 50%        | 25%        | 2.0        | 10        | 106.502       |
| 100(b)       | 500        | 2000       | 5%        | 25%        | 75%        | 5.0        | 10        | 305.840       |

Como é possível verificar, em vários casos o AG retornou melhores resultados do que aqueles obtidos pelo algoritmo guloso simples. A configuração de parâmetros que produziu os melhores resultados (linha da Tabela 5.2 em negrito) para a instância `brock200_4` será detalhada no final desta subseção; seus resultados estão descritos na Tabela 5.3, na qual a coluna “Gerações” enumera as gerações do AG até o ótimo ser atingido. Na coluna “Sementes” está descrito o número *estimado* de heurísticas geradas<sup>4</sup>. Na coluna “Trocias” indica-se o número de alterações efetuadas pela heurística de pós-processamento *gcv*m (descrita na Subseção 5.2.1) sobre os vértices. Na coluna “Ind.” está descrito o tamanho da heurística que obteve o melhor resultado. Por fim, a última coluna da tabela demonstra o tempo de duração dos testes (expresso em minutos). Novamente, são fornecidos os valores mínimo, máximo e médio para os experimentos com o método *SAHe-CVM+gcv*m.

Ao comparar os valores apresentados nas Tabelas 5.1 e 5.3, é possível notar que os resultados encontrados pelo *SAHeCVM+cv*m são melhores do que aqueles encontrados pela aplicação da heurística de *gcv*m pura, apontando para um futuro

<sup>4</sup>O número *estimado* de heurísticas geradas pelo Algoritmo Genético corresponde ao produto dos valores dos parâmetros: número de gerações, tamanho da população e número de testes para cada heurística.



promissor, confirmado através da realização de novos experimentos com as outras heurísticas de pós-processamento implementadas (descritas na Subseção 5.2.1). Os resultados desses experimentos são descritos a seguir.

Tabela 5.3: Resultados do *SAHeCVM+gcv*m para `brock200_4` [64].

|              | Gerações   | Sementes         | Ind.       | Trocas      | Tempo(m)      |
|--------------|------------|------------------|------------|-------------|---------------|
| Mínimo       | 29         | ~ 137750         | 278        | 76          | 5.664         |
| Máximo       | 1805       | ~ 8573750        | 222        | 87          | 338.967       |
| <b>Média</b> | <b>499</b> | ~ <b>2370250</b> | <b>243</b> | <b>86.5</b> | <b>93.545</b> |

### Conjunto de Parâmetros do AG

A partir dos resultados alcançados com a calibração do modelo, é possível destacar algumas características desejáveis para a realização dos próximos experimentos. Dentre elas, verificou-se que o tamanho máximo das heurísticas geradas não deve ser muito maior que o número de vértices; a reinicialização da população deve ser realizada em intervalos curtos, e a taxa de *crossover* deve ser baixa. Além disso, a taxa de mutação dos genes deve ser alta.

Com base nos resultados encontrados na Subseção 5.2.3, os parâmetros que representam os melhores resultados encontrados são descritos a seguir:

- o AG será executado sobre cada instância, até que o valor ótimo seja encontrado ou o tempo máximo (quatro horas) seja alcançado;
- reinicialização da população após 50 gerações;
- cada geração contém  $K = 100$  indivíduos;
- cada indivíduo contém até  $1.5n$  genes (para  $n = |V|$ );
- a taxa mínima de variáveis distintas em cada indivíduo é 40% do número total de variáveis da instância;
- número de inicializações aleatórias para os estados dos vértices com cada heurística,  $t = 10$ ;
- o elitismo da população é 5%;

- a decisão entre *crossover* e mutação possui taxa de 25% para a escolha do primeiro;
- a mutação ocorre em até 75% dos genes;
- foi escolhido  $\Phi = 5$  como probabilidade de seleção (Equação 4.5).

#### 5.2.4 Resultados Experimentais

A fase de calibração do modelo foi responsável por definir o conjunto de parâmetros utilizados nos experimentos finais do problema CVM. Nesta primeira etapa, apenas a heurística de pós-processamento *gcv* foi utilizada. Entretanto, foram criadas algumas variações da heurística gulosa (*gcv*) para a segunda etapa de experimentos (heurísticas *gwcv*, *hcv* e *hwcv*). Os resultados apresentados nesta subseção referem-se a elas, utilizadas em conjunto com o *SAHe-CVM* (como rotinas de pós-processamento), ou sozinhas (como *heurísticas puras*). Neste último caso, cada heurística é executada repetidamente, cada vez com uma nova associação aleatória de valores verdade para os vértices, até satisfazer o critério de parada: valor ótimo encontrado ou tempo máximo atingido.

Assim como ocorreu nos experimentos realizados com o *SAHe-SAT*, o resultado reportado pelo *SAHe-CVM* refere-se ao melhor indivíduo da última população que foi preenchida. Ao encontrar o valor ótimo ou atingir o tempo máximo, previamente estipulado, o algoritmo recebe um sinal, sendo encerrado no momento em que a próxima população de indivíduos for preenchida. Com o objetivo de comparar o desempenho do *SAHe-CVM*, tanto quanto possível, o resultado reportado para a *heurística pura* corresponde ao melhor resultado obtido a cada 100 repetições.

Nos experimentos, foram utilizadas todas as 80 instâncias do conjunto de 1996 [65] e todas as 41 do conjunto de 2010 [62]. O tempo limite, alocado para cada instância do *SAHe-CVM* e para cada uma das *heurísticas puras*, foi de 240 minutos ou 4 horas, sempre com configuração de hardware e software idênticas, e com acesso exclusivo ao sistema.

Todas as instâncias do conjunto de 2010 possuem o valor ótimo conhecido, porquanto tais instâncias tenham sido geradas a partir de um valor ótimo pré-determinado. Entretanto, 26 instâncias do conjunto de 1996 não possuem um valor

ótimo conhecido.

Como apontado anteriormente, os casos em que o valor ótimo não é conhecido são de maior interesse para análise do *SAHe-CVM*, pois correspondem aos casos mais difíceis e que ainda não foram resolvidos por um método exato. Desta forma, as Tabelas 5.4 a 5.7, respectivamente para  $Hp = gcvm$ ,  $Hp = gwcv$ ,  $Hp = hcvm$  e  $Hp = hwcv$ , descrevem os resultados sobre as instâncias difíceis do conjunto de 1996. Cada tabela contém uma linha para cada uma das 26 instâncias, com uma linha horizontal separando as famílias. Para cada instância estão descritos o número de vértices, bem como o número de arestas e os resultados para as estratégias *SAHe-CVM+Hp* e *Hp-pura* em questão. Estes resultados são relativos ao número de vértices na cobertura e ao tempo em que esta solução foi encontrada pela primeira vez, durante o tempo limite de 240 minutos.

Tabela 5.4: Resultados do *SAHe-CVM* e  $Hp = gcvm$ .

| Família             | Instância            | Vértices | Arestas | <i>SAHe-CVM+gcvm</i> |               | <i>gcvm</i> |          |
|---------------------|----------------------|----------|---------|----------------------|---------------|-------------|----------|
|                     |                      |          |         | Cobertura            | Tempo(m)      | Cobertura   | Tempo(m) |
| <i>Cn.x</i>         | C125.9               | 125      | 787     | <b>91</b>            | <b>0.001</b>  | 91          | 0.065    |
|                     | C250.9               | 250      | 3141    | <b>206</b>           | 15.389        | 208         | 15.618   |
|                     | C500.9               | 500      | 12418   | <b>448</b>           | 18.158        | 449         | 22.031   |
|                     | C1000.9              | 1000     | 49421   | <b>941</b>           | 105.085       | 942         | 0.120    |
|                     | C2000.5              | 2000     | 999164  | <b>1986</b>          | <b>20.626</b> | 1986        | 22.063   |
|                     | C2000.9              | 2000     | 199468  | <b>1935</b>          | 20.891        | 1936        | 11.553   |
|                     | C4000.5              | 4000     | 3997732 | 3985                 | 174.480       | 3985        | 164.458  |
| DSJ <i>Cn.x</i>     | DSJC500.5            | 500      | 63264   | 487                  | 14.790        | 487         | 13.117   |
|                     | DSJC1000.5           | 1000     | 249674  | <b>986</b>           | <b>2.712</b>  | 986         | 205.580  |
| <i>hammingW-d</i>   | <i>hamming10-4</i>   | 1024     | 89600   | <b>988</b>           | 44.174        | 990         | 1.051    |
| <i>johnsonW-w-d</i> | <i>johnson32-2-4</i> | 496      | 14880   | 480                  | 0.031         | 480         | 0.000    |
| <i>kellerd</i>      | <i>keller6</i>       | 3361     | 1026582 | <b>3320</b>          | 180.345       | 3323        | 13.914   |
| <i>MANN_ax</i>      | <i>MANN_a81</i>      | 3321     | 6480    | <b>2240</b>          | <b>0.159</b>  | 2240        | 30.749   |
| <i>p_hatn-x</i>     | <i>p_hat500-3</i>    | 500      | 30950   | <b>450</b>           | 83.408        | 451         | 210.405  |
|                     | <i>p_hat700-1</i>    | 700      | 183651  | <b>689</b>           | <b>2.092</b>  | 689         | 6.614    |
|                     | <i>p_hat700-2</i>    | 700      | 122922  | 657                  | 12.972        | 656         | 184.272  |
|                     | <i>p_hat700-3</i>    | 700      | 61640   | <b>640</b>           | 92.064        | 641         | 97.152   |
|                     | <i>p_hat1000-1</i>   | 1000     | 377247  | 990                  | 11.628        | 990         | 6.321    |
|                     | <i>p_hat1000-2</i>   | 1000     | 254701  | <b>955</b>           | 156.891       | 956         | 3.143    |
|                     | <i>p_hat1000-3</i>   | 1000     | 127754  | <b>937</b>           | 150.598       | 939         | 23.397   |
|                     | <i>p_hat1500-1</i>   | 1500     | 839327  | 1489                 | 59.134        | 1489        | 57.577   |
|                     | <i>p_hat1500-2</i>   | 1500     | 555290  | <b>1440</b>          | 44.548        | 1442        | 207.045  |
|                     | <i>p_hat1500-3</i>   | 1500     | 277006  | <b>1412</b>          | 127.723       | 1414        | 14.328   |
| <i>sanrn_p</i>      | <i>sanr200_0.9</i>   | 200      | 2037    | <b>158</b>           | 0.950         | 160         | 3.594    |
|                     | <i>sanr400_0.5</i>   | 400      | 39816   | <b>387</b>           | <b>0.890</b>  | 387         | 3.861    |
|                     | <i>sanr400_0.7</i>   | 400      | 23931   | <b>379</b>           | <b>8.808</b>  | 379         | 142.667  |

Algumas entradas nas tabelas estão destacadas (em negrito) para indicar que o *SAHe-CVM+Hp* encontrou uma solução estritamente melhor que a solução encontrada pela *Hp* pura ou, ainda, uma solução contendo o mesmo número de vértices, mas encontrada em um tempo menor pelo *SAHe-CVM+Hp*. No primeiro caso, destacou-se apenas o número de vértices. No segundo caso, o tempo também foi destacado.

Tabela 5.5: Resultados do *SAHe-CVM* e  $H_p = gwcvm$ .

| Família             | Instância     | Vértices | Arestas | <i>SAHe-CVM+gwcvm</i> |               | <i>gwcvm</i> |          |
|---------------------|---------------|----------|---------|-----------------------|---------------|--------------|----------|
|                     |               |          |         | Cobertura             | Tempo(m)      | Cobertura    | Tempo(m) |
| <i>Cn.x</i>         | C125.9        | 125      | 787     | <b>91</b>             | <b>0.042</b>  | 91           | 0.682    |
|                     | C250.9        | 250      | 3141    | <b>206</b>            | 16.032        | 208          | 8.939    |
|                     | C500.9        | 500      | 12418   | <b>450</b>            | 153.739       | 451          | 132.932  |
|                     | C1000.9       | 1000     | 49421   | <b>944</b>            | 71.264        | 946          | 2.612    |
|                     | C2000.5       | 2000     | 999164  | <b>1986</b>           | <b>79.646</b> | 1986         | 123.434  |
|                     | C2000.9       | 2000     | 199468  | <b>1939</b>           | 57.082        | 1941         | 26.544   |
|                     | C4000.5       | 4000     | 3997732 | 3986                  | 172.973       | 3986         | 1.638    |
| <i>DSJCN.x</i>      | DSJC500.5     | 500      | 63264   | <b>488</b>            | 12.345        | 489          | 0.381    |
|                     | DSJC1000.5    | 1000     | 249674  | <b>987</b>            | <b>2.757</b>  | 987          | 26.608   |
| <i>hammingW-d</i>   | hamming10-4   | 1024     | 89600   | <b>990</b>            | 3.041         | 991          | 25.238   |
| <i>johnsonW-w-d</i> | johnson32-2-4 | 496      | 14880   | 480                   | 0.032         | 480          | 0.001    |
| <i>kellerd</i>      | keller6       | 3361     | 1026582 | <b>3319</b>           | 221.182       | 3321         | 114.238  |
| <i>MANN_ax</i>      | MANN_a81      | 3321     | 6480    | 2240                  | 0.930         | 2240         | 0.617    |
| <i>p_hatn-x</i>     | p_hat500-3    | 500      | 30950   | <b>453</b>            | 49.203        | 456          | 4.309    |
|                     | p_hat700-1    | 700      | 183651  | <b>690</b>            | 68.904        | 691          | 6.136    |
|                     | p_hat700-2    | 700      | 122922  | <b>660</b>            | 26.130        | 661          | 45.998   |
|                     | p_hat700-3    | 700      | 61640   | <b>642</b>            | 47.179        | 646          | 140.313  |
|                     | p_hat1000-1   | 1000     | 377247  | <b>990</b>            | 39.693        | 991          | 6.380    |
|                     | p_hat1000-2   | 1000     | 254701  | <b>960</b>            | <b>15.797</b> | 960          | 62.116   |
|                     | p_hat1000-3   | 1000     | 127754  | <b>940</b>            | 154.780       | 944          | 83.012   |
|                     | p_hat1500-1   | 1500     | 839327  | <b>1490</b>           | <b>19.281</b> | 1490         | 19.628   |
|                     | p_hat1500-2   | 1500     | 555290  | <b>1445</b>           | 105.686       | 1446         | 150.191  |
| <i>sanrn_p</i>      | sanr200_0.9   | 200      | 2037    | <b>158</b>            | 228.260       | 160          | 18.703   |
|                     | sanr400_0.5   | 400      | 39816   | <b>387</b>            | 18.955        | 388          | 24.300   |
|                     | sanr400_0.7   | 400      | 23931   | <b>380</b>            | 16.906        | 381          | 0.387    |

Tabela 5.6: Resultados do *SAHe-CVM* e  $H_p = hcvm$ .

| Família             | Instância     | Vértices | Arestas     | <i>SAHe-CVM+hcvm</i> |                | <i>hcvm</i> |          |
|---------------------|---------------|----------|-------------|----------------------|----------------|-------------|----------|
|                     |               |          |             | Cobertura            | Tempo(m)       | Cobertura   | Tempo(m) |
| <i>Cn.x</i>         | C125.9        | 125      | 787         | <b>91</b>            | <b>0.018</b>   | 91          | 0.022    |
|                     | C250.9        | 250      | 3141        | <b>206</b>           | 27.151         | 207         | 33.147   |
|                     | C500.9        | 500      | 12418       | <b>448</b>           | 5.708          | 449         | 0.158    |
|                     | C1000.9       | 1000     | 49421       | <b>941</b>           | <b>3.942</b>   | 941         | 183.862  |
|                     | C2000.5       | 2000     | 999164      | 1986                 | 20.872         | 1985        | 205.851  |
|                     | C2000.9       | 2000     | 199468      | <b>1934</b>          | 202.645        | 1935        | 27.839   |
|                     | C4000.5       | 4000     | 3997732     | <b>3986</b>          | <b>143.234</b> | 3986        | 174.875  |
| <i>DSJCN.x</i>      | DSJC500.5     | 500      | 63264       | <b>487</b>           | <b>6.181</b>   | 487         | 169.341  |
|                     | DSJC1000.5    | 1000     | 249674      | <b>986</b>           | <b>90.974</b>  | 986         | 241.171  |
| <i>hammingW-d</i>   | hamming10-4   | 1024     | 89600       | 988                  | 111.069        | 988         | 1.055    |
| <i>johnsonW-w-d</i> | johnson32-2-4 | 496      | 14880       | 480                  | 0.032          | 480         | 0.001    |
| <i>kellerd</i>      | keller6       | 3361     | 1026582     | 3322                 | 41.383         | 3321        | 137.409  |
| <i>MANN_ax</i>      | MANN_a81      | 3321     | 6480        | <b>2240</b>          | <b>0.382</b>   | 2240        | 0.715    |
| <i>p_hatn-x</i>     | p_hat500-3    | 500      | 30950       | <b>451</b>           | 3.597          | 452         | 5.728    |
|                     | p_hat700-1    | 700      | 183651      | 689                  | 2.208          | 689         | 2.060    |
|                     | p_hat700-2    | 700      | 122922      | <b>656</b>           | 103.665        | 658         | 223.147  |
|                     | p_hat700-3    | 700      | 61640       | <b>640</b>           | 66.922         | 642         | 2.605    |
|                     | p_hat1000-1   | 1000     | 377247      | 990                  | 17.303         | 990         | 6.384    |
|                     | p_hat1000-2   | 1000     | 254701      | <b>955</b>           | 231.785        | 957         | 8.771    |
|                     | p_hat1000-3   | 1000     | 127754      | <b>937</b>           | 143.357        | 939         | 4.656    |
|                     | p_hat1500-1   | 1500     | 839327      | 1489                 | 73.571         | 1489        | 20.955   |
|                     | p_hat1500-2   | 1500     | 555290      | <b>1438</b>          | 89.805         | 1440        | 137.879  |
| p_hat1500-3         | 1500          | 277006   | <b>1413</b> | 160.109              | 1416           | 6.143       |          |
| <i>sanrn_p</i>      | sanr200_0.9   | 200      | 2037        | <b>158</b>           | 35.405         | 159         | 114.668  |
|                     | sanr400_0.5   | 400      | 39816       | <b>387</b>           | <b>0.394</b>   | 387         | 4.615    |
|                     | sanr400_0.7   | 400      | 23931       | <b>379</b>           | 15.745         | 380         | 7.102    |

Tabela 5.7: Resultados do *SAHe-CVM* e  $Hp = hwcvm$ .

| Família             | Instância     | Vértices | Arestas | <i>SAHe-CVM+hwcvm</i> |               | <i>hwcvm</i> |          |
|---------------------|---------------|----------|---------|-----------------------|---------------|--------------|----------|
|                     |               |          |         | Cobertura             | Tempo(m)      | Cobertura    | Tempo(m) |
| <i>Cn.x</i>         | C125.9        | 125      | 787     | <b>91</b>             | <b>0.130</b>  | 91           | 0.131    |
|                     | C250.9        | 250      | 3141    | <b>207</b>            | 13.434        | 209          | 29.921   |
|                     | C500.9        | 500      | 12418   | <b>450</b>            | 50.512        | 452          | 44.527   |
|                     | C1000.9       | 1000     | 49421   | <b>944</b>            | 50.673        | 946          | 4.437    |
|                     | C2000.5       | 2000     | 999164  | <b>1986</b>           | 40.310        | 1987         | 20.801   |
|                     | C2000.9       | 2000     | 199468  | <b>1939</b>           | <b>31.443</b> | 1939         | 46.475   |
|                     | C4000.5       | 4000     | 3997732 | 3986                  | 143.716       | 3985         | 490.951  |
| <i>DSJCN.x</i>      | DSJC500.5     | 500      | 63264   | <b>487</b>            | 0.637         | 488          | 0.354    |
|                     | DSJC1000.5    | 1000     | 249674  | 987                   | 73.119        | 987          | 17.066   |
| <i>hammingW-d</i>   | hamming10-4   | 1024     | 89600   | <b>990</b>            | <b>10.063</b> | 990          | 107.323  |
| <i>johnsonW-w-d</i> | johnson32-2-4 | 496      | 14880   | 480                   | 0.032         | 480          | 0.000    |
| <i>kellerd</i>      | keller6       | 3361     | 1026582 | <b>3320</b>           | 154.570       | 3321         | 84.543   |
| <i>MANN_ax</i>      | MANN_a81      | 3321     | 6480    | <b>2240</b>           | <b>5.393</b>  | 2240         | 8.661    |
| <i>p_hatn-x</i>     | p_hat500-3    | 500      | 30950   | <b>453</b>            | 88.264        | 455          | 13.665   |
|                     | p_hat700-1    | 700      | 183651  | <b>689</b>            | 173.653       | 690          | 37.282   |
|                     | p_hat700-2    | 700      | 122922  | <b>660</b>            | 14.187        | 661          | 107.743  |
|                     | p_hat700-3    | 700      | 61640   | <b>643</b>            | 13.224        | 645          | 165.579  |
|                     | p_hat1000-1   | 1000     | 377247  | <b>990</b>            | 129.956       | 991          | 6.403    |
|                     | p_hat1000-2   | 1000     | 254701  | <b>959</b>            | 6.448         | 961          | 143.639  |
|                     | p_hat1000-3   | 1000     | 127754  | <b>942</b>            | 107.102       | 944          | 24.010   |
|                     | p_hat1500-1   | 1500     | 839327  | <b>1490</b>           | <b>37.787</b> | 1490         | 193.833  |
|                     | p_hat1500-2   | 1500     | 555290  | <b>1444</b>           | 63.698        | 1446         | 28.078   |
|                     | p_hat1500-3   | 1500     | 277006  | <b>1421</b>           | 115.484       | 1424         | 144.977  |
| <i>sanrn_p</i>      | sanr200_0.9   | 200      | 2037    | <b>158</b>            | 77.743        | 160          | 69.390   |
|                     | sanr400_0.5   | 400      | 39816   | <b>388</b>            | <b>0.193</b>  | 388          | 98.975   |
|                     | sanr400_0.7   | 400      | 23931   | <b>380</b>            | 7.740         | 381          | 6.403    |

O número de instâncias destacadas gera a razão apresentada na linha “Final” da Tabela 5.8. As demais linhas apresentam a mesma informação para diversas fatias de tempo; cada linha contém o tempo limite (em minutos) e a razão de ganho do *SAHe-CVM+Hp* para o conjunto de 1996. Cada coluna contém a razão de ganho do *SAHe-CVM+Hp* combinado com alguma heurística de pós-processamento (*gcv*, *gwcvm*, *hcv*, *hwcvm*), aplicada sobre um conjunto de instâncias.

Algumas entradas nas tabelas estão destacadas (em negrito) para indicar que o *SAHe-CVM+Hp* obteve um desempenho superior à *heurística pura* em mais de 50% das instâncias. Verifica-se, claramente, que o *SAHe-CVM+Hp* obteve um bom desempenho sobre uma fração significativa das instâncias. Uma comparação mais detalhada dos valores presentes na linha “Final” da Tabela 5.8 é apresentada através dos gráficos da Figura 5.1, onde é possível notar que menos de 20% das instâncias o resultado encontrado pela aplicação da *Hp-pura* foi superior ao resultado da sua aplicação em conjunto com o método *SAHe-CVM*. Também pode ser observado que as variantes do *SAHe-CVM* que utilizam o mecanismo de *random-walk* (*Hp’s gvcvm* e *hwcvm*) apresentaram um desempenho superior ao apresentado pelas variantes que não utilizam esse mecanismo, indicando que o mecanismo de *random-walk* potencializa o desempenho do método criado.

Tabela 5.8: Taxa de sucesso do *SAHe-CVM* por tempo limite, sobre as instâncias do conjunto de 1996 sem valor ótimo conhecido.

| Conjunto de 1996 |                     |                      |                     |                      |
|------------------|---------------------|----------------------|---------------------|----------------------|
| Tempo(m)         | <i>SAHe-CVM+gcv</i> | <i>SAHe-CVM+gvcv</i> | <i>SAHe-CVM+hcv</i> | <i>SAHe-CVM+hvcv</i> |
| 10               | <b>0.769</b>        | <b>0.808</b>         | <b>0.731</b>        | <b>0.923</b>         |
| 20               | <b>0.731</b>        | <b>0.885</b>         | <b>0.731</b>        | <b>0.846</b>         |
| 30               | <b>0.731</b>        | <b>0.885</b>         | <b>0.692</b>        | <b>0.846</b>         |
| 40               | <b>0.731</b>        | <b>0.885</b>         | <b>0.692</b>        | <b>0.808</b>         |
| 50               | <b>0.731</b>        | <b>0.885</b>         | <b>0.731</b>        | <b>0.808</b>         |
| 60               | <b>0.731</b>        | <b>0.885</b>         | <b>0.731</b>        | <b>0.808</b>         |
| 70               | <b>0.731</b>        | <b>0.885</b>         | <b>0.731</b>        | <b>0.846</b>         |
| 80               | <b>0.731</b>        | <b>0.885</b>         | <b>0.731</b>        | <b>0.846</b>         |
| 90               | <b>0.731</b>        | <b>0.885</b>         | <b>0.731</b>        | <b>0.846</b>         |
| 100              | <b>0.731</b>        | <b>0.885</b>         | <b>0.731</b>        | <b>0.846</b>         |
| 110              | <b>0.769</b>        | <b>0.885</b>         | <b>0.731</b>        | <b>0.846</b>         |
| 120              | <b>0.769</b>        | <b>0.885</b>         | <b>0.731</b>        | <b>0.846</b>         |
| 130              | <b>0.808</b>        | <b>0.885</b>         | <b>0.731</b>        | <b>0.885</b>         |
| 140              | <b>0.808</b>        | <b>0.885</b>         | <b>0.692</b>        | <b>0.885</b>         |
| 150              | <b>0.808</b>        | <b>0.885</b>         | <b>0.692</b>        | <b>0.885</b>         |
| 160              | <b>0.846</b>        | <b>0.885</b>         | <b>0.692</b>        | <b>0.885</b>         |
| 170              | <b>0.808</b>        | <b>0.885</b>         | <b>0.692</b>        | <b>0.885</b>         |
| 180              | <b>0.808</b>        | <b>0.885</b>         | <b>0.692</b>        | <b>0.923</b>         |
| 190              | <b>0.769</b>        | <b>0.885</b>         | <b>0.692</b>        | <b>0.923</b>         |
| 200              | <b>0.769</b>        | <b>0.885</b>         | <b>0.692</b>        | <b>0.923</b>         |
| 210              | <b>0.769</b>        | <b>0.885</b>         | <b>0.731</b>        | <b>0.923</b>         |
| 220              | <b>0.769</b>        | <b>0.885</b>         | <b>0.731</b>        | <b>0.923</b>         |
| 230              | <b>0.769</b>        | <b>0.885</b>         | <b>0.731</b>        | <b>0.923</b>         |
| 240              | <b>0.769</b>        | <b>0.885</b>         | <b>0.731</b>        | <b>0.923</b>         |
| Final            | <b>0.769</b>        | <b>0.885</b>         | <b>0.731</b>        | <b>0.885</b>         |

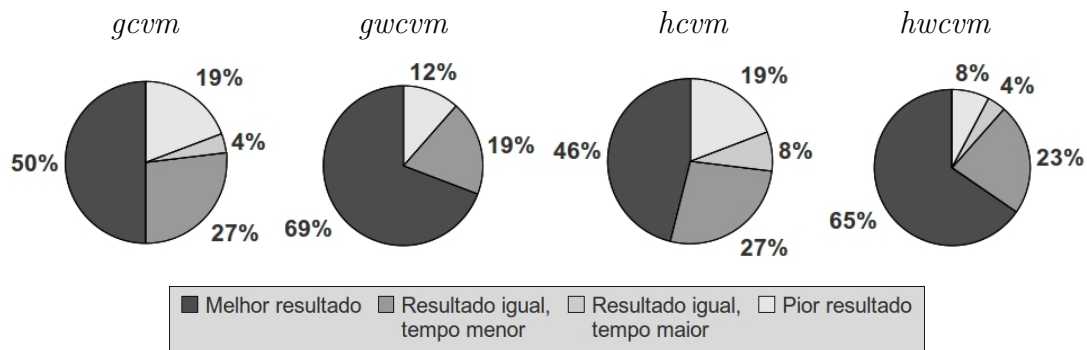


Figura 5.1: Comparação dos valores obtidos entre *SAHe-CVM+Hp's* e *Hp's-puras*, sobre as instâncias do conjunto de 1996, cujo valor ótimo é desconhecido.

Ao efetuar uma análise semelhante sobre todas as instâncias de ambos os conjuntos (incluindo resultados omitidos nas Tabelas 5.4 até 5.7), tem-se as razões presentes na Tabela 5.9 e os gráficos das Figuras 5.2 e 5.3, demonstrando que o *SAHe-CVM+Hp*, combinado com alguma heurística de pós-processamento ( $Hp_1 = gcv$ ,  $Hp_2 = gvcv$ ,  $Hp_3 = hcv$ ,  $Hp_4 = hvcv$ ), obteve um desempenho um pouco inferior sobre todas as instâncias do conjunto de 1996. Entretanto, obteve um excelente desempenho, próximo de 90%, sobre todas as instâncias do conjunto de 2010. Novamente, pode ser observado que as variantes do *SAHe-CVM* com *random-walk* produziram melhores resultados que as demais variantes.

Tabela 5.9: Taxa de sucesso do *SAHe-CVM* por tempo limite, sobre todas as instâncias do conjunto de 1996 e todas as instâncias do conjunto de 2010.

| Tempo(m) | Conjunto de 1996 |        |        |        | Conjunto de 2010 |        |        |        |
|----------|------------------|--------|--------|--------|------------------|--------|--------|--------|
|          | $Hp_1$           | $Hp_2$ | $Hp_3$ | $Hp_4$ | $Hp_1$           | $Hp_2$ | $Hp_3$ | $Hp_4$ |
| 10       | 0.575            | 0.675  | 0.600  | 0.688  | 0.780            | 0.756  | 0.683  | 0.707  |
| 20       | 0.587            | 0.700  | 0.637  | 0.650  | 0.805            | 0.780  | 0.780  | 0.756  |
| 30       | 0.587            | 0.712  | 0.625  | 0.675  | 0.805            | 0.829  | 0.829  | 0.780  |
| 40       | 0.587            | 0.712  | 0.650  | 0.675  | 0.854            | 0.878  | 0.829  | 0.854  |
| 50       | 0.600            | 0.725  | 0.663  | 0.663  | 0.854            | 0.878  | 0.878  | 0.878  |
| 60       | 0.600            | 0.725  | 0.663  | 0.663  | 0.854            | 0.927  | 0.878  | 0.902  |
| 70       | 0.613            | 0.725  | 0.663  | 0.675  | 0.854            | 0.927  | 0.878  | 0.927  |
| 80       | 0.613            | 0.725  | 0.663  | 0.675  | 0.878            | 0.927  | 0.878  | 0.927  |
| 90       | 0.625            | 0.725  | 0.663  | 0.675  | 0.878            | 0.927  | 0.878  | 0.951  |
| 100      | 0.625            | 0.725  | 0.663  | 0.675  | 0.878            | 0.927  | 0.878  | 0.951  |
| 110      | 0.637            | 0.725  | 0.675  | 0.675  | 0.878            | 0.927  | 0.878  | 0.927  |
| 120      | 0.637            | 0.725  | 0.675  | 0.675  | 0.878            | 0.927  | 0.878  | 0.927  |
| 130      | 0.650            | 0.712  | 0.675  | 0.688  | 0.878            | 0.927  | 0.878  | 0.927  |
| 140      | 0.650            | 0.712  | 0.663  | 0.688  | 0.878            | 0.927  | 0.878  | 0.927  |
| 150      | 0.650            | 0.712  | 0.663  | 0.675  | 0.878            | 0.927  | 0.878  | 0.927  |
| 160      | 0.663            | 0.712  | 0.663  | 0.675  | 0.878            | 0.927  | 0.878  | 0.927  |
| 170      | 0.650            | 0.712  | 0.663  | 0.675  | 0.878            | 0.927  | 0.878  | 0.927  |
| 180      | 0.650            | 0.712  | 0.663  | 0.700  | 0.878            | 0.927  | 0.902  | 0.927  |
| 190      | 0.637            | 0.712  | 0.663  | 0.700  | 0.878            | 0.927  | 0.902  | 0.927  |
| 200      | 0.637            | 0.725  | 0.663  | 0.712  | 0.878            | 0.927  | 0.902  | 0.927  |
| 210      | 0.637            | 0.725  | 0.675  | 0.712  | 0.878            | 0.951  | 0.902  | 0.927  |
| 220      | 0.637            | 0.738  | 0.675  | 0.712  | 0.878            | 0.951  | 0.902  | 0.927  |
| 230      | 0.637            | 0.738  | 0.675  | 0.712  | 0.878            | 0.951  | 0.902  | 0.927  |
| 240      | 0.637            | 0.738  | 0.675  | 0.712  | 0.878            | 0.951  | 0.902  | 0.927  |
| Final    | 0.637            | 0.738  | 0.675  | 0.700  | 0.878            | 0.951  | 0.902  | 0.927  |

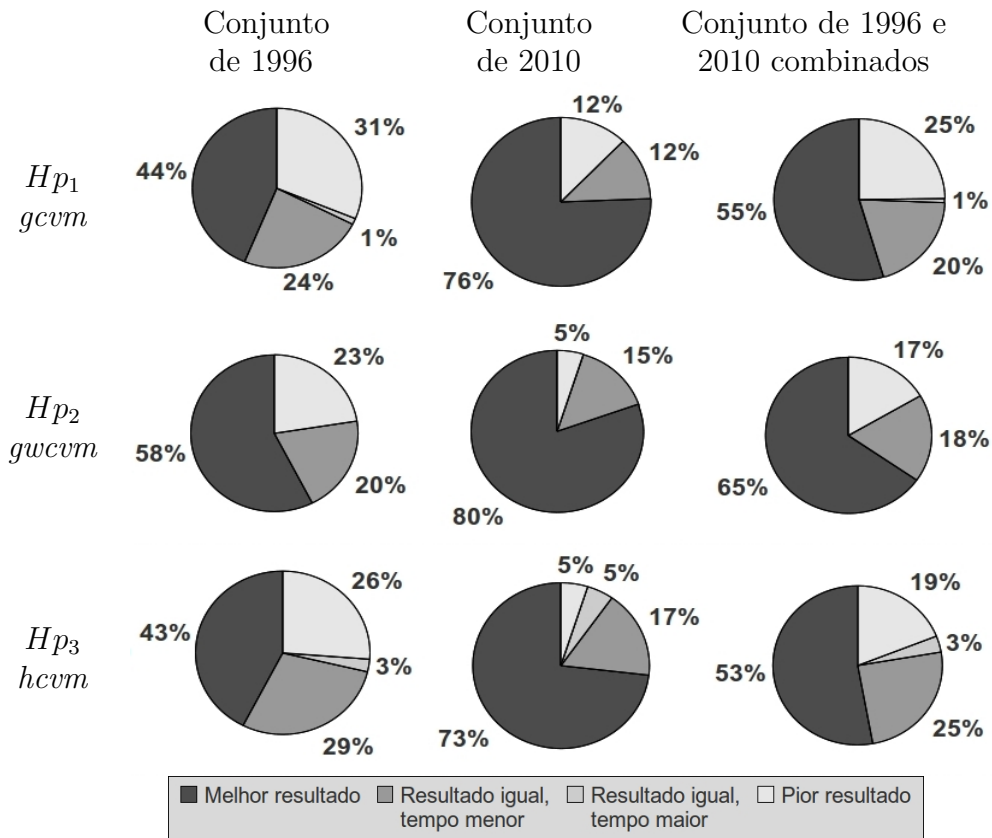


Figura 5.2: Comparação dos valores obtidos entre *SAHe-SAT+Hp's* e *Hp's-puras*, sobre todas as 80 instâncias do conjunto de 1996 e todas as 41 de 2010.

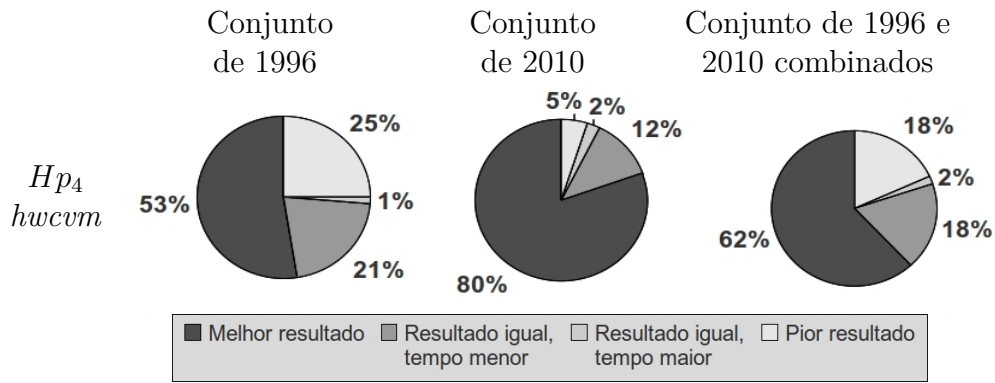


Figura 5.3: Comparação dos valores obtidos entre *SAHe-SAT+Hp's* e *Hp's-puras*, sobre todas as 80 instâncias do conjunto de 1996 e todas as 41 de 2010. (continuação)

Os resultados sobre o conjunto total de instâncias, que deram origem as razões da Tabela 5.9 e aos gráficos das Figuras 5.2 e 5.3, estão descritos nas Tabelas D.1 a D.12, do Anexo D.

Por fim, o *Second DIMACS Challenge* [22] descreve um conjunto de heurísticas para o problema CVM e seus equivalentes. Essas heurísticas servem, mesmo que de uma forma indireta, como base de referência para se fazer comparações sobre o desempenho dos resultados. Por este motivo, os resultados obtidos pelo uso do método *SAHe-CVM* foram comparados aos resultados obtidos por algumas heurísticas presentes no *Second DIMACS Challenge* e por outras duas heurísticas, *OCH* [48] e *WAO* [23], descritos a seguir.

