

# **Estratégia para Escalonamento de Tarefas em Dois Estágios para Ambientes Heterogêneos**

**José Viterbo Filho**

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Computação Aplicada e Automação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre. Área de concentração: Processamento Distribuído e Paralelo.

Orientador: Profa. Maria Cristina Silva Boeres. PhD.

Niterói, 13 de outubro de 2004.

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

V843 Viterbo Filho, José

Estratégia para escalonamento de tarefas em dois estágios para ambientes heterogêneos / José Viterbo Filho. – Niterói, RJ : [s.n.], 2004.

160 f.

Orientador: Maria Cristina Silva Boeres.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal Fluminense, 2004.

1. Escalonamento de tarefas. 2. Heurísticas - Escalonamento. 3. Algoritmos de escalonamento. 4. Cluster de processadores. I. Título.

CDD 005.136

**JOSÉ VITERBO FILHO**

**ESTRATÉGIA PARA ESCALONAMENTO DE TAREFAS  
EM DOIS ESTÁGIOS PARA AMBIENTES  
HETEROGÊNEOS**

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Computação Aplicada e Automação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre. Área de concentração: Processamento Distribuído e Paralelo.

Aprovada em 13/10/2004

---

Profa. Maria Cristina Silva Boeres, PhD / IC-UFF (Presidente)

---

Profa. Lúcia Maria de Assumpção Drummond, DSc / IC-UFF

---

Prof. Felipe Maia Galvão França, PhD / COPPE Sistemas - UFRJ

NITERÓI

2004

*Ao meu pai, que não tivemos a alegria de  
ter conosco nesse momento.*

# Agradecimentos

Agradeço a todos que de algum modo contribuíram para que eu chegasse até aqui. Aos meus pais, que desde o início proporcionaram condições para meu desenvolvimento pessoal, e em especial à minha mãe, que me incutiu a vontade de ir cada dia mais além. Aos meus irmãos, sempre prontos e dispostos a ajudar. Ao meu amigo Arantes, que me convenceu da importância de dar este passo. Aos amigos Cris, Carlos, Sérgio, André, Lúcio, Lis, Lúcia, Anderson, Waldir, Cabbet, Glauber, Daniela, Clóvis, Arnaldo e tantos outros, que, de alguma forma, estiveram o tempo todo por perto. Ao meu sócio e amigo Nelson, que assumiu uma grande sobrecarga de trabalho para permitir que eu me dedicasse ao curso, e com quem, portanto, divido essa realização. Ao Wagner, pelo apoio significativo na reta final. Aos colegas e amigos Helder, Eduardo, Rodrigo, Stênio, Eyder, Deolinda, Alexandre, Adriana, Cristiane, Geisa, Reinaldo, Jaques, Daniela e Aline, que formaram um ambiente de camaradagem, tornando bastante agradável o convívio na universidade. Aos professores Isabel Cafezeiro, Michael Stanton, Vinod Rebello e, em especial, Lucia Drummond, de quem tive o privilégio de ser aluno e que estimularam meu interesse em prosseguir estudando. E, finalmente, à minha orientadora Cristina Boeres, que me guiou calma e pacientemente ao longo de todo esse período, transmitindo não apenas informações, experiência e conhecimento mas também tranquilidade e confiança. Seu apoio foi determinante para que eu fosse bem-sucedido.

**Resumo da Dissertação de Mestrado apresentada à UFF como requisito parcial para a obtenção do grau de Mestre em Computação Aplicada e Automação (M.Sc.)**

Estratégia para Escalonamento de Tarefas em Dois Estágios para  
Ambientes Heterogêneos

José Viterbo Filho

Outubro, 2004

Orientador: Profa. Maria Cristina Silva Boeres, PhD.  
Pós-Graduação em Computação Aplicada e Automação

O presente trabalho aborda o problema do escalonamento de tarefas de uma aplicação paralela em um conjunto de processadores heterogêneos de um sistema distribuído. Devido à sua grande importância, o escalonamento de tarefas é objeto de um grande número de estudos, e diversas técnicas já foram apresentadas para a solução deste problema que é NP-completo em sua forma geral. Algoritmos do tipo *list scheduling* que levam em consideração a heterogeneidade do ambiente de execução podem parecer, a princípio, mais propícios para o escalonamento de tarefas em ambientes heterogêneos como as grades computacionais. Entretanto, sabe-se