### Heurística *Squeeze*

A Tabela 5.10 apresenta os resultados obtidos pela heurística *Squeeze* [40] e pelo método *SAHe-CVM* sobre instâncias do conjunto de 1996. São apresentadas apenas as instâncias (grafos), para as quais a heurística *Squeeze* encontrou uma resposta. Cada linha da tabela corresponde a uma instância, com uma linha horizontal separando as famílias. Para cada instância, está descrito o valor ótimo (quando conhecido), bem como os resultados encontrados pela heurística *Squeeze* e pelas quatro variantes do *SAHe-CVM*, a citar: *SAHe-CVM+gcv*, *SAHe-CVM+gwcvm*, *SAHe-CVM+hcv* e *SAHe-CVM+hwcv*. Estes resultados são relativos ao tamanho da cobertura de vértices encontrada pelo método em questão. Algumas entradas na tabela estão destacadas (em negrito) para indicar que uma determinada variante



do *SAHe-CVM* encontrou uma solução melhor ou igual à solução encontrada pelo *Squeeze*.

Tabela 5.10: Comparação de resultados entre *Squeeze* e *SAHe-CVM*.

| Família             | Grafo          | Ótimo | <i>Squeeze</i> | <i>SAHe-CVM+</i> |              |             |              |
|---------------------|----------------|-------|----------------|------------------|--------------|-------------|--------------|
|                     |                |       |                | <i>gcvm</i>      | <i>gwcvm</i> | <i>hcvm</i> | <i>hwcvm</i> |
| <i>brockn_x</i>     | brock200_1     | 179   | 179            | <b>179</b>       | <b>179</b>   | <b>179</b>  | <b>179</b>   |
|                     | brock200_2     | 188   | 188            | <b>188</b>       | <b>188</b>   | <b>188</b>  | <b>188</b>   |
|                     | brock200_3     | 185   | 185            | 186              | <b>185</b>   | 186         | <b>185</b>   |
|                     | brock200_4     | 183   | 183            | <b>183</b>       | <b>183</b>   | <b>183</b>  | <b>183</b>   |
| <i>Cn.d</i>         | C125.9         |       | 91             | <b>91</b>        | <b>91</b>    | <b>91</b>   | <b>91</b>    |
|                     | C250.9         |       | 206            | <b>206</b>       | <b>206</b>   | <b>206</b>  | 207          |
| <i>c-fatn-c</i>     | c-fat200-1     | 188   | 188            | <b>188</b>       | <b>188</b>   | <b>188</b>  | <b>188</b>   |
|                     | c-fat200-2     | 176   | 176            | <b>176</b>       | <b>176</b>   | <b>176</b>  | <b>176</b>   |
|                     | c-fat200-5     | 142   | 142            | <b>142</b>       | <b>142</b>   | <b>142</b>  | <b>142</b>   |
|                     | c-fat500-1     | 486   | 486            | <b>486</b>       | <b>486</b>   | <b>486</b>  | <b>486</b>   |
|                     | c-fat500-2     | 474   | 474            | <b>474</b>       | <b>474</b>   | <b>474</b>  | <b>474</b>   |
|                     | c-fat500-5     | 436   | 436            | <b>436</b>       | <b>436</b>   | <b>436</b>  | <b>436</b>   |
|                     | c-fat500-10    | 374   | 374            | <b>374</b>       | <b>374</b>   | <b>374</b>  | <b>374</b>   |
| <i>DSJcn_x</i>      | DSJC500.5      |       | 487            | <b>487</b>       | 488          | <b>487</b>  | <b>487</b>   |
| <i>gen_n_pp_c</i>   | gen200_p0.9_44 | 156   | 156            | <b>156</b>       | <b>156</b>   | <b>156</b>  | 157          |
|                     | gen200_p0.9_55 | 145   | 145            | <b>145</b>       | <b>145</b>   | <b>145</b>  | <b>145</b>   |
| <i>hammingW-d</i>   | hamming6-2     | 32    | 32             | <b>32</b>        | <b>32</b>    | <b>32</b>   | <b>32</b>    |
|                     | hamming6-4     | 60    | 60             | <b>60</b>        | <b>60</b>    | <b>60</b>   | <b>60</b>    |
|                     | hamming8-2     | 128   | 128            | <b>128</b>       | <b>128</b>   | <b>128</b>  | <b>128</b>   |
|                     | hamming8-4     | 240   | 240            | <b>240</b>       | <b>240</b>   | <b>240</b>  | <b>240</b>   |
|                     | hamming10-2    | 512   | 512            | <b>512</b>       | 530          | <b>512</b>  | 530          |
| <i>johnsonW-w-d</i> | johnson8-2-4   | 24    | 24             | <b>24</b>        | <b>24</b>    | <b>24</b>   | <b>24</b>    |
|                     | johnson8-4-4   | 56    | 56             | <b>56</b>        | <b>56</b>    | <b>56</b>   | <b>56</b>    |
|                     | johnson16-2-4  | 112   | 112            | <b>112</b>       | <b>112</b>   | <b>112</b>  | <b>112</b>   |
|                     | johnson32-2-4  | 480   | 480            | <b>480</b>       | <b>480</b>   | <b>480</b>  | <b>480</b>   |
| <i>kellerd</i>      | keller4        | 160   | 160            | <b>160</b>       | <b>160</b>   | <b>160</b>  | <b>160</b>   |
| <i>MANN_ax</i>      | MANN_a9        | 29    | 29             | <b>29</b>        | <b>29</b>    | <b>29</b>   | <b>29</b>    |
|                     | MANN_a27       | 252   | 252            | 255              | 256          | 255         | 256          |
| <i>p_hatn-x</i>     | p_hat300-1     | 292   | 292            | <b>292</b>       | <b>292</b>   | <b>292</b>  | <b>292</b>   |
|                     | p_hat300-2     | 275   | 275            | <b>275</b>       | <b>275</b>   | <b>275</b>  | <b>275</b>   |
|                     | p_hat300-3     | 264   | 264            | <b>264</b>       | 265          | <b>264</b>  | 265          |
|                     | p_hat500-1     | 491   | 491            | <b>491</b>       | <b>491</b>   | <b>491</b>  | <b>491</b>   |
|                     | p_hat500-2     | 464   | 464            | <b>464</b>       | 466          | <b>464</b>  | 467          |
|                     | p_hat700-1     |       | 689            | <b>689</b>       | 690          | <b>689</b>  | <b>689</b>   |
| <i>sann_f_x</i>     | san200_0.7_1   | 170   | 170            | <b>170</b>       | <b>170</b>   | 184         | <b>170</b>   |
|                     | san200_0.7_2   | 182   | 182            | 186              | 186          | 186         | 186          |
|                     | san200_0.9_1   | 130   | 130            | <b>130</b>       | <b>130</b>   | <b>130</b>  | <b>130</b>   |
|                     | san200_0.9_2   | 140   | 140            | <b>140</b>       | <b>140</b>   | <b>140</b>  | <b>140</b>   |
|                     | san200_0.9_3   | 156   | 156            | <b>156</b>       | <b>156</b>   | <b>156</b>  | <b>156</b>   |
|                     | san400_0.9_1   | 300   | 300            | <b>300</b>       | <b>300</b>   | <b>300</b>  | <b>300</b>   |
| <i>sanrn_p</i>      | sanr200_0.7    | 182   | 182            | <b>182</b>       | <b>182</b>   | <b>182</b>  | <b>182</b>   |
|                     | sanr200_0.9    |       | 159            | <b>158</b>       | <b>158</b>   | <b>158</b>  | <b>158</b>   |
|                     | sanr400_0.7    |       | 380            | <b>379</b>       | <b>380</b>   | <b>379</b>  | <b>380</b>   |

Percebe-se, claramente, que a maior parte dos resultados do método *SAHe-CVM* é igual ou superior aos resultados encontrados pela heurística *Squeeze*. Com o objetivo de deixar esta relação de proporcionalidade mais clara, foram gerados os gráficos da Figura 5.4. Eles, demonstram proporcionalmente a relação entre o número coberturas encontradas pelo método *SAHe-CVM+Hp* que são melhores (legenda “Melhor”), iguais (legenda “Igual”) ou piores (legenda “Pior”) que as coberturas encontradas pelo método *Squeeze*. Os primeiros quatro gráficos ilustram este comparativo para cada uma das quatro variantes do método *SAHe-CVM*; o quinto

gráfico compara, também proporcionalmente, os resultados do método *Squeeze* com os melhores resultados encontrados pelas quatro variantes do método *SAHe-CVM*.

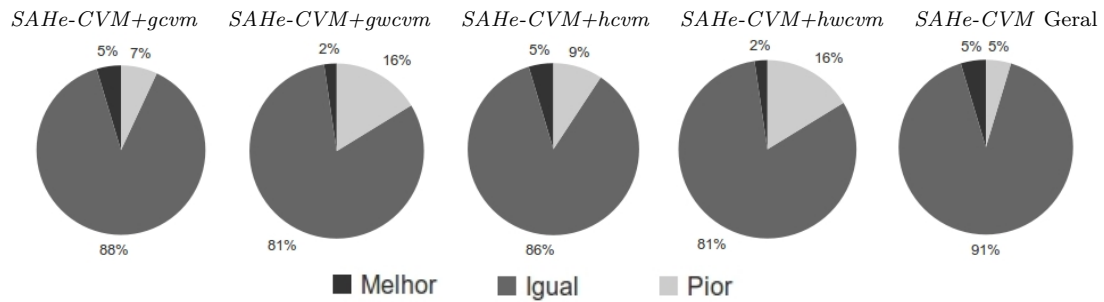


Figura 5.4: Comparação dos valores obtidos entre *Squeeze* e *SAHe-CVM*.

Nota-se que, na totalidade dos gráficos, o maior número de ocorrências corresponde a valores iguais para as respostas dos métodos. Existe, também, um grupo pequeno de instâncias em que o método *Squeeze* obteve um resultado melhor que o método *SAHe-CVM*, e um número ainda menor de instâncias em que o *SAHe-CVM* obteve melhores resultados.

### Heurística *MIPO*

A Tabela 5.11 e a Figura 5.5 descrevem informações análogas às apresentadas para a heurística *Squeeze*. Na Tabela 5.11, é exibida uma comparação entre os resultados apresentados pela heurística *MIPO* [41] e os resultados obtidos nos testes experimentais das variações do *SAHe-CVM*, destacando-se (em negrito) os casos em que este último método encontrou resultados iguais ou melhores que o primeiro. Na Figura 5.5, os gráficos demonstram, proporcionalmente, a relação entre o número de coberturas encontradas pelas variantes do método *SAHe-CVM* que são melhores, iguais e piores que as encontradas pelo método *MIPO*.

É possível notar que o método *SAHe-CVM* foi igual ou inferior à heurística *MIPO* na totalidade dos casos. Importante destacar que mesmo nesta condição, o tamanho da cobertura de vértices encontrada pelos métodos foi o mesmo em mais de 80% dos casos.

Tabela 5.11: Comparação de resultados entre *MIPO* e *SAHe-CVM*.

| Familia                     | Grafo                 | Ótimo | <i>B&amp;C</i> | <i>SAHe-CVM</i> |            |            |            |
|-----------------------------|-----------------------|-------|----------------|-----------------|------------|------------|------------|
|                             |                       |       |                | <i>gcv</i>      | <i>gwc</i> | <i>hcv</i> | <i>hwc</i> |
| <i>brockn_x</i>             | <i>brock200_2</i>     | 188   | 188            | <b>188</b>      | <b>188</b> | <b>188</b> | <b>188</b> |
| <i>Cn.d</i>                 | <i>C125.9</i>         |       | 91             | <b>91</b>       | <b>91</b>  | <b>91</b>  | <b>91</b>  |
| <i>c-fatn-c</i>             | <i>c-fat200-1</i>     | 188   | 188            | <b>188</b>      | <b>188</b> | <b>188</b> | <b>188</b> |
|                             | <i>c-fat200-2</i>     | 176   | 176            | <b>176</b>      | <b>176</b> | <b>176</b> | <b>176</b> |
|                             | <i>c-fat200-5</i>     | 142   | 142            | <b>142</b>      | <b>142</b> | <b>142</b> | <b>142</b> |
| <i>gen<sub>n</sub>-pp-c</i> | <i>gen200_p0.9_44</i> | 156   | 156            | <b>156</b>      | <b>156</b> | <b>156</b> | 157        |
|                             | <i>gen200_p0.9_55</i> | 145   | 145            | <b>145</b>      | <b>145</b> | <b>145</b> | <b>145</b> |
| <i>hammingW-d</i>           | <i>hamming6-2</i>     | 32    | 32             | <b>32</b>       | <b>32</b>  | <b>32</b>  | <b>32</b>  |
|                             | <i>hamming6-4</i>     | 60    | 60             | <b>60</b>       | <b>60</b>  | <b>60</b>  | <b>60</b>  |
|                             | <i>hamming8-2</i>     | 128   | 128            | <b>128</b>      | <b>128</b> | <b>128</b> | <b>128</b> |
|                             | <i>hamming8-4</i>     | 240   | 240            | <b>240</b>      | <b>240</b> | <b>240</b> | <b>240</b> |
| <i>johnsonW-w-d</i>         | <i>johnson8-2-4</i>   | 24    | 24             | <b>24</b>       | <b>24</b>  | <b>24</b>  | <b>24</b>  |
|                             | <i>johnson8-4-4</i>   | 56    | 56             | <b>56</b>       | <b>56</b>  | <b>56</b>  | <b>56</b>  |
|                             | <i>johnson16-2-4</i>  | 112   | 112            | <b>112</b>      | <b>112</b> | <b>112</b> | <b>112</b> |
| <i>kellerd</i>              | <i>keller4</i>        | 160   | 160            | <b>160</b>      | <b>160</b> | <b>160</b> | <b>160</b> |
| <i>p_hatn-x</i>             | <i>p_hat300-1</i>     | 292   | 292            | <b>292</b>      | <b>292</b> | <b>292</b> | <b>292</b> |
| <i>sann_f_x</i>             | <i>san200_0.7_1</i>   | 170   | 170            | <b>170</b>      | <b>170</b> | 184        | <b>170</b> |
|                             | <i>san200_0.7_2</i>   | 182   | 182            | 186             | 186        | 186        | 186        |
|                             | <i>san200_0.9_1</i>   | 130   | 130            | <b>130</b>      | <b>130</b> | <b>130</b> | <b>130</b> |
|                             | <i>san200_0.9_2</i>   | 140   | 140            | <b>140</b>      | <b>140</b> | <b>140</b> | <b>140</b> |
|                             | <i>san200_0.9_3</i>   | 156   | 156            | <b>156</b>      | <b>156</b> | <b>156</b> | <b>156</b> |
|                             | <i>san400_0.5_1</i>   | 387   | 387            | 392             | 392        | 392        | 392        |
|                             | <i>san400_0.7_2</i>   | 370   | 370            | 384             | 383        | 384        | 384        |

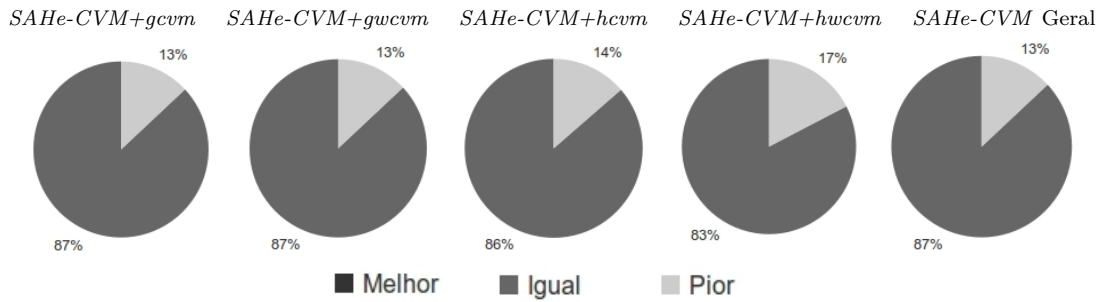


Figura 5.5: Comparação dos valores obtidos entre *MIPO* e *SAHe-CVM*.

### Heurística *RB-Clique*

Em caso análogo, a Tabela 5.12 e a Figura 5.6 descrevem um comparativo dos resultados obtidos pela heurística *RB-clique* [43] e pelas as variantes do *SAHe-CVM*.

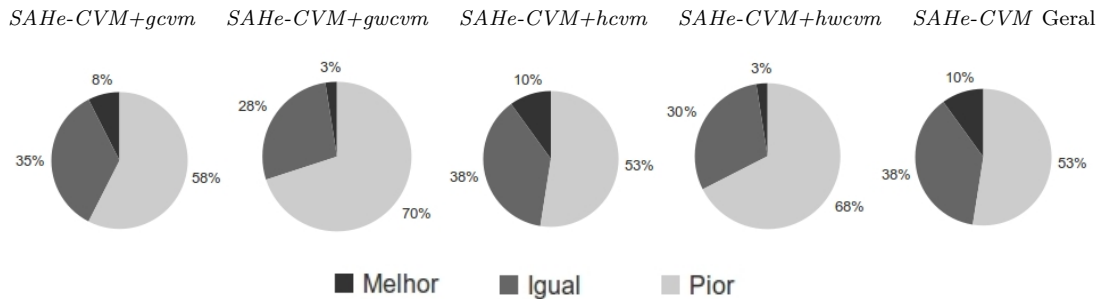


Figura 5.6: Comparação dos valores obtidos entre *RB-clique* e *SAHe-CVM*.

Tabela 5.12: Comparação de resultados entre *RB-clique* e *SAHe-CVM*.

| Família                               | Grafo                   | Ótimo | <i>RB-Clique</i> | <i>SAHe-CVM</i> |             |            |             |
|---------------------------------------|-------------------------|-------|------------------|-----------------|-------------|------------|-------------|
|                                       |                         |       |                  | <i>gcv</i>      | <i>gwcv</i> | <i>hcv</i> | <i>hwcv</i> |
| <i>brockn<sub>x</sub></i>             | brock200_2              | 188   | 188              | <b>188</b>      | <b>188</b>  | <b>188</b> | <b>188</b>  |
|                                       | brock200_4              | 183   | 183              | <b>183</b>      | <b>183</b>  | <b>183</b> | <b>183</b>  |
|                                       | brock400_2              | 371   | 375              | 376             | 376         | <b>375</b> | 376         |
|                                       | brock400_4              | 367   | 367              | 375             | 377         | 376        | 375         |
|                                       | brock800_2              | 776   | 779              | 780             | 781         | 781        | 781         |
|                                       | brock800_4              | 774   | 779              | 781             | 781         | 781        | 781         |
| <i>Cn.d</i>                           | C125.9                  |       | 91               | <b>91</b>       | <b>91</b>   | <b>91</b>  | <b>91</b>   |
|                                       | C250.9                  |       | 206              | <b>206</b>      | <b>206</b>  | <b>206</b> | 207         |
|                                       | C500.9                  |       | 445              | 448             | 450         | 448        | 450         |
|                                       | C1000.9                 |       | 935              | 941             | 944         | 941        | 944         |
|                                       | C2000.5                 |       | 1985             | 1986            | 1986        | 1986       | 1986        |
|                                       | C2000.9                 |       | 1926             | 1935            | 1939        | 1934       | 1939        |
|                                       | C4000.5                 |       | 3982             | 3985            | 3986        | 3986       | 3986        |
|                                       |                         |       |                  |                 |             |            |             |
| <i>DSJcn<sub>x</sub></i>              | DSJC500.5               |       | 487              | <b>487</b>      | 488         | <b>487</b> | <b>487</b>  |
|                                       | DSJC1000.5              |       | 986              | <b>986</b>      | 987         | <b>986</b> | 987         |
| <i>gen<sub>n</sub>_pp<sub>c</sub></i> | gen200_p0.9_44          | 156   | 158              | <b>156</b>      | <b>156</b>  | <b>156</b> | <b>157</b>  |
|                                       | gen200_p0.9_55          | 145   | 145              | <b>145</b>      | <b>145</b>  | <b>145</b> | <b>145</b>  |
|                                       | gen400_p0.9_55          | 345   | 348              | 351             | 353         | 351        | 352         |
|                                       | gen400_p0.9_65          | 335   | 340              | <b>340</b>      | 348         | <b>339</b> | 344         |
|                                       | gen400_p0.9_75          | 325   | 326              | <b>325</b>      | 335         | <b>325</b> | 334         |
| <i>hamming<sub>W-d</sub></i>          | hamming8-4              | 240   | 240              | <b>240</b>      | <b>240</b>  | <b>240</b> | <b>240</b>  |
|                                       | hamming10-4             |       | 984              | 988             | 990         | 988        | 990         |
| <i>keller<sub>d</sub></i>             | keller4                 | 160   | 160              | <b>160</b>      | <b>160</b>  | <b>160</b> | <b>160</b>  |
|                                       | keller5                 | 749   | 749              | 753             | 753         | 754        | 752         |
|                                       | keller6                 |       | 3307             | 3320            | 3319        | 3322       | 3320        |
| <i>MANN<sub>ax</sub></i>              | MANN_a27                | 252   | 252              | 255             | 256         | 255        | 256         |
|                                       | MANN_a45                | 690   | 691              | 702             | 702         | 702        | 703         |
|                                       | MANN_a81                |       | 2224             | 2240            | 2240        | 2240       | 2240        |
| <i>p<sub>hat</sub>n<sub>x</sub></i>   | p <sub>hat</sub> 300-1  | 292   | 292              | <b>292</b>      | <b>292</b>  | <b>292</b> | <b>292</b>  |
|                                       | p <sub>hat</sub> 300-2  | 275   | 275              | <b>275</b>      | <b>275</b>  | <b>275</b> | <b>275</b>  |
|                                       | p <sub>hat</sub> 300-3  | 264   | 265              | <b>264</b>      | <b>265</b>  | <b>264</b> | <b>265</b>  |
|                                       | p <sub>hat</sub> 700-1  |       | 689              | <b>689</b>      | 690         | <b>689</b> | <b>689</b>  |
|                                       | p <sub>hat</sub> 700-2  |       | 656              | 657             | 660         | <b>656</b> | 660         |
|                                       | p <sub>hat</sub> 700-3  |       | 638              | 640             | 642         | 640        | 643         |
|                                       | p <sub>hat</sub> 1000-1 |       | 990              | <b>990</b>      | <b>990</b>  | <b>990</b> | <b>990</b>  |
|                                       | p <sub>hat</sub> 1000-2 |       | 954              | 955             | 960         | 955        | 959         |
|                                       | p <sub>hat</sub> 1000-3 |       | 935              | 937             | 940         | 937        | 942         |
|                                       | p <sub>hat</sub> 1500-1 |       | 1488             | 1489            | 1490        | 1489       | 1490        |
|                                       | p <sub>hat</sub> 1500-2 |       | 1436             | 1440            | 1445        | 1438       | 1444        |
|                                       | p <sub>hat</sub> 1500-3 |       | 1407             | 1412            | 1423        | 1413       | 1421        |

Diferente do que ocorreu com as heurísticas *Squeeze* e *MIPO*, a heurística *RB-clique* apresentou melhores resultados que a totalidade das variantes do método *SAHe-CVM*, para uma quantidade bem expressiva das instâncias (mais de 50%). Além disso, nota-se que esta superioridade é ainda maior sobre as instâncias em que o ótimo é desconhecido (em 17 de 21 instâncias, ou em 80% delas).

### Heurísticas *XSD*

JAGOTA e SANCHIS [45] apresentam os resultados encontrados por uma família de heurísticas, chamadas aqui de *XSD*, sobre o conjunto de 1996. A Tabela 5.13 e a Figura 5.7 apresentam informações análogas às tabelas e figuras anteriores, considerando o melhor resultado encontrado por todo o conjunto de heurísticas *XSD*.

Tabela 5.13: Comparação de resultados entre *XSD* e *SAHe-CVM*.

| Família         | Grafo         | Ótimo | <i>XSD</i> | <i>SAHe-CVM</i> |              |            |              |
|-----------------|---------------|-------|------------|-----------------|--------------|------------|--------------|
|                 |               |       |            | <i>gcv</i>      | <i>gucvm</i> | <i>hcv</i> | <i>hucvm</i> |
| brock $n_x$     | brock200_1    | 179   | 180        | <b>179</b>      | <b>179</b>   | <b>179</b> | <b>179</b>   |
|                 | brock200_2    | 188   | 190        | <b>188</b>      | <b>188</b>   | <b>188</b> | <b>188</b>   |
|                 | brock200_3    | 185   | 185        | 186             | <b>185</b>   | 186        | <b>185</b>   |
|                 | brock200_4    | 183   | 184        | <b>183</b>      | <b>183</b>   | <b>183</b> | <b>183</b>   |
|                 | brock400_1    | 373   | 376        | <b>376</b>      | <b>376</b>   | <b>375</b> | 377          |
|                 | brock400_2    | 371   | 376        | <b>376</b>      | <b>376</b>   | <b>375</b> | <b>376</b>   |
|                 | brock400_3    | 369   | 376        | <b>376</b>      | <b>376</b>   | <b>375</b> | <b>376</b>   |
|                 | brock400_4    | 367   | 376        | <b>375</b>      | 377          | <b>376</b> | <b>375</b>   |
|                 | brock800_1    | 777   | 780        | <b>780</b>      | 781          | <b>780</b> | 781          |
|                 | brock800_2    | 776   | 781        | <b>780</b>      | <b>781</b>   | <b>781</b> | <b>781</b>   |
| c-fat $n_c$     | brock800_3    | 775   | 781        | <b>781</b>      | 782          | <b>780</b> | <b>780</b>   |
|                 | brock800_4    | 774   | 781        | <b>781</b>      | <b>781</b>   | <b>781</b> | <b>781</b>   |
|                 | c-fat200-1    | 188   | 188        | <b>188</b>      | <b>188</b>   | <b>188</b> | <b>188</b>   |
|                 | c-fat200-2    | 176   | 176        | <b>176</b>      | <b>176</b>   | <b>176</b> | <b>176</b>   |
|                 | c-fat200-5    | 142   | 142        | <b>142</b>      | <b>142</b>   | <b>142</b> | <b>142</b>   |
|                 | c-fat500-1    | 486   | 486        | <b>486</b>      | <b>486</b>   | <b>486</b> | <b>486</b>   |
| hamming $W-d$   | c-fat500-2    | 474   | 474        | <b>474</b>      | <b>474</b>   | <b>474</b> | <b>474</b>   |
|                 | c-fat500-5    | 436   | 436        | <b>436</b>      | <b>436</b>   | <b>436</b> | <b>436</b>   |
|                 | c-fat500-10   | 374   | 374        | <b>374</b>      | <b>374</b>   | <b>374</b> | <b>374</b>   |
|                 | hamming6-2    | 32    | 32         | <b>32</b>       | <b>32</b>    | <b>32</b>  | <b>32</b>    |
|                 | hamming6-4    | 60    | 60         | <b>60</b>       | <b>60</b>    | <b>60</b>  | <b>60</b>    |
| johnson $W-w-d$ | hamming8-2    | 128   | 128        | <b>128</b>      | <b>128</b>   | <b>128</b> | <b>128</b>   |
|                 | hamming8-4    | 240   | 240        | <b>240</b>      | <b>240</b>   | <b>240</b> | <b>240</b>   |
|                 | hamming10-2   | 512   | 512        | <b>512</b>      | 530          | <b>512</b> | 530          |
|                 | hamming10-4   |       | 988        | <b>988</b>      | 990          | <b>988</b> | 990          |
|                 | johnson8-2-4  | 24    | 24         | <b>24</b>       | <b>24</b>    | <b>24</b>  | <b>24</b>    |
| keller $d$      | johnson8-4-4  | 56    | 56         | <b>56</b>       | <b>56</b>    | <b>56</b>  | <b>56</b>    |
|                 | johnson16-2-4 | 112   | 112        | <b>112</b>      | <b>112</b>   | <b>112</b> | <b>112</b>   |
|                 | johnson32-2-4 |       | 480        | <b>480</b>      | <b>480</b>   | <b>480</b> | <b>480</b>   |
| MANN_ $ax$      | keller4       | 160   | 160        | <b>160</b>      | <b>160</b>   | <b>160</b> | <b>160</b>   |
|                 | keller5       | 749   | 750        | 753             | 753          | 754        | 752          |
|                 | keller6       |       | 3310       | 3320            | 3319         | 3322       | 3320         |
| p_hat $n_x$     | MANN_a27      | 252   | 252        | 255             | 256          | 255        | 256          |
|                 | p_hat300-1    | 292   | 292        | <b>292</b>      | <b>292</b>   | <b>292</b> | <b>292</b>   |
|                 | p_hat300-2    | 275   | 275        | <b>275</b>      | <b>275</b>   | <b>275</b> | <b>275</b>   |
| sann_ $f_x$     | p_hat300-3    | 264   | 266        | <b>264</b>      | <b>265</b>   | <b>264</b> | <b>265</b>   |
|                 | san200_0.7_1  | 170   | 170        | <b>170</b>      | <b>170</b>   | 184        | <b>170</b>   |
|                 | san200_0.7_2  | 182   | 185        | 186             | 186          | 186        | 186          |
|                 | san200_0.9_1  | 130   | 130        | <b>130</b>      | <b>130</b>   | <b>130</b> | <b>130</b>   |
|                 | san200_0.9_2  | 140   | 140        | <b>140</b>      | <b>140</b>   | <b>140</b> | <b>140</b>   |
|                 | san200_0.9_3  | 156   | 164        | <b>156</b>      | <b>156</b>   | <b>156</b> | <b>156</b>   |
|                 | san400_0.5_1  | 387   | 391        | 392             | 392          | 392        | 392          |
|                 | san400_0.7_1  | 360   | 367        | 379             | 378          | <b>0</b>   | 379          |
|                 | san400_0.7_2  | 370   | 381        | 384             | 383          | <b>0</b>   | 384          |
|                 | san400_0.7_3  | 378   | 384        | 385             | 386          | 385        | <b>384</b>   |
| sanr $n_p$      | san400_0.9_1  | 300   | 300        | <b>300</b>      | <b>300</b>   | <b>300</b> | <b>300</b>   |
|                 | san1000       | 985   | 990        | 992             | 992          | 992        | 992          |
|                 | sanr200_0.7   | 182   | 182        | <b>182</b>      | <b>182</b>   | <b>182</b> | <b>182</b>   |
|                 | sanr200_0.9   |       | 159        | <b>158</b>      | <b>158</b>   | <b>158</b> | <b>158</b>   |
|                 | sanr400_0.5   |       | 388        | <b>387</b>      | <b>387</b>   | <b>387</b> | <b>388</b>   |
|                 | sanr400_0.7   |       | 379        | <b>379</b>      | 380          | <b>379</b> | 380          |

É possível notar que, quando comparado aos melhores resultados das heurísticas *XSD*, as variantes do método *SAHe-CVM* apresentam bom desempenho. Além disso, a Figura 5.7 demonstra que o desempenho das variantes do *SAHe-CVM* foi superior para uma porção considerável das instâncias; melhor ainda, o quinto gráfico dessa figura, demonstra que ao considerar os melhores resultados de todas as variantes do *SAHe-CVM*, este conjunto de heurísticas apresenta mais resultados melhores (25%) do que resultados piores (16%).

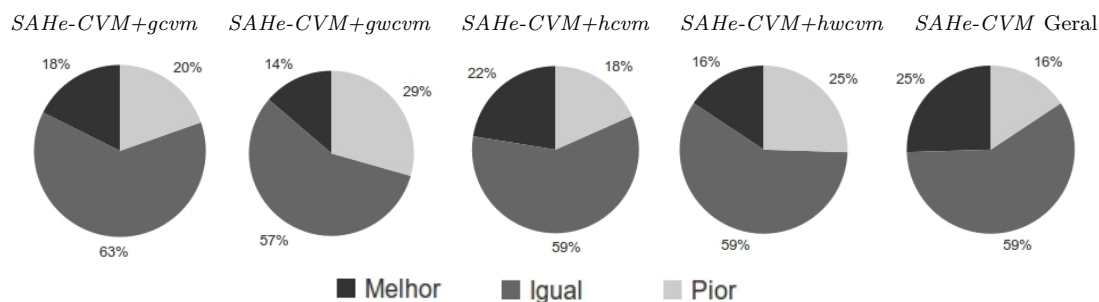


Figura 5.7: Comparação dos valores obtidos entre *XSD* e *SAHe-CVM*.

### Heurística *CBH*

A Tabela 5.14 e a Figura 5.8 demonstram um desempenho ainda melhor de todas as variantes do *SAHe-CVM*, quando comparado aos resultados encontrados pela heurística determinística *CBH* [46].

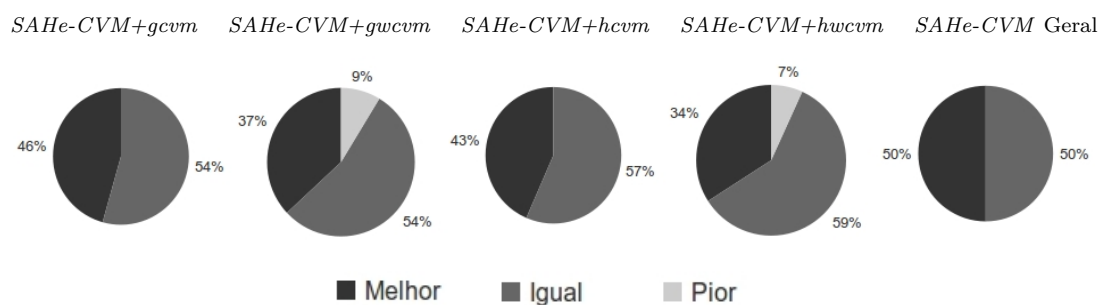


Figura 5.8: Comparação dos valores obtidos entre *CBH* e *SAHe-CVM*.

Nota-se que os resultados da heurística *CBH* foram no máximo iguais aos resultados das variantes *SAHe-CVM+gwcvm* e *SAHe-CVM+hcvvm* e, ainda, apenas na minoria dos casos, obteve resultados melhores que as variantes do *SAHe-CVM*, que utilizam o mecanismo de *random walk* na rotina de pós-processamento.

É possível notar, inclusive, que, ao comparar os resultados da heurística *CBH* aos melhores resultados do *SAHe-CVM* (último gráfico da Figura 5.8), este método apresenta resultados estritamente melhores em 50% dos casos.

Tabela 5.14: Comparação de resultados entre *CBH* e *SAHe-CVM*.

| Família                        | Grafo           | Ótimo | <i>CBH</i> | <i>SAHe-CVM</i> |             |            |             |
|--------------------------------|-----------------|-------|------------|-----------------|-------------|------------|-------------|
|                                |                 |       |            | <i>gcv</i>      | <i>gwcv</i> | <i>hcv</i> | <i>hwcv</i> |
| <i>brock<sub>n-x</sub></i>     | brock200_1      | 179   | 180        | <b>179</b>      | <b>179</b>  | <b>179</b> | <b>179</b>  |
|                                | brock200_2      | 188   | 188        | <b>188</b>      | <b>188</b>  | <b>188</b> | <b>188</b>  |
|                                | brock200_3      | 185   | 186        | <b>186</b>      | <b>185</b>  | <b>186</b> | <b>185</b>  |
|                                | brock200_4      | 183   | 184        | <b>183</b>      | <b>183</b>  | <b>183</b> | <b>183</b>  |
|                                | brock400_1      | 373   | 377        | <b>376</b>      | <b>376</b>  | <b>375</b> | <b>377</b>  |
|                                | brock400_2      | 371   | 376        | <b>376</b>      | <b>376</b>  | <b>375</b> | <b>376</b>  |
|                                | brock400_3      | 369   | 377        | <b>376</b>      | <b>376</b>  | <b>375</b> | <b>376</b>  |
|                                | brock400_4      | 367   | 376        | <b>375</b>      | 377         | <b>376</b> | <b>375</b>  |
| <i>c-fat<sub>n-c</sub></i>     | c-fat200-1      | 188   | 188        | <b>188</b>      | <b>188</b>  | <b>188</b> | <b>188</b>  |
|                                | c-fat200-2      | 176   | 176        | <b>176</b>      | <b>176</b>  | <b>176</b> | <b>176</b>  |
|                                | c-fat200-5      | 142   | 142        | <b>142</b>      | <b>142</b>  | <b>142</b> | <b>142</b>  |
|                                | c-fat500-1      | 486   | 486        | <b>486</b>      | <b>486</b>  | <b>486</b> | <b>486</b>  |
|                                | c-fat500-2      | 474   | 474        | <b>474</b>      | <b>474</b>  | <b>474</b> | <b>474</b>  |
|                                | c-fat500-5      | 436   | 436        | <b>436</b>      | <b>436</b>  | <b>436</b> | <b>436</b>  |
|                                | c-fat500-10     | 374   | 374        | <b>374</b>      | <b>374</b>  | <b>374</b> | <b>374</b>  |
| <i>hamming<sub>W-d</sub></i>   | hamming6-2      | 32    | 32         | <b>32</b>       | <b>32</b>   | <b>32</b>  | <b>32</b>   |
|                                | hamming6-4      | 60    | 60         | <b>60</b>       | <b>60</b>   | <b>60</b>  | <b>60</b>   |
|                                | hamming8-2      | 128   | 128        | <b>128</b>      | <b>128</b>  | <b>128</b> | <b>128</b>  |
|                                | hamming8-4      | 240   | 240        | <b>240</b>      | <b>240</b>  | <b>240</b> | <b>240</b>  |
| <i>johnson<sub>W-w-d</sub></i> | johnson8-2-4    | 24    | 24         | <b>24</b>       | <b>24</b>   | <b>24</b>  | <b>24</b>   |
|                                | johnson8-4-4    | 56    | 56         | <b>56</b>       | <b>56</b>   | <b>56</b>  | <b>56</b>   |
|                                | johnson16-2-4   | 112   | 112        | <b>112</b>      | <b>112</b>  | <b>112</b> | <b>112</b>  |
|                                | johnson32-2-4   |       | 480        | <b>480</b>      | <b>480</b>  | <b>480</b> | <b>480</b>  |
| <i>keller<sub>d</sub></i>      | <i>keller4</i>  | 160   | 161        | <b>160</b>      | <b>160</b>  | <b>160</b> | <b>160</b>  |
| <i>MANN<sub>ax</sub></i>       | <i>MANN_a9</i>  | 29    | 29         | <b>29</b>       | <b>29</b>   | <b>29</b>  | <b>29</b>   |
|                                | <i>MANN_a27</i> | 252   | 257        | <b>255</b>      | <b>256</b>  | <b>255</b> | <b>256</b>  |
| <i>p_hat<sub>n-x</sub></i>     | p_hat300-1      | 292   | 292        | <b>292</b>      | <b>292</b>  | <b>292</b> | <b>292</b>  |
|                                | p_hat300-2      | 275   | 275        | <b>275</b>      | <b>275</b>  | <b>275</b> | <b>275</b>  |
|                                | p_hat300-3      | 264   | 264        | <b>264</b>      | 265         | <b>264</b> | 265         |
|                                | p_hat500-1      | 491   | 491        | <b>491</b>      | <b>491</b>  | <b>491</b> | <b>491</b>  |
|                                | p_hat500-2      | 464   | 465        | <b>464</b>      | 466         | <b>464</b> | 467         |
|                                | p_hat500-3      |       | 451        | <b>450</b>      | 453         | <b>451</b> | 453         |
|                                |                 |       |            |                 |             |            |             |
| <i>san<sub>n-f-x</sub></i>     | san200_0.7_1    | 170   | 185        | <b>170</b>      | <b>170</b>  | <b>184</b> | <b>170</b>  |
|                                | san200_0.7_2    | 182   | 188        | <b>186</b>      | <b>186</b>  | <b>186</b> | <b>186</b>  |
|                                | san200_0.9_1    | 130   | 154        | <b>130</b>      | <b>130</b>  | <b>130</b> | <b>130</b>  |
|                                | san200_0.9_2    | 140   | 164        | <b>140</b>      | <b>140</b>  | <b>140</b> | <b>140</b>  |
|                                | san200_0.9_3    | 156   | 170        | <b>156</b>      | <b>156</b>  | <b>156</b> | <b>156</b>  |
|                                | san400_0.5_1    | 387   | 392        | <b>392</b>      | <b>392</b>  | <b>392</b> | <b>392</b>  |
|                                | san400_0.7_1    | 360   | 380        | <b>379</b>      | <b>378</b>  |            | <b>379</b>  |
|                                | san400_0.7_2    | 370   | 385        | <b>384</b>      | <b>383</b>  |            | <b>384</b>  |
|                                | san400_0.7_3    | 378   | 386        | <b>385</b>      | <b>386</b>  | <b>385</b> | <b>384</b>  |
|                                | san400_0.9_1    | 300   | 350        | <b>300</b>      | <b>300</b>  | <b>300</b> | <b>300</b>  |
| <i>sanr<sub>n-p</sub></i>      | sanr200_0.7     | 182   | 182        | <b>182</b>      | <b>182</b>  | <b>182</b> | <b>182</b>  |
|                                | sanr200_0.9     |       | 159        | <b>158</b>      | <b>158</b>  | <b>158</b> | <b>158</b>  |
|                                | sanr400_0.5     |       | 388        | <b>387</b>      | <b>387</b>  | <b>387</b> | <b>388</b>  |
|                                | sanr400_0.7     |       | 380        | <b>379</b>      | <b>380</b>  | <b>379</b> | <b>380</b>  |

## Heurística *ATA*

A Tabela 5.15 é análoga à Tabela 5.10 e apresenta a comparação entre os resultados da heurística *ATA* e os resultados obtidos nos testes experimentais com algumas variantes do método *SAHe-CVM*. De modo complementar, os gráficos presentes na Figura 5.9 demonstram relações de proporcionalidade entre as coberturas encontradas pelo método *ATA* e pelas variantes do método *SAHe-CVM*.

Nota-se que a heurística *ATA* possui melhor desempenho para uma porção expressiva de instâncias do que as variantes do método *SAHe-CVM*. Este fato é mais claro para as variantes do *SAHe-CVM* que utilizam o mecanismo de *random walk*

na rotina de pós-processamento, onde a heurística *ATA* obteve um desempenho melhor em 59% das instâncias sobre a variante *SAHe-CVM+gvcvm*, e em 49% das instâncias sobre a variante *SAHe-CVM+hvcvm*.

Tabela 5.15: Comparação de resultados entre *ATA* e *SAHe-CVM*.

| Família                              | Grafo          | Ótimo | <i>ATA</i> | <i>SAHe-CVM</i> |              |              |              |
|--------------------------------------|----------------|-------|------------|-----------------|--------------|--------------|--------------|
|                                      |                |       |            | <i>gvcvm</i>    | <i>gvcvm</i> | <i>hvcvm</i> | <i>hvcvm</i> |
| <i>brockn<sub>x</sub></i>            | brock200_2     | 188   | 189        | <b>188</b>      | <b>188</b>   | <b>188</b>   | <b>188</b>   |
|                                      | brock200_4     | 183   | 184        | <b>183</b>      | <b>183</b>   | <b>183</b>   | <b>183</b>   |
|                                      | brock400_2     | 371   | 375        | 376             | 376          | <b>375</b>   | 376          |
|                                      | brock400_4     | 367   | 375        | <b>375</b>      | 377          | 376          | <b>375</b>   |
|                                      | brock800_2     | 776   | 780        | <b>780</b>      | 781          | 781          | 781          |
|                                      | brock800_4     | 774   | 779        | 781             | 781          | 781          | 781          |
| <i>Cn.d</i>                          | C125.9         |       | 91         | <b>91</b>       | <b>91</b>    | <b>91</b>    | <b>91</b>    |
|                                      | C250.9         |       | 207        | <b>206</b>      | <b>206</b>   | <b>206</b>   | <b>207</b>   |
|                                      | C500.9         |       | 445        | 448             | 450          | 448          | 450          |
|                                      | C1000.9        |       | 936        | 941             | 944          | 941          | 944          |
|                                      | C2000.5        |       | 1985       | 1986            | 1986         | 1986         | 1986         |
|                                      | C2000.9        |       | 1930       | 1935            | 1939         | 1934         | 1939         |
|                                      | C4000.5        |       | 3984       | 3985            | 3986         | 3986         | 3986         |
| <i>DSJCN<sub>x</sub></i>             | DSJC500.5      |       | 487        | <b>487</b>      | 488          | <b>487</b>   | <b>487</b>   |
|                                      | DSJC1000.5     |       | 987        | <b>986</b>      | <b>987</b>   | <b>986</b>   | <b>987</b>   |
| <i>gen<sub>n</sub>pp<sub>c</sub></i> | gen200_p0.9_44 | 156   | 162        | <b>156</b>      | <b>156</b>   | <b>156</b>   | <b>157</b>   |
|                                      | gen200_p0.9_55 | 145   | 145        | <b>145</b>      | <b>145</b>   | <b>145</b>   | <b>145</b>   |
|                                      | gen400_p0.9_55 | 345   | 352        | <b>351</b>      | 353          | <b>351</b>   | <b>352</b>   |
|                                      | gen400_p0.9_65 | 335   | 366        | <b>340</b>      | <b>348</b>   | <b>339</b>   | <b>344</b>   |
|                                      | gen400_p0.9_75 | 325   | 361        | <b>325</b>      | <b>335</b>   | <b>325</b>   | <b>334</b>   |
| <i>hamming<sub>w-d</sub></i>         | hamming8-4     | 240   | 240        | <b>240</b>      | <b>240</b>   | <b>240</b>   | <b>240</b>   |
|                                      | hamming10-4    |       | 992        | <b>988</b>      | <b>990</b>   | <b>988</b>   | <b>990</b>   |
| <i>keller<sub>d</sub></i>            | keller4        | 160   | 160        | <b>160</b>      | <b>160</b>   | <b>160</b>   | <b>160</b>   |
|                                      | keller5        | 749   | 749        | 753             | 753          | 754          | 752          |
|                                      | keller6        |       | 3322       | <b>3320</b>     | <b>3319</b>  | <b>3322</b>  | <b>3320</b>  |
| <i>MANN<sub>a<sub>x</sub></sub></i>  | MANN_a27       | 252   | 252        | 255             | 256          | 255          | 256          |
|                                      | MANN_a45       | 690   | 695        | 702             | 702          | 702          | 703          |
|                                      | MANN_a81       |       | 2225       | 2240            | 2240         | 2240         | 2240         |
| <i>p<sub>hat</sub><sub>n-x</sub></i> | p_hat300-1     | 292   | 292        | <b>292</b>      | <b>292</b>   | <b>292</b>   | <b>292</b>   |
|                                      | p_hat300-2     | 275   | 275        | <b>275</b>      | <b>275</b>   | <b>275</b>   | <b>275</b>   |
|                                      | p_hat300-3     | 264   | 264        | <b>264</b>      | 265          | <b>264</b>   | 265          |
|                                      | p_hat700-1     |       | 689        | <b>689</b>      | 690          | <b>689</b>   | <b>689</b>   |
|                                      | p_hat700-2     |       | 658        | <b>657</b>      | 660          | <b>656</b>   | 660          |
|                                      | p_hat700-3     |       | 641        | <b>640</b>      | 642          | <b>640</b>   | 643          |
|                                      | p_hat1500-1    |       | 1489       | <b>1489</b>     | 1490         | <b>1489</b>  | 1490         |
|                                      | p_hat1500-2    |       | 1438       | 1440            | 1445         | <b>1438</b>  | 1444         |
|                                      | p_hat1500-3    |       | 1411       | 1412            | 1423         | 1413         | 1421         |

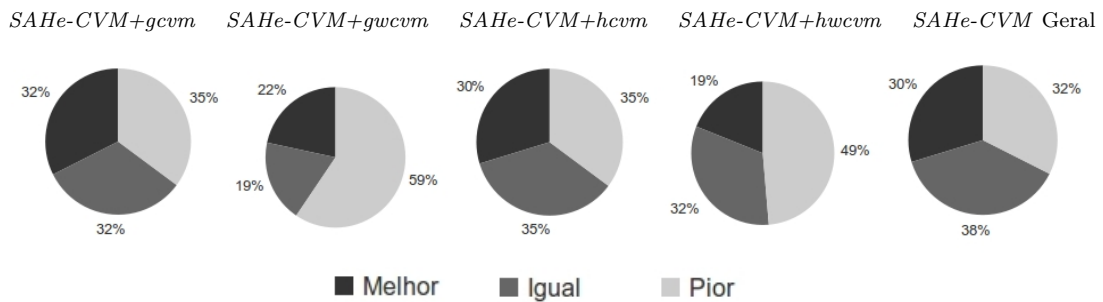


Figura 5.9: Comparação dos valores obtidos entre *ATA* e *SAHe-CVM*.



## Heurística *OCH*

Como descrito na Subseção 2.2.3, a heurística *OCH* [48] consiste em um algoritmo genético, cujos indivíduos são soluções candidatas para uma dada instância do problema do conjunto independente máximo.

A Tabela 5.16 e a Figura 5.10 apresentam informações análogas às tabelas e figuras anteriores, considerando-se o resultados da heurísticas *OCH*.

Tabela 5.16: Comparação de resultados entre *OCH* e *SAHe-CVM*.

| Família                    | Grafo         | Ótimo | <i>OCH</i> | <i>SAHe-CVM</i> |              |             |              |
|----------------------------|---------------|-------|------------|-----------------|--------------|-------------|--------------|
|                            |               |       |            | <i>gcvm</i>     | <i>gwcvm</i> | <i>hcvm</i> | <i>hwcvm</i> |
| <i>brockn<sub>x</sub></i>  | brock200_1    | 179   | 179        | <b>179</b>      | <b>179</b>   | <b>179</b>  | <b>179</b>   |
|                            | brock200_2    | 188   | 189        | <b>188</b>      | <b>188</b>   | <b>188</b>  | <b>188</b>   |
|                            | brock200_3    | 185   | 186        | <b>186</b>      | <b>185</b>   | <b>186</b>  | <b>185</b>   |
|                            | brock200_4    | 183   | 184        | <b>183</b>      | <b>183</b>   | <b>183</b>  | <b>183</b>   |
|                            | brock400_1    | 373   | 376        | <b>376</b>      | <b>376</b>   | <b>375</b>  | 377          |
|                            | brock400_2    | 371   | 376        | <b>376</b>      | <b>376</b>   | <b>375</b>  | <b>376</b>   |
|                            | brock400_3    | 369   | 376        | <b>376</b>      | <b>376</b>   | <b>375</b>  | <b>376</b>   |
| <i>c-fatn-c</i>            | brock400_4    | 367   | 376        | <b>375</b>      | <b>377</b>   | <b>376</b>  | <b>375</b>   |
|                            | c-fat200-1    | 188   | 188        | <b>188</b>      | <b>188</b>   | <b>188</b>  | <b>188</b>   |
|                            | c-fat200-2    | 176   | 176        | <b>176</b>      | <b>176</b>   | <b>176</b>  | <b>176</b>   |
|                            | c-fat200-5    | 142   | 142        | <b>142</b>      | <b>142</b>   | <b>142</b>  | <b>142</b>   |
|                            | c-fat500-1    | 486   | 486        | <b>486</b>      | <b>486</b>   | <b>486</b>  | <b>486</b>   |
|                            | c-fat500-2    | 474   | 474        | <b>474</b>      | <b>474</b>   | <b>474</b>  | <b>474</b>   |
|                            | c-fat500-5    | 436   | 436        | <b>436</b>      | <b>436</b>   | <b>436</b>  | <b>436</b>   |
| <i>hammingW-d</i>          | c-fat500-10   | 374   | 374        | <b>374</b>      | <b>374</b>   | <b>374</b>  | <b>374</b>   |
|                            | hamming6-2    | 32    | 32         | <b>32</b>       | <b>32</b>    | <b>32</b>   | <b>32</b>    |
|                            | hamming6-4    | 60    | 60         | <b>60</b>       | <b>60</b>    | <b>60</b>   | <b>60</b>    |
|                            | hamming8-2    | 128   | 128        | <b>128</b>      | <b>128</b>   | <b>128</b>  | <b>128</b>   |
| <i>johnsonW-w-d</i>        | hamming8-4    | 240   | 240        | <b>240</b>      | <b>240</b>   | <b>240</b>  | <b>240</b>   |
|                            | johnson8-2-4  | 24    | 24         | <b>24</b>       | <b>24</b>    | <b>24</b>   | <b>24</b>    |
|                            | johnson8-4-4  | 56    | 56         | <b>56</b>       | <b>56</b>    | <b>56</b>   | <b>56</b>    |
|                            | johnson16-2-4 | 112   | 112        | <b>112</b>      | <b>112</b>   | <b>112</b>  | <b>112</b>   |
| <i>kellerd</i>             | johnson32-2-4 |       | 480        | <b>480</b>      | <b>480</b>   | <b>480</b>  | <b>480</b>   |
|                            | keller4       | 160   | 160        | <b>160</b>      | <b>160</b>   | <b>160</b>  | <b>160</b>   |
| <i>MANN_ax</i>             | MANN_a9       | 29    | 29         | <b>29</b>       | <b>29</b>    | <b>29</b>   | <b>29</b>    |
|                            | MANN_a27      | 252   | 252        | 255             | 256          | 255         | 256          |
| <i>p_hatn-x</i>            | p_hat300-1    | 292   | 292        | <b>292</b>      | <b>292</b>   | <b>292</b>  | <b>292</b>   |
|                            | p_hat300-2    | 275   | 275        | <b>275</b>      | <b>275</b>   | <b>275</b>  | <b>275</b>   |
|                            | p_hat300-3    | 264   | 264        | <b>264</b>      | 265          | <b>264</b>  | 265          |
|                            | p_hat500-1    | 491   | 491        | <b>491</b>      | <b>491</b>   | <b>491</b>  | <b>491</b>   |
|                            | p_hat500-2    | 464   | 464        | <b>464</b>      | 466          | <b>464</b>  | 467          |
|                            | p_hat500-3    | 451   | 451        | <b>450</b>      | 453          | <b>451</b>  | 453          |
| <i>san<sub>n</sub>_f_x</i> | san200_0.7_1  | 170   | 170        | <b>170</b>      | <b>170</b>   | 184         | <b>170</b>   |
|                            | san200_0.7_2  | 182   | 185        | 186             | 186          | 186         | 186          |
|                            | san200_0.9_1  | 130   | 130        | <b>130</b>      | <b>130</b>   | <b>130</b>  | <b>130</b>   |
|                            | san200_0.9_2  | 140   | 140        | <b>140</b>      | <b>140</b>   | <b>140</b>  | <b>140</b>   |
|                            | san200_0.9_3  | 156   | 164        | <b>156</b>      | <b>156</b>   | <b>156</b>  | <b>156</b>   |
|                            | san400_0.5_1  | 387   | 387        | 392             | 392          | 392         | 392          |
|                            | san400_0.7_1  | 360   | 360        | 379             | 378          |             | 379          |
|                            | san400_0.7_2  | 370   | 370        | 384             | 383          |             | 384          |
|                            | san400_0.7_3  | 378   | 384        | 385             | 386          | 385         | <b>384</b>   |
| san400_0.9_1               | 300           | 300   | <b>300</b> | <b>300</b>      | <b>300</b>   | <b>300</b>  |              |
| <i>sanr<sub>n</sub>_p</i>  | sanr200_0.7   | 182   | 182        | <b>182</b>      | <b>182</b>   | <b>182</b>  | <b>182</b>   |
|                            | sanr200_0.9   |       | 158        | <b>158</b>      | <b>158</b>   | <b>158</b>  | <b>158</b>   |
|                            | sanr400_0.5   |       | 388        | <b>387</b>      | <b>387</b>   | <b>387</b>  | <b>388</b>   |
|                            | sanr400_0.7   |       | 380        | <b>379</b>      | <b>380</b>   | <b>379</b>  | <b>380</b>   |

É possível notar que, quando comparado aos resultados da heurística *OCH*, as variantes do método *SAHe-CVM* apresentam bom desempenho. Além disso, a

Figura 5.10 demonstra que o número de instâncias, em que o melhor resultado foi encontrado pelo *SAHe-CVM*, corresponde a mais que o dobro do que o número de instâncias cujo melhor resultado foi encontrado pelo método *OCH*.

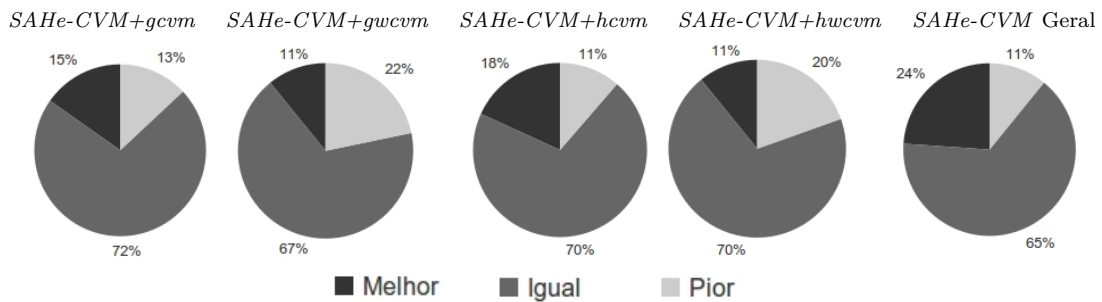


Figura 5.10: Comparação dos valores obtidos entre *OCH* e *SAHe-CVM*.

### Heurística *WAO*

A heurística *WAO* [23] é relativa também a um algoritmo genético para resolução do problema do conjunto independente máximo.

A Tabela 5.17 e a Figura 5.11 demonstram um ótimo desempenho da totalidade de variantes do *SAHe-CVM*, se comparado aos resultados da heurística *WAO*.

Nota-se, ainda, que os resultados dessa heurística foram melhores em, no máximo, 11% das instâncias. Mais ainda, a heurística *WAO* foi superior aos melhores resultados do método *SAHe-CVM* em apenas 2% das instâncias testadas.

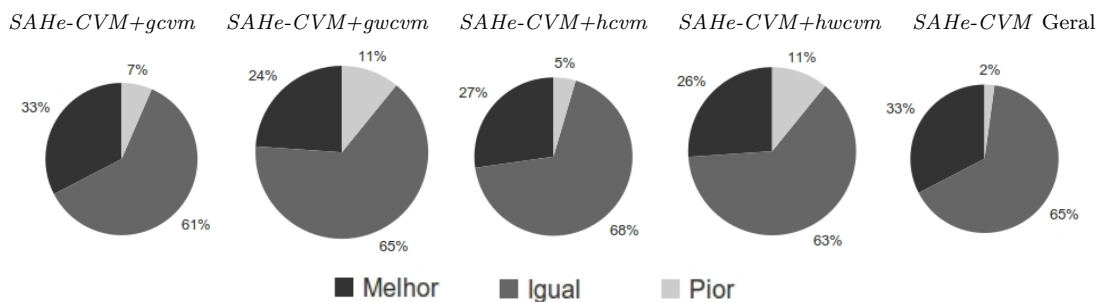


Figura 5.11: Comparação dos valores obtidos entre *WAO* e *SAHe-CVM*.

Tabela 5.17: Comparação de resultados entre *WAO* e *SAHe-CVM*.

| Família                        | Grafo           | Ótimo | <i>WAO</i> | <i>SAHe-CVM</i> |             |            |             |
|--------------------------------|-----------------|-------|------------|-----------------|-------------|------------|-------------|
|                                |                 |       |            | <i>gcv</i>      | <i>gwcv</i> | <i>hcv</i> | <i>hwcv</i> |
| <i>brock<sub>n-x</sub></i>     | brock200_1      | 179   | 181        | <b>179</b>      | <b>179</b>  | <b>179</b> | <b>179</b>  |
|                                | brock200_2      | 188   | 190        | <b>188</b>      | <b>188</b>  | <b>188</b> | <b>188</b>  |
|                                | brock200_3      | 185   | 187        | <b>186</b>      | <b>185</b>  | <b>186</b> | <b>185</b>  |
|                                | brock200_4      | 183   | 185        | <b>183</b>      | <b>183</b>  | <b>183</b> | <b>183</b>  |
|                                | brock400_1      | 373   | 378        | <b>376</b>      | <b>376</b>  | <b>375</b> | <b>377</b>  |
|                                | brock400_2      | 371   | 378        | <b>376</b>      | <b>376</b>  | <b>375</b> | <b>376</b>  |
|                                | brock400_3      | 369   | 378        | <b>376</b>      | <b>376</b>  | <b>375</b> | <b>376</b>  |
|                                | brock400_4      | 367   | 377        | <b>375</b>      | <b>377</b>  | <b>376</b> | <b>375</b>  |
| <i>c-fat<sub>n-c</sub></i>     | c-fat200-1      | 188   | 188        | <b>188</b>      | <b>188</b>  | <b>188</b> | <b>188</b>  |
|                                | c-fat200-2      | 176   | 176        | <b>176</b>      | <b>176</b>  | <b>176</b> | <b>176</b>  |
|                                | c-fat200-5      | 142   | 142        | <b>142</b>      | <b>142</b>  | <b>142</b> | <b>142</b>  |
|                                | c-fat500-1      | 486   | 486        | <b>486</b>      | <b>486</b>  | <b>486</b> | <b>486</b>  |
|                                | c-fat500-2      | 474   | 474        | <b>474</b>      | <b>474</b>  | <b>474</b> | <b>474</b>  |
|                                | c-fat500-5      | 436   | 436        | <b>436</b>      | <b>436</b>  | <b>436</b> | <b>436</b>  |
|                                | c-fat500-10     | 374   | 374        | <b>374</b>      | <b>374</b>  | <b>374</b> | <b>374</b>  |
| <i>hamming<sub>W-d</sub></i>   | hamming6-2      | 32    | 32         | <b>32</b>       | <b>32</b>   | <b>32</b>  | <b>32</b>   |
|                                | hamming6-4      | 60    | 60         | <b>60</b>       | <b>60</b>   | <b>60</b>  | <b>60</b>   |
|                                | hamming8-2      | 128   | 128        | <b>128</b>      | <b>128</b>  | <b>128</b> | <b>128</b>  |
|                                | hamming8-4      | 240   | 240        | <b>240</b>      | <b>240</b>  | <b>240</b> | <b>240</b>  |
| <i>johnson<sub>W-w-d</sub></i> | johnson8-2-4    | 24    | 24         | <b>24</b>       | <b>24</b>   | <b>24</b>  | <b>24</b>   |
|                                | johnson8-4-4    | 56    | 56         | <b>56</b>       | <b>56</b>   | <b>56</b>  | <b>56</b>   |
|                                | johnson16-2-4   | 112   | 112        | <b>112</b>      | <b>112</b>  | <b>112</b> | <b>112</b>  |
|                                | johnson32-2-4   |       | 480        | <b>480</b>      | <b>480</b>  | <b>480</b> | <b>480</b>  |
| <i>keller<sub>d</sub></i>      | <i>keller4</i>  | 160   | 160        | <b>160</b>      | <b>160</b>  | <b>160</b> | <b>160</b>  |
| <i>MANN<sub>ax</sub></i>       | <i>MANN_a9</i>  | 29    | 29         | <b>29</b>       | <b>29</b>   | <b>29</b>  | <b>29</b>   |
|                                | <i>MANN_a27</i> | 252   | 252        | 255             | 256         | 255        | 256         |
| <i>p_hat<sub>n-x</sub></i>     | p_hat300-1      | 292   | 292        | <b>292</b>      | <b>292</b>  | <b>292</b> | <b>292</b>  |
|                                | p_hat300-2      | 275   | 275        | <b>275</b>      | <b>275</b>  | <b>275</b> | <b>275</b>  |
|                                | p_hat300-3      | 264   | 264        | <b>264</b>      | 265         | <b>264</b> | 265         |
|                                | p_hat500-1      | 491   | 491        | <b>491</b>      | <b>491</b>  | <b>491</b> | <b>491</b>  |
|                                | p_hat500-2      | 464   | 465        | <b>464</b>      | 466         | <b>464</b> | 467         |
|                                | p_hat500-3      |       | 451        | <b>450</b>      | 453         | <b>451</b> | 453         |
|                                |                 |       |            |                 |             |            |             |
| <i>san<sub>n-f-x</sub></i>     | san200_0.7_1    | 170   | 184        | <b>170</b>      | <b>170</b>  | <b>184</b> | <b>170</b>  |
|                                | san200_0.7_2    | 182   | 186        | <b>186</b>      | <b>186</b>  | <b>186</b> | <b>186</b>  |
|                                | san200_0.9_1    | 130   | 130        | <b>130</b>      | <b>130</b>  | <b>130</b> | <b>130</b>  |
|                                | san200_0.9_2    | 140   | 140        | <b>140</b>      | <b>140</b>  | <b>140</b> | <b>140</b>  |
|                                | san200_0.9_3    | 156   | 156        | <b>156</b>      | <b>156</b>  | <b>156</b> | <b>156</b>  |
|                                | san400_0.5_1    | 387   | 392        | <b>392</b>      | <b>392</b>  | <b>392</b> | <b>392</b>  |
|                                | san400_0.7_1    | 360   | 380        | <b>379</b>      | <b>378</b>  | 0          | <b>379</b>  |
|                                | san400_0.7_2    | 370   | 383        | 384             | <b>383</b>  | 0          | 384         |
|                                | san400_0.7_3    | 378   | 384        | 385             | 386         | 385        | <b>384</b>  |
|                                | san400_0.9_1    | 300   | 302        | <b>300</b>      | <b>300</b>  | <b>300</b> | <b>300</b>  |
| <i>sanr<sub>n-p</sub></i>      | sanr200_0.7     | 182   | 182        | <b>182</b>      | <b>182</b>  | <b>182</b> | <b>182</b>  |
|                                | sanr200_0.9     |       | 158        | <b>158</b>      | <b>158</b>  | <b>158</b> | <b>158</b>  |
|                                | sanr400_0.5     |       | 389        | <b>387</b>      | <b>387</b>  | <b>387</b> | <b>388</b>  |
|                                | sanr400_0.7     |       | 380        | <b>379</b>      | <b>380</b>  | <b>379</b> | <b>380</b>  |

### Heurística *DLS-MC*

A heurística *DLS-MC*, descrita por PULLAN e HOSS [4], também apresenta resultados experimentais sobre o conjunto de 1996. Grande parte dos resultados descritos por essa heurística são melhores do aqueles alcançados pelas variantes do método *SAHe-CVM*.

A *DLS-MC* encontrou o melhor resultado conhecido para quase 100% das instâncias do conjunto de 1996; fato que não ocorreu com o método *SAHe-CVM*. A Figura 5.12 descreve as relações de proporcionalidade entre as coberturas encontradas pelo método *DLS-MC* e pelas variantes do método *SAHe-CVM*.

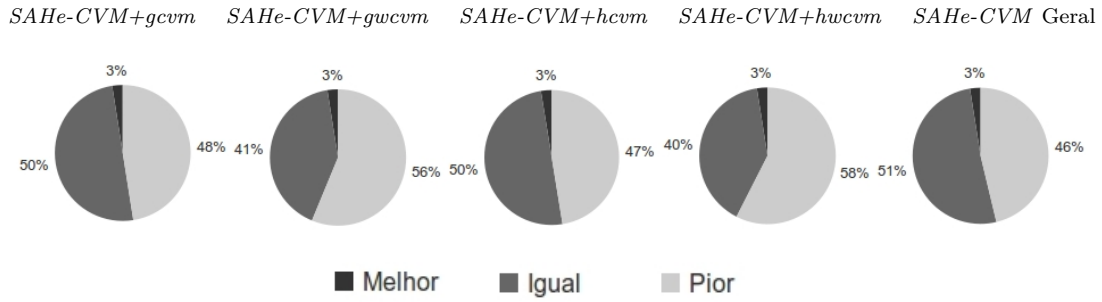


Figura 5.12: Comparação dos valores obtidos entre *DLS-MC* e *SAHe-CVM*.

Nota-se que os resultados da heurística *DLS-MC* foram melhores em pelo menos 46% das instâncias. Mais ainda, que a heurística *DLS-MC* foi inferior aos resultados das variantes do método *SAHe-CVM* em apenas 3% das instâncias testadas, a saber: sobre duas instâncias da família *johnsonW-w-d* (*johnson16-2-4* e *johnson32-2-4*). Para a instância *johnson16-2-4*, o método *DLS-MC* encontrou uma cobertura de tamanho 116, enquanto que o método *SAHe-CVM* encontrou uma cobertura de tamanho 112. Valor ótimo. Já para a instância *johnson32-2-4*, os valores de cobertura encontrados foram 482 e 480, respectivamente, para *DLS-MC* e *SAHe-CVM*.

### 5.3 Considerações Finais

Foram devidamente expostos neste capítulo a formulação e descrição do modelo de seleção automática de heurísticas para o problema CVM, denominado *SAHe-CVM*, o qual maximiza os resultados alcançados por outras heurísticas, utilizadas aqui como rotinas de pós-processamento (*Hp*).

Com o objetivo de realizar a análise experimental do método *SAHe-CVM*, foram implementadas algumas variantes de métodos heurísticos gulosos (*gcvm*, *gwcvm*, *hcvm* e *hwcvm*). Como já fora mencionado anteriormente, estas variantes são bem conhecidas e comumente utilizadas nas soluções para problemas em grafos e otimização combinatória.

Apesar de ser um problema largamente estudado, as heurísticas de pós-processamento implementadas não são os melhores métodos heurísticos conhecidos para o problema CVM. Desta forma, as heurísticas geradas pelo mecanismo evolucionário presente no método *SAHe-CVM* puderam contribuir mais decisivamente

para o aumento do desempenho das rotinas de pós-processamento, resultando em melhores soluções para as instâncias. Vale ressaltar, que este aumento de desempenho gerou melhores soluções ou, ainda, soluções iguais em tempo menor, em mais de 60% das instâncias do conjunto de 1996 [22], e em mais de 80% das instâncias do conjunto de 2010 [62].

Outro fato que confirma a melhora nos resultados, foi a análise comparativa dos resultados do método *SAHe-CVM* com os resultados de outras heurísticas, aplicados ao mesmo conjunto de instâncias. Nesta comparação foi possível verificar que o método *SAHe-CVM* obteve desempenho igual ou superior em mais de 50% das instâncias, quando comparado com os resultados apresentados pelos métodos *Squeeze* (96%), *MIPO* (87%), *XSD* (84%), *CBH* (100%), *ATA* (68%), *OCH* (89%), *WAO* (98%) e *DLS-MC* (54%), sendo inferior apenas ao método *RB-clique* (47%).

Vale ressaltar que o método *SAHe-CVM*, além do método *DLS-MC* citado, resolveu todas as 80 instâncias do conjunto de 1996, enquanto que os demais métodos resolveram, na média, pouco mais da metade dessas instâncias.

Esta abordagem será novamente trabalhada no Capítulo 6, onde algumas conjecturas sobre novos testes e possíveis modificações na abordagem serão elencadas.

# Capítulo 6

## Considerações, Conclusões e Trabalhos Futuros

A partir da análise e do desenvolvimento de algoritmos híbridos para seleção automática de heurísticas aplicáveis a problemas NP-difíceis, constatou-se que é possível utilizar as potencialidades dos diversos métodos heurísticos em cooperação para resolução desses problemas.

Pela análise teórica de um método de seleção de algoritmos, foram constatadas algumas inconsistências no algoritmo e na sua complexidade, presentes na formulação original. A correção dessas inconsistências serviu, primeiramente, para comprovar que é possível desenvolver mecanismos de seleção capazes de efetuar a análise de características presentes na instância e decidir, com base no resultado dessa análise, entre a execução de um algoritmo exato ou o cálculo de uma solução aproximada. Verificou-se, também, que em virtude dessa análise prévia, o *seletor* melhorou as propriedades teóricas dos métodos utilizados, quando comparado a aplicação, dos mesmos, sem o uso da rotina de seleção.

A análise experimental do método, supramencionado, demonstrou que, apesar de proporcionar a melhora das propriedades teóricas dos métodos envolvidos – como a redução no tempo de execução do método exato, a melhora na complexidade da busca por uma solução exata e a melhora na taxa de aproximação do método aproximado –, o *seletor* efetuava uma escolha prematura pelo algoritmo aproximado. Estas descobertas serviram, em um primeiro momento, para o desenvolvimento de uma nova formulação capaz de evitar a escolha prematura do *seletor* original, proporcio-

nando maior possibilidade de resolução pelo algoritmo exato. Outro resultado dessa análise experimental, foi a constatação de que o sucesso teórico dos mecanismos desenvolvidos, nem sempre representa sucesso equivalente na prática.

Em face dos primeiros resultados alcançados, decidiu-se por desenvolver um método experimental de seleção automática de heurísticas (*SAHe*) para os problemas Max-SAT e CVM. Além disso, por meio do estudo de trabalhos relacionados a esses problemas, pode-se constatar que grande parte desses trabalhos exploravam a característica de maximização (ou minimização) por análise local. Concluiu-se, então, que a escolha ou desenvolvimento de algoritmos que façam uso dessas características poderia indicar um caminho para uma solução do PSA para estes problemas.

O principal componente do modelo proposto, chamado de *heurística*, consiste na definição da sequência de variáveis (vértices) a serem visitadas e na ação tomada sobre elas, com base na característica de minimização local, acima mencionada.

Após a execução da heurística, uma rotina de pós-processamento (que também utiliza a mesma característica) é utilizada para finalizar a busca por uma solução viável ótima.

Com a realização de experimentos computacionais sobre instâncias bem conhecidas e referenciadas na literatura, constatou-se que a aplicação do mecanismo evolucionário do AG gerou boas heurísticas, que conduziram as outras heurísticas (utilizadas como rotinas de pós-processamento) a melhores soluções em mais de 50% das instâncias testadas, quando comparadas ao seu uso isolado.

Em face da inexistência de implementações disponíveis dos melhores métodos heurísticos para o problema CVM, um conjunto reduzido de rotinas de pós-processamento simples foram implementadas. A saber, essas rotinas de pós-processamento possuem mecanismos equivalentes aos métodos heurísticos que obtiveram os piores resultados no processo de calibração do método *SAHe-SAT*. Mesmo assim, os resultados encontrados pelo *SAHe-CVM* em conjunto com estas rotinas foram bons, quando comparados aos melhores métodos heurísticos existentes para o problema CVM.

Com relação a um panorama geral, não foi possível relacionar o desempenho do método criado à características de instâncias ou famílias. Por outro lado, os resultados experimentais demonstram que o método desenvolvido possui desempe-

nho robusto sobre todos os conjuntos de instâncias, o que indica que o seletor de heurísticas atingiu o seu objetivo, independente do contexto.

Outra característica importante, apontada pelos experimentos, foi a diminuição dos passos executados pelas heurísticas de pós-processamento até atingir-se a solução ótima. Demonstrando que o mecanismo criado retirou o caráter aleatório das condições iniciais de busca dessas heurísticas, fazendo-as operar em condições iniciais mais favoráveis. Assim, recomenda-se o uso do mecanismo criado em conjunto com métodos heurísticos que façam uso de condições iniciais aleatórias de busca.

Os resultados experimentais também demonstram que, para o conjunto de instâncias consideradas difíceis, o modelo proposto obteve um bom desempenho sobre uma fração significativa das instâncias dos conjuntos de instâncias testadas. Desta maneira, conclui-se que o modelo proposto aqui é recomendado para a resolução de instâncias difíceis.

## 6.1 Trabalhos Futuros

Como apontado anteriormente, nos experimentos computacionais do método desenvolvido para o problema CVM, apenas um conjunto reduzido de rotinas de pós-processamento simples foram implementadas. Desta forma, acredita-se que ao implementar métodos heurísticos mais eficientes como rotinas de pós-processamento do método *SAHe-CVM*, melhores resultados poderão ser atingidos.

Outro fato que pode ser explorado, futuramente, é a modificação do método proposto para que as rotinas de pós-processamento sejam diferentes para cada indivíduo do Algoritmo Genético e que a sua escolha, também faça parte do processo evolucionário do método.

Acreditamos, também, que futuros esforços podem ser usados proveitosamente sobre soluções similares para outros problemas que, assim como Max-SAT e CVM, podem ser expressados como um problema de otimização de restrições sobre variáveis binárias. Como exemplos, tem-se o Conjunto Independente Máximo e o Conjunto Dominante Mínimo sobre grafos, ambos admitindo formulações bem conhecidas deste tipo [37].

No contexto referente aos problemas de *reconhecimento de padrões* e de *clas-*



*sificação*, diversos métodos heurísticos têm sido desenvolvidos, entre eles, as *redes neurais artificiais (RNA's)*. Um dos objetivos das pesquisas sobre *RNA's* é o desenvolvimento de morfologias neurais matemáticas capazes de realizar funções diversas. Em sua grande maioria, modelos neurais são compostos de muitos elementos não lineares que operam em paralelo e que são classificados de acordo com padrões ligados à biologia. Por causa de diferenças entre algumas ou às vezes todas as entidades envolvidas, diferentes estruturas de *RNA's* têm sido desenvolvidas por pesquisadores ([66] e [67] descrevem algumas estruturas de *RNA's*).