que heurísticas de aglomeração produzem escalonamentos eficientes para ambientes homogêneos, em particular quando os custos de comunicação são mais elevados que a média dos custos de computação. O objetivo deste trabalho é estudar a aplicabilidade de algoritmos baseados em metodologia de aglomeração com replicação já proposta, para o escalonamento de tarefas em grades computacionais. Este trabalho apresenta uma estratégia de escalonamento de aplicações em um número limitado de processadores heterogêneos em dois estágios, chamada de Clusterização para Processadores Heterogêneos (CPH). No primeiro estágio, as coleções de tarefas são construídas com o objetivo de amenizar os efeitos dos custos de comunicação na execução da aplicação. O segundo estágio, o Algoritmo de Mapeamento de Coleções (AMC), proposto neste trabalho de Dissertação, implementa uma política de mapeamento de coleções nos processadores heterogêneos, de forma a minimizar o *makespan* da aplicação. O algoritmo de mapeamento proposto é do tipo *list scheduling* e explora tanto características das coleções quanto do sistema alvo. Para validação da sua aplicabilidade, os resultados produzidos por CPH foram comparados aos resultados produzidos por dois outros algoritmos amplamente estudados na literatura, específicos para ambiente heterogêneos e número limitado de processadores. A análise dos resultados mostra a vantagem em utilizar tal abordagem, principalmente quando os custos de comunicação são dominantes.

**Abstract of Thesis presented to UFF as a partial fulfillment of the requirements for the degree of Master of Science in Computer Science(M.Sc.)**

A Two Stages Cluster-based Strategy for Scheduling Tasks on  
Heterogeneous Processors

José Viterbo Filho

October, 2004

Advisor: Prof. Maria Cristina Silva Boeres, Ph.D  
Department: Computer Science Engineering

This work focuses on the problem of scheduling tasks of a parallel application onto a heterogeneous set of processors in a distributed system. Due to its importance, the task scheduling problem has been the subject of a large amount of studies and several techniques have already been proposed for solving this problem, which is known to be NP-complete in its general form. At first, list scheduling algorithms, which account for the heterogeneity of the target systems, may seem more appropriate for scheduling tasks onto a heterogeneous environment such as computational grids. However, it is known that clustering heuristics produce efficient schedulings for homogeneous environments, particularly when the communication costs are higher



than the mean computation costs. The purpose of this work is to study the applicability of formerly proposed cluster-based algorithms with replication for scheduling tasks onto computational grids. This work presents a two stage strategy for scheduling applications in a bounded set of heterogeneous processors called Clustering for Heterogeneous Processors (CHP). In the first stage, clusters of tasks are created to minimise the communications costs. The second stage, called Cluster Mapping Algorithm (CMA), which is proposed in this work, implements policies for mapping clusters on heterogeneous processors so that the makespan of the application is minimised. The proposed algorithm has a list scheduling framework and exploits the features of both the clusters and the target system. To validate the applicability of the algorithm, the results produced by CHP were compared to those produced by two other well known algorithms from the literature, which are specific for a bounded number of heterogeneous processors. The analysis of the results shows the advantage of applying the proposed approach, particularly when the communication costs are high.

# Palavras-chave

1. Aglomeração de tarefas
2. Heurística de escalonamento estático
3. Ambientes de processadores heterogêneos
4. Latência de comunicação

# Índice

|                                                       |           |
|-------------------------------------------------------|-----------|
| <b>Resumo</b>                                         | <b>iv</b> |
| <b>Abstract</b>                                       | <b>vi</b> |
| <b>1 Introdução</b>                                   | <b>1</b>  |
| 1.1 Organização . . . . .                             | 5         |
| <b>2 Definição do Problema</b>                        | <b>6</b>  |
| 2.1 Modelo da Aplicação . . . . .                     | 6         |
| 2.2 Modelo Arquitetural . . . . .                     | 7         |
| 2.2.1 Modelo de Comunicação . . . . .                 | 8         |
| 2.3 Granularidade . . . . .                           | 9         |
| 2.4 Escalonamento de Tarefas . . . . .                | 10        |
| 2.5 Atributos de um GAD . . . . .                     | 10        |
| 2.6 Resumo . . . . .                                  | 13        |
| <b>3 Heurísticas de Escalonamento</b>                 | <b>15</b> |
| 3.1 Heurísticas de Escalonamento de Tarefas . . . . . | 15        |
| 3.2 Replicação de Tarefas . . . . .                   | 18        |