Existem muitas variações e tipos de *RNA's*. A melhor escolha para um determinado problema deverá passar pela definição de várias características e componentes, tais como: disposição dos neurônios, tipo de comunicação, tamanho da rede, existência ou não de camadas, número de neurônios em cada camada, tipo de função que define o comportamento do neurônio e tipo do aprendizado.

O processo de escolha de uma determinada *RNA* é realizado, geralmente, por um especialista no problema em que ela deverá ser empregada ou, manualmente, através do teste de um subconjunto reduzido de configurações distintas. Com base nos mecanismos de seleção automática de heurísticas desenvolvido nesta tese, e, nos resultados alcançados, surge a possibilidade de aplicação do mesmo conceito para a escolha e definição automática das características e componentes de uma *RNA's*, para a resolução de problemas de *reconhecimento de padrões* e de *classificação*.

# Referências Bibliográficas

- [1] VASSILEVSKA, V., WILLIAMS, R., WOO, S. L. M. “Confronting hardness using a hybrid approach”. In: *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*, pp. 1–10, New York, NY, 2006. ACM Press.
- [2] GUO, H. *Algorithm Selection for Sorting and Probabilistic Inference: A Machine-Learning Approach*. Tese de Doutorado, Kansas State University, 2003.
- [3] ROPKE, S., PISINGER, D. “An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows”, *Transportation Science*, v. 40, n. 4, pp. 455–472, 2006.
- [4] PULLAN, W., HOOS, H. H. “Dynamic local search for the maximum clique problem”, *Journal of Artificial Intelligence Research*, v. 25, pp. 159–185, 2006.
- [5] VAZIRANI, V. V. *Approximation Algorithms*. Springer, 2001.
- [6] CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L., et al. *Introduction to Algorithms*. 3rd ed. Cambridge, Massachusetts, The MIT Press, 2009.
- [7] BURKE, E. K., KENDALL, G. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer, 2005.
- [8] RICE, J. R. “The algorithm selection problem”. In: Rubinoff, M., Yovits, M. C. (Eds.), *Advances in Computers*, v. 15, Academic Press, pp. 65–118, New York, NY, 1976.

- [9] LEYTON-BROWN, K., NUDELMAN, E., ANDREW, G., et al. “Boosting as a metaphor for algorithm design”. In: Rossi, F. (Ed.), *Principles and Practice of Constraint Programming*, v. 2833, *Lecture Notes in Computer Science*, pp. 899–903, Berlin, Germany, 2003. Springer-Verlag.
- [10] BRODLEY, C. E. “Addressing the selective superiority problem: Automatic Algorithm/Model Class Selection”. In: *The Tenth International Machine Learning Conference*, pp. 17–24. Tufts University, 1993.
- [11] HARTMANN, A. K., WEIGT, M. *Phase Transitions in Combinatorial Optimization Problems: Basics, Algorithms and Statistical Mechanics*. Weinheim, Germany, WileyVCH, 2005.
- [12] LAGOUDAKIS, M. G., LITTMAN, M. L., PARR, R. “Selecting the right algorithm”. In: *Proceedings of the 2001 AAAI Fall Symposium Series: Using Uncertainty within Computation*, pp. 74–75. AAAI Press, 2001.
- [13] GOMES, C. P., SELMAN, B. “Algorithm portfolios”, *Artificial Intelligence*, v. 126(1-2), pp. 43–62, 2001.
- [14] FINK, E. “How to solve it automatically: Selection among problem-solving methods”. In: Simmons, R. G., Veloso, M. M., Smith, S. (Eds.), *Proceedings of Fourth International Conference on Artificial Intelligence Planning Systems*, pp. 128–136, Menlo Park, CA, 1998. AAAI Press.
- [15] RUSSELL, S., SUBRAMANIAN, D. “Provably Bounded-Optimal Agents”, *Journal of Artificial Intelligence Research*, v. 2, pp. 575–609, 1995.
- [16] MINTON, S. “Automatically configuring constraint satisfaction programs: A case study”, *Constraints*, v. 1, n. 1-2, pp. 7–43, 1996.
- [17] SHAW, P. “Using constraint programming and local search methods to solve vehicle routing problems”. In: *Fourth international conference on principles and practice of constraint programming*, v. 1520, *Lecture Notes in Computer Science*, pp. 417–31. Springer-Verlag, 1998.

- [18] PISINGER, D., ROPKE, S. “Large Neighborhood Search”. In: Gendreau, M., Potvin, Y. (Eds.), *Handbook of Metaheuristics*, v. 146, Springer Science+Business Media, pp. 399–419, 2010.
- [19] TOMPKINS, D. A. D., HOOS, H. H. “UBCSAT: An implementation and experimentation environment for SLS algorithms for SAT and MAX-SAT”. In: Hoos, H. H., Mitchell, D. G. (Eds.), *Theory and Applications of Satisfiability Testing*, v. 3542, *Lecture Notes in Computer Science*, pp. 306–320, Berlin, Germany, 2005. Springer Verlag.
- [20] BERRE, D. L., SIMON, L. “Fifty-Five Solvers in Vancouver: The SAT 2004 competition”. In: Mitchell, D. G. (Ed.), *Theory and Applications of Satisfiability Testing*, v. 3542, *Lecture Notes in Computer Science*, pp. 321–344, Berlin, Germany, 2005. Springer-Verlag.
- [21] ARGELICH, J., LI, C. M., MANYÀ, F., et al. “Third Evaluation of Max-SAT Solvers (MaxSAT-2008)”. 2008. <http://www.maxsat.udl.cat/08/>  
Acesso em: 7 de julho de 2009, 11:00:00.
- [22] JOHNSON, D. S., TRICK, M. A. *Cliques, Coloring, and Satisfiability*, v. 26. American Mathematical Society, 1996.
- [23] BARBOSA, V. C., CAMPOS, L. C. D. “A Novel Evolutionary Formulation of the Maximum Independent Set Problem”, *Journal of Combinatorial Optimization*, v. 8, pp. 419–437, 2004.
- [24] PAPADIMITRIOU, C. H. *Computational Complexity*. University of California, San Diego, Addison-Wesley, 1994.
- [25] DANTSIN, E., GAVRILOVICH, M., HIRSCH, E., et al. “MAX SAT approximation beyond the limits of polynomial-time approximation”. In: *Annals of Pure and Applied Logic*, v. 113, pp. 81–94, 2001.
- [26] LI, C. M., MANYÀ, F., PLANES, J. “New Inference Rules for Max-SAT”, *Journal of Artificial Intelligence Research*, v. 30, pp. 321–359, 2007.

- [27] ARGELICH, J., LI, C. M., MANYÀ, F., et al. “Second Evaluation of Max-SAT Solvers (MaxSAT-2007)”. 2007. <http://www.maxsat.udl.cat/>. Acesso em: 7 de julho de 2009, 11:00:00.
- [28] MANQUINHO, V. M., SILVA, J. P. M., PLANES, J. “Algorithms for weighted boolean optimization”. In: *Theory and Applications of Satisfiability Testing*, v. 5584, *Lecture Notes in Computer Science*, pp. 495–508, Berlin, Germany, 2009. Springer-Verlag.
- [29] SELMAN, B., LEVESQUE, H., MITCHELL, D. “A new method for solving hard satisfiability problems”. In: *Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 459–465, Menlo Park, CA, 1992. AAAI Press.
- [30] SELMAN, B., KAUTZ, H. “Domain-independent extensions to GSAT: Solving large structured variables”. In: *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pp. 290–295, San Mateo, CA, 1993. Morgan Kaufmann.
- [31] SELMAN, B., KAUTZ, H., COHEN, B. “Noise Strategies for Improving Local Search”. In: *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pp. 337–343. AAAI Press, 1994.
- [32] MCALLESTER, D., SELMAN, B., KAUTZ, H. “Evidence for invariants in local search”. In: *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pp. 321–326, Menlo Park, CA, 1997. AAAI Press.
- [33] HOOS, H. H. “An adaptive noise mechanism for WalkSAT”. In: *Proceedings of the Eighteenth National Conference in Artificial Intelligence*, pp. 655–660, Menlo Park, CA, 2002. AAAI Press.
- [34] HUTTER, F., TOMPKINS, D. A. D., HOOS, H. H. “Scaling and probabilistic smoothing: Efficient dynamic local search for SAT”. In: van Hentenryck, P. (Ed.), *Principles and Practice of Constraint Programming*, v. 2470, *Lecture Notes in Computer Science*, pp. 233–248, Berlin, Germany, 2002. Springer-Verlag.

- [35] TOMPKINS, D. A. D., HOOS, H. H. “Warped landscapes and random acts of SAT solving”. In: *Proceedings of the Eighth International Symposium on Artificial Intelligence and Mathematics*, Piscataway, NJ, 2004. RUTCOR.
- [36] THORNTON, J., BAIN, S., SATTAR, A., et al. “A Two Level Local Search for MAX-SAT Problems with Hard and Soft Constraints”. In: *The 15th Australian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence*, v. 2557, pp. 603–614, London, UK, 2002. Springer-Verlag.
- [37] BARBOSA, V. C., GAFNI, E. “A Distributed Implementation of Simulated Annealing”, *Journal of Parallel and Distributed Computing*, v. 6, pp. 411–434, 1989.
- [38] MANNINO, C., SASSANO, A. “Edge projection and the maximum cardinality stable set problem”. In: JOHNSON e TRICK [22], pp. 205–219.
- [39] LOVÁSZ, L., PLUMMER, M. D. *Matching Theory*. Amsterdam, The Netherlands, North-Holland, 1986.
- [40] BOURJOLLY, J.-M., GILL, P., LAPORTE, G., et al. “An exact quadratic 0-1 algorithm for the stable set problem”. In: JOHNSON e TRICK [22], pp. 53–73.
- [41] BALAS, E., CERIA, S., CORNUÉJOLS, G., et al. “Polyhedral methods for the maximum clique problem”. In: JOHNSON e TRICK [22], pp. 11–28.
- [42] BALAS, E., NIEHAUS, W. “Finding large cliques in arbitrary graphs by bipartite matching”. In: JOHNSON e TRICK [22], pp. 29–51.
- [43] GOLDBERG, M. K., RIVENBURGH, R. D. “Constructing cliques using restricted backtracking”. In: JOHNSON e TRICK [22], pp. 89–101.
- [44] SORIANO, P., GENDREAU, M. “Tabu search algorithms for the maximum clique problem”. In: JOHNSON e TRICK [22], pp. 221–241.
- [45] JAGOTA, A., SANCHIS, L., GANESAN, R. “Approximately solving Maximum Clique using neural network and related heuristics”. In: JOHNSON e TRICK [22], pp. 169–204.

- [46] GIBBONS, L. E., HEARN, D. H., PARDALOS, P. M. “A continuous based heuristic for the maximum clique problem”. In: JOHNSON e TRICK [22], pp. 103–124.
- [47] GROSSMAN, T. “Applying the INN model to the Maximum Clique problem”. In: JOHNSON e TRICK [22], pp. 125–145.
- [48] AGGARWAL, C. C., ORLIN, J. B., TAI, R. P. “Optimized Crossover for the Independent Set Problem”, *Operations Research*, v. 45, n. 2, pp. 226–234, mar. 1997.
- [49] KOTTECHA, K., GAMBHAVA, N. “A Hybrid Genetic Algorithm for Minimum Vertex Cover Problem”. In: *Indian International Conference on Artificial Intelligence*, pp. 904–913, 2003.
- [50] MITCHELL, M. *An Introduction to Genetic Algorithms*. Cambridge, MA, The MIT Press, 1996.
- [51] CAI, S., SU, K., CHEN, Q. “EWLS: A New Local Search for Minimum Vertex Cover”. In: *The Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, pp. 45–50, Menlo Park, California, 2010. The AAAI Press.
- [52] “The International Conferences on Theory and Applications of Satisfiability Testing (SAT)”. 2008. <http://www.satisfiability.org/>. Acesso em: 7 de julho de 2009, 11:00:00.
- [53] “DIMACS Challenge-Satisfiability: Suggested Format”. 1993. <ftp://dimacs.rutgers.edu/pub/challenge/satisfiability/>. Acesso em: 7 de julho de 2009, 10:00:00.
- [54] MONASSON, R., ZECCHINA, R., KIRKPATRICK, S., et al. “Determining computacional complexity from characteristic ‘phase transitions””, *Nature*, v. 400, n. 8, pp. 133–137, 1999.
- [55] SHEN, H., ZHANG, H. “An empirical study of max-2-sat phase transitions”. In: *Electronic Notes in Discrete Mathematics*, v. 16, pp. 80–92, Ottawa, CA, 2003. Elsevier B.V.

- [56] GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. MA, Addison-Wesley Reading, 1989.
- [57] KIRKPATRICK, S., GELATT, C. D., VECCHI, M. P. “Optimization by simulated annealing”, *Science*, v. 220, pp. 671–680, 1983.
- [58] RIGO JR., L. O. *Evolução de Autômatos Celulares para a Previsão de Séries Temporais Correlacionadas*. Tese de Mestrado, Departamento de Engenharia de Sistemas e Computação / COPPE - UFRJ, Rio de Janeiro, Brasil, Abril 2005.
- [59] MAZURE, B., SAIS, L., GREGOIRE, E. “Tabu search for SAT”. In: *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, pp. 281–285, Menlo Park, CA, 1997. AAAI Press.
- [60] GENT, I. P., WALSH, T. “Towards an understanding of hill-climbing procedures for SAT”. In: *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pp. 28–33, Menlo Park, CA, 1993. AAAI Press.
- [61] GENT, I. P., WALSH, T. “Unsatisfied variables in local search”. In: Hallam, J. (Ed.), *Hybrid Problems, Hybrid Solutions*, pp. 73–85, Amsterdam, The Netherlands, 1995. IOS Press.
- [62] XU, K. “Benchmarks with Hidden Optimum Solutions for Graph Problems (Maximum Clique, Maximum Independent Set, Minimum Vertex Cover and Vertex Coloring)”. 2010. <http://www.nlsde.buaa.edu.cn/~kexu/benchmarks/graph-benchmarks.htm>. Acesso em: 10 de janeiro de 2011, 20:00:00.
- [63] XU, K., LI, W. “Exact Phase Transitions in Random Constraint Satisfaction Problems”, *Journal of Artificial Intelligence Research*, v. 12, pp. 93–103, 2000.
- [64] BROCKINGTON, M., CULBERSON, J. C. “Camouflaging Independent Sets in Quasi-Random Graphs”. In: Johnson, D. S., Trick, M. A. (Eds.), *Cliques, Coloring, and Satisfiability*, v. 26, pp. 75–88. American Mathematical Society, 1996.



- [65] TRICK, M. A. “Appendix: Second DIMACS Implementation Challenge”. In: JOHNSON e TRICK [22], pp. 653–657.
- [66] RUSSELL, S. J., NORVIG, P. *Artificial intelligence: a modern approach*. Prentice Hall series in artificial intelligence. Prentice Hall, 2003.
- [67] HAYKIN, S. S. *Redes Neurais - Princípios e Prática*. Bookman, 2001.

## Apêndice A

### Instâncias de *Benchmark* para o Problema *Max-SAT*

Tabela A.1: Conjunto de instâncias de 2004 [20].

| Instância    | $n$   | $m$    | <i>Algoritmo Exato MSUnCore</i> |          |
|--------------|-------|--------|---------------------------------|----------|
|              |       |        | Num. cls. sat.                  | Tempo(m) |
| alu4mul      | 4736  | 30465  | 30464                           | 336.849  |
| c3540mul     | 5248  | 33199  |                                 |          |
| c6288mul     | 9540  | 61421  |                                 |          |
| c7552mul     | 11282 | 69529  | 69528                           | 18.577   |
| C880mul      | 1612  | 9373   | 9372                            | 26.570   |
| dalumul      | 9426  | 59991  |                                 |          |
| desmul       | 28902 | 179895 | 179894                          | 75.617   |
| frg1mul      | 3230  | 20575  |                                 |          |
| frg2mul      | 10316 | 62943  | 62942                           | 101.746  |
| i10mul       | 12998 | 77941  | 77940                           | 311.919  |
| i8mul        | 14524 | 91139  | 91138                           | 565.855  |
| k2mul        | 11680 | 74581  |                                 |          |
| rotmul       | 5980  | 35229  | 35228                           | 22.781   |
| term1mul     | 3504  | 22229  | 22228                           | 826.832  |
| vdumul       | 5444  | 34509  | 34508                           | 99.402   |
| x1mul        | 8760  | 55571  |                                 |          |
| am_4_4       | 433   | 1458   | 1457                            | 0.208    |
| am_5_5       | 1076  | 3677   | 3676                            | 4.284    |
| am_6_6       | 2269  | 7814   |                                 |          |
| am_7_7       | 4264  | 14751  |                                 |          |
| am_8_8       | 7361  | 25538  |                                 |          |
| am_9_9       | 11908 | 41393  |                                 |          |
| li-exam-61   | 28147 | 108436 |                                 |          |
| li-exam-62   | 28147 | 108436 |                                 |          |
| li-exam-63   | 28147 | 108436 |                                 |          |
| li-exam-64   | 28147 | 108436 |                                 |          |
| li-test4-100 | 36809 | 142491 |                                 |          |
| li-test4-101 | 36809 | 142491 |                                 |          |
| li-test4-94  | 36809 | 142491 |                                 |          |
| li-test4-95  | 36809 | 142491 |                                 |          |
| li-test4-96  | 36809 | 142491 |                                 |          |
| li-test4-97  | 36809 | 142491 |                                 |          |
| li-test4-98  | 36809 | 142491 |                                 |          |
| li-test4-99  | 36809 | 142491 |                                 |          |
| ferry10      | 2958  | 20791  | 20791                           | 0.104    |
| ferry10u     | 2883  | 20259  | 20259                           | 0.076    |
| ferry11      | 3562  | 26105  | 26105                           | 0.104    |
| ferry11u     | 3480  | 25499  | 25499                           | 0.068    |
| ferry12      | 4222  | 32199  | 32199                           | 1.232    |
| ferry12u     | 4133  | 31515  | 31515                           | 1.276    |
| ferry8       | 1918  | 12311  | 12311                           | 0.092    |
| ferry8u      | 1857  | 11915  | 11915                           | 0.044    |
| ferry9       | 2410  | 16209  | 16209                           | 0.060    |
| ferry9u      | 2342  | 15747  | 15747                           | 0.072    |
| gripper10    | 2438  | 19700  | 19700                           | 11.073   |
| gripper10u   | 2312  | 18666  |                                 |          |
| gripper11    | 3222  | 27199  | 27199                           | 68.644   |
| gripper11u   | 3084  | 26019  |                                 |          |
| gripper12    | 3502  | 30746  | 30746                           | 72.477   |
| gripper12u   | 3352  | 29412  |                                 |          |

Tabela A.2: Conjunto de instâncias de 2004 [20]. (continuação)

| Instância           | $n$    | $m$     | Algoritmo Exato <i>MSUnCore</i> |          |
|---------------------|--------|---------|---------------------------------|----------|
|                     |        |         | Num. cls. sat.                  | Tempo(m) |
| gripper13           | 4430   | 40461   | 40461                           | 243.327  |
| gripper13u          | 4268   | 38965   |                                 |          |
| gripper14           | 4758   | 45056   | 45056                           | 625.603  |
| gripper14u          | 4584   | 43390   |                                 |          |
| hanoi4              | 1404   | 18058   | 18058                           | 1.016    |
| hanoi4u             | 1312   | 16856   | 16855                           | 2.388    |
| hanoi5              | 4434   | 76235   | 76235                           | 31.366   |
| hanoi5u             | 4292   | 73777   | 73776                           | 69.952   |
| abp1-1-k31          | 14809  | 48483   | 48482                           | 114.259  |
| abp4-1-k31          | 14809  | 48483   | 48482                           | 77.765   |
| bc56-sensors-1-k391 | 561371 | 1778987 |                                 |          |
| bc56-sensors-2-k592 | 850398 | 2694319 |                                 |          |
| bc57-sensors-1-k303 | 435701 | 1379987 |                                 |          |
| dme-03-1-k247       | 261352 | 773077  |                                 |          |
| guidance-1-k56      | 98746  | 307346  | 307345                          | 3979.900 |
| motors-stuck-1-k407 | 654766 | 2068742 |                                 |          |
| motors-stuck-2-k314 | 505536 | 1596837 |                                 |          |
| motors-stuck-2-k315 | 507145 | 1601920 | 1601920                         | 2493.320 |
| valves-gates-1-k617 | 985042 | 3113540 |                                 |          |
| 6pipe_6_ooo         | 17064  | 545612  | 545611                          | 210.885  |
| 6pipe               | 15800  | 394739  |                                 |          |
| 7pipe               | 23910  | 751118  |                                 |          |
| cnf-r4-b1-k1.1-comp | 2424   | 14812   | 14812                           | 163.602  |
| cnf-r4-b1-k1.2-comp | 2424   | 14812   | 14812                           | 190.244  |
| cnt10               | 20470  | 68561   | 68561                           | 30.226   |
| comb1               | 5910   | 16804   |                                 |          |
| comb2               | 31933  | 112462  | 112461                          | 47.635   |
| comb3               | 4774   | 16331   | 16330                           | 206.113  |
| dp11u10             | 9197   | 25271   | 25270                           | 1266.940 |
| dp12u11             | 11137  | 30792   |                                 |          |
| f2clk_40            | 27568  | 80439   | 80438                           | 365.115  |
| f2clk_50            | 34678  | 101319  |                                 |          |
| fifo8_300           | 194762 | 530713  |                                 |          |
| fifo8_400           | 259762 | 707913  | 707912                          | 8325.730 |
| homer17             | 286    | 1742    |                                 |          |
| homer18             | 308    | 2030    |                                 |          |
| homer19             | 330    | 2340    |                                 |          |
| homer20             | 440    | 4220    |                                 |          |
| ip36                | 47273  | 153368  | 153367                          | 19.273   |
| ip38                | 49967  | 162142  | 162141                          | 33.622   |
| ip50                | 66131  | 214786  | 214785                          | 46.955   |
| k2fix_gr_2pinvar_w8 | 3771   | 270136  |                                 |          |
| k2fix_gr_2pinvar_w9 | 5028   | 307674  |                                 |          |
| k2fix_gr_2pin_w8    | 9882   | 295998  |                                 |          |
| k2fix_gr_2pin_w9    | 13176  | 345426  |                                 |          |
| k2fix_gr_rcs_w8     | 10056  | 271393  |                                 |          |
| sha1                | 61377  | 255417  |                                 |          |
| sha2                | 61377  | 255417  |                                 |          |
| w08_14              | 120367 | 425316  | 425316                          | 171.335  |
| w08_15              | 132555 | 469519  | 469519                          | 245.107  |

Tabela A.3: Conjunto de instâncias de 2008 [21].

| Instância                            | $n$    | $m$     | <i>Algoritmo Exato MSUnCore</i> |          |
|--------------------------------------|--------|---------|---------------------------------|----------|
|                                      |        |         | Num. cls. sat.                  | Tempo(m) |
| b14_opt_bug2_vec1-gate-0             | 130328 | 402707  | 402706                          | 4.660    |
| b15-bug-fourvec-gate-0               | 581064 | 1712690 | 1712686                         | 109.439  |
| b15-bug-onevec-gate-0                | 121836 | 359040  | 359039                          | 11.365   |
| c1_DD_s3_f1_e2_v1-bug-fourvec-gate-0 | 391897 | 989885  | 989881                          | 6.416    |
| c1_DD_s3_f1_e2_v1-bug-onevec-gate-0  | 102234 | 258294  | 258293                          | 0.980    |
| c2_DD_s3_f1_e2_v1-bug-fourvec-gate-0 | 400085 | 1121810 | 1121806                         | 13.165   |
| c2_DD_s3_f1_e2_v1-bug-onevec-gate-0  | 84525  | 236942  | 236941                          | 1.720    |
| c3_DD_s3_f1_e1_v1-bug-fourvec-gate-0 | 33540  | 86944   | 86940                           | 0.472    |
| c3_DD_s3_f1_e1_v1-bug-onevec-gate-0  | 8385   | 21736   | 21735                           | 0.052    |
| c4_DD_s3_f1_e1_v1-bug-gate-0         | 797728 | 2011216 | 2011208                         | 26.146   |
| c4_DD_s3_f1_e2_v1-bug-fourvec-gate-0 | 448465 | 1130672 | 1130668                         | 9.381    |
| c4_DD_s3_f1_e2_v1-bug-onevec-gate-0  | 131584 | 331754  | 331753                          | 1.212    |
| c5315-bug-gate-0                     | 1880   | 5049    | 5048                            | 0.020    |
| c5_DD_s3_f1_e1_v1-bug-fourvec-gate-0 | 100472 | 270492  | 270488                          | 1.360    |
| c5_DD_s3_f1_e1_v1-bug-gate-0         | 200944 | 540984  | 540976                          | 5.080    |
| c5_DD_s3_f1_e1_v1-bug-onevec-gate-0  | 25118  | 67623   | 67622                           | 0.204    |
| c5_DD_s3_f1_e1_v2-bug-gate-0         | 200944 | 540984  | 540976                          | 4.348    |
| c6288-bug-gate-0                     | 3462   | 9285    | 9284                            | 0.364    |
| c6_DD_s3_f1_e1_v1-bug-fourvec-gate-0 | 170019 | 454050  | 454046                          | 3.324    |
| c6_DD_s3_f1_e1_v1-bug-gate-0         | 298058 | 795900  | 795892                          | 7.680    |
| c6_DD_s3_f1_e1_v1-bug-onevec-gate-0  | 44079  | 117720  | 117719                          | 0.412    |
| c6_DD_s3_f1_e2_v1-bug-fourvec-gate-0 | 170019 | 454050  | 454046                          | 3.664    |
| c7552-bug-gate-0                     | 2640   | 7008    | 7007                            | 0.140    |
| divider-problem.dimacs_10            | 234004 | 760965  | 760965                          | 5.148    |
| divider-problem.dimacs_11            | 215964 | 709377  | 709375                          | 31.438   |
| divider-problem.dimacs_12            | 229482 | 751921  | 751920                          | 8.169    |
| divider-problem.dimacs_1             | 215676 | 708801  | 708800                          | 7.664    |
| divider-problem.dimacs_2             | 228874 | 750705  | 750703                          | 190.536  |
| divider-problem.dimacs_3             | 216900 | 711249  | 711248                          | 7.076    |
| divider-problem.dimacs_4             | 225340 | 743637  | 743636                          | 9.569    |
| divider-problem.dimacs_5             | 228874 | 750705  | 750703                          | 62.028   |
| divider-problem.dimacs_6             | 234004 | 760965  | 760965                          | 6.100    |
| divider-problem.dimacs_7             | 239640 | 787745  | 787744                          | 8.257    |
| divider-problem.dimacs_8             | 246943 | 810105  | 810103                          | 70.308   |
| divider-problem.dimacs_9             | 222851 | 730905  | 730904                          | 8.701    |
| dividers10                           | 45552  | 162874  | 162872                          | 23.081   |
| dividers11                           | 45552  | 162982  | 162981                          | 3.436    |
| dividers1                            | 45552  | 162982  | 162981                          | 2.188    |
| dividers2                            | 45552  | 162982  | 162981                          | 0.836    |
| dividers3                            | 45552  | 162982  | 162981                          | 3.012    |
| dividers4                            | 45552  | 162983  | 162981                          | 2.196    |
| dividers5                            | 45552  | 162982  | 162981                          | 2.180    |
| dividers6_hack                       | 35376  | 132699  | 132698                          | 1.996    |
| dividers7                            | 64532  | 231332  | 231331                          | 1.996    |
| dividers8                            | 45552  | 162982  | 162981                          | 1.948    |
| dividers9                            | 45552  | 162982  | 162981                          | 2.300    |
| dividers_multivec1                   | 106128 | 397650  | 397648                          | 3.504    |
| fpu1-problem                         | 119286 | 409139  | 409138                          | 2.764    |
| fpu2-problem                         | 160184 | 548699  | 548698                          | 3.328    |
| fpu3_hack-problem                    | 96384  | 347489  | 347488                          | 2.316    |
| fpu3-problem                         | 120114 | 410963  | 410962                          | 2.776    |
| fpu4-problem                         | 120150 | 411276  | 411275                          | 2.756    |
| fpu5-problem                         | 320400 | 1099039 | 1099038                         | 5.568    |
| fpu6-problem                         | 160160 | 548748  | 548748                          | 3.116    |
| fpu7_hack-problem                    | 96438  | 347843  | 347842                          | 3.860    |
| fpu7-problem                         | 120174 | 411323  | 411322                          | 4.024    |

Tabela A.4: Conjunto de instâncias de 2008 [21]. (continuação)

| Instância                        | $n$     | $m$      | Algoritmo Exato MSUnCore |          |
|----------------------------------|---------|----------|--------------------------|----------|
|                                  |         |          | Num. cls. sat.           | Tempo(m) |
| fpu8-problem                     | 160232  | 548848   | 548847                   | 4.768    |
| fpu_fsm1-problem                 | 160200  | 548843   | 548842                   | 3.436    |
| fpu_multivec1-problem            | 257168  | 928310   | 928308                   | 5.252    |
| i2c_master1                      | 82429   | 285987   | 285986                   | 6.328    |
| i2c_master2                      | 63816   | 221320   | 221319                   | 7.064    |
| i2c-problem.dimacs_25            | 521672  | 1581471  | 1581469                  | 20.829   |
| i2c-problem.dimacs_26            | 397668  | 1205454  | 1205452                  | 38.962   |
| mem_ctrl1                        | 1128648 | 4422185  | 4422184                  | 18.505   |
| mem_ctrl2_blackbox_mc_dp-problem | 1974822 | 6795573  | 6795570                  | 101.818  |
| mem_ctrl1-problem                | 4426323 | 15983633 | 15983632                 | 75.217   |
| mot_comb1._red-gate-0            | 2159    | 5326     | 5325                     | 0.016    |
| mot_comb2._red-gate-0            | 5484    | 13894    | 13893                    | 0.044    |
| mot_comb3._red-gate-0            | 11265   | 29520    | 29519                    | 0.092    |
| mrisc_mem2wire1                  | 168960  | 641598   | 641597                   | 4.296    |
| mrisc_mem2wire-problem           | 844900  | 2905976  | 2905975                  | 13.877   |
| rsdecoder1_blackbox_CSEEBlock    | 277950  | 806460   | 806456                   | 819.601  |
| rsdecoder1_blackbox_KESBlock     | 707330  | 1106376  |                          |          |
| rsdecoder1                       | 238056  | 935187   | 935186                   | 14.629   |
| rsdecoder2_blackbox-problem      | 927691  | 2728843  | 2728843                  | 60.836   |
| rsdecoder2                       | 415480  | 1632526  | 1632525                  | 130.564  |
| rsdecoder3                       | 238290  | 936006   | 936005                   | 14.785   |
| rsdecoder4                       | 237783  | 933978   |                          |          |
| rsdecoder5                       | 238290  | 936006   | 936004                   | 124.051  |
| rsdecoder6                       | 238290  | 936006   | 936003                   | 1321.980 |
| rsdecoder_fsm1                   | 238290  | 936006   | 936005                   | 14.597   |
| rsdecoder_fsm2                   | 238290  | 936006   | 936004                   | 43.539   |
| rsdecoder_multivec1              | 394446  | 1626312  | 1626308                  | 913.573  |
| rsdecoder_multivec1-problem      | 627993  | 2125620  | 2125616                  | 71.024   |
| rsdecoder-problem.dimacs_31      | 1197376 | 3863287  | 3863286                  | 27.698   |
| rsdecoder-problem.dimacs_34      | 226040  | 728516   | 728515                   | 8.561    |
| rsdecoder-problem.dimacs_36      | 1220616 | 3938467  | 3938466                  | 32.178   |
| rsdecoder-problem.dimacs_37      | 1513544 | 4909231  | 4909230                  | 42.783   |
| rsdecoder-problem.dimacs_38      | 1198012 | 3865513  |                          |          |
| rsdecoder-problem.dimacs_39      | 1199602 | 3868693  | 3868692                  | 29.018   |
| rsdecoder-problem.dimacs_40      | 1220616 | 3938467  | 3938466                  | 215.393  |
| rsdecoder-problem.dimacs_41      | 1186710 | 3829036  |                          |          |
| s15850-bug-fourvec-gate-0        | 88544   | 206252   | 206248                   | 1.072    |
| s15850-bug-onevec-gate-0         | 22136   | 51563    | 51562                    | 0.164    |
| s38584-bug-onevec-gate-0         | 314272  | 819830   | 819829                   | 3.860    |
| SM_AS_TOP_buggy1                 | 145900  | 694438   | 694381                   | 284.381  |
| SM_MAIN_MEM_buggy1               | 870975  | 3812147  |                          |          |
| SM_RX_TOP                        | 235456  | 934091   | 934085                   | 19.865   |
| spi2                             | 124260  | 515813   | 515812                   | 14.089   |
| spi-problem                      | 628192  | 2234134  | 2234133                  | 27.178   |
| wb1                              | 49525   | 140091   | 139873                   | 13.285   |
| wb2                              | 49490   | 140056   | 139468                   | 40.651   |
| wb_4m8s1                         | 463080  | 1759150  |                          |          |
| wb_4m8s3                         | 463080  | 1759150  | 1759142                  | 13.309   |
| wb_4m8s4                         | 463080  | 1759150  |                          |          |
| wb_4m8s-problem.dimacs_47        | 2691648 | 8517027  |                          |          |
| wb_4m8s-problem.dimacs_48        | 2766036 | 8774655  | 8774647                  | 78.929   |
| wb_4m8s-problem.dimacs_49        | 2785108 | 8812799  |                          |          |
| wb_conmax1                       | 277950  | 1221020  |                          |          |
| wb_conmax3                       | 277950  | 1221020  |                          |          |
| wb-problem.dimacs_45             | 309491  | 806440   | 806426                   | 10.201   |
| wb-problem.dimacs_46             | 300846  | 789283   | 788807                   | 243.539  |

## Apêndice B

# Resultados Experimentais Completos do *SAT*

Tabela B.1: Resultados completos do *SAHe-SAT* e  $H_p = \text{Adapt novelty+}$ , para instâncias do conjunto de 2004 [20].

| Instância    | $n$   | $m$    | <i>SAHe-SAT+Adapt novelty+</i> |               | <i>Adapt novelty+</i> |          |
|--------------|-------|--------|--------------------------------|---------------|-----------------------|----------|
|              |       |        | Num. cls. sat.                 | Tempo(m)      | Num. cls. sat.        | Tempo(m) |
| alu4mul      | 4736  | 30465  | <b>30456</b>                   | 36.976        | 30449                 | 93.340   |
| c3540mul     | 5248  | 33199  | <b>33158</b>                   | 230.755       | 33156                 | 80.479   |
| c6288mul     | 9540  | 61421  | <b>61385</b>                   | 1.100         | 61382                 | 36.361   |
| c7552mul     | 11282 | 69529  | 69307                          | 74.947        | 69318                 | 47.749   |
| C880mul      | 1612  | 9373   | <b>9372</b>                    | <b>0.194</b>  | 9372                  | 0.508    |
| dalumul      | 9426  | 59991  | <b>59924</b>                   | 50.854        | 59920                 | 44.117   |
| desmul       | 28902 | 179895 | 179695                         | 69.812        | 179697                | 63.611   |
| frg1mul      | 3230  | 20575  | <b>20574</b>                   | <b>0.495</b>  | 20574                 | 1.290    |
| frg2mul      | 10316 | 62943  | 62891                          | 48.930        | 62893                 | 119.656  |
| i10mul       | 12998 | 77941  | <b>77769</b>                   | 52.594        | 77758                 | 10.510   |
| i8mul        | 14524 | 91139  | 91044                          | 28.027        | 91044                 | 16.167   |
| k2mul        | 11680 | 74581  | <b>74418</b>                   | 112.313       | 74407                 | 16.932   |
| rotmul       | 5980  | 35229  | <b>35227</b>                   | 133.946       | 35226                 | 113.432  |
| term1mul     | 3504  | 22229  | <b>22227</b>                   | <b>63.729</b> | 22227                 | 72.490   |
| vdamul       | 5444  | 34509  | <b>34471</b>                   | 100.078       | 34468                 | 108.568  |
| x1mul        | 8760  | 55571  | <b>55570</b>                   | <b>1.145</b>  | 55570                 | 2.477    |
| am_4_4       | 433   | 1458   | <b>1457</b>                    | <b>0.170</b>  | 1457                  | 0.232    |
| am_5_5       | 1076  | 3677   | <b>3676</b>                    | <b>0.095</b>  | 3676                  | 0.251    |
| am_6_6       | 2269  | 7814   | <b>7813</b>                    | <b>0.103</b>  | 7813                  | 0.267    |
| am_7_7       | 4264  | 14751  | <b>14750</b>                   | <b>0.363</b>  | 14750                 | 0.558    |
| am_8_8       | 7361  | 25538  | <b>25525</b>                   | 237.715       | 25523                 | 180.424  |
| am_9_9       | 11908 | 41393  | <b>41298</b>                   | 81.680        | 41293                 | 53.522   |
| li-exam-61   | 28147 | 108436 | 108027                         | 36.642        | 108045                | 85.697   |
| li-exam-62   | 28147 | 108436 | 108031                         | 45.118        | 108032                | 17.074   |
| li-exam-63   | 28147 | 108436 | <b>108033</b>                  | 17.467        | 108026                | 49.837   |
| li-exam-64   | 28147 | 108436 | <b>108041</b>                  | 80.617        | 108037                | 49.076   |
| li-test4-100 | 36809 | 142491 | 141896                         | 60.697        | 141902                | 35.010   |
| li-test4-101 | 36809 | 142491 | 141894                         | 44.930        | 141904                | 105.123  |
| li-test4-94  | 36809 | 142491 | 141894                         | 36.740        | 141908                | 37.992   |
| li-test4-95  | 36809 | 142491 | <b>141923</b>                  | 91.495        | 141904                | 95.294   |
| li-test4-96  | 36809 | 142491 | 141902                         | 100.277       | 141908                | 64.973   |
| li-test4-97  | 36809 | 142491 | 141908                         | 62.687        | 141920                | 33.317   |
| li-test4-98  | 36809 | 142491 | 141892                         | 79.786        | 141907                | 52.417   |
| li-test4-99  | 36809 | 142491 | <b>141911</b>                  | 41.042        | 141906                | 12.445   |
| ferry10      | 2958  | 20791  | <b>20791</b>                   | <b>0.159</b>  | 20791                 | 0.391    |
| ferry10u     | 2883  | 20259  | <b>20259</b>                   | <b>0.362</b>  | 20259                 | 0.386    |
| ferry11      | 3562  | 26105  | 26105                          | 1.345         | 26105                 | 1.209    |
| ferry11u     | 3480  | 25499  | <b>25499</b>                   | <b>0.438</b>  | 25499                 | 0.803    |
| ferry12      | 4222  | 32199  | 32198                          | 61.143        | 32198                 | 32.646   |
| ferry12u     | 4133  | 31515  | 31513                          | 29.650        | 31513                 | 2.501    |
| ferry8       | 1918  | 12311  | <b>12311</b>                   | <b>0.140</b>  | 12311                 | 0.343    |
| ferry8u      | 1857  | 11915  | <b>11915</b>                   | <b>0.136</b>  | 11915                 | 0.338    |
| ferry9       | 2410  | 16209  | <b>16209</b>                   | <b>0.147</b>  | 16209                 | 0.490    |
| ferry9u      | 2342  | 15747  | <b>15747</b>                   | <b>0.147</b>  | 15747                 | 0.365    |
| gripper10    | 2438  | 19700  | <b>19699</b>                   | <b>1.726</b>  | 19699                 | 22.047   |
| gripper10u   | 2312  | 18666  | <b>18665</b>                   | <b>1.468</b>  | 18665                 | 5.913    |
| gripper11    | 3222  | 27199  | <b>27198</b>                   | <b>0.732</b>  | 27198                 | 23.529   |
| gripper11u   | 3084  | 26019  | 26018                          | 27.940        | 26018                 | 15.914   |
| gripper12    | 3502  | 30746  | <b>30745</b>                   | <b>5.423</b>  | 30745                 | 9.139    |
| gripper12u   | 3352  | 29412  | 29411                          | 71.192        | 29411                 | 56.930   |



Tabela B.2: Resultados completos do *SAHe-SAT* e  $H_p = \text{Adaptnovelty+}$ , para instâncias do conjunto de 2004 [20]. (continuação)

| Instância           | $n$    | $m$     | <i>SAHe-SAT+Adaptnovelty+</i> |               | <i>Adaptnovelty+</i> |          |
|---------------------|--------|---------|-------------------------------|---------------|----------------------|----------|
|                     |        |         | Num. cls. sat.                | Tempo(m)      | Num. cls. sat.       | Tempo(m) |
| gripper13           | 4430   | 40461   | 40459                         | 9.671         | 40459                | 1.356    |
| gripper13u          | 4268   | 38965   | 38963                         | 1.809         | 38963                | 0.454    |
| gripper14           | 4758   | 45056   | <b>45055</b>                  | 59.306        | 45054                | 1.376    |
| gripper14u          | 4584   | 43390   | <b>43389</b>                  | 59.523        | 43388                | 0.456    |
| hanoi4              | 1404   | 18058   | 18057                         | 3.436         | 18057                | 3.314    |
| hanoi4u             | 1312   | 16856   | <b>16855</b>                  | <b>0.222</b>  | 16855                | 1.086    |
| hanoi5              | 4434   | 76235   | 76232                         | 27.134        | 76232                | 7.171    |
| hanoi5u             | 4292   | 73777   | <b>73774</b>                  | <b>0.870</b>  | 73774                | 3.974    |
| abp1-1-k31          | 14809  | 48483   | 48152                         | 14.261        | 48156                | 9.895    |
| abp4-1-k31          | 14809  | 48483   | <b>48156</b>                  | 102.109       | 48155                | 36.851   |
| bc56-sensors-1-k391 | 561371 | 1778987 | 1623366                       | 85.786        | 1623430              | 36.854   |
| bc56-sensors-2-k592 | 850398 | 2694319 | <b>2394072</b>                | 122.112       | 2393894              | 6.102    |
| bc57-sensors-1-k303 | 435701 | 1379987 | <b>1282461</b>                | 100.646       | 1282338              | 34.813   |
| dme-03-1-k247       | 261352 | 773077  | <b>747079</b>                 | 102.536       | 746975               | 75.608   |
| guidance-1-k56      | 98746  | 307346  | 303767                        | 28.828        | 303813               | 110.182  |
| motors-stuck-1-k407 | 654766 | 2068742 | <b>1867495</b>                | 70.249        | 1867266              | 56.079   |
| motors-stuck-2-k314 | 505536 | 1596837 | <b>1467190</b>                | 94.221        | 1467153              | 104.632  |
| motors-stuck-2-k315 | 507145 | 1601920 | 1471468                       | 91.534        | 1471586              | 82.470   |
| valves-gates-1-k617 | 985042 | 3113540 | <b>2742563</b>                | 71.558        | 2742310              | 18.078   |
| 6pipe_6_ooo         | 17064  | 545612  | <b>545262</b>                 | 44.949        | 545208               | 181.612  |
| 6pipe               | 15800  | 394739  | 393814                        | 82.111        | 393829               | 81.745   |
| 7pipe               | 23910  | 751118  | 749667                        | 125.703       | 749670               | 145.208  |
| cnf-r4-b1-k1.1-comp | 2424   | 14812   | <b>14803</b>                  | <b>36.140</b> | 14803                | 37.408   |
| cnf-r4-b1-k1.2-comp | 2424   | 14812   | <b>14804</b>                  | 110.681       | 14802                | 19.694   |
| cnt10               | 20470  | 68561   | <b>68454</b>                  | 20.007        | 68452                | 7.950    |
| comb1               | 5910   | 16804   | <b>16761</b>                  | 51.819        | 16759                | 31.832   |
| comb2               | 31933  | 112462  | <b>111953</b>                 | <b>24.537</b> | 111953               | 39.263   |
| comb3               | 4774   | 16331   | 16305                         | 34.559        | 16307                | 110.225  |
| dp11u10             | 9197   | 25271   | 25230                         | 54.236        | 25232                | 2.847    |
| dp12u11             | 11137  | 30792   | <b>30723</b>                  | 95.372        | 30722                | 12.257   |
| f2clk_40            | 27568  | 80439   | 79842                         | 80.439        | 79852                | 0.654    |
| f2clk_50            | 34678  | 101319  | 100430                        | 42.333        | 100435               | 3.063    |
| fifo8_300           | 194762 | 530713  | <b>516445</b>                 | 104.923       | 516380               | 35.441   |
| fifo8_400           | 259762 | 707913  | <b>677018</b>                 | 101.843       | 676998               | 70.469   |
| homer17             | 286    | 1742    | <b>1738</b>                   | <b>0.134</b>  | 1738                 | 0.330    |
| homer18             | 308    | 2030    | <b>2024</b>                   | <b>0.138</b>  | 2024                 | 0.345    |
| homer19             | 330    | 2340    | <b>2332</b>                   | <b>0.146</b>  | 2332                 | 0.367    |
| homer20             | 440    | 4220    | <b>4202</b>                   | <b>0.175</b>  | 4202                 | 0.440    |
| ip36                | 47273  | 153368  | <b>151793</b>                 | 77.624        | 151783               | 41.247   |
| ip38                | 49967  | 162142  | <b>160477</b>                 | 31.826        | 160454               | 112.384  |
| ip50                | 66131  | 214786  | 212433                        | 94.922        | 212435               | 21.855   |
| k2fix_gr_2pinvar_w8 | 3771   | 270136  | <b>269951</b>                 | 118.768       | 269930               | 39.335   |
| k2fix_gr_2pinvar_w9 | 5028   | 307674  | <b>307562</b>                 | 101.736       | 307559               | 28.284   |
| k2fix_gr_2pin_w8    | 9882   | 295998  | 295637                        | 21.429        | 295645               | 11.987   |
| k2fix_gr_2pin_w9    | 13176  | 345426  | 345118                        | 11.300        | 345150               | 51.661   |
| k2fix_gr_rcs_w8     | 10056  | 271393  | 271287                        | 87.388        | 271290               | 39.640   |
| sha1                | 61377  | 255417  | <b>252631</b>                 | 99.706        | 252617               | 120.679  |
| sha2                | 61377  | 255417  | <b>252633</b>                 | 7.391         | 252630               | 88.739   |
| w08_14              | 120367 | 425316  | 0                             | 0.000         | 420133               | 29.329   |
| w08_15              | 132555 | 469519  | <b>463541</b>                 | 100.429       | 463527               | 118.654  |

Tabela B.3: Resultados completos do *SAHe-SAT* e  $H_p = \text{Adapt novelty+}$ , para instâncias do conjunto de 2008 [21].

| Instância                     | $n$    | $m$     | <i>SAHe-SAT+Adapt novelty+</i> |               | <i>Adapt novelty+</i> |          |
|-------------------------------|--------|---------|--------------------------------|---------------|-----------------------|----------|
|                               |        |         | Num. cls. sat.                 | Tempo(m)      | Num. cls. sat.        | Tempo(m) |
| b14_opt_bug2_vec1             | 130328 | 402707  | <b>395297</b>                  | 96.925        | 395265                | 158.398  |
| b15-bug-fourvec               | 581064 | 1712690 | <b>1530911</b>                 | 236.245       | 1530735               | 110.504  |
| b15-bug-onevec                | 121836 | 359040  | 352429                         | 159.404       | 352453                | 168.748  |
| c1_DD_s3_f1_e2_v1-bug-fourvec | 391897 | 989885  | <b>900746</b>                  | 224.761       | 900504                | 212.998  |
| c1_DD_s3_f1_e2_v1-bug-onevec  | 102234 | 258294  | 255314                         | 190.684       | 255428                | 22.543   |
| c2_DD_s3_f1_e2_v1-bug-fourvec | 400085 | 1121810 | <b>1035518</b>                 | 101.263       | 1035402               | 210.120  |
| c2_DD_s3_f1_e2_v1-bug-onevec  | 84525  | 236942  | 234065                         | 26.966        | 234087                | 51.369   |
| c3_DD_s3_f1_e1_v1-bug-fourvec | 33540  | 86944   | <b>86281</b>                   | 81.745        | 86261                 | 90.182   |
| c3_DD_s3_f1_e1_v1-bug-onevec  | 8385   | 21736   | 21702                          | 192.328       | 21704                 | 220.192  |
| c4_DD_s3_f1_e1_v1-bug         | 797728 | 2011216 | 1697012                        | 159.714       | 1697053               | 226.966  |
| c4_DD_s3_f1_e2_v1-bug-fourvec | 448465 | 1130672 | <b>1005427</b>                 | 240.564       | 1005239               | 226.922  |
| c4_DD_s3_f1_e2_v1-bug-onevec  | 131584 | 331754  | 326175                         | 7.964         | 326219                | 217.096  |
| c5315-bug                     | 1880   | 5049    | <b>5048</b>                    | <b>0.216</b>  | 5048                  | 0.613    |
| c5_DD_s3_f1_e1_v1-bug-fourvec | 100472 | 270492  | <b>268010</b>                  | 107.333       | 267987                | 172.481  |
| c5_DD_s3_f1_e1_v1-bug         | 200944 | 540984  | 523186                         | 112.637       | 523266                | 39.113   |
| c5_DD_s3_f1_e1_v1-bug-onevec  | 25118  | 67623   | 67289                          | 203.837       | 67290                 | 61.713   |
| c5_DD_s3_f1_e1_v2-bug         | 200944 | 540984  | 523157                         | 3.709         | 523278                | 6.203    |
| c6288-bug                     | 3462   | 9285    | <b>9265</b>                    | 19.976        | 9264                  | 53.442   |
| c6_DD_s3_f1_e1_v1-bug-fourvec | 170019 | 454050  | 443876                         | 128.338       | 443963                | 82.426   |
| c6_DD_s3_f1_e1_v1-bug         | 298058 | 795900  | <b>746680</b>                  | 90.884        | 746674                | 82.761   |
| c6_DD_s3_f1_e1_v1-bug-onevec  | 44079  | 117720  | <b>116899</b>                  | <b>73.860</b> | 116899                | 226.679  |
| c6_DD_s3_f1_e2_v1-bug-fourvec | 170019 | 454050  | <b>443944</b>                  | 32.006        | 443907                | 30.341   |
| c7552-bug                     | 2640   | 7008    | <b>7007</b>                    | <b>6.066</b>  | 7007                  | 14.828   |
| divider-problem.dimacs_10     | 234004 | 760965  | <b>736083</b>                  | 16.605        | 735881                | 206.661  |
| divider-problem.dimacs_11     | 215964 | 709377  | 688929                         | 183.271       | 689020                | 117.469  |
| divider-problem.dimacs_12     | 229482 | 751921  | <b>728508</b>                  | 72.902        | 728426                | 195.159  |
| divider-problem.dimacs_1      | 215676 | 708801  | 688496                         | 194.052       | 688596                | 203.367  |
| divider-problem.dimacs_2      | 228874 | 750705  | 727348                         | 53.420        | 727403                | 212.322  |
| divider-problem.dimacs_3      | 216900 | 711249  | 690689                         | 133.517       | 690701                | 167.010  |
| divider-problem.dimacs_4      | 225340 | 743637  | 720908                         | 206.363       | 721019                | 186.418  |
| divider-problem.dimacs_5      | 228874 | 750705  | 727475                         | 16.642        | 727540                | 75.915   |
| divider-problem.dimacs_6      | 234004 | 760965  | 735979                         | 170.099       | 736185                | 169.546  |
| divider-problem.dimacs_7      | 239640 | 787745  | <b>761363</b>                  | 230.672       | 761183                | 193.335  |
| divider-problem.dimacs_8      | 246943 | 810105  | 781544                         | 21.442        | 781545                | 20.326   |
| divider-problem.dimacs_9      | 222851 | 730905  | <b>708962</b>                  | 195.044       | 708900                | 38.601   |
| dividers10                    | 45552  | 162874  | 161038                         | 207.399       | 161042                | 218.989  |
| dividers11                    | 45552  | 162982  | 161063                         | 112.532       | 161109                | 24.027   |
| dividers1                     | 45552  | 162982  | 161075                         | 73.872        | 161121                | 36.622   |
| dividers2                     | 45552  | 162982  | 161035                         | 109.057       | 161046                | 118.619  |
| dividers3                     | 45552  | 162982  | 161057                         | 59.334        | 161062                | 235.267  |
| dividers4                     | 45552  | 162983  | <b>161057</b>                  | 203.428       | 161055                | 76.611   |
| dividers5                     | 45552  | 162982  | <b>161085</b>                  | 162.448       | 161074                | 166.849  |
| dividers6_hack                | 35376  | 132699  | 131173                         | 107.073       | 131192                | 97.655   |
| dividers7                     | 64532  | 231332  | 228351                         | 77.134        | 228369                | 152.435  |
| dividers8                     | 45552  | 162982  | <b>161064</b>                  | 71.438        | 161063                | 210.246  |
| dividers9                     | 45552  | 162982  | 161037                         | 115.748       | 161062                | 1.918    |
| dividers_multivec1            | 106128 | 397650  | <b>392175</b>                  | 221.467       | 392158                | 147.194  |
| fpu1-problem                  | 119286 | 409139  | <b>404463</b>                  | 17.435        | 404406                | 114.422  |
| fpu2-problem                  | 160184 | 548699  | 539664                         | 15.861        | 539683                | 241.104  |
| fpu3_hack-problem             | 96384  | 347489  | 343730                         | 67.259        | 343731                | 40.214   |
| fpu3-problem                  | 120114 | 410963  | <b>406169</b>                  | 222.714       | 406146                | 147.895  |
| fpu4-problem                  | 120150 | 411276  | <b>406475</b>                  | 217.102       | 406470                | 105.681  |
| fpu5-problem                  | 320400 | 1099039 | 1043131                        | 192.739       | 1043190               | 15.354   |
| fpu6-problem                  | 160160 | 548748  | <b>539809</b>                  | 197.190       | 539713                | 215.420  |
| fpu7_hack-problem             | 96438  | 347843  | <b>344070</b>                  | 130.244       | 344056                | 27.766   |
| fpu7-problem                  | 120174 | 411323  | <b>406556</b>                  | 215.447       | 406523                | 153.119  |

Tabela B.4: Resultados completos do *SAHe-SAT* e  $H_p = \text{AdaptNovelty+}$ , para instâncias do conjunto de 2008 [21]. (continuação)

| Instância                    | $n$     | $m$      | <i>SAHe-SAT+AdaptNovelty+</i> |              | <i>AdaptNovelty+</i> |          |
|------------------------------|---------|----------|-------------------------------|--------------|----------------------|----------|
|                              |         |          | Num. cls. sat.                | Tempo(m)     | Num. cls. sat.       | Tempo(m) |
| fpu8-problem                 | 160232  | 548848   | <b>539840</b>                 | 9.440        | 539784               | 4.572    |
| fpu_fsm1-problem             | 160200  | 548843   | 539744                        | 216.627      | 539806               | 211.266  |
| fpu_multivec1-problem        | 257168  | 928310   | 894852                        | 14.906       | 894952               | 129.178  |
| i2c_master1                  | 82429   | 285987   | 283185                        | 40.401       | 283197               | 221.071  |
| i2c_master2                  | 63816   | 221320   | <b>219251</b>                 | 111.203      | 219239               | 187.125  |
| i2c-problem.dimacs_25        | 521672  | 1581471  | <b>1429341</b>                | 69.259       | 1429324              | 116.999  |
| i2c-problem.dimacs_26        | 397668  | 1205454  | <b>1112855</b>                | 226.232      | 1112711              | 239.086  |
| mem_ctrl1                    | 1128648 | 4422185  | 3906995                       | 78.482       | 3907098              | 82.346   |
| mem_ctrl2_blackbox_mc        | 1974822 | 6795573  | <b>5878336</b>                | 130.616      | 5878253              | 203.392  |
| mem_ctrl1-problem            | 4426323 | 15983633 | 13559852                      | 55.397       | 13560010             | 50.407   |
| mot_comb1._red               | 2159    | 5326     | <b>5325</b>                   | <b>0.107</b> | 5325                 | 0.180    |
| mot_comb2._red               | 5484    | 13894    | <b>13893</b>                  | <b>0.117</b> | 13893                | 0.189    |
| mot_comb3._red               | 11265   | 29520    | 29454                         | 45.318       | 29458                | 52.136   |
| mrisc_mem2wire1              | 168960  | 641598   | <b>630077</b>                 | 139.177      | 630006               | 108.443  |
| mrisc_mem2wire-problem       | 844900  | 2905976  | 2587564                       | 121.285      | 2587626              | 211.196  |
| rsdecoder1_blackbox_CSEblock | 277950  | 806460   | 775465                        | 24.148       | 775482               | 16.454   |
| rsdecoder1_blackbox_KESblock | 707330  | 1106376  | <b>1042885</b>                | 167.766      | 1042849              | 35.251   |
| rsdecoder1                   | 238056  | 935187   | 906159                        | 68.772       | 906221               | 10.916   |
| rsdecoder2_blackbox-problem  | 927691  | 2728843  | 2425422                       | 222.385      | 2425441              | 211.746  |
| rsdecoder2                   | 415480  | 1632526  | 1537118                       | 17.885       | 1537292              | 132.670  |
| rsdecoder3                   | 238290  | 936006   | 906863                        | 4.751        | 906951               | 222.119  |
| rsdecoder4                   | 237783  | 933978   | <b>905052</b>                 | 133.385      | 905038               | 152.771  |
| rsdecoder5                   | 238290  | 936006   | 906879                        | 159.503      | 906960               | 241.537  |
| rsdecoder6                   | 238290  | 936006   | 906957                        | 216.650      | 906973               | 227.383  |
| rsdecoder_fsm1               | 238290  | 936006   | 906894                        | 97.870       | 906942               | 205.129  |
| rsdecoder_fsm2               | 238290  | 936006   | 906817                        | 178.571      | 906944               | 8.660    |
| rsdecoder_multivec1          | 394446  | 1626312  | 1537494                       | 167.034      | 1537543              | 133.547  |
| rsdecoder_multivec1-problem  | 627993  | 2125620  | <b>1934408</b>                | 149.937      | 1933918              | 11.755   |
| rsdecoder-problem.dimacs_31  | 1197376 | 3863287  | 3372239                       | 146.073      | 3372413              | 12.187   |
| rsdecoder-problem.dimacs_34  | 226040  | 728516   | 703618                        | 72.163       | 703682               | 120.048  |
| rsdecoder-problem.dimacs_36  | 1220616 | 3938467  | 3435017                       | 98.291       | 3435165              | 6.493    |
| rsdecoder-problem.dimacs_37  | 1513544 | 4909231  | <b>4245328</b>                | 157.286      | 4245100              | 236.780  |
| rsdecoder-problem.dimacs_38  | 1198012 | 3865513  | <b>3374563</b>                | 220.802      | 3374454              | 99.393   |
| rsdecoder-problem.dimacs_39  | 1199602 | 3868693  | <b>3376675</b>                | 41.434       | 3376591              | 117.372  |
| rsdecoder-problem.dimacs_40  | 1220616 | 3938467  | <b>3435054</b>                | 46.702       | 3434849              | 23.473   |
| rsdecoder-problem.dimacs_41  | 1186710 | 3829036  | <b>3343967</b>                | 55.766       | 3343848              | 229.876  |
| s15850-bug-fourvec           | 88544   | 206252   | 204066                        | 152.969      | 204090               | 46.128   |
| s15850-bug-onevec            | 22136   | 51563    | <b>51368</b>                  | 19.605       | 51363                | 218.871  |
| s38584-bug-onevec            | 314272  | 819830   | <b>766491</b>                 | 197.614      | 766461               | 126.718  |
| SM_AS_TOP_buggy1             | 145900  | 694438   | <b>689519</b>                 | 37.990       | 689463               | 25.557   |
| SM_MAIN_MEM_buggy1           | 870975  | 3812147  | <b>3431928</b>                | 15.243       | 3431544              | 100.193  |
| SM_RX_TOP                    | 235456  | 934091   | 908956                        | 20.369       | 909037               | 132.722  |
| spi2                         | 124260  | 515813   | 509364                        | 108.727      | 509402               | 220.824  |
| spi-problem                  | 628192  | 2234134  | <b>2027989</b>                | 63.453       | 2027614              | 200.614  |
| wb1                          | 49525   | 140091   | 138602                        | 106.928      | 138603               | 168.792  |
| wb2                          | 49490   | 140056   | 138143                        | 71.258       | 138151               | 51.200   |
| wb_4m8s1                     | 463080  | 1759150  | 1637425                       | 6.646        | 1637511              | 209.569  |
| wb_4m8s3                     | 463080  | 1759150  | 1637071                       | 81.734       | 1637186              | 148.129  |
| wb_4m8s4                     | 463080  | 1759150  | <b>1636787</b>                | 15.974       | 1636765              | 189.417  |
| wb_4m8s-problem.dimacs_47    | 2691648 | 8517027  | 7189162                       | 95.320       | 7189244              | 19.575   |
| wb_4m8s-problem.dimacs_48    | 2766036 | 8774655  | 7403607                       | 67.450       | 7403873              | 234.528  |
| wb_4m8s-problem.dimacs_49    | 2785108 | 8812799  | <b>7432769</b>                | 20.241       | 7432645              | 216.399  |
| wb_conmax1                   | 277950  | 1221020  | 1175797                       | 96.978       | 1175846              | 104.276  |
| wb_conmax3                   | 277950  | 1221020  | 1175833                       | 12.656       | 1175942              | 14.228   |
| wb-problem.dimacs_45         | 309491  | 806440   | 758762                        | 64.776       | 758819               | 44.686   |
| wb-problem.dimacs_46         | 300846  | 789283   | 743686                        | 95.522       | 743721               | 72.191   |

Tabela B.5: Resultados completos do *SAHe-SAT* e  $Hp = Novelty$ , para instâncias do conjunto de 2004 [20].

| Instância    | $n$   | $m$    | <i>SAHe-SAT+Novelty</i> |               | <i>Novelty</i> |          |
|--------------|-------|--------|-------------------------|---------------|----------------|----------|
|              |       |        | Num. cls. sat.          | Tempo(m)      | Num. cls. sat. | Tempo(m) |
| alu4mul      | 4736  | 30465  | 30455                   | 137.892       | 30457          | 201.983  |
| c3540mul     | 5248  | 33199  | 33176                   | 17.965        | 33183          | 187.185  |
| c6288mul     | 9540  | 61421  | 61379                   | 81.437        | 61384          | 103.835  |
| c7552mul     | 11282 | 69529  | <b>69423</b>            | 45.515        | 69421          | 64.015   |
| C880mul      | 1612  | 9373   | <b>9372</b>             | <b>0.195</b>  | 9372           | 0.513    |
| dalumul      | 9426  | 59991  | 59975                   | 62.621        | 59976          | 77.171   |
| desmul       | 28902 | 179895 | <b>179821</b>           | 37.407        | 179817         | 67.515   |
| frg1mul      | 3230  | 20575  | <b>20574</b>            | <b>0.543</b>  | 20574          | 1.390    |
| frg2mul      | 10316 | 62943  | <b>62922</b>            | <b>7.082</b>  | 62922          | 82.769   |
| i10mul       | 12998 | 77941  | 77840                   | 92.730        | 77854          | 114.425  |
| i8mul        | 14524 | 91139  | <b>91119</b>            | 70.068        | 91112          | 122.295  |
| k2mul        | 11680 | 74581  | <b>74524</b>            | <b>1.729</b>  | 74524          | 96.870   |
| rotmul       | 5980  | 35229  | <b>35228</b>            | <b>1.596</b>  | 35228          | 9.976    |
| term1mul     | 3504  | 22229  | 22228                   | 100.641       | 22228          | 31.442   |
| vdamul       | 5444  | 34509  | 34487                   | 64.092        | 34491          | 195.162  |
| x1mul        | 8760  | 55571  | <b>55570</b>            | <b>1.550</b>  | 55570          | 5.383    |
| am_4_4       | 433   | 1458   | <b>1457</b>             | <b>0.146</b>  | 1457           | 0.226    |
| am_5_5       | 1076  | 3677   | <b>3676</b>             | <b>0.093</b>  | 3676           | 0.250    |
| am_6_6       | 2269  | 7814   | <b>7813</b>             | <b>0.263</b>  | 7813           | 0.267    |
| am_7_7       | 4264  | 14751  | <b>14744</b>            | 55.268        | 14739          | 120.361  |
| am_8_8       | 7361  | 25538  | <b>25337</b>            | 69.707        | 25334          | 203.760  |
| am_9_9       | 11908 | 41393  | <b>40982</b>            | 1.827         | 40970          | 36.322   |
| li-exam-61   | 28147 | 108436 | 108011                  | 47.997        | 108041         | 22.993   |
| li-exam-62   | 28147 | 108436 | <b>108018</b>           | 112.380       | 108008         | 100.084  |
| li-exam-63   | 28147 | 108436 | <b>108011</b>           | 101.528       | 108003         | 19.108   |
| li-exam-64   | 28147 | 108436 | 107996                  | 70.593        | 108013         | 34.106   |
| li-test4-100 | 36809 | 142491 | 141844                  | 34.564        | 141870         | 88.830   |
| li-test4-101 | 36809 | 142491 | 141858                  | 40.733        | 141865         | 31.374   |
| li-test4-94  | 36809 | 142491 | <b>141863</b>           | 2.556         | 141859         | 74.018   |
| li-test4-95  | 36809 | 142491 | 141865                  | 63.408        | 141868         | 16.500   |
| li-test4-96  | 36809 | 142491 | <b>141865</b>           | 101.795       | 141851         | 112.326  |
| li-test4-97  | 36809 | 142491 | 141855                  | 29.113        | 141861         | 91.341   |
| li-test4-98  | 36809 | 142491 | <b>141862</b>           | 40.632        | 141852         | 90.647   |
| li-test4-99  | 36809 | 142491 | 141859                  | 12.537        | 141870         | 102.007  |
| ferry10      | 2958  | 20791  | <b>20791</b>            | <b>0.196</b>  | 20791          | 0.339    |
| ferry10u     | 2883  | 20259  | <b>20259</b>            | <b>0.271</b>  | 20259          | 0.341    |
| ferry11      | 3562  | 26105  | <b>26105</b>            | <b>0.146</b>  | 26105          | 0.363    |
| ferry11u     | 3480  | 25499  | <b>25499</b>            | <b>0.146</b>  | 25499          | 0.356    |
| ferry12      | 4222  | 32199  | <b>32199</b>            | <b>9.707</b>  | 32199          | 21.036   |
| ferry12u     | 4133  | 31515  | <b>31515</b>            | <b>22.224</b> | 31515          | 65.811   |
| ferry8       | 1918  | 12311  | 12311                   | 0.326         | 12311          | 0.310    |
| ferry8u      | 1857  | 11915  | 11915                   | 0.625         | 11915          | 0.308    |
| ferry9       | 2410  | 16209  | <b>16209</b>            | <b>0.133</b>  | 16209          | 0.329    |
| ferry9u      | 2342  | 15747  | <b>15747</b>            | <b>0.133</b>  | 15747          | 0.331    |
| gripper10    | 2438  | 19700  | <b>19697</b>            | <b>2.353</b>  | 19697          | 8.277    |
| gripper10u   | 2312  | 18666  | 18663                   | 0.877         | 18664          | 76.493   |
| gripper11    | 3222  | 27199  | <b>27197</b>            | <b>42.493</b> | 27197          | 76.256   |
| gripper11u   | 3084  | 26019  | <b>26017</b>            | 15.145        | 26016          | 6.459    |
| gripper12    | 3502  | 30746  | 30743                   | 57.930        | 30743          | 20.463   |
| gripper12u   | 3352  | 29412  | <b>29409</b>            | <b>14.485</b> | 29409          | 29.184   |

Tabela B.6: Resultados completos do *SAHe-SAT* e  $Hp = Novelty$ , para instâncias do conjunto de 2004 [20]. (continuação)

| Instância           | $n$    | $m$     | <i>SAHe-SAT+Novelty</i> |               | <i>Novelty</i> |          |
|---------------------|--------|---------|-------------------------|---------------|----------------|----------|
|                     |        |         | Num. cls. sat.          | Tempo(m)      | Num. cls. sat. | Tempo(m) |
| gripper13           | 4430   | 40461   | 40458                   | 27.590        | 40458          | 17.129   |
| gripper13u          | 4268   | 38965   | <b>38962</b>            | <b>87.752</b> | 38962          | 118.531  |
| gripper14           | 4758   | 45056   | <b>45053</b>            | 23.516        | 45052          | 5.395    |
| gripper14u          | 4584   | 43390   | 43386                   | 23.381        | 43387          | 97.279   |
| hanoi4              | 1404   | 18058   | <b>18057</b>            | <b>0.217</b>  | 18057          | 0.728    |
| hanoi4u             | 1312   | 16856   | <b>16855</b>            | <b>0.214</b>  | 16855          | 0.531    |
| hanoi5              | 4434   | 76235   | 76232                   | 63.667        | 76232          | 24.495   |
| hanoi5u             | 4292   | 73777   | <b>73774</b>            | <b>1.397</b>  | 73774          | 5.657    |
| abp1-1-k31          | 14809  | 48483   | 48320                   | 3.183         | 48330          | 67.040   |
| abp4-1-k31          | 14809  | 48483   | <b>48325</b>            | 65.573        | 48324          | 61.416   |
| bc56-sensors-1-k391 | 561371 | 1778987 | <b>1600354</b>          | 88.186        | 1600149        | 86.918   |
| bc56-sensors-2-k592 | 850398 | 2694319 | 2366874                 | 48.252        | 2366923        | 13.745   |
| bc57-sensors-1-k303 | 435701 | 1379987 | <b>1262046</b>          | 79.846        | 1262043        | 14.386   |
| dme-03-1-k247       | 261352 | 773077  | <b>736229</b>           | 49.434        | 736147         | 106.616  |
| guidance-1-k56      | 98746  | 307346  | 302986                  | 109.738       | 302990         | 62.197   |
| motors-stuck-1-k407 | 654766 | 2068742 | <b>1842645</b>          | 112.121       | 1842299        | 7.294    |
| motors-stuck-2-k314 | 505536 | 1596837 | 1445274                 | 53.357        | 1445323        | 57.764   |
| motors-stuck-2-k315 | 507145 | 1601920 | <b>1449734</b>          | 79.784        | 1449545        | 53.990   |
| valves-gates-1-k617 | 985042 | 3113540 | 2714448                 | 58.357        | 2714681        | 30.268   |
| 6pipe_6_ooo         | 17064  | 545612  | <b>545611</b>           | <b>41.648</b> | 545611         | 433.499  |
| 6pipe               | 15800  | 394739  | 394722                  | 112.168       | 394723         | 241.421  |
| 7pipe               | 23910  | 751118  | 751077                  | 187.551       | 751088         | 1860.378 |
| cnf-r4-b1-k1.1-comp | 2424   | 14812   | 14787                   | 27.977        | 14788          | 78.056   |
| cnf-r4-b1-k1.2-comp | 2424   | 14812   | 14787                   | 11.071        | 14794          | 20.610   |
| cnt10               | 20470  | 68561   | 68359                   | 108.621       | 68366          | 9.462    |
| comb1               | 5910   | 16804   | <b>16756</b>            | 89.910        | 16755          | 101.368  |
| comb2               | 31933  | 112462  | 112116                  | 2.038         | 112133         | 95.826   |
| comb3               | 4774   | 16331   | 16303                   | 88.723        | 16304          | 6.255    |
| dp11u10             | 9197   | 25271   | <b>25268</b>            | 27.466        | 25267          | 4.244    |
| dp12u11             | 11137  | 30792   | 30785                   | 11.239        | 30789          | 17.881   |
| f2clk_40            | 27568  | 80439   | <b>80000</b>            | 5.994         | 79986          | 2.874    |
| f2clk_50            | 34678  | 101319  | 100629                  | 18.233        | 100668         | 29.506   |
| fifo8_300           | 194762 | 530713  | 506281                  | 72.155        | 506329         | 49.814   |
| fifo8_400           | 259762 | 707913  | <b>660375</b>           | 64.274        | 660293         | 57.484   |
| homer17             | 286    | 1742    | <b>1738</b>             | <b>0.126</b>  | 1738           | 0.317    |
| homer18             | 308    | 2030    | <b>2024</b>             | <b>0.131</b>  | 2024           | 0.331    |
| homer19             | 330    | 2340    | <b>2332</b>             | <b>0.142</b>  | 2332           | 0.349    |
| homer20             | 440    | 4220    | <b>4202</b>             | <b>0.170</b>  | 4202           | 0.429    |
| ip36                | 47273  | 153368  | <b>152340</b>           | 57.125        | 152334         | 64.625   |
| ip38                | 49967  | 162142  | 160995                  | 106.594       | 161004         | 12.770   |
| ip50                | 66131  | 214786  | <b>212679</b>           | 26.520        | 212678         | 76.055   |
| k2fix_gr_2pinvar_w8 | 3771   | 270136  | <b>269918</b>           | 47.861        | 269916         | 116.270  |
| k2fix_gr_2pinvar_w9 | 5028   | 307674  | 307563                  | 43.461        | 307565         | 42.008   |
| k2fix_gr_2pin_w8    | 9882   | 295998  | 295684                  | 69.188        | 295691         | 5.697    |
| k2fix_gr_2pin_w9    | 13176  | 345426  | 345243                  | 87.169        | 345255         | 81.536   |
| k2fix_gr_rcs_w8     | 10056  | 271393  | <b>271296</b>           | 17.291        | 271292         | 58.947   |
| sha1                | 61377  | 255417  | 251884                  | 126.938       | 251927         | 48.348   |
| sha2                | 61377  | 255417  | <b>251938</b>           | 87.569        | 251918         | 72.757   |
| w08_14              | 120367 | 425316  | -                       | -             | 417775         | 120.289  |
| w08_15              | 132555 | 469519  | 460087                  | 45.941        | 460125         | 80.069   |

Tabela B.7: Resultados completos do *SAHe-SAT* e  $Hp = Novelty$ , para instâncias do conjunto de 2008 [21].