|          |                                                            |           |
|----------|------------------------------------------------------------|-----------|
| 3.3      | Escalonamento em Ambientes Heterogêneos . . . . .          | 19        |
| 3.3.1    | HEFT . . . . .                                             | 22        |
| 3.3.2    | LDBS . . . . .                                             | 24        |
| 3.4      | Resumo . . . . .                                           | 27        |
| <b>4</b> | <b>Estratégia de Escalonamento em Dois Estágios</b>        | <b>28</b> |
| 4.1      | Primeiro Estágio: HAL . . . . .                            | 31        |
| 4.2      | Atributos de um GAD Escalonado . . . . .                   | 35        |
| 4.3      | Segundo Estágio: AMC . . . . .                             | 37        |
| 4.3.1    | Escolha da Próxima Coleção . . . . .                       | 41        |
| 4.3.2    | Escolha do Processador . . . . .                           | 43        |
| 4.3.3    | Alocação de uma Coleção $c$ ao Processador $p_j$ . . . . . | 47        |
| 4.3.4    | Otimização . . . . .                                       | 50        |
| 4.4      | Resumo . . . . .                                           | 52        |
| <b>5</b> | <b>Resultados: Uma Análise Experimental</b>                | <b>53</b> |
| 5.1      | Análise experimental . . . . .                             | 53        |
| 5.2      | Ambientes Não Restritos . . . . .                          | 55        |
| 5.2.1    | GADs Unitários . . . . .                                   | 56        |
|          | Árvores Binárias Invertidas . . . . .                      | 56        |
|          | Árvores Binárias . . . . .                                 | 60        |
|          | Diamantes . . . . .                                        | 63        |
|          | GADs Randômicos . . . . .                                  | 64        |
| 5.2.2    | Grafos representantes de aplicações reais . . . . .        | 66        |

|                                                                                |            |
|--------------------------------------------------------------------------------|------------|
| Eliminação de Gauss . . . . .                                                  | 66         |
| Dinâmica Molecular . . . . .                                                   | 70         |
| 5.2.3 Grafos PSG . . . . .                                                     | 72         |
| 5.2.4 Número de Processadores . . . . .                                        | 74         |
| 5.3 Ambientes Restritos . . . . .                                              | 75         |
| 5.4 Resumo . . . . .                                                           | 76         |
| <b>6 Conclusões</b>                                                            | <b>77</b>  |
| <b>A Resultados para Árvores Binárias Invertidas</b>                           | <b>81</b>  |
| <b>B Resultados para Árvore Binárias</b>                                       | <b>85</b>  |
| <b>C Resultados para GADs Diamantes</b>                                        | <b>88</b>  |
| <b>D Resultados para GADs Randômicos</b>                                       | <b>91</b>  |
| <b>E Resultados para GADs representantes da Eliminação de Gauss</b>            | <b>94</b>  |
| <b>F Resultados para o GAD representante do Problema da Dinâmica Molecular</b> | <b>97</b>  |
| <b>G Resultados para os GADs PSG</b>                                           | <b>101</b> |
| <b>H Resultados para Ambientes Restritos</b>                                   | <b>108</b> |
| <b>I Notação</b>                                                               | <b>120</b> |
| <b>Bibliografia</b>                                                            | <b>122</b> |

# Lista de Figuras

|     |                                                                                                                                                                                    |    |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1 | GAD de uma aplicação paralela com 10 tarefas. . . . .                                                                                                                              | 7  |
| 3.1 | (a) GAD de uma aplicação paralela com 4 tarefas e (b) o respectivo escalonamento em 2 processadores homogêneos com replicação de tarefas. . . . .                                  | 19 |
| 4.1 | Diagrama de blocos do escalonamento de tarefas para ambientes heterogêneos em dois estágios. . . . .                                                                               | 29 |
| 4.2 | (a) GAD de uma aplicação hipotética com 10 tarefas e (b) o conjunto resultante de coleções necessárias definidas pelo primeiro estágio (HAL). . . . .                              | 35 |
| 4.3 | (a) Árvore binária com 7 tarefas, (b) o escalonamento HAL correspondente, e (c) mapeamento em um ambiente heterogêneo. . . . .                                                     | 39 |
| 4.4 | (a) GAD de uma aplicação, (b) o escalonamento HAL e (c) a escolha da coleção $c_i$ durante a primeira iteração de AMC. . . . .                                                     | 43 |
| 4.5 | (a) Árvore binária com 7 tarefas, (b) o escalonamento HAL correspondente, (c) e (d) passos intermediários do mapeamento do grafo em que $c_2$ é testada em $p_1$ e $p_0$ . . . . . | 47 |
| 4.6 | (a) GAD de uma aplicação, (b) escalonamento fornecido por HAL e (c) mapeamento produzido por AMC. . . . .                                                                          | 49 |

|     |                                                                                                                                                                                                                             |    |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 4.7 | (a) GAD de uma árvore binária, (b) o respectivo escalonamento produzido pelo primeiro estágio, (c) o mapeamento produzido por AMC antes da eliminação de réplicas, (d) e depois da eliminação. . . . .                      | 51 |
| 5.1 | Uma comparação dos escalonamentos gerados por CPH com HEFT e LDBS, em relação aos makespans, em termos de resultados piores ( $>$ ), iguais ( $=$ ) ou melhores ( $<$ ) para GADs <i>árvore binária invertida</i> . . . . . | 