| Instância                     | $n$    | $m$     | <i>SAHe-SAT+Novelty</i> |               | <i>Novelty</i> |          |
|-------------------------------|--------|---------|-------------------------|---------------|----------------|----------|
|                               |        |         | Num. cls. sat.          | Tempo(m)      | Num. cls. sat. | Tempo(m) |
| b14_opt_bug2_vec1             | 130328 | 402707  | <b>392574</b>           | 89.986        | 392571         | 28.673   |
| b15-bug-fourvec               | 581064 | 1712690 | 1510462                 | 90.387        | 1510602        | 32.348   |
| b15-bug-onevec                | 121836 | 359040  | <b>350273</b>           | 229.313       | 350183         | 176.749  |
| c1_DD_s3_f1_e2_v1-bug-fourvec | 391897 | 989885  | 885395                  | 57.153        | 885405         | 102.570  |
| c1_DD_s3_f1_e2_v1-bug-onevec  | 102234 | 258294  | 254011                  | 186.256       | 254051         | 208.404  |
| c2_DD_s3_f1_e2_v1-bug-fourvec | 400085 | 1121810 | 1023328                 | 60.074        | 1023402        | 215.318  |
| c2_DD_s3_f1_e2_v1-bug-onevec  | 84525  | 236942  | 233608                  | 211.002       | 233616         | 115.981  |
| c3_DD_s3_f1_e1_v1-bug-fourvec | 33540  | 86944   | <b>86719</b>            | 97.032        | 86693          | 94.134   |
| c3_DD_s3_f1_e1_v1-bug-onevec  | 8385   | 21736   | <b>21735</b>            | <b>0.150</b>  | 21735          | 0.253    |
| c4_DD_s3_f1_e1_v1-bug         | 797728 | 2011216 | 1667714                 | 135.834       | 1667910        | 178.432  |
| c4_DD_s3_f1_e2_v1-bug-fourvec | 448465 | 1130672 | 981851                  | 216.918       | 982040         | 124.342  |
| c4_DD_s3_f1_e2_v1-bug-onevec  | 131584 | 331754  | <b>321529</b>           | 129.637       | 321447         | 237.029  |
| c5315-bug                     | 1880   | 5049    | <b>5048</b>             | <b>14.164</b> | 5048           | 19.739   |
| c5_DD_s3_f1_e1_v1-bug-fourvec | 100472 | 270492  | <b>266620</b>           | 90.086        | 266613         | 121.399  |
| c5_DD_s3_f1_e1_v1-bug         | 200944 | 540984  | <b>517892</b>           | 88.095        | 517753         | 136.871  |
| c5_DD_s3_f1_e1_v1-bug-onevec  | 25118  | 67623   | 67290                   | 196.716       | 67299          | 14.201   |
| c5_DD_s3_f1_e1_v2-bug         | 200944 | 540984  | <b>517862</b>           | 138.831       | 517738         | 108.599  |
| c6288-bug                     | 3462   | 9285    | 9241                    | 99.305        | 9252           | 63.328   |
| c6_DD_s3_f1_e1_v1-bug-fourvec | 170019 | 454050  | 439674                  | 159.633       | 439687         | 107.092  |
| c6_DD_s3_f1_e1_v1-bug         | 298058 | 795900  | 736381                  | 232.411       | 736411         | 69.905   |
| c6_DD_s3_f1_e1_v1-bug-onevec  | 44079  | 117720  | <b>116994</b>           | 139.751       | 116975         | 195.447  |
| c6_DD_s3_f1_e2_v1-bug-fourvec | 170019 | 454050  | <b>439685</b>           | 8.630         | 439625         | 229.621  |
| c7552-bug                     | 2640   | 7008    | 6992                    | 223.341       | 6995           | 96.806   |
| divider-problem.dimacs_10     | 234004 | 760965  | <b>728047</b>           | 108.251       | 728024         | 175.123  |
| divider-problem.dimacs_11     | 215964 | 709377  | 681705                  | 114.592       | 681740         | 21.149   |
| divider-problem.dimacs_12     | 229482 | 751921  | <b>720223</b>           | 73.439        | 720208         | 207.576  |
| divider-problem.dimacs_1      | 215676 | 708801  | <b>681257</b>           | 8.759         | 681167         | 15.463   |
| divider-problem.dimacs_2      | 228874 | 750705  | <b>719308</b>           | 138.306       | 719266         | 182.423  |
| divider-problem.dimacs_3      | 216900 | 711249  | <b>683482</b>           | 88.685        | 683278         | 153.251  |
| divider-problem.dimacs_4      | 225340 | 743637  | <b>713305</b>           | 195.022       | 713221         | 195.329  |
| divider-problem.dimacs_5      | 228874 | 750705  | 719381                  | 121.959       | 719492         | 236.451  |
| divider-problem.dimacs_6      | 234004 | 760965  | 727989                  | 103.294       | 728052         | 69.044   |
| divider-problem.dimacs_7      | 239640 | 787745  | <b>752882</b>           | 114.347       | 752873         | 25.689   |
| divider-problem.dimacs_8      | 246943 | 810105  | 772798                  | 204.592       | 772832         | 95.456   |
| divider-problem.dimacs_9      | 222851 | 730905  | 701250                  | 220.196       | 701272         | 223.376  |
| dividers10                    | 45552  | 162874  | <b>161376</b>           | 230.581       | 161366         | 112.262  |
| dividers11                    | 45552  | 162982  | 161395                  | 62.771        | 161468         | 41.262   |
| dividers1                     | 45552  | 162982  | <b>161436</b>           | 218.344       | 161420         | 234.819  |
| dividers2                     | 45552  | 162982  | 161379                  | 144.179       | 161390         | 218.237  |
| dividers3                     | 45552  | 162982  | 161407                  | 181.772       | 161408         | 227.842  |
| dividers4                     | 45552  | 162983  | <b>161471</b>           | 67.838        | 161390         | 135.486  |
| dividers5                     | 45552  | 162982  | <b>161424</b>           | 16.476        | 161415         | 82.897   |
| dividers6_hack                | 35376  | 132699  | 131539                  | 4.493         | 131548         | 120.972  |
| dividers7                     | 64532  | 231332  | <b>228534</b>           | 192.707       | 228520         | 89.926   |
| dividers8                     | 45552  | 162982  | <b>161419</b>           | 85.305        | 161394         | 216.958  |
| dividers9                     | 45552  | 162982  | 161379                  | 163.963       | 161385         | 214.114  |
| dividers_multivec1            | 106128 | 397650  | 390616                  | 57.206        | 390660         | 116.049  |
| fpu1-problem                  | 119286 | 409139  | <b>403327</b>           | 28.486        | 403323         | 33.856   |
| fpu2-problem                  | 160184 | 548699  | 535932                  | 89.976        | 535937         | 98.751   |
| fpu3_hack-problem             | 96384  | 347489  | <b>343435</b>           | 109.465       | 343401         | 39.469   |
| fpu3-problem                  | 120114 | 410963  | <b>405017</b>           | 230.171       | 405014         | 211.225  |
| fpu4-problem                  | 120150 | 411276  | <b>405330</b>           | 139.481       | 405276         | 113.795  |
| fpu5-problem                  | 320400 | 1099039 | <b>1031880</b>          | 113.593       | 1031799        | 109.368  |
| fpu6-problem                  | 160160 | 548748  | 535939                  | 28.578        | 535943         | 72.515   |
| fpu7_hack-problem             | 96438  | 347843  | 343764                  | 159.117       | 343765         | 26.512   |
| fpu7-problem                  | 120174 | 411323  | <b>405352</b>           | 167.325       | 405312         | 151.464  |

Tabela B.8: Resultados completos do *SAHe-SAT* e  $Hp = Novelty$ , para instâncias do conjunto de 2008 [21]. (continuação)

| Instância                    | $n$     | $m$      | <i>SAHe-SAT+Novelty</i> |              | <i>Novelty</i> |          |
|------------------------------|---------|----------|-------------------------|--------------|----------------|----------|
|                              |         |          | Num. cls. sat.          | Tempo(m)     | Num. cls. sat. | Tempo(m) |
| fpu8-problem                 | 160232  | 548848   | 536073                  | 87.091       | 536114         | 133.969  |
| fpu_fsm1-problem             | 160200  | 548843   | <b>536081</b>           | 205.639      | 536003         | 3.356    |
| fpu_multivec1-problem        | 257168  | 928310   | 886273                  | 173.934      | 886492         | 18.856   |
| i2c_master1                  | 82429   | 285987   | 282940                  | 14.780       | 282974         | 193.288  |
| i2c_master2                  | 63816   | 221320   | 219413                  | 238.509      | 219489         | 23.529   |
| i2c-problem.dimacs_25        | 521672  | 1581471  | <b>1414549</b>          | 199.919      | 1414478        | 214.350  |
| i2c-problem.dimacs_26        | 397668  | 1205454  | 1099960                 | 206.009      | 1100100        | 241.601  |
| mem_ctrl1                    | 1128648 | 4422185  | 3892737                 | 36.186       | 3893701        | 43.323   |
| mem_ctrl2_blackbox_mc        | 1974822 | 6795573  | <b>5859581</b>          | 136.212      | 5859138        | 247.110  |
| mem_ctrl1-problem            | 4426323 | 15983633 | <b>13521310</b>         | 199.292      | 13520341       | 31.445   |
| mot_comb1._red               | 2159    | 5326     | <b>5325</b>             | <b>0.101</b> | 5325           | 0.172    |
| mot_comb2._red               | 5484    | 13894    | <b>13893</b>            | <b>0.112</b> | 13893          | 0.188    |
| mot_comb3._red               | 11265   | 29520    | 29517                   | 71.408       | 29517          | 24.655   |
| mrisc_mem2wire1              | 168960  | 641598   | 625150                  | 89.432       | 625254         | 192.191  |
| mrisc_mem2wire1-problem      | 844900  | 2905976  | 2567902                 | 72.364       | 2568308        | 150.885  |
| rsdecoder1_blackbox_CSEBlock | 277950  | 806460   | 766425                  | 229.027      | 766512         | 221.293  |
| rsdecoder1_blackbox_KESBlock | 707330  | 1106376  | 1030117                 | 203.708      | 1030149        | 97.698   |
| rsdecoder1                   | 238056  | 935187   | 897662                  | 226.635      | 897680         | 204.164  |
| rsdecoder2_blackbox-problem  | 927691  | 2728843  | <b>2405269</b>          | 210.601      | 2405107        | 134.806  |
| rsdecoder2                   | 415480  | 1632526  | <b>1523950</b>          | 187.564      | 1523942        | 78.253   |
| rsdecoder3                   | 238290  | 936006   | <b>898379</b>           | 19.474       | 898368         | 29.397   |
| rsdecoder4                   | 237783  | 933978   | 896371                  | 140.864      | 896441         | 51.221   |
| rsdecoder5                   | 238290  | 936006   | <b>898465</b>           | 231.140      | 898417         | 230.944  |
| rsdecoder6                   | 238290  | 936006   | <b>898470</b>           | 38.079       | 898368         | 121.189  |
| rsdecoder_fsm1               | 238290  | 936006   | <b>898393</b>           | 84.508       | 898245         | 143.533  |
| rsdecoder_fsm2               | 238290  | 936006   | 898274                  | 173.553      | 898383         | 99.086   |
| rsdecoder_multivec1          | 394446  | 1626312  | <b>1525348</b>          | 18.255       | 1525212        | 212.275  |
| rsdecoder_multivec1-problem  | 627993  | 2125620  | <b>1916475</b>          | 60.583       | 1916202        | 225.536  |
| rsdecoder-problem.dimacs_31  | 1197376 | 3863287  | <b>3349140</b>          | 50.309       | 3348950        | 53.565   |
| rsdecoder-problem.dimacs_34  | 226040  | 728516   | 695416                  | 51.909       | 695416         | 43.199   |
| rsdecoder-problem.dimacs_36  | 1220616 | 3938467  | <b>3411760</b>          | 36.004       | 3411417        | 146.963  |
| rsdecoder-problem.dimacs_37  | 1513544 | 4909231  | 4219561                 | 115.407      | 4220079        | 68.545   |
| rsdecoder-problem.dimacs_38  | 1198012 | 3865513  | 3350860                 | 203.150      | 3351073        | 46.902   |
| rsdecoder-problem.dimacs_39  | 1199602 | 3868693  | <b>3353400</b>          | 239.815      | 3352866        | 185.949  |
| rsdecoder-problem.dimacs_40  | 1220616 | 3938467  | <b>3411562</b>          | 210.638      | 3411243        | 26.653   |
| rsdecoder-problem.dimacs_41  | 1186710 | 3829036  | <b>3320522</b>          | 182.019      | 3320305        | 224.181  |
| s15850-bug-fourvec           | 88544   | 206252   | <b>203159</b>           | 33.943       | 203155         | 84.878   |
| s15850-bug-onevec            | 22136   | 51563    | 51476                   | 138.018      | 51481          | 188.969  |
| s38584-bug-onevec            | 314272  | 819830   | <b>754228</b>           | 206.175      | 754212         | 43.689   |
| SM_AS_TOP_buggy1             | 145900  | 694438   | <b>687024</b>           | 123.921      | 686982         | 169.495  |
| SM_MAIN_MEM_buggy1           | 870975  | 3812147  | <b>3416883</b>          | 137.656      | 3416837        | 37.047   |
| SM_RX_TOP                    | 235456  | 934091   | <b>901957</b>           | 156.618      | 901871         | 200.077  |
| spi2                         | 124260  | 515813   | <b>508060</b>           | 82.002       | 508039         | 84.299   |
| spi-problem                  | 628192  | 2234134  | <b>2012585</b>          | 138.896      | 2012445        | 64.527   |
| wb1                          | 49525   | 140091   | <b>139168</b>           | 175.293      | 139157         | 36.955   |
| wb2                          | 49490   | 140056   | <b>138647</b>           | 78.838       | 138633         | 63.734   |
| wb_4m8s1                     | 463080  | 1759150  | <b>1624751</b>          | 17.008       | 1624585        | 207.639  |
| wb_4m8s3                     | 463080  | 1759150  | <b>1623998</b>          | 122.166      | 1623823        | 185.209  |
| wb_4m8s4                     | 463080  | 1759150  | <b>1624017</b>          | 4.333        | 1623808        | 164.746  |
| wb_4m8s-problem.dimacs_47    | 2691648 | 8517027  | <b>7159756</b>          | 56.460       | 7159458        | 53.487   |
| wb_4m8s-problem.dimacs_48    | 2766036 | 8774655  | <b>7373635</b>          | 63.090       | 7373540        | 97.320   |
| wb_4m8s-problem.dimacs_49    | 2785108 | 8812799  | <b>7402599</b>          | 144.220      | 7402249        | 183.336  |
| wb_conmax1                   | 277950  | 1221020  | 1168273                 | 23.774       | 1168351        | 200.654  |
| wb_conmax3                   | 277950  | 1221020  | <b>1168628</b>          | 212.218      | 1168466        | 147.748  |
| wb-problem.dimacs_45         | 309491  | 806440   | <b>745066</b>           | 58.044       | 744802         | 115.301  |
| wb-problem.dimacs_46         | 300846  | 789283   | 730001                  | 5.619        | 730092         | 58.469   |

Tabela B.9: Resultados completos do *SAHe-SAT* e  $H_p = \text{Walksat-tabu}$ , para instâncias do conjunto de 2004 [20].

| Instância    | $n$   | $m$    | <i>SAHe-SAT+Walksat-tabu</i> |                | <i>Walksat-tabu</i> |          |
|--------------|-------|--------|------------------------------|----------------|---------------------|----------|
|              |       |        | Num. cls. sat.               | Tempo(m)       | Num. cls. sat.      | Tempo(m) |
| alu4mul      | 4736  | 30465  | <b>30433</b>                 | <b>34.552</b>  | 30433               | 190.191  |
| c3540mul     | 5248  | 33199  | 33170                        | 147.363        | 33170               | 129.171  |
| c6288mul     | 9540  | 61421  | <b>61392</b>                 | 3.884          | 61389               | 23.451   |
| c7552mul     | 11282 | 69529  | <b>69408</b>                 | 31.186         | 69398               | 118.727  |
| C880mul      | 1612  | 9373   | <b>9372</b>                  | <b>3.522</b>   | 9372                | 5.606    |
| dalumul      | 9426  | 59991  | 59896                        | 2.730          | 59910               | 2.546    |
| desmul       | 28902 | 179895 | <b>179639</b>                | 18.186         | 179614              | 55.090   |
| frg1mul      | 3230  | 20575  | <b>20574</b>                 | 141.607        | 20573               | 106.643  |
| frg2mul      | 10316 | 62943  | <b>62897</b>                 | 93.458         | 62893               | 10.831   |
| i10mul       | 12998 | 77941  | <b>77857</b>                 | <b>27.995</b>  | 77857               | 35.209   |
| i8mul        | 14524 | 91139  | 91031                        | 28.980         | 91035               | 66.652   |
| k2mul        | 11680 | 74581  | <b>74351</b>                 | <b>114.570</b> | 74351               | 119.464  |
| rotmul       | 5980  | 35229  | <b>35221</b>                 | 26.747         | 35220               | 125.972  |
| term1mul     | 3504  | 22229  | 22215                        | 179.062        | 22217               | 135.601  |
| vdamul       | 5444  | 34509  | <b>34497</b>                 | <b>22.727</b>  | 34497               | 158.719  |
| x1mul        | 8760  | 55571  | 55563                        | 88.259         | 55564               | 67.999   |
| am_4_4       | 433   | 1458   | <b>1457</b>                  | <b>0.137</b>   | 1457                | 0.192    |
| am_5_5       | 1076  | 3677   | <b>3676</b>                  | <b>0.416</b>   | 3676                | 0.443    |
| am_6_6       | 2269  | 7814   | 7812                         | 233.708        | 7812                | 121.373  |
| am_7_7       | 4264  | 14751  | <b>14734</b>                 | 162.201        | 14731               | 130.805  |
| am_8_8       | 7361  | 25538  | 25466                        | 19.596         | 25477               | 137.402  |
| am_9_9       | 11908 | 41393  | <b>41197</b>                 | 62.596         | 41190               | 89.407   |
| li-exam-61   | 28147 | 108436 | <b>107996</b>                | 69.841         | 107982              | 31.107   |
| li-exam-62   | 28147 | 108436 | 107986                       | 74.585         | 107992              | 53.727   |
| li-exam-63   | 28147 | 108436 | 107980                       | 48.127         | 107983              | 100.824  |
| li-exam-64   | 28147 | 108436 | 107985                       | 3.946          | 107991              | 79.596   |
| li-test4-100 | 36809 | 142491 | 141790                       | 39.717         | 141793              | 66.380   |
| li-test4-101 | 36809 | 142491 | <b>141804</b>                | 19.039         | 141800              | 11.141   |
| li-test4-94  | 36809 | 142491 | <b>141806</b>                | 2.853          | 141780              | 44.018   |
| li-test4-95  | 36809 | 142491 | <b>141796</b>                | 67.216         | 141790              | 72.285   |
| li-test4-96  | 36809 | 142491 | <b>141804</b>                | 24.995         | 141791              | 33.641   |
| li-test4-97  | 36809 | 142491 | 141791                       | 26.641         | 141805              | 22.784   |
| li-test4-98  | 36809 | 142491 | 141787                       | 92.233         | 141791              | 54.528   |
| li-test4-99  | 36809 | 142491 | 141780                       | 96.418         | 141795              | 111.411  |
| ferry10      | 2958  | 20791  | <b>20791</b>                 | <b>0.114</b>   | 20791               | 0.283    |
| ferry10u     | 2883  | 20259  | <b>20259</b>                 | <b>0.113</b>   | 20259               | 0.281    |
| ferry11      | 3562  | 26105  | <b>26105</b>                 | <b>0.118</b>   | 26105               | 0.291    |
| ferry11u     | 3480  | 25499  | <b>25499</b>                 | <b>0.118</b>   | 25499               | 0.295    |
| ferry12      | 4222  | 32199  | 32198                        | 5.985          | 32199               | 26.006   |
| ferry12u     | 4133  | 31515  | <b>31515</b>                 | <b>66.261</b>  | 31515               | 118.714  |
| ferry8       | 1918  | 12311  | <b>12311</b>                 | <b>0.186</b>   | 12311               | 0.511    |
| ferry8u      | 1857  | 11915  | 11915                        | 2.186          | 11915               | 0.255    |
| ferry9       | 2410  | 16209  | <b>16209</b>                 | <b>0.191</b>   | 16209               | 0.362    |
| ferry9u      | 2342  | 15747  | <b>15747</b>                 | <b>0.106</b>   | 15747               | 0.265    |
| gripper10    | 2438  | 19700  | <b>19696</b>                 | <b>11.001</b>  | 19696               | 75.355   |
| gripper10u   | 2312  | 18666  | <b>18662</b>                 | <b>42.287</b>  | 18662               | 58.256   |
| gripper11    | 3222  | 27199  | 27194                        | 5.751          | 27194               | 3.375    |
| gripper11u   | 3084  | 26019  | <b>26014</b>                 | <b>2.383</b>   | 26014               | 10.450   |
| gripper12    | 3502  | 30746  | <b>30740</b>                 | <b>2.009</b>   | 30740               | 7.379    |
| gripper12u   | 3352  | 29412  | <b>29406</b>                 | <b>2.104</b>   | 29406               | 2.788    |



Tabela B.10: Resultados completos do *SAHe-SAT* e  $H_p = Walksat-tabu$ , para instâncias do conjunto de 2004 [20]. (continuação)

| Instância           | $n$    | $m$     | <i>SAHe-SAT+Walksat-tabu</i> |              | <i>Walksat-tabu</i> |          |
|---------------------|--------|---------|------------------------------|--------------|---------------------|----------|
|                     |        |         | Num. cls. sat.               | Tempo(m)     | Num. cls. sat.      | Tempo(m) |
| gripper13           | 4430   | 40461   | 40455                        | 25.375       | 40455               | 2.241    |
| gripper13u          | 4268   | 38965   | <b>38960</b>                 | 88.907       | 38959               | 50.868   |
| gripper14           | 4758   | 45056   | <b>45050</b>                 | 86.154       | 45049               | 4.608    |
| gripper14u          | 4584   | 43390   | <b>43383</b>                 | <b>2.521</b> | 43383               | 93.167   |
| hanoi4              | 1404   | 18058   | <b>18057</b>                 | <b>0.087</b> | 18057               | 0.206    |
| hanoi4u             | 1312   | 16856   | <b>16855</b>                 | <b>0.081</b> | 16855               | 0.198    |
| hanoi5              | 4434   | 76235   | <b>76234</b>                 | <b>0.164</b> | 76234               | 0.379    |
| hanoi5u             | 4292   | 73777   | <b>73776</b>                 | <b>0.159</b> | 73776               | 0.365    |
| abp1-1-k31          | 14809  | 48483   | 48083                        | 9.948        | 48091               | 95.909   |
| abp4-1-k31          | 14809  | 48483   | 48085                        | 40.963       | 48094               | 50.278   |
| bc56-sensors-1-k391 | 561371 | 1778987 | <b>1600245</b>               | 36.329       | 1599963             | 43.683   |
| bc56-sensors-2-k592 | 850398 | 2694319 | <b>2361372</b>               | 33.625       | 2361225             | 39.390   |
| bc57-sensors-1-k303 | 435701 | 1379987 | <b>1264444</b>               | 33.863       | 1264065             | 73.125   |
| dme-03-1-k247       | 261352 | 773077  | <b>740068</b>                | 15.573       | 739924              | 53.964   |
| guidance-1-k56      | 98746  | 307346  | <b>302312</b>                | 18.488       | 302265              | 31.383   |
| motors-stuck-1-k407 | 654766 | 2068742 | <b>1840520</b>               | 79.129       | 1840216             | 68.053   |
| motors-stuck-2-k314 | 505536 | 1596837 | <b>1446222</b>               | 85.452       | 1445841             | 80.114   |
| motors-stuck-2-k315 | 507145 | 1601920 | <b>1450608</b>               | 96.998       | 1450263             | 102.878  |
| valves-gates-1-k617 | 985042 | 3113540 | <b>2706084</b>               | 77.354       | 2705761             | 53.865   |
| 6pipe_6_ooo         | 17064  | 545612  | 545597                       | 126.860      | 545601              | 97.804   |
| 6pipe               | 15800  | 394739  | 394727                       | 48.824       | 394730              | 156.274  |
| 7pipe               | 23910  | 751118  | <b>751108</b>                | 135.140      | 751099              | 878.490  |
| cnf-r4-b1-k1.1-comp | 2424   | 14812   | 14783                        | 54.175       | 14785               | 61.864   |
| cnf-r4-b1-k1.2-comp | 2424   | 14812   | <b>14784</b>                 | 37.851       | 14783               | 14.315   |
| cnt10               | 20470  | 68561   | <b>68315</b>                 | 15.612       | 68306               | 19.907   |
| comb1               | 5910   | 16804   | 16716                        | 116.380      | 16717               | 9.986    |
| comb2               | 31933  | 112462  | <b>111841</b>                | 65.936       | 111815              | 12.575   |
| comb3               | 4774   | 16331   | 16289                        | 8.601        | 16294               | 61.983   |
| dp11u10             | 9197   | 25271   | <b>25259</b>                 | 6.564        | 25258               | 91.325   |
| dp12u11             | 11137  | 30792   | <b>30775</b>                 | 22.338       | 30773               | 27.777   |
| f2clk_40            | 27568  | 80439   | 79556                        | 46.729       | 79565               | 95.691   |
| f2clk_50            | 34678  | 101319  | <b>100087</b>                | 49.688       | 100077              | 96.540   |
| fifo8_300           | 194762 | 530713  | 509196                       | 2.029        | 509252              | 56.291   |
| fifo8_400           | 259762 | 707913  | <b>666252</b>                | 120.056      | 666133              | 31.959   |
| homer17             | 286    | 1742    | <b>1738</b>                  | <b>0.105</b> | 1738                | 0.268    |
| homer18             | 308    | 2030    | <b>2024</b>                  | <b>0.112</b> | 2024                | 0.285    |
| homer19             | 330    | 2340    | <b>2332</b>                  | <b>0.119</b> | 2332                | 0.309    |
| homer20             | 440    | 4220    | <b>4202</b>                  | <b>0.152</b> | 4202                | 0.374    |
| ip36                | 47273  | 153368  | <b>151602</b>                | 116.826      | 151601              | 96.127   |
| ip38                | 49967  | 162142  | <b>160263</b>                | 79.261       | 160209              | 82.170   |
| ip50                | 66131  | 214786  | <b>211950</b>                | 22.018       | 211896              | 100.215  |
| k2fix_gr_2pinvar_w8 | 3771   | 270136  | <b>269860</b>                | 73.909       | 269859              | 83.152   |
| k2fix_gr_2pinvar_w9 | 5028   | 307674  | <b>307490</b>                | 31.671       | 307488              | 117.265  |
| k2fix_gr_2pin_w8    | 9882   | 295998  | 295544                       | 12.155       | 295554              | 43.311   |
| k2fix_gr_2pin_w9    | 13176  | 345426  | <b>345065</b>                | 3.299        | 345044              | 47.063   |
| k2fix_gr_rcs_w8     | 10056  | 271393  | 271302                       | 81.870       | 271302              | 64.336   |
| sha1                | 61377  | 255417  | 251374                       | 30.878       | 251376              | 90.648   |
| sha2                | 61377  | 255417  | 251390                       | 7.669        | 251392              | 16.520   |
| w08_14              | 120367 | 425316  | 0                            | 0.000        | 417608              | 32.191   |
| w08_15              | 132555 | 469519  | <b>460477</b>                | 51.923       | 460435              | 93.114   |

Tabela B.11: Resultados completos do *SAHe-SAT* e  $H_p = \text{Walksat-tabu}$ , para instâncias do conjunto de 2008 [21].

| Instância                     | $n$    | $m$     | <i>SAHe-SAT+Walksat-tabu</i> |               | <i>Walksat-tabu</i> |          |
|-------------------------------|--------|---------|------------------------------|---------------|---------------------|----------|
|                               |        |         | Num. cls. sat.               | Tempo(m)      | Num. cls. sat.      | Tempo(m) |
| b14_opt_bug2_vec1             | 130328 | 402707  | <b>394137</b>                | 104.515       | 394050              | 66.790   |
| b15-bug-fourvec               | 581064 | 1712690 | 1504893                      | 111.709       | 1505178             | 126.398  |
| b15-bug-onevec                | 121836 | 359040  | <b>351594</b>                | 84.999        | 351575              | 129.655  |
| c1_DD_s3_f1_e2_v1-bug-fourvec | 391897 | 989885  | 891535                       | 105.397       | 891640              | 133.470  |
| c1_DD_s3_f1_e2_v1-bug-onevec  | 102234 | 258294  | <b>253466</b>                | 153.792       | 253398              | 53.660   |
| c2_DD_s3_f1_e2_v1-bug-fourvec | 400085 | 1121810 | <b>1022211</b>               | 83.062        | 1022164             | 224.172  |
| c2_DD_s3_f1_e2_v1-bug-onevec  | 84525  | 236942  | 232989                       | 51.955        | 232993              | 18.176   |
| c3_DD_s3_f1_e1_v1-bug-fourvec | 33540  | 86944   | 86238                        | 177.649       | 86241               | 34.934   |
| c3_DD_s3_f1_e1_v1-bug-onevec  | 8385   | 21736   | 21735                        | 8.065         | 21735               | 3.501    |
| c4_DD_s3_f1_e1_v1-bug         | 797728 | 2011216 | 1680756                      | 216.117       | 1680933             | 155.047  |
| c4_DD_s3_f1_e2_v1-bug-fourvec | 448465 | 1130672 | <b>996648</b>                | 133.687       | 996595              | 57.160   |
| c4_DD_s3_f1_e2_v1-bug-onevec  | 131584 | 331754  | <b>321759</b>                | 232.730       | 321560              | 212.594  |
| c5315-bug                     | 1880   | 5049    | <b>5042</b>                  | <b>25.289</b> | 5042                | 54.334   |
| c5_DD_s3_f1_e1_v1-bug-fourvec | 100472 | 270492  | <b>266189</b>                | 35.301        | 266128              | 182.176  |
| c5_DD_s3_f1_e1_v1-bug         | 200944 | 540984  | 519875                       | 34.947        | 520105              | 91.952   |
| c5_DD_s3_f1_e1_v1-bug-onevec  | 25118  | 67623   | <b>67236</b>                 | 25.873        | 67216               | 203.280  |
| c5_DD_s3_f1_e1_v2-bug         | 200944 | 540984  | <b>519926</b>                | 191.911       | 519826              | 44.409   |
| c6288-bug                     | 3462   | 9285    | 9251                         | 30.285        | 9253                | 52.196   |
| c6_DD_s3_f1_e1_v1-bug-fourvec | 170019 | 454050  | <b>439720</b>                | 59.282        | 439710              | 119.551  |
| c6_DD_s3_f1_e1_v1-bug         | 298058 | 795900  | 738012                       | 203.303       | 738073              | 31.146   |
| c6_DD_s3_f1_e1_v1-bug-onevec  | 44079  | 117720  | <b>116679</b>                | 115.254       | 116654              | 93.016   |
| c6_DD_s3_f1_e2_v1-bug-fourvec | 170019 | 454050  | <b>439728</b>                | 135.239       | 439711              | 63.171   |
| c7552-bug                     | 2640   | 7008    | <b>6988</b>                  | 232.742       | 6986                | 57.326   |
| divider-problem.dimacs_10     | 234004 | 760965  | <b>726738</b>                | 200.540       | 726558              | 60.677   |
| divider-problem.dimacs_11     | 215964 | 709377  | <b>680891</b>                | 9.658         | 680269              | 161.695  |
| divider-problem.dimacs_12     | 229482 | 751921  | <b>719132</b>                | 35.368        | 719027              | 83.150   |
| divider-problem.dimacs_1      | 215676 | 708801  | <b>679937</b>                | 227.562       | 679751              | 240.573  |
| divider-problem.dimacs_2      | 228874 | 750705  | 717833                       | 223.271       | 718017              | 37.677   |
| divider-problem.dimacs_3      | 216900 | 711249  | <b>682014</b>                | 57.971        | 681934              | 136.947  |
| divider-problem.dimacs_4      | 225340 | 743637  | 711882                       | 16.019        | 712154              | 154.804  |
| divider-problem.dimacs_5      | 228874 | 750705  | <b>718295</b>                | 200.313       | 718205              | 76.567   |
| divider-problem.dimacs_6      | 234004 | 760965  | <b>727453</b>                | 158.987       | 726924              | 175.403  |
| divider-problem.dimacs_7      | 239640 | 787745  | <b>751378</b>                | 120.835       | 751338              | 122.551  |
| divider-problem.dimacs_8      | 246943 | 810105  | <b>771779</b>                | 171.078       | 771537              | 129.333  |
| divider-problem.dimacs_9      | 222851 | 730905  | 700227                       | 92.137        | 700733              | 229.514  |
| dividers10                    | 45552  | 162874  | <b>160466</b>                | 103.219       | 160445              | 139.450  |
| dividers11                    | 45552  | 162982  | 160446                       | 233.654       | 160454              | 208.052  |
| dividers1                     | 45552  | 162982  | <b>160493</b>                | 18.744        | 160469              | 204.659  |
| dividers2                     | 45552  | 162982  | <b>160449</b>                | 31.287        | 160423              | 100.755  |
| dividers3                     | 45552  | 162982  | <b>160442</b>                | 56.768        | 160424              | 195.332  |
| dividers4                     | 45552  | 162983  | 160431                       | 60.532        | 160441              | 201.775  |
| dividers5                     | 45552  | 162982  | <b>160471</b>                | 180.420       | 160462              | 5.440    |
| dividers6_hack                | 35376  | 132699  | <b>130735</b>                | 232.832       | 130717              | 228.903  |
| dividers7                     | 64532  | 231332  | 227256                       | 34.722        | 227273              | 203.883  |
| dividers8                     | 45552  | 162982  | 160446                       | 94.180        | 160461              | 176.923  |
| dividers9                     | 45552  | 162982  | <b>160474</b>                | 85.106        | 160446              | 22.648   |
| dividers_multivec1            | 106128 | 397650  | <b>388875</b>                | 231.651       | 388855              | 123.486  |
| fpu1-problem                  | 119286 | 409139  | 402565                       | 145.601       | 402602              | 15.742   |
| fpu2-problem                  | 160184 | 548699  | <b>535735</b>                | 144.603       | 535601              | 30.024   |
| fpu3_hack-problem             | 96384  | 347489  | <b>342762</b>                | 79.972        | 342703              | 54.552   |
| fpu3-problem                  | 120114 | 410963  | 404203                       | 61.993        | 404309              | 169.380  |
| fpu4-problem                  | 120150 | 411276  | <b>404546</b>                | 173.913       | 404520              | 36.461   |
| fpu5-problem                  | 320400 | 1099039 | 1030460                      | 237.310       | 1030821             | 99.454   |
| fpu6-problem                  | 160160 | 548748  | 535740                       | 98.594        | 535812              | 175.243  |
| fpu7_hack-problem             | 96438  | 347843  | <b>343099</b>                | 127.156       | 343059              | 47.227   |
| fpu7-problem                  | 120174 | 411323  | 404589                       | 12.263        | 404590              | 114.741  |

Tabela B.12: Resultados completos do *SAHe-SAT* e  $H_p = \text{Walksat-tabu}$ , para instâncias do conjunto de 2008 [21]. (continuação)

| Instância                    | $n$     | $m$      | <i>SAHe-SAT+Walksat-tabu</i> |               | <i>Walksat-tabu</i> |          |
|------------------------------|---------|----------|------------------------------|---------------|---------------------|----------|
|                              |         |          | Num. cls. sat.               | Tempo(m)      | Num. cls. sat.      | Tempo(m) |
| fpu8-problem                 | 160232  | 548848   | 535725                       | 197.044       | 535794              | 55.199   |
| fpu_fsm1-problem             | 160200  | 548843   | 535759                       | 159.065       | 535829              | 71.503   |
| fpu_multivec1-problem        | 257168  | 928310   | 885142                       | 226.594       | 885216              | 39.087   |
| i2c_master1                  | 82429   | 285987   | <b>282384</b>                | 109.260       | 282373              | 7.912    |
| i2c_master2                  | 63816   | 221320   | 218950                       | 238.739       | 218962              | 40.045   |
| i2c-problem.dimacs_25        | 521672  | 1581471  | <b>1407812</b>               | 105.219       | 1407762             | 241.536  |
| i2c-problem.dimacs_26        | 397668  | 1205454  | <b>1096361</b>               | 50.621        | 1096214             | 111.637  |
| mem_ctrl1                    | 1128648 | 4422185  | <b>3841059</b>               | 147.699       | 3840819             | 112.611  |
| mem_ctrl2_blackbox_mc        | 1974822 | 6795573  | <b>5783405</b>               | 160.157       | 5783022             | 10.915   |
| mem_ctrl1-problem            | 4426323 | 15983633 | <b>13369401</b>              | 95.888        | 13368307            | 69.162   |
| mot_comb1._red               | 2159    | 5326     | <b>5325</b>                  | <b>0.149</b>  | 5325                | 0.149    |
| mot_comb2._red               | 5484    | 13894    | <b>13893</b>                 | <b>4.427</b>  | 13893               | 5.928    |
| mot_comb3._red               | 11265   | 29520    | <b>29519</b>                 | <b>0.111</b>  | 29519               | 0.389    |
| mrisc_mem2wire1              | 168960  | 641598   | <b>626476</b>                | <b>72.096</b> | 626476              | 116.855  |
| mrisc_mem2wire-problem       | 844900  | 2905976  | <b>2553334</b>               | 29.392        | 2553238             | 163.772  |
| rsdecoder1_blackbox_CSEBlock | 277950  | 806460   | <b>766775</b>                | 169.292       | 766751              | 107.658  |
| rsdecoder1_blackbox_KESBlock | 707330  | 1106376  | 1028280                      | 55.013        | 1028328             | 33.828   |
| rsdecoder1                   | 238056  | 935187   | <b>897339</b>                | 175.377       | 897264              | 97.510   |
| rsdecoder2_blackbox-problem  | 927691  | 2728843  | <b>2394396</b>               | 103.360       | 2394157             | 238.797  |
| rsdecoder2                   | 415480  | 1632526  | 1515454                      | 236.352       | 1516070             | 175.242  |
| rsdecoder3                   | 238290  | 936006   | <b>898162</b>                | 141.375       | 897896              | 56.708   |
| rsdecoder4                   | 237783  | 933978   | <b>896382</b>                | 100.396       | 896184              | 137.977  |
| rsdecoder5                   | 238290  | 936006   | 897775                       | 46.303        | 897868              | 125.092  |
| rsdecoder6                   | 238290  | 936006   | 897842                       | 209.745       | 898031              | 151.217  |
| rsdecoder_fsm1               | 238290  | 936006   | <b>898045</b>                | 67.392        | 898039              | 109.960  |
| rsdecoder_fsm2               | 238290  | 936006   | 897840                       | 49.467        | 898046              | 220.503  |
| rsdecoder_multivec1          | 394446  | 1626312  | 1518707                      | 164.149       | 1518907             | 230.765  |
| rsdecoder_multivec1-problem  | 627993  | 2125620  | <b>1910474</b>               | 215.828       | 1910444             | 209.890  |
| rsdecoder-problem.dimacs_31  | 1197376 | 3863287  | <b>3332841</b>               | 110.504       | 3332701             | 236.733  |
| rsdecoder-problem.dimacs_34  | 226040  | 728516   | 695848                       | 31.672        | 696112              | 187.987  |
| rsdecoder-problem.dimacs_36  | 1220616 | 3938467  | 3394883                      | 112.451       | 3395043             | 61.259   |
| rsdecoder-problem.dimacs_37  | 1513544 | 4909231  | <b>4198207</b>               | 120.698       | 4198120             | 142.439  |
| rsdecoder-problem.dimacs_38  | 1198012 | 3865513  | 3334250                      | 34.227        | 3334644             | 145.764  |
| rsdecoder-problem.dimacs_39  | 1199602 | 3868693  | 3336867                      | 97.965        | 3337426             | 36.939   |
| rsdecoder-problem.dimacs_40  | 1220616 | 3938467  | 3394438                      | 57.056        | 3394903             | 36.678   |
| rsdecoder-problem.dimacs_41  | 1186710 | 3829036  | 3304320                      | 65.006        | 3304476             | 4.832    |
| s15850-bug-fourvec           | 88544   | 206252   | <b>201642</b>                | 79.018        | 201602              | 5.580    |
| s15850-bug-onevec            | 22136   | 51563    | 51239                        | 59.672        | 51262               | 2.905    |
| s38584-bug-onevec            | 314272  | 819830   | <b>758357</b>                | 32.595        | 758353              | 164.935  |
| SM_AS_TOP_buggy1             | 145900  | 694438   | <b>684931</b>                | 110.533       | 684929              | 21.146   |
| SM_MAIN_MEM_buggy1           | 870975  | 3812147  | 3386126                      | 37.134        | 3386263             | 58.493   |
| SM_RX_TOP                    | 235456  | 934091   | <b>898872</b>                | 52.982        | 898823              | 237.606  |
| spi2                         | 124260  | 515813   | 507223                       | 68.573        | 507226              | 212.581  |
| spi-problem                  | 628192  | 2234134  | 2001122                      | 6.127         | 2001465             | 68.969   |
| wb1                          | 49525   | 140091   | <b>138631</b>                | 33.385        | 138625              | 183.149  |
| wb2                          | 49490   | 140056   | 138197                       | 3.969         | 138218              | 232.134  |
| wb_4m8s1                     | 463080  | 1759150  | <b>1612444</b>               | 64.671        | 1611909             | 95.642   |
| wb_4m8s3                     | 463080  | 1759150  | 1611306                      | 28.162        | 1611462             | 228.215  |
| wb_4m8s4                     | 463080  | 1759150  | 1611219                      | 113.248       | 1611361             | 56.378   |
| wb_4m8s-problem.dimacs_47    | 2691648 | 8517027  | 7133234                      | 33.361        | 7133580             | 44.962   |
| wb_4m8s-problem.dimacs_48    | 2766036 | 8774655  | 7346985                      | 48.814        | 7347258             | 2.952    |
| wb_4m8s-problem.dimacs_49    | 2785108 | 8812799  | 7374846                      | 3.899         | 7374923             | 181.258  |
| wb_conmax1                   | 277950  | 1221020  | <b>1157042</b>               | 60.081        | 1156892             | 201.127  |
| wb_conmax3                   | 277950  | 1221020  | <b>1157179</b>               | 64.238        | 1157029             | 186.785  |
| wb-problem.dimacs_45         | 309491  | 806440   | 749516                       | 118.965       | 749683              | 112.174  |
| wb-problem.dimacs_46         | 300846  | 789283   | 733757                       | 94.903        | 733832              | 33.351   |

## Apêndice C

### Instâncias de *Benchmark* para o Problema *CVM*

Tabela C.1: Conjunto de instâncias de 1996 [65].

| Família                          | Instância      | Vértices | Arestas | Ótimo |
|----------------------------------|----------------|----------|---------|-------|
| <i>brock<math>n_x</math></i>     | brock200_1     | 200      | 5066    | 179   |
|                                  | brock200_2     | 200      | 10024   | 188   |
|                                  | brock200_3     | 200      | 7852    | 185   |
|                                  | brock200_4     | 200      | 6811    | 183   |
|                                  | brock400_1     | 400      | 20077   | 373   |
|                                  | brock400_2     | 400      | 20014   | 371   |
|                                  | brock400_3     | 400      | 20119   | 369   |
|                                  | brock400_4     | 400      | 20035   | 367   |
|                                  | brock800_1     | 800      | 112095  | 777   |
|                                  | brock800_2     | 800      | 111434  | 776   |
|                                  | brock800_3     | 800      | 112267  | 775   |
|                                  | brock800_4     | 800      | 111957  | 774   |
| <i>C<math>n.d</math></i>         | C125.9         | 125      | 787     |       |
|                                  | C250.9         | 250      | 3141    |       |
|                                  | C500.9         | 500      | 12418   |       |
|                                  | C1000.9        | 1000     | 49421   |       |
|                                  | C2000.5        | 2000     | 999164  |       |
|                                  | C2000.9        | 2000     | 199468  |       |
|                                  | C4000.5        | 4000     | 3997732 |       |
| <i>c-fat<math>n-c</math></i>     | c-fat200-1     | 200      | 18366   | 188   |
|                                  | c-fat200-2     | 200      | 16665   | 176   |
|                                  | c-fat200-5     | 200      | 11427   | 142   |
|                                  | c-fat500-1     | 500      | 120291  | 486   |
|                                  | c-fat500-2     | 500      | 115611  | 474   |
|                                  | c-fat500-5     | 500      | 101559  | 436   |
|                                  | c-fat500-10    | 500      | 78123   | 374   |
| <i>DSJC<math>n_x</math></i>      | DSJC500.5      | 500      | 63264   |       |
|                                  | DSJC1000.5     | 1000     | 249674  |       |
| <i>gen<math>n_{pp_c}</math></i>  | gen200_p0.9_44 | 200      | 1990    | 156   |
|                                  | gen200_p0.9_55 | 200      | 1990    | 145   |
|                                  | gen400_p0.9_55 | 400      | 7980    | 345   |
|                                  | gen400_p0.9_65 | 400      | 7980    | 335   |
|                                  | gen400_p0.9_75 | 400      | 7980    | 325   |
| <i>hamming<math>w-d</math></i>   | hamming6-2     | 64       | 192     | 32    |
|                                  | hamming6-4     | 64       | 1312    | 60    |
|                                  | hamming8-2     | 256      | 1024    | 128   |
|                                  | hamming8-4     | 256      | 11776   | 240   |
|                                  | hamming10-2    | 1024     | 5120    | 512   |
|                                  | hamming10-4    | 1024     | 89600   |       |
| <i>johnson<math>w-w-d</math></i> | johnson8-2-4   | 28       | 168     | 24    |
|                                  | johnson8-4-4   | 70       | 560     | 56    |
|                                  | johnson16-2-4  | 120      | 1680    | 112   |
|                                  | johnson32-2-4  | 496      | 14880   |       |
| <i>keller<math>d</math></i>      | keller4        | 171      | 5100    | 160   |
|                                  | keller5        | 776      | 74710   | 749   |
|                                  | keller6        | 3361     | 1026582 |       |
| <i>MANN<math>ax</math></i>       | MANN_a9        | 45       | 72      | 29    |
|                                  | MANN_a27       | 378      | 702     | 252   |
|                                  | MANN_a45       | 1035     | 1980    | 690   |
|                                  | MANN_a81       | 3321     | 6480    |       |
| <i>p_hat<math>n-x</math></i>     | p_hat300-1     | 300      | 33917   | 292   |
|                                  | p_hat300-2     | 300      | 22922   | 275   |
|                                  | p_hat300-3     | 300      | 11460   | 264   |
|                                  | p_hat500-1     | 500      | 93181   | 491   |
|                                  | p_hat500-2     | 500      | 61804   | 464   |
|                                  | p_hat500-3     | 500      | 30950   |       |
|                                  | p_hat700-1     | 700      | 183651  |       |
|                                  | p_hat700-2     | 700      | 122922  |       |
|                                  | p_hat700-3     | 700      | 61640   |       |
|                                  | p_hat1000-1    | 1000     | 377247  |       |
|                                  | p_hat1000-2    | 1000     | 254701  |       |
|                                  | p_hat1000-3    | 1000     | 127754  |       |
|                                  | p_hat1500-1    | 1500     | 839327  |       |
|                                  | p_hat1500-2    | 1500     | 555290  |       |
|                                  | p_hat1500-3    | 1500     | 277006  |       |

Tabela C.2: Conjunto de instâncias de 1996 [65]. (continuação)

| Família                    | Instância                 | Vértices           | Arestas | Ótimo |     |
|----------------------------|---------------------------|--------------------|---------|-------|-----|
| <i>san<sub>n</sub>_f_x</i> | <i>san200_0.7_1</i>       | 200                | 5970    | 170   |     |
|                            | <i>san200_0.7_2</i>       | 200                | 5970    | 182   |     |
|                            | <i>san200_0.9_1</i>       | 200                | 1990    | 130   |     |
|                            | <i>san200_0.9_2</i>       | 200                | 1990    | 140   |     |
|                            | <i>san200_0.9_3</i>       | 200                | 1990    | 156   |     |
|                            | <i>san400_0.5_1</i>       | 400                | 39900   | 387   |     |
|                            | <i>san400_0.7_1</i>       | 400                | 23940   | 360   |     |
|                            | <i>san400_0.7_2</i>       | 400                | 23940   | 370   |     |
|                            | <i>san400_0.7_3</i>       | 400                | 23940   | 378   |     |
|                            | <i>san400_0.9_1</i>       | 400                | 7980    | 300   |     |
|                            | <i>san1000</i>            | 1000               | 249000  | 985   |     |
|                            | <i>sanr<sub>n</sub>_p</i> | <i>sanr200_0.7</i> | 200     | 6032  | 182 |
|                            |                           | <i>sanr200_0.9</i> | 200     | 2037  |     |
| <i>sanr400_0.5</i>         |                           | 400                | 39816   |       |     |
| <i>sanr400_0.7</i>         |                           | 400                | 23931   |       |     |

Tabela C.3: Conjunto de instâncias de 2010 [62].

| Instância         | Vértices | Arestas | Ótimo |
|-------------------|----------|---------|-------|
| <i>frb30-15-1</i> | 450      | 17827   | 420   |
| <i>frb30-15-2</i> | 450      | 17874   | 420   |
| <i>frb30-15-3</i> | 450      | 17809   | 420   |
| <i>frb30-15-4</i> | 450      | 17831   | 420   |
| <i>frb30-15-5</i> | 450      | 17794   | 420   |
| <i>frb35-17-1</i> | 595      | 27856   | 560   |
| <i>frb35-17-2</i> | 595      | 27847   | 560   |
| <i>frb35-17-3</i> | 595      | 27931   | 560   |
| <i>frb35-17-4</i> | 595      | 27842   | 560   |
| <i>frb35-17-5</i> | 595      | 28143   | 560   |
| <i>frb40-19-1</i> | 760      | 41314   | 720   |
| <i>frb40-19-2</i> | 760      | 41263   | 720   |
| <i>frb40-19-3</i> | 760      | 41095   | 720   |
| <i>frb40-19-4</i> | 760      | 41605   | 720   |
| <i>frb40-19-5</i> | 760      | 41619   | 720   |
| <i>frb45-21-1</i> | 945      | 59186   | 900   |
| <i>frb45-21-2</i> | 945      | 58624   | 900   |
| <i>frb45-21-3</i> | 945      | 58245   | 900   |
| <i>frb45-21-4</i> | 945      | 58549   | 900   |
| <i>frb45-21-5</i> | 945      | 58579   | 900   |
| <i>frb50-23-1</i> | 1150     | 80072   | 1100  |
| <i>frb50-23-2</i> | 1150     | 80851   | 1100  |
| <i>frb50-23-3</i> | 1150     | 81068   | 1100  |
| <i>frb50-23-4</i> | 1150     | 80258   | 1100  |
| <i>frb50-23-5</i> | 1150     | 80035   | 1100  |
| <i>frb53-24-1</i> | 1272     | 94227   | 1219  |
| <i>frb53-24-2</i> | 1272     | 94289   | 1219  |
| <i>frb53-24-3</i> | 1272     | 94127   | 1219  |
| <i>frb53-24-4</i> | 1272     | 94308   | 1219  |
| <i>frb53-24-5</i> | 1272     | 94226   | 1219  |
| <i>frb56-25-1</i> | 1400     | 109676  | 1344  |
| <i>frb56-25-2</i> | 1400     | 109401  | 1344  |
| <i>frb56-25-3</i> | 1400     | 109379  | 1344  |
| <i>frb56-25-4</i> | 1400     | 110038  | 1344  |
| <i>frb56-25-5</i> | 1400     | 109601  | 1344  |
| <i>frb59-26-1</i> | 1534     | 126555  | 1475  |
| <i>frb59-26-2</i> | 1534     | 126163  | 1475  |
| <i>frb59-26-3</i> | 1534     | 126082  | 1475  |
| <i>frb59-26-4</i> | 1534     | 127011  | 1475  |
| <i>frb59-26-5</i> | 1534     | 125982  | 1475  |
| <i>frb100-40</i>  | 4000     | 572774  | 3900  |

# Apêndice D

## Resultados Experimentais

### Completos do *SHe-CVM*

Tabela D.1: Resultados completos do *SAHe-CVM* e  $H_p = gcvm$ , para instâncias do conjunto de 1996 [65].

| Família      | Instância      | Vértices | Arestas     | <i>SAHe-CVM+gcvm</i> |               | <i>gcvm</i> |          |
|--------------|----------------|----------|-------------|----------------------|---------------|-------------|----------|
|              |                |          |             | Cobertura            | Tempo(m)      | Cobertura   | Tempo(m) |
| brockn_x     | brock200_1     | 200      | 5066        | <b>179</b>           | 40.248        | 180         | 0.006    |
|              | brock200_2     | 200      | 10024       | <b>188</b>           | 25.483        | 189         | 0.429    |
|              | brock200_3     | 200      | 7852        | <b>186</b>           | <b>0.256</b>  | 186         | 3.597    |
|              | brock200_4     | 200      | 6811        | <b>183</b>           | 65.039        | 184         | 0.011    |
|              | brock400_1     | 400      | 20077       | <b>376</b>           | 0.861         | 377         | 0.043    |
|              | brock400_2     | 400      | 20014       | <b>376</b>           | 1.493         | 377         | 0.624    |
|              | brock400_3     | 400      | 20119       | <b>376</b>           | <b>4.515</b>  | 376         | 172.387  |
|              | brock400_4     | 400      | 20035       | <b>375</b>           | 51.253        | 377         | 5.006    |
|              | brock800_1     | 800      | 112095      | <b>780</b>           | 27.779        | 781         | 53.855   |
|              | brock800_2     | 800      | 111434      | <b>780</b>           | 51.566        | 781         | 0.562    |
| brock800_3   | 800            | 112267   | <b>781</b>  | 8.483                | 782           | 7.335       |          |
| brock800_4   | 800            | 111957   | 781         | 17.337               | 781           | 10.369      |          |
| Cn.d         | C125.9         | 125      | 787         | <b>91</b>            | <b>0.001</b>  | 91          | 0.065    |
|              | C250.9         | 250      | 3141        | <b>206</b>           | 15.389        | 208         | 15.618   |
|              | C500.9         | 500      | 12418       | <b>448</b>           | 18.158        | 449         | 22.031   |
|              | C1000.9        | 1000     | 49421       | <b>941</b>           | 105.085       | 942         | 0.120    |
|              | C2000.5        | 2000     | 999164      | <b>1986</b>          | <b>20.626</b> | 1986        | 22.063   |
|              | C2000.9        | 2000     | 199468      | <b>1935</b>          | 20.891        | 1936        | 11.553   |
|              | C4000.5        | 4000     | 3997732     | 3985                 | 174.480       | 3985        | 164.458  |
| c-fatn-c     | c-fat200-1     | 200      | 18366       | 188                  | 0.060         | 188         | 0.002    |
|              | c-fat200-2     | 200      | 16665       | 176                  | 0.059         | 176         | 0.012    |
|              | c-fat200-5     | 200      | 11427       | 142                  | 0.028         | 142         | 0.001    |
|              | c-fat500-1     | 500      | 120291      | 486                  | 1.086         | 486         | 0.013    |
|              | c-fat500-2     | 500      | 115611      | 474                  | 1.100         | 474         | 0.021    |
|              | c-fat500-5     | 500      | 101559      | 436                  | 0.862         | 436         | 0.011    |
|              | c-fat500-10    | 500      | 78123       | 374                  | 0.506         | 374         | 0.005    |
| DSJCn_x      | DSJC500.5      | 500      | 63264       | 487                  | 14.790        | 487         | 13.117   |
|              | DSJC1000.5     | 1000     | 249674      | <b>986</b>           | <b>2.712</b>  | 986         | 205.580  |
| genn_pp_c    | gen200_p0.9_44 | 200      | 1990        | <b>156</b>           | 9.452         | 157         | 2.222    |
|              | gen200_p0.9_55 | 200      | 1990        | <b>145</b>           | <b>0.027</b>  | 145         | 30.381   |
|              | gen400_p0.9_55 | 400      | 7980        | <b>351</b>           | 21.220        | 353         | 0.009    |
|              | gen400_p0.9_65 | 400      | 7980        | <b>340</b>           | 41.615        | 350         | 23.147   |
|              | gen400_p0.9_75 | 400      | 7980        | <b>325</b>           | 21.581        | 329         | 19.344   |
| hammingW-d   | hamming6-2     | 64       | 192         | 32                   | 0.000         | 32          | 0.000    |
|              | hamming6-4     | 64       | 1312        | 60                   | 0.002         | 60          | 0.000    |
|              | hamming8-2     | 256      | 1024        | 128                  | 0.002         | 128         | 0.000    |
|              | hamming8-4     | 256      | 11776       | 240                  | 0.031         | 240         | 0.001    |
|              | hamming10-2    | 1024     | 5120        | 512                  | 2.703         | 512         | 0.667    |
|              | hamming10-4    | 1024     | 89600       | <b>988</b>           | 44.174        | 990         | 1.051    |
| johnsonW-w-d | johnson8-2-4   | 28       | 168         | 24                   | 0.000         | 24          | 0.000    |
|              | johnson8-4-4   | 70       | 560         | 56                   | 0.001         | 56          | 0.000    |
|              | johnson16-2-4  | 120      | 1680        | 112                  | 0.003         | 112         | 0.000    |
|              | johnson32-2-4  | 496      | 14880       | 480                  | 0.031         | 480         | 0.000    |
| kellerd      | keller4        | 171      | 5100        | 160                  | 0.011         | 160         | 0.004    |
|              | keller5        | 776      | 74710       | <b>753</b>           | 84.902        | 754         | 59.171   |
|              | keller6        | 3361     | 1026582     | <b>3320</b>          | 180.345       | 3323        | 13.914   |
| MANN_ax      | MANN_a9        | 45       | 72          | 29                   | 0.000         | 29          | 0.000    |
|              | MANN_a27       | 378      | 702         | <b>255</b>           | 41.167        | 256         | 58.745   |
|              | MANN_a45       | 1035     | 1980        | <b>702</b>           | 2.040         | 703         | 0.046    |
|              | MANN_a81       | 3321     | 6480        | <b>2240</b>          | <b>0.159</b>  | 2240        | 30.749   |
| p_hatn-x     | p_hat300-1     | 300      | 33917       | 292                  | 0.153         | 292         | 0.002    |
|              | p_hat300-2     | 300      | 22922       | 275                  | 0.079         | 275         | 0.048    |
|              | p_hat300-3     | 300      | 11460       | <b>264</b>           | 23.885        | 266         | 1.402    |
|              | p_hat500-1     | 500      | 93181       | 491                  | 0.669         | 491         | 0.275    |
|              | p_hat500-2     | 500      | 61804       | <b>464</b>           | <b>3.870</b>  | 464         | 10.123   |
|              | p_hat500-3     | 500      | 30950       | <b>450</b>           | 83.408        | 451         | 210.405  |
|              | p_hat700-1     | 700      | 183651      | <b>689</b>           | <b>2.092</b>  | 689         | 6.614    |
|              | p_hat700-2     | 700      | 122922      | 657                  | 12.972        | 656         | 184.272  |
|              | p_hat700-3     | 700      | 61640       | <b>640</b>           | 92.064        | 641         | 97.152   |
|              | p_hat1000-1    | 1000     | 377247      | 990                  | 11.628        | 990         | 6.321    |
|              | p_hat1000-2    | 1000     | 254701      | <b>955</b>           | 156.891       | 956         | 3.143    |
|              | p_hat1000-3    | 1000     | 127754      | <b>937</b>           | 150.598       | 939         | 23.397   |
|              | p_hat1500-1    | 1500     | 839327      | 1489                 | 59.134        | 1489        | 57.577   |
|              | p_hat1500-2    | 1500     | 555290      | <b>1440</b>          | 44.548        | 1442        | 207.045  |
| p_hat1500-3  | 1500           | 277006   | <b>1412</b> | 127.723              | 1414          | 14.328      |          |



Tabela D.2: Resultados completos do *SAHe-CVM* e  $H_p = gcvm$ , para instâncias do conjunto de 1996 [65]. (continuação)

| Família         | Instância           | Vértices | Arestas | <i>SAHe-CVM+gcvm</i> |              | <i>gcvm</i> |          |
|-----------------|---------------------|----------|---------|----------------------|--------------|-------------|----------|
|                 |                     |          |         | Cobertura            | Tempo(m)     | Cobertura   | Tempo(m) |
| <i>sanr_f_x</i> | <i>san200_0.7_1</i> | 200      | 5970    | <b>170</b>           | 59.638       | 184         | 0.766    |
|                 | <i>san200_0.7_2</i> | 200      | 5970    | <b>186</b>           | 6.613        | 187         | 0.291    |
|                 | <i>san200_0.9_1</i> | 200      | 1990    | <b>130</b>           | <b>0.204</b> | 130         | 165.130  |
|                 | <i>san200_0.9_2</i> | 200      | 1990    | <b>140</b>           | <b>0.095</b> | 140         | 0.485    |
|                 | <i>san200_0.9_3</i> | 200      | 1990    | <b>156</b>           | 4.128        | 163         | 12.397   |
|                 | <i>san400_0.5_1</i> | 400      | 39900   | 392                  | 2.200        | 392         | 0.947    |
|                 | <i>san400_0.7_1</i> | 400      | 23940   | <b>379</b>           | 2.532        | 380         | 0.001    |
|                 | <i>san400_0.7_2</i> | 400      | 23940   | <b>384</b>           | <b>0.308</b> | 384         | 4.803    |
|                 | <i>san400_0.7_3</i> | 400      | 23940   | <b>385</b>           | 43.915       | 387         | 0.001    |
|                 | <i>san400_0.9_1</i> | 400      | 7980    | <b>300</b>           | <b>0.351</b> | 300         | 3.266    |
|                 | <i>san1000</i>      | 1000     | 249000  | 992                  | 2.351        | 992         | 0.031    |
| <i>sanrn_p</i>  | <i>sanr200_0.7</i>  | 200      | 6032    | <b>182</b>           | <b>0.296</b> | 182         | 6.781    |
|                 | <i>sanr200_0.9</i>  | 200      | 2037    | <b>158</b>           | 0.950        | 160         | 3.594    |
|                 | <i>sanr400_0.5</i>  | 400      | 39816   | <b>387</b>           | <b>0.890</b> | 387         | 3.861    |
|                 | <i>sanr400_0.7</i>  | 400      | 23931   | <b>379</b>           | <b>8.808</b> | 379         | 142.667  |

Tabela D.3: Resultados completos do *SAHe-CVM* e  $H_p = gcvm$ , para instâncias do conjunto de 2010 [62].

| Instância         | Vértices | Arestas | <i>SAHe-CVM+gcvm</i> |               | <i>gcvm</i> |          |
|-------------------|----------|---------|----------------------|---------------|-------------|----------|
|                   |          |         | Cobertura            | Tempo(m)      | Cobertura   | Tempo(m) |
| <i>frb30-15-1</i> | 450      | 17827   | <b>423</b>           | <b>5.803</b>  | 423         | 102.355  |
| <i>frb30-15-2</i> | 450      | 17874   | <b>423</b>           | 44.054        | 424         | 13.227   |
| <i>frb30-15-3</i> | 450      | 17809   | <b>422</b>           | 40.490        | 424         | 0.196    |
| <i>frb30-15-4</i> | 450      | 17831   | <b>422</b>           | 32.221        | 424         | 10.658   |
| <i>frb30-15-5</i> | 450      | 17794   | <b>422</b>           | 84.083        | 423         | 50.859   |
| <i>frb35-17-1</i> | 595      | 27856   | 564                  | 8.337         | 564         | 0.185    |
| <i>frb35-17-2</i> | 595      | 27847   | <b>564</b>           | 10.470        | 565         | 18.699   |
| <i>frb35-17-3</i> | 595      | 27931   | <b>564</b>           | 6.885         | 565         | 6.033    |
| <i>frb35-17-4</i> | 595      | 27842   | <b>564</b>           | 10.279        | 565         | 49.838   |
| <i>frb35-17-5</i> | 595      | 28143   | <b>563</b>           | 75.905        | 564         | 10.088   |
| <i>frb40-19-1</i> | 760      | 41314   | <b>726</b>           | 20.532        | 728         | 0.313    |
| <i>frb40-19-2</i> | 760      | 41263   | <b>725</b>           | 23.693        | 726         | 2.163    |
| <i>frb40-19-3</i> | 760      | 41095   | <b>725</b>           | 53.915        | 726         | 16.510   |
| <i>frb40-19-4</i> | 760      | 41605   | <b>725</b>           | <b>13.163</b> | 725         | 72.371   |
| <i>frb40-19-5</i> | 760      | 41619   | <b>724</b>           | 102.314       | 725         | 42.050   |
| <i>frb45-21-1</i> | 945      | 59186   | <b>907</b>           | 34.938        | 908         | 17.054   |
| <i>frb45-21-2</i> | 945      | 58624   | <b>906</b>           | 32.132        | 907         | 1.054    |
| <i>frb45-21-3</i> | 945      | 58245   | <b>907</b>           | 42.596        | 908         | 30.822   |
| <i>frb45-21-4</i> | 945      | 58549   | <b>906</b>           | 98.836        | 907         | 135.014  |
| <i>frb45-21-5</i> | 945      | 58579   | <b>906</b>           | 114.280       | 907         | 31.152   |
| <i>frb50-23-1</i> | 1150     | 80072   | <b>1108</b>          | 94.508        | 1109        | 61.222   |
| <i>frb50-23-2</i> | 1150     | 80851   | <b>1108</b>          | 33.036        | 1109        | 2.931    |
| <i>frb50-23-3</i> | 1150     | 81068   | <b>1107</b>          | 158.462       | 1109        | 0.273    |
| <i>frb50-23-4</i> | 1150     | 80258   | <b>1108</b>          | 142.950       | 1110        | 4.333    |
| <i>frb50-23-5</i> | 1150     | 80035   | 1108                 | 115.882       | 1108        | 20.722   |
| <i>frb53-24-1</i> | 1272     | 94227   | <b>1229</b>          | 73.381        | 1230        | 0.359    |
| <i>frb53-24-2</i> | 1272     | 94289   | <b>1228</b>          | <b>11.386</b> | 1228        | 39.768   |
| <i>frb53-24-3</i> | 1272     | 94127   | <b>1227</b>          | 91.868        | 1229        | 5.342    |
| <i>frb53-24-4</i> | 1272     | 94308   | <b>1228</b>          | 101.991       | 1229        | 208.500  |
| <i>frb53-24-5</i> | 1272     | 94226   | <b>1228</b>          | 37.186        | 1229        | 48.273   |
| <i>frb56-25-1</i> | 1400     | 109676  | <b>1355</b>          | <b>34.334</b> | 1355        | 71.437   |
| <i>frb56-25-2</i> | 1400     | 109401  | <b>1354</b>          | 33.612        | 1355        | 83.294   |
| <i>frb56-25-3</i> | 1400     | 109379  | <b>1354</b>          | <b>13.450</b> | 1354        | 129.483  |
| <i>frb56-25-4</i> | 1400     | 110038  | 1354                 | 24.139        | 1354        | 11.570   |
| <i>frb56-25-5</i> | 1400     | 109601  | 1354                 | 121.015       | 1354        | 101.759  |
| <i>frb59-26-1</i> | 1534     | 126555  | 1486                 | 111.628       | 1486        | 41.082   |
| <i>frb59-26-2</i> | 1534     | 126163  | <b>1485</b>          | 186.128       | 1488        | 0.540    |
| <i>frb59-26-3</i> | 1534     | 126082  | <b>1487</b>          | 63.521        | 1488        | 27.218   |
| <i>frb59-26-4</i> | 1534     | 127011  | <b>1486</b>          | 6.750         | 1487        | 14.009   |
| <i>frb59-26-5</i> | 1534     | 125982  | <b>1485</b>          | 2.483         | 1487        | 60.330   |
| <i>frb100-40</i>  | 4000     | 572774  | <b>3926</b>          | 12.170        | 3928        | 3.772    |

Tabela D.4: Resultados completos do *SAHe-CVM* e  $H_p = gwcvm$ , para instâncias do conjunto de 1996 [65].

| Família      | Instância      | Vértices | Arestas     | <i>SAHe-CVM+gwcvm</i> |               | <i>gwcvm</i> |          |
|--------------|----------------|----------|-------------|-----------------------|---------------|--------------|----------|
|              |                |          |             | Cobertura             | Tempo(m)      | Cobertura    | Tempo(m) |
| brockn_x     | brock200_1     | 200      | 5066        | <b>179</b>            | 20.972        | 180          | 63.772   |
|              | brock200_2     | 200      | 10024       | <b>188</b>            | <b>0.029</b>  | 188          | 77.637   |
|              | brock200_3     | 200      | 7852        | <b>185</b>            | 14.097        | 186          | 4.721    |
|              | brock200_4     | 200      | 6811        | <b>183</b>            | 4.907         | 185          | 0.587    |
|              | brock400_1     | 400      | 20077       | <b>376</b>            | 218.458       | 377          | 12.901   |
|              | brock400_2     | 400      | 20014       | <b>376</b>            | 99.938        | 378          | 3.947    |
|              | brock400_3     | 400      | 20119       | <b>376</b>            | 155.181       | 378          | 1.583    |
|              | brock400_4     | 400      | 20035       | <b>377</b>            | 37.471        | 378          | 3.417    |
|              | brock800_1     | 800      | 112095      | <b>781</b>            | 192.816       | 782          | 12.153   |
|              | brock800_2     | 800      | 111434      | <b>781</b>            | 116.204       | 781          | 57.500   |
| brock800_3   | 800            | 112267   | <b>782</b>  | <b>0.767</b>          | 782           | 38.635       |          |
| brock800_4   | 800            | 111957   | <b>781</b>  | <b>85.860</b>         | 781           | 99.837       |          |
| Cn.d         | C125.9         | 125      | 787         | <b>91</b>             | <b>0.042</b>  | 91           | 0.682    |
|              | C250.9         | 250      | 3141        | <b>206</b>            | 16.032        | 208          | 8.939    |
|              | C500.9         | 500      | 12418       | <b>450</b>            | 153.739       | 451          | 132.932  |
|              | C1000.9        | 1000     | 49421       | <b>944</b>            | 71.264        | 946          | 2.612    |
|              | C2000.5        | 2000     | 999164      | <b>1986</b>           | <b>79.646</b> | 1986         | 123.434  |
|              | C2000.9        | 2000     | 199468      | <b>1939</b>           | 57.082        | 1941         | 26.544   |
|              | C4000.5        | 4000     | 3997732     | 3986                  | 172.973       | 3986         | 1.638    |
| c-fatn-c     | c-fat200-1     | 200      | 18366       | 188                   | 0.070         | 188          | 0.001    |
|              | c-fat200-2     | 200      | 16665       | 176                   | 0.063         | 176          | 0.015    |
|              | c-fat200-5     | 200      | 11427       | 142                   | 0.038         | 142          | 0.000    |
|              | c-fat500-1     | 500      | 120291      | 486                   | 1.261         | 486          | 0.066    |
|              | c-fat500-2     | 500      | 115611      | 474                   | 1.105         | 474          | 0.105    |
|              | c-fat500-5     | 500      | 101559      | 436                   | 0.970         | 436          | 0.032    |
|              | c-fat500-10    | 500      | 78123       | 374                   | 0.703         | 374          | 0.027    |
| DSJCn_x      | DSJC500.5      | 500      | 63264       | <b>488</b>            | 12.345        | 489          | 0.381    |
|              | DSJC1000.5     | 1000     | 249674      | <b>987</b>            | <b>2.757</b>  | 987          | 26.608   |
| gen_n_pp_c   | gen200_p0.9_44 | 200      | 1990        | <b>156</b>            | 63.817        | 162          | 2.456    |
|              | gen200_p0.9_55 | 200      | 1990        | <b>145</b>            | 6.102         | 148          | 2.546    |
|              | gen400_p0.9_55 | 400      | 7980        | <b>353</b>            | 8.578         | 354          | 195.876  |
|              | gen400_p0.9_65 | 400      | 7980        | <b>348</b>            | 200.536       | 352          | 58.319   |
|              | gen400_p0.9_75 | 400      | 7980        | <b>335</b>            | 56.805        | 343          | 78.083   |
| hammingW-d   | hamming6-2     | 64       | 192         | 32                    | 0.000         | 32           | 0.000    |
|              | hamming6-4     | 64       | 1312        | 60                    | 0.002         | 60           | 0.000    |
|              | hamming8-2     | 256      | 1024        | <b>128</b>            | <b>0.014</b>  | 128          | 0.161    |
|              | hamming8-4     | 256      | 11776       | <b>240</b>            | <b>0.031</b>  | 240          | 0.131    |
|              | hamming10-2    | 1024     | 5120        | <b>530</b>            | 156.047       | 569          | 0.235    |
|              | hamming10-4    | 1024     | 89600       | <b>990</b>            | 3.041         | 991          | 25.238   |
| johnsonW-w-d | johnson8-2-4   | 28       | 168         | 24                    | 0.000         | 24           | 0.000    |
|              | johnson8-4-4   | 70       | 560         | 56                    | 0.001         | 56           | 0.000    |
|              | johnson16-2-4  | 120      | 1680        | 112                   | 0.003         | 112          | 0.000    |
|              | johnson32-2-4  | 496      | 14880       | 480                   | 0.032         | 480          | 0.001    |
| kellerd      | keller4        | 171      | 5100        | 160                   | 0.011         | 160          | 0.000    |
|              | keller5        | 776      | 74710       | <b>753</b>            | 89.067        | 755          | 0.347    |
|              | keller6        | 3361     | 1026582     | <b>3319</b>           | 221.182       | 3321         | 114.238  |
| MANN_ax      | MANN_a9        | 45       | 72          | 29                    | 0.000         | 29           | 0.000    |
|              | MANN_a27       | 378      | 702         | <b>256</b>            | 27.119        | 257          | 35.579   |
|              | MANN_a45       | 1035     | 1980        | <b>702</b>            | 196.473       | 703          | 0.050    |
|              | MANN_a81       | 3321     | 6480        | 2240                  | 0.930         | 2240         | 0.617    |
| p_hatn-x     | p_hat300-1     | 300      | 33917       | 292                   | 0.445         | 292          | 0.138    |
|              | p_hat300-2     | 300      | 22922       | <b>275</b>            | 6.685         | 276          | 14.678   |
|              | p_hat300-3     | 300      | 11460       | <b>265</b>            | 74.367        | 267          | 2.223    |
|              | p_hat500-1     | 500      | 93181       | <b>491</b>            | <b>1.320</b>  | 491          | 4.818    |
|              | p_hat500-2     | 500      | 61804       | 466                   | 169.013       | 466          | 126.944  |
|              | p_hat500-3     | 500      | 30950       | <b>453</b>            | 49.203        | 456          | 4.309    |
|              | p_hat700-1     | 700      | 183651      | <b>690</b>            | 68.904        | 691          | 6.136    |
|              | p_hat700-2     | 700      | 122922      | <b>660</b>            | 26.130        | 661          | 45.998   |
|              | p_hat700-3     | 700      | 61640       | <b>642</b>            | 47.179        | 646          | 140.313  |
|              | p_hat1000-1    | 1000     | 377247      | <b>990</b>            | 39.693        | 991          | 6.380    |
|              | p_hat1000-2    | 1000     | 254701      | <b>960</b>            | <b>15.797</b> | 960          | 62.116   |
|              | p_hat1000-3    | 1000     | 127754      | <b>940</b>            | 154.780       | 944          | 83.012   |
|              | p_hat1500-1    | 1500     | 839327      | <b>1490</b>           | <b>19.281</b> | 1490         | 19.628   |
|              | p_hat1500-2    | 1500     | 555290      | <b>1445</b>           | 105.686       | 1446         | 150.191  |
| p_hat1500-3  | 1500           | 277006   | <b>1423</b> | 133.433               | 1426          | 88.109       |          |

Tabela D.5: Resultados completos do *SAHe-CVM* e  $H_p = gwcvm$ , para instâncias do conjunto de 1996 [65]. (continuação)

| Família                    | Instância           | Vértices | Arestas | <i>SAHe-CVM+gwcvm</i> |              | <i>gwcvm</i> |          |
|----------------------------|---------------------|----------|---------|-----------------------|--------------|--------------|----------|
|                            |                     |          |         | Cobertura             | Tempo(m)     | Cobertura    | Tempo(m) |
| <i>san<sub>n</sub>_f_x</i> | <i>san200_0.7_1</i> | 200      | 5970    | <b>170</b>            | 2.442        | 183          | 10.587   |
|                            | <i>san200_0.7_2</i> | 200      | 5970    | <b>186</b>            | 20.654       | 187          | 0.009    |
|                            | <i>san200_0.9_1</i> | 200      | 1990    | <b>130</b>            | <b>0.191</b> | 130          | 199.279  |
|                            | <i>san200_0.9_2</i> | 200      | 1990    | <b>140</b>            | 2.467        | 141          | 37.308   |
|                            | <i>san200_0.9_3</i> | 200      | 1990    | <b>156</b>            | 171.693      | 163          | 72.565   |
|                            | <i>san400_0.5_1</i> | 400      | 39900   | <b>392</b>            | <b>7.091</b> | 392          | 13.535   |
|                            | <i>san400_0.7_1</i> | 400      | 23940   | <b>378</b>            | 222.975      | 380          | 0.001    |
|                            | <i>san400_0.7_2</i> | 400      | 23940   | <b>383</b>            | 6.043        | 384          | 1.235    |
|                            | <i>san400_0.7_3</i> | 400      | 23940   | <b>386</b>            | 0.998        | 387          | 0.061    |
|                            | <i>san400_0.9_1</i> | 400      | 7980    | <b>300</b>            | 43.319       | 321          | 75.284   |
|                            | <i>san1000</i>      | 1000     | 249000  | 992                   | 2.735        | 992          | 0.029    |
| <i>sanr<sub>n</sub>_p</i>  | <i>sanr200_0.7</i>  | 200      | 6032    | <b>182</b>            | 5.906        | 183          | 0.391    |
|                            | <i>sanr200_0.9</i>  | 200      | 2037    | <b>158</b>            | 228.260      | 160          | 18.703   |
|                            | <i>sanr400_0.5</i>  | 400      | 39816   | <b>387</b>            | 18.955       | 388          | 24.300   |
|                            | <i>sanr400_0.7</i>  | 400      | 23931   | <b>380</b>            | 16.906       | 381          | 0.387    |

Tabela D.6: Resultados completos do *SAHe-CVM* e  $H_p = gwcvm$ , para instâncias do conjunto de 2010 [62].

| Instância         | Vértices | Arestas | <i>SAHe-CVM+gwcvm</i> |                | <i>gwcvm</i> |          |
|-------------------|----------|---------|-----------------------|----------------|--------------|----------|
|                   |          |         | Cobertura             | Tempo(m)       | Cobertura    | Tempo(m) |
| <i>frb30-15-1</i> | 450      | 17827   | <b>423</b>            | 123.286        | 424          | 82.816   |
| <i>frb30-15-2</i> | 450      | 17874   | <b>423</b>            | 160.258        | 425          | 6.957    |
| <i>frb30-15-3</i> | 450      | 17809   | <b>423</b>            | 123.775        | 424          | 37.104   |
| <i>frb30-15-4</i> | 450      | 17831   | <b>424</b>            | <b>3.732</b>   | 424          | 205.429  |
| <i>frb30-15-5</i> | 450      | 17794   | <b>423</b>            | 46.800         | 424          | 1.801    |
| <i>frb35-17-1</i> | 595      | 27856   | <b>564</b>            | 23.870         | 566          | 1.363    |
| <i>frb35-17-2</i> | 595      | 27847   | <b>564</b>            | 101.871        | 566          | 1.345    |
| <i>frb35-17-3</i> | 595      | 27931   | <b>564</b>            | 67.385         | 565          | 74.816   |
| <i>frb35-17-4</i> | 595      | 27842   | <b>564</b>            | 134.228        | 566          | 0.181    |
| <i>frb35-17-5</i> | 595      | 28143   | <b>564</b>            | <b>110.908</b> | 564          | 124.459  |
| <i>frb40-19-1</i> | 760      | 41314   | 727                   | 1.697          | 727          | 1.072    |
| <i>frb40-19-2</i> | 760      | 41263   | <b>726</b>            | 14.699         | 727          | 17.279   |
| <i>frb40-19-3</i> | 760      | 41095   | <b>726</b>            | 115.444        | 728          | 1.117    |
| <i>frb40-19-4</i> | 760      | 41605   | <b>726</b>            | 31.195         | 727          | 183.318  |
| <i>frb40-19-5</i> | 760      | 41619   | <b>726</b>            | 17.652         | 727          | 3.901    |
| <i>frb45-21-1</i> | 945      | 59186   | <b>907</b>            | 24.350         | 909          | 8.600    |
| <i>frb45-21-2</i> | 945      | 58624   | <b>908</b>            | <b>47.938</b>  | 908          | 82.481   |
| <i>frb45-21-3</i> | 945      | 58245   | <b>907</b>            | 171.836        | 909          | 56.859   |
| <i>frb45-21-4</i> | 945      | 58549   | <b>907</b>            | 19.866         | 909          | 14.911   |
| <i>frb45-21-5</i> | 945      | 58579   | <b>907</b>            | 154.076        | 908          | 57.486   |
| <i>frb50-23-1</i> | 1150     | 80072   | <b>1109</b>           | 196.738        | 1110         | 74.307   |
| <i>frb50-23-2</i> | 1150     | 80851   | <b>1109</b>           | 207.415        | 1110         | 2.808    |
| <i>frb50-23-3</i> | 1150     | 81068   | <b>1109</b>           | 54.792         | 1110         | 4.504    |
| <i>frb50-23-4</i> | 1150     | 80258   | <b>1109</b>           | 76.037         | 1110         | 69.146   |
| <i>frb50-23-5</i> | 1150     | 80035   | <b>1109</b>           | 146.050        | 1110         | 103.688  |
| <i>frb53-24-1</i> | 1272     | 94227   | <b>1229</b>           | 160.478        | 1230         | 241.089  |
| <i>frb53-24-2</i> | 1272     | 94289   | <b>1229</b>           | <b>9.988</b>   | 1229         | 56.334   |
| <i>frb53-24-3</i> | 1272     | 94127   | <b>1229</b>           | <b>29.716</b>  | 1229         | 158.695  |
| <i>frb53-24-4</i> | 1272     | 94308   | <b>1229</b>           | 100.515        | 1230         | 38.649   |
| <i>frb53-24-5</i> | 1272     | 94226   | <b>1229</b>           | 9.400          | 1230         | 9.610    |
| <i>frb56-25-1</i> | 1400     | 109676  | <b>1355</b>           | 58.560         | 1356         | 2.521    |
| <i>frb56-25-2</i> | 1400     | 109401  | <b>1354</b>           | 141.936        | 1357         | 0.462    |
| <i>frb56-25-3</i> | 1400     | 109379  | <b>1355</b>           | 10.175         | 1356         | 87.696   |
| <i>frb56-25-4</i> | 1400     | 110038  | 1355                  | 26.664         | 1355         | 23.246   |
| <i>frb56-25-5</i> | 1400     | 109601  | <b>1355</b>           | 172.122        | 1356         | 2.095    |
| <i>frb59-26-1</i> | 1534     | 126555  | <b>1487</b>           | <b>32.525</b>  | 1487         | 235.905  |
| <i>frb59-26-2</i> | 1534     | 126163  | <b>1487</b>           | 80.126         | 1488         | 7.617    |
| <i>frb59-26-3</i> | 1534     | 126082  | <b>1487</b>           | 127.654        | 1489         | 11.483   |
| <i>frb59-26-4</i> | 1534     | 127011  | <b>1486</b>           | 232.985        | 1489         | 1.535    |
| <i>frb59-26-5</i> | 1534     | 125982  | <b>1487</b>           | 6.944          | 1488         | 4.209    |
| <i>frb100-40</i>  | 4000     | 572774  | <b>3926</b>           | 32.008         | 3927         | 0.047    |

Tabela D.7: Resultados completos do *SAHe-CVM* e  $H_p = hcvm$ , para instâncias do conjunto de 1996 [65].

| Família             | Instância      | Vértices | Arestas | <i>SAHe-CVM+hcvm</i> |                | <i>hcvm</i> |          |
|---------------------|----------------|----------|---------|----------------------|----------------|-------------|----------|
|                     |                |          |         | Cobertura            | Tempo(m)       | Cobertura   | Tempo(m) |
| <i>brockn_x</i>     | brock200_1     | 200      | 5066    | <b>179</b>           | 16.289         | 180         | 0.223    |
|                     | brock200_2     | 200      | 10024   | <b>188</b>           | 7.008          | 189         | 1.134    |
|                     | brock200_3     | 200      | 7852    | <b>186</b>           | <b>0.048</b>   | 186         | 0.140    |
|                     | brock200_4     | 200      | 6811    | <b>183</b>           | 11.827         | 184         | 4.038    |
|                     | brock400_1     | 400      | 20077   | <b>375</b>           | 128.977        | 377         | 2.497    |
|                     | brock400_2     | 400      | 20014   | <b>375</b>           | 16.353         | 377         | 1.876    |
|                     | brock400_3     | 400      | 20119   | <b>375</b>           | 175.693        | 377         | 4.743    |
|                     | brock400_4     | 400      | 20035   | <b>376</b>           | <b>0.323</b>   | 376         | 8.830    |
|                     | brock800_1     | 800      | 112095  | <b>780</b>           | 81.786         | 781         | 15.687   |
|                     | brock800_2     | 800      | 111434  | <b>781</b>           | <b>4.984</b>   | 781         | 30.759   |
|                     | brock800_3     | 800      | 112267  | <b>780</b>           | 32.723         | 781         | 10.470   |
|                     | brock800_4     | 800      | 111957  | <b>781</b>           | <b>10.028</b>  | 781         | 82.665   |
| <i>Cn.d</i>         | C125.9         | 125      | 787     | <b>91</b>            | <b>0.018</b>   | 91          | 0.022    |
|                     | C250.9         | 250      | 3141    | <b>206</b>           | 27.151         | 207         | 33.147   |
|                     | C500.9         | 500      | 12418   | <b>448</b>           | 5.708          | 449         | 0.158    |
|                     | C1000.9        | 1000     | 49421   | <b>941</b>           | <b>3.942</b>   | 941         | 183.862  |
|                     | C2000.5        | 2000     | 999164  | 1986                 | 20.872         | 1985        | 205.851  |
|                     | C2000.9        | 2000     | 199468  | <b>1934</b>          | 202.645        | 1935        | 27.839   |
|                     | C4000.5        | 4000     | 3997732 | <b>3986</b>          | <b>143.234</b> | 3986        | 174.875  |
| <i>c-fatn-c</i>     | c-fat200-1     | 200      | 18366   | 188                  | 0.061          | 188         | 0.002    |
|                     | c-fat200-2     | 200      | 16665   | 176                  | 0.055          | 176         | 0.002    |
|                     | c-fat200-5     | 200      | 11427   | 142                  | 0.028          | 142         | 0.000    |
|                     | c-fat500-1     | 500      | 120291  | 486                  | 1.097          | 486         | 0.013    |
|                     | c-fat500-2     | 500      | 115611  | 474                  | 1.029          | 474         | 0.012    |
|                     | c-fat500-5     | 500      | 101559  | 436                  | 0.859          | 436         | 0.010    |
|                     | c-fat500-10    | 500      | 78123   | 374                  | 0.513          | 374         | 0.011    |
| <i>DSJCn_x</i>      | DSJC500.5      | 500      | 63264   | <b>487</b>           | <b>6.181</b>   | 487         | 169.341  |
|                     | DSJC1000.5     | 1000     | 249674  | <b>986</b>           | <b>90.974</b>  | 986         | 241.171  |
| <i>genn_pp_c</i>    | gen200_p0.9_44 | 200      | 1990    | <b>156</b>           | 2.126          | 157         | 5.706    |
|                     | gen200_p0.9_55 | 200      | 1990    | 145                  | 0.190          | 145         | 0.159    |
|                     | gen400_p0.9_55 | 400      | 7980    | <b>351</b>           | 54.900         | 353         | 1.777    |
|                     | gen400_p0.9_65 | 400      | 7980    | <b>339</b>           | 119.871        | 348         | 34.537   |
|                     | gen400_p0.9_75 | 400      | 7980    | <b>325</b>           | 0.886          | 337         | 0.849    |
| <i>hammingW-d</i>   | hamming6-2     | 64       | 192     | 32                   | 0.000          | 32          | 0.000    |
|                     | hamming6-4     | 64       | 1312    | 60                   | 0.002          | 60          | 0.000    |
|                     | hamming8-2     | 256      | 1024    | 128                  | 0.002          | 128         | 0.000    |
|                     | hamming8-4     | 256      | 11776   | 240                  | 0.031          | 240         | 0.000    |
|                     | hamming10-2    | 1024     | 5120    | <b>512</b>           | <b>0.263</b>   | 512         | 0.519    |
|                     | hamming10-4    | 1024     | 89600   | 988                  | 111.069        | 988         | 1.055    |
| <i>johnsonW-w-d</i> | johnson8-2-4   | 28       | 168     | 24                   | 0.000          | 24          | 0.000    |
|                     | johnson8-4-4   | 70       | 560     | 56                   | 0.001          | 56          | 0.000    |
|                     | johnson16-2-4  | 120      | 1680    | 112                  | 0.003          | 112         | 0.000    |
|                     | johnson32-2-4  | 496      | 14880   | 480                  | 0.032          | 480         | 0.001    |
| <i>kellerd</i>      | keller4        | 171      | 5100    | <b>160</b>           | <b>0.030</b>   | 160         | 0.060    |
|                     | keller5        | 776      | 74710   | <b>754</b>           | <b>26.054</b>  | 754         | 164.981  |
|                     | keller6        | 3361     | 1026582 | 3322                 | 41.383         | 3321        | 137.409  |
| <i>MANN_ax</i>      | MANN_a9        | 45       | 72      | 29                   | 0.000          | 29          | 0.000    |
|                     | MANN_a27       | 378      | 702     | <b>255</b>           | 8.391          | 256         | 9.648    |
|                     | MANN_a45       | 1035     | 1980    | <b>702</b>           | 39.343         | 703         | 0.005    |
|                     | MANN_a81       | 3321     | 6480    | <b>2240</b>          | <b>0.382</b>   | 2240        | 0.715    |
| <i>p_hatn-x</i>     | p_hat300-1     | 300      | 33917   | 292                  | 0.154          | 292         | 0.110    |
|                     | p_hat300-2     | 300      | 22922   | <b>275</b>           | <b>0.094</b>   | 275         | 2.810    |
|                     | p_hat300-3     | 300      | 11460   | <b>264</b>           | 11.857         | 265         | 0.309    |
|                     | p_hat500-1     | 500      | 93181   | 491                  | 0.666          | 491         | 0.514    |
|                     | p_hat500-2     | 500      | 61804   | <b>464</b>           | 4.247          | 465         | 3.744    |
|                     | p_hat500-3     | 500      | 30950   | <b>451</b>           | 3.597          | 452         | 5.728    |
|                     | p_hat700-1     | 700      | 183651  | 689                  | 2.208          | 689         | 2.060    |
|                     | p_hat700-2     | 700      | 122922  | <b>656</b>           | 103.665        | 658         | 223.147  |
|                     | p_hat700-3     | 700      | 61640   | <b>640</b>           | 66.922         | 642         | 2.605    |
|                     | p_hat1000-1    | 1000     | 377247  | 990                  | 17.303         | 990         | 6.384    |
|                     | p_hat1000-2    | 1000     | 254701  | <b>955</b>           | 231.785        | 957         | 8.771    |
|                     | p_hat1000-3    | 1000     | 127754  | <b>937</b>           | 143.357        | 939         | 4.656    |
|                     | p_hat1500-1    | 1500     | 839327  | 1489                 | 73.571         | 1489        | 20.955   |
|                     | p_hat1500-2    | 1500     | 555290  | <b>1438</b>          | 89.805         | 1440        | 137.879  |
|                     | p_hat1500-3    | 1500     | 277006  | <b>1413</b>          | 160.109        | 1416        | 6.143    |

Tabela D.8: Resultados completos do *SAHe-CVM* e  $Hp = hcvm$ , para instâncias do conjunto de 1996 [65]. (continuação)

| Família                    | Instância           | Vértices | Arestas | <i>SAHe-CVM+hcvm</i> |               | <i>hcvm</i> |          |
|----------------------------|---------------------|----------|---------|----------------------|---------------|-------------|----------|
|                            |                     |          |         | Cobertura            | Tempo(m)      | Cobertura   | Tempo(m) |
| <i>san<sub>n</sub>_f_x</i> | <i>san200_0.7_1</i> | 200      | 5970    | <b>184</b>           | <b>0.269</b>  | 184         | 0.616    |
|                            | <i>san200_0.7_2</i> | 200      | 5970    | <b>186</b>           | 1.154         | 187         | 0.647    |
|                            | <i>san200_0.9_1</i> | 200      | 1990    | <b>130</b>           | 0.192         | 132         | 99.388   |
|                            | <i>san200_0.9_2</i> | 200      | 1990    | <b>140</b>           | <b>0.166</b>  | 140         | 0.298    |
|                            | <i>san200_0.9_3</i> | 200      | 1990    | <b>156</b>           | 7.239         | 160         | 38.495   |
|                            | <i>san400_0.5_1</i> | 400      | 39900   | <b>392</b>           | <b>12.319</b> | 392         | 64.697   |
|                            | <i>san400_0.7_1</i> | 400      | 23940   | <b>0</b>             | 0.000         | 380         | 0.001    |
|                            | <i>san400_0.7_2</i> | 400      | 23940   | <b>0</b>             | 0.000         | 384         | 0.062    |
|                            | <i>san400_0.7_3</i> | 400      | 23940   | <b>385</b>           | 101.872       | 386         | 0.882    |
|                            | <i>san400_0.9_1</i> | 400      | 7980    | <b>300</b>           | <b>0.943</b>  | 300         | 6.443    |
|                            | <i>san1000</i>      | 1000     | 249000  | 992                  | 2.695         | 992         | 0.031    |
| <i>sanr<sub>n</sub>_p</i>  | <i>sanr200_0.7</i>  | 200      | 6032    | <b>182</b>           | <b>0.187</b>  | 182         | 0.825    |
|                            | <i>sanr200_0.9</i>  | 200      | 2037    | <b>158</b>           | 35.405        | 159         | 114.668  |
|                            | <i>sanr400_0.5</i>  | 400      | 39816   | <b>387</b>           | <b>0.394</b>  | 387         | 4.615    |
|                            | <i>sanr400_0.7</i>  | 400      | 23931   | <b>379</b>           | 15.745        | 380         | 7.102    |

Tabela D.9: Resultados completos do *SAHe-CVM* e  $Hp = hcvm$ , para instâncias do conjunto de 2010 [62].

| Instância         | Vértices | Arestas | <i>SAHe-CVM+hcvm</i> |                | <i>hcvm</i> |          |
|-------------------|----------|---------|----------------------|----------------|-------------|----------|
|                   |          |         | Cobertura            | Tempo(m)       | Cobertura   | Tempo(m) |
| <i>frb30-15-1</i> | 450      | 17827   | <b>423</b>           | 11.941         | 424         | 0.186    |
| <i>frb30-15-2</i> | 450      | 17874   | <b>423</b>           | 27.787         | 424         | 7.826    |
| <i>frb30-15-3</i> | 450      | 17809   | <b>422</b>           | 49.356         | 424         | 2.390    |
| <i>frb30-15-4</i> | 450      | 17831   | <b>423</b>           | 3.553          | 424         | 4.720    |
| <i>frb30-15-5</i> | 450      | 17794   | <b>422</b>           | 21.230         | 423         | 1.395    |
| <i>frb35-17-1</i> | 595      | 27856   | <b>564</b>           | 45.366         | 565         | 1.400    |
| <i>frb35-17-2</i> | 595      | 27847   | <b>564</b>           | 33.363         | 565         | 3.117    |
| <i>frb35-17-3</i> | 595      | 27931   | <b>564</b>           | 23.398         | 565         | 8.692    |
| <i>frb35-17-4</i> | 595      | 27842   | <b>564</b>           | 14.003         | 565         | 1.208    |
| <i>frb35-17-5</i> | 595      | 28143   | <b>564</b>           | <b>8.354</b>   | 564         | 48.042   |
| <i>frb40-19-1</i> | 760      | 41314   | <b>726</b>           | 68.031         | 727         | 26.325   |
| <i>frb40-19-2</i> | 760      | 41263   | <b>724</b>           | 153.022        | 726         | 51.218   |
| <i>frb40-19-3</i> | 760      | 41095   | <b>725</b>           | 156.876        | 726         | 19.687   |
| <i>frb40-19-4</i> | 760      | 41605   | <b>725</b>           | 83.979         | 726         | 5.053    |
| <i>frb40-19-5</i> | 760      | 41619   | <b>725</b>           | 33.379         | 726         | 12.577   |
| <i>frb45-21-1</i> | 945      | 59186   | <b>907</b>           | 11.349         | 908         | 6.744    |
| <i>frb45-21-2</i> | 945      | 58624   | <b>907</b>           | <b>7.680</b>   | 907         | 86.525   |
| <i>frb45-21-3</i> | 945      | 58245   | <b>907</b>           | 40.498         | 908         | 7.794    |
| <i>frb45-21-4</i> | 945      | 58549   | 906                  | 101.525        | 905         | 123.752  |
| <i>frb45-21-5</i> | 945      | 58579   | <b>907</b>           | <b>15.288</b>  | 907         | 31.694   |
| <i>frb50-23-1</i> | 1150     | 80072   | <b>1108</b>          | 17.478         | 1109        | 210.827  |
| <i>frb50-23-2</i> | 1150     | 80851   | 1108                 | 6.282          | 1107        | 9.795    |
| <i>frb50-23-3</i> | 1150     | 81068   | <b>1107</b>          | 139.772        | 1108        | 24.889   |
| <i>frb50-23-4</i> | 1150     | 80258   | 1108                 | 54.686         | 1108        | 18.201   |
| <i>frb50-23-5</i> | 1150     | 80035   | <b>1108</b>          | <b>35.870</b>  | 1108        | 141.540  |
| <i>frb53-24-1</i> | 1272     | 94227   | <b>1229</b>          | 123.038        | 1230        | 1.007    |
| <i>frb53-24-2</i> | 1272     | 94289   | <b>1228</b>          | 43.004         | 1229        | 1.086    |
| <i>frb53-24-3</i> | 1272     | 94127   | <b>1228</b>          | <b>177.348</b> | 1228        | 236.349  |
| <i>frb53-24-4</i> | 1272     | 94308   | <b>1229</b>          | 9.324          | 1230        | 3.351    |
| <i>frb53-24-5</i> | 1272     | 94226   | <b>1228</b>          | 172.530        | 1230        | 0.335    |
| <i>frb56-25-1</i> | 1400     | 109676  | 1356                 | 24.023         | 1356        | 9.372    |
| <i>frb56-25-2</i> | 1400     | 109401  | <b>1354</b>          | 5.636          | 1355        | 18.682   |
| <i>frb56-25-3</i> | 1400     | 109379  | <b>1354</b>          | 57.414         | 1355        | 27.047   |
| <i>frb56-25-4</i> | 1400     | 110038  | <b>1353</b>          | 98.093         | 1354        | 86.999   |
| <i>frb56-25-5</i> | 1400     | 109601  | <b>1354</b>          | 87.203         | 1355        | 70.379   |
| <i>frb59-26-1</i> | 1534     | 126555  | <b>1486</b>          | 173.395        | 1487        | 0.517    |
| <i>frb59-26-2</i> | 1534     | 126163  | <b>1486</b>          | <b>1.736</b>   | 1486        | 63.423   |
| <i>frb59-26-3</i> | 1534     | 126082  | <b>1487</b>          | 18.229         | 1488        | 0.505    |
| <i>frb59-26-4</i> | 1534     | 127011  | <b>1486</b>          | <b>75.598</b>  | 1486        | 141.629  |
| <i>frb59-26-5</i> | 1534     | 125982  | <b>1486</b>          | 48.438         | 1487        | 219.812  |
| <i>frb100-40</i>  | 4000     | 572774  | <b>3926</b>          | 32.354         | 3927        | 56.345   |

Tabela D.10: Resultados completos do *SAHe-CVM* e  $H_p = hwcvm$ , para instâncias do conjunto de 1996 [65].

| Família      | Instância      | Vértices | Arestas     | <i>SAHe-CVM+hwcvm</i> |               | <i>hwcvm</i> |          |
|--------------|----------------|----------|-------------|-----------------------|---------------|--------------|----------|
|              |                |          |             | Cobertura             | Tempo(m)      | Cobertura    | Tempo(m) |
| brockn_x     | brock200_1     | 200      | 5066        | 179                   | 16.797        | 179          | 14.083   |
|              | brock200_2     | 200      | 10024       | <b>188</b>            | 0.931         | 189          | 38.917   |
|              | brock200_3     | 200      | 7852        | <b>185</b>            | 95.130        | 186          | 4.694    |
|              | brock200_4     | 200      | 6811        | <b>183</b>            | 1.586         | 184          | 0.104    |
|              | brock400_1     | 400      | 20077       | <b>377</b>            | <b>20.270</b> | 377          | 72.112   |
|              | brock400_2     | 400      | 20014       | <b>376</b>            | 80.072        | 378          | 55.462   |
|              | brock400_3     | 400      | 20119       | <b>376</b>            | 69.340        | 378          | 1.694    |
|              | brock400_4     | 400      | 20035       | <b>375</b>            | 104.001       | 378          | 8.306    |
|              | brock800_1     | 800      | 112095      | 781                   | 133.876       | 781          | 46.294   |
|              | brock800_2     | 800      | 111434      | <b>781</b>            | 58.437        | 782          | 33.609   |
| brock800_3   | 800            | 112267   | <b>780</b>  | 39.008                | 782           | 7.040        |          |
| brock800_4   | 800            | 111957   | <b>781</b>  | 31.237                | 782           | 7.497        |          |
| Cn.d         | C125.9         | 125      | 787         | <b>91</b>             | <b>0.130</b>  | 91           | 0.131    |
|              | C250.9         | 250      | 3141        | <b>207</b>            | 13.434        | 209          | 29.921   |
|              | C500.9         | 500      | 12418       | <b>450</b>            | 50.512        | 452          | 44.527   |
|              | C1000.9        | 1000     | 49421       | <b>944</b>            | 50.673        | 946          | 4.437    |
|              | C2000.5        | 2000     | 999164      | <b>1986</b>           | 40.310        | 1987         | 20.801   |
|              | C2000.9        | 2000     | 199468      | <b>1939</b>           | <b>31.443</b> | 1939         | 46.475   |
|              | C4000.5        | 4000     | 3997732     | 3986                  | 143.716       | 3985         | 490.951  |
| c-fatn-c     | c-fat200-1     | 200      | 18366       | 188                   | 0.065         | 188          | 0.003    |
|              | c-fat200-2     | 200      | 16665       | 176                   | 0.062         | 176          | 0.013    |
|              | c-fat200-5     | 200      | 11427       | 142                   | 0.036         | 142          | 0.001    |
|              | c-fat500-1     | 500      | 120291      | 486                   | 1.151         | 486          | 0.013    |
|              | c-fat500-2     | 500      | 115611      | 474                   | 1.108         | 474          | 0.012    |
|              | c-fat500-5     | 500      | 101559      | 436                   | 0.989         | 436          | 0.020    |
|              | c-fat500-10    | 500      | 78123       | 374                   | 0.651         | 374          | 0.024    |
| DSJCN_x      | DSJC500.5      | 500      | 63264       | <b>487</b>            | 0.637         | 488          | 0.354    |
|              | DSJC1000.5     | 1000     | 249674      | 987                   | 73.119        | 987          | 17.066   |
| genn_pp_c    | gen200_p0.9_44 | 200      | 1990        | <b>157</b>            | 13.553        | 160          | 164.822  |
|              | gen200_p0.9_55 | 200      | 1990        | <b>145</b>            | 0.558         | 148          | 115.482  |
|              | gen400_p0.9_55 | 400      | 7980        | <b>352</b>            | 125.241       | 355          | 5.438    |
|              | gen400_p0.9_65 | 400      | 7980        | <b>344</b>            | 198.329       | 347          | 147.190  |
|              | gen400_p0.9_75 | 400      | 7980        | <b>334</b>            | 48.928        | 347          | 127.699  |
| hammingW-d   | hamming6-2     | 64       | 192         | 32                    | 0.000         | 32           | 0.000    |
|              | hamming6-4     | 64       | 1312        | 60                    | 0.002         | 60           | 0.000    |
|              | hamming8-2     | 256      | 1024        | 128                   | 0.188         | 128          | 0.000    |
|              | hamming8-4     | 256      | 11776       | 240                   | 0.119         | 240          | 0.009    |
|              | hamming10-2    | 1024     | 5120        | <b>530</b>            | 19.338        | 586          | 187.174  |
|              | hamming10-4    | 1024     | 89600       | <b>990</b>            | <b>10.063</b> | 990          | 107.323  |
| johnsonW-w-d | johnson8-2-4   | 28       | 168         | 24                    | 0.000         | 24           | 0.000    |
|              | johnson8-4-4   | 70       | 560         | 56                    | 0.001         | 56           | 0.000    |
|              | johnson16-2-4  | 120      | 1680        | 112                   | 0.003         | 112          | 0.000    |
|              | johnson32-2-4  | 496      | 14880       | 480                   | 0.032         | 480          | 0.000    |
| kellerd      | keller4        | 171      | 5100        | <b>160</b>            | <b>0.012</b>  | 160          | 0.015    |
|              | keller5        | 776      | 74710       | <b>752</b>            | 178.541       | 753          | 47.256   |
|              | keller6        | 3361     | 1026582     | <b>3320</b>           | 154.570       | 3321         | 84.543   |
| MANN_ax      | MANN_a9        | 45       | 72          | 29                    | 0.000         | 29           | 0.000    |
|              | MANN_a27       | 378      | 702         | <b>256</b>            | 18.306        | 257          | 0.392    |
|              | MANN_a45       | 1035     | 1980        | <b>703</b>            | <b>0.291</b>  | 703          | 179.475  |
|              | MANN_a81       | 3321     | 6480        | <b>2240</b>           | <b>5.393</b>  | 2240         | 8.661    |
| p_hatn-x     | p_hat300-1     | 300      | 33917       | <b>292</b>            | <b>0.875</b>  | 292          | 2.474    |
|              | p_hat300-2     | 300      | 22922       | <b>275</b>            | 2.062         | 276          | 31.301   |
|              | p_hat300-3     | 300      | 11460       | <b>265</b>            | 6.437         | 266          | 2.190    |
|              | p_hat500-1     | 500      | 93181       | <b>491</b>            | <b>1.327</b>  | 491          | 11.218   |
|              | p_hat500-2     | 500      | 61804       | <b>467</b>            | 6.744         | 468          | 49.244   |
|              | p_hat500-3     | 500      | 30950       | <b>453</b>            | 88.264        | 455          | 13.665   |
|              | p_hat700-1     | 700      | 183651      | <b>689</b>            | 173.653       | 690          | 37.282   |
|              | p_hat700-2     | 700      | 122922      | <b>660</b>            | 14.187        | 661          | 107.743  |
|              | p_hat700-3     | 700      | 61640       | <b>643</b>            | 13.224        | 645          | 165.579  |
|              | p_hat1000-1    | 1000     | 377247      | <b>990</b>            | 129.956       | 991          | 6.403    |
|              | p_hat1000-2    | 1000     | 254701      | <b>959</b>            | 6.448         | 961          | 143.639  |
|              | p_hat1000-3    | 1000     | 127754      | <b>942</b>            | 107.102       | 944          | 24.010   |
|              | p_hat1500-1    | 1500     | 839327      | <b>1490</b>           | <b>37.787</b> | 1490         | 193.833  |
|              | p_hat1500-2    | 1500     | 555290      | <b>1444</b>           | 63.698        | 1446         | 28.078   |
| p_hat1500-3  | 1500           | 277006   | <b>1421</b> | 115.484               | 1424          | 144.977      |          |

Tabela D.11: Resultados completos do *SAHe-CVM* e  $H_p = hwcvm$ , para instâncias do conjunto de 1996 [65]. (continuação)

| Família         | Instância           | Vértices | Arestas | <i>SAHe-CVM+hwcvm</i> |              | <i>hwcvm</i> |          |
|-----------------|---------------------|----------|---------|-----------------------|--------------|--------------|----------|
|                 |                     |          |         | Cobertura             | Tempo(m)     | Cobertura    | Tempo(m) |
| <i>sanr_f_x</i> | <i>san200_0.7_1</i> | 200      | 5970    | <b>170</b>            | <b>2.078</b> | 170          | 60.261   |
|                 | <i>san200_0.7_2</i> | 200      | 5970    | <b>186</b>            | 4.604        | 187          | 0.025    |
|                 | <i>san200_0.9_1</i> | 200      | 1990    | <b>130</b>            | 0.239        | 134          | 59.810   |
|                 | <i>san200_0.9_2</i> | 200      | 1990    | <b>140</b>            | <b>1.147</b> | 140          | 85.086   |
|                 | <i>san200_0.9_3</i> | 200      | 1990    | <b>156</b>            | 131.269      | 164          | 91.250   |
|                 | <i>san400_0.5_1</i> | 400      | 39900   | <b>392</b>            | <b>5.360</b> | 392          | 80.512   |
|                 | <i>san400_0.7_1</i> | 400      | 23940   | 379                   | 10.156       | 379          | 5.328    |
|                 | <i>san400_0.7_2</i> | 400      | 23940   | 384                   | 0.292        | 384          | 0.060    |
|                 | <i>san400_0.7_3</i> | 400      | 23940   | <b>384</b>            | 4.107        | 386          | 30.245   |
|                 | <i>san400_0.9_1</i> | 400      | 7980    | 300                   | 30.029       | 300          | 0.372    |
|                 | <i>san1000</i>      | 1000     | 249000  | 992                   | 2.757        | 992          | 0.031    |
| <i>sanrn_p</i>  | <i>sanr200_0.7</i>  | 200      | 6032    | <b>182</b>            | 2.385        | 183          | 0.889    |
|                 | <i>sanr200_0.9</i>  | 200      | 2037    | <b>158</b>            | 77.743       | 160          | 69.390   |
|                 | <i>sanr400_0.5</i>  | 400      | 39816   | <b>388</b>            | <b>0.193</b> | 388          | 98.975   |
|                 | <i>sanr400_0.7</i>  | 400      | 23931   | <b>380</b>            | 7.740        | 381          | 6.403    |

Tabela D.12: Resultados completos do *SAHe-CVM* e  $H_p = hwcvm$ , para instâncias do conjunto de 2010 [62].

| Instância         | Vértices | Arestas | <i>SAHe-CVM+hwcvm</i> |               | <i>hwcvm</i> |          |
|-------------------|----------|---------|-----------------------|---------------|--------------|----------|
|                   |          |         | Cobertura             | Tempo(m)      | Cobertura    | Tempo(m) |
| <i>frb30-15-1</i> | 450      | 17827   | <b>423</b>            | 111.817       | 425          | 0.191    |
| <i>frb30-15-2</i> | 450      | 17874   | <b>424</b>            | 3.256         | 425          | 1.247    |
| <i>frb30-15-3</i> | 450      | 17809   | <b>423</b>            | 146.187       | 424          | 201.126  |
| <i>frb30-15-4</i> | 450      | 17831   | <b>423</b>            | 158.126       | 425          | 0.084    |
| <i>frb30-15-5</i> | 450      | 17794   | <b>423</b>            | <b>32.501</b> | 423          | 91.681   |
| <i>frb35-17-1</i> | 595      | 27856   | <b>565</b>            | 4.245         | 566          | 13.867   |
| <i>frb35-17-2</i> | 595      | 27847   | <b>564</b>            | 13.367        | 566          | 4.605    |
| <i>frb35-17-3</i> | 595      | 27931   | 564                   | 5.812         | 563          | 105.759  |
| <i>frb35-17-4</i> | 595      | 27842   | <b>565</b>            | 61.053        | 566          | 2.677    |
| <i>frb35-17-5</i> | 595      | 28143   | 565                   | 15.591        | 565          | 9.885    |
| <i>frb40-19-1</i> | 760      | 41314   | <b>726</b>            | 179.226       | 727          | 182.181  |
| <i>frb40-19-2</i> | 760      | 41263   | <b>726</b>            | 146.510       | 727          | 34.340   |
| <i>frb40-19-3</i> | 760      | 41095   | <b>726</b>            | 227.170       | 727          | 0.638    |
| <i>frb40-19-4</i> | 760      | 41605   | <b>726</b>            | 16.172        | 728          | 0.344    |
| <i>frb40-19-5</i> | 760      | 41619   | <b>726</b>            | 1.525         | 727          | 1.725    |
| <i>frb45-21-1</i> | 945      | 59186   | <b>908</b>            | 5.927         | 909          | 0.536    |
| <i>frb45-21-2</i> | 945      | 58624   | <b>907</b>            | 23.973        | 909          | 1.146    |
| <i>frb45-21-3</i> | 945      | 58245   | <b>907</b>            | 184.222       | 909          | 49.483   |
| <i>frb45-21-4</i> | 945      | 58549   | <b>907</b>            | 22.340        | 909          | 0.003    |
| <i>frb45-21-5</i> | 945      | 58579   | <b>907</b>            | 104.109       | 909          | 27.584   |
| <i>frb50-23-1</i> | 1150     | 80072   | <b>1110</b>           | 17.875        | 1111         | 15.949   |
| <i>frb50-23-2</i> | 1150     | 80851   | <b>1109</b>           | 20.896        | 1110         | 2.864    |
| <i>frb50-23-3</i> | 1150     | 81068   | <b>1109</b>           | 22.506        | 1111         | 0.300    |
| <i>frb50-23-4</i> | 1150     | 80258   | <b>1109</b>           | 70.360        | 1110         | 13.116   |
| <i>frb50-23-5</i> | 1150     | 80035   | <b>1109</b>           | 104.710       | 1111         | 13.358   |
| <i>frb53-24-1</i> | 1272     | 94227   | <b>1230</b>           | <b>0.832</b>  | 1230         | 20.084   |
| <i>frb53-24-2</i> | 1272     | 94289   | <b>1229</b>           | 46.775        | 1230         | 16.654   |
| <i>frb53-24-3</i> | 1272     | 94127   | <b>1229</b>           | 46.448        | 1230         | 27.221   |
| <i>frb53-24-4</i> | 1272     | 94308   | <b>1230</b>           | <b>3.258</b>  | 1230         | 220.097  |
| <i>frb53-24-5</i> | 1272     | 94226   | <b>1229</b>           | 38.559        | 1231         | 13.056   |
| <i>frb56-25-1</i> | 1400     | 109676  | <b>1356</b>           | 36.174        | 1357         | 0.427    |
| <i>frb56-25-2</i> | 1400     | 109401  | <b>1355</b>           | <b>97.717</b> | 1355         | 125.947  |
| <i>frb56-25-3</i> | 1400     | 109379  | <b>1355</b>           | 42.054        | 1356         | 2.726    |
| <i>frb56-25-4</i> | 1400     | 110038  | <b>1355</b>           | 22.876        | 1356         | 66.991   |
| <i>frb56-25-5</i> | 1400     | 109601  | <b>1355</b>           | 65.690        | 1356         | 6.273    |
| <i>frb59-26-1</i> | 1534     | 126555  | 1487                  | 192.989       | 1487         | 79.380   |
| <i>frb59-26-2</i> | 1534     | 126163  | <b>1487</b>           | 33.575        | 1488         | 148.599  |
| <i>frb59-26-3</i> | 1534     | 126082  | <b>1487</b>           | 49.754        | 1489         | 0.541    |
| <i>frb59-26-4</i> | 1534     | 127011  | <b>1487</b>           | 50.007        | 1488         | 1.548    |
| <i>frb59-26-5</i> | 1534     | 125982  | <b>1487</b>           | <b>62.391</b> | 1487         | 127.229  |
| <i>frb100-40</i>  | 4000     | 572774  | <b>3926</b>           | 86.979        | 3927         | 4.145    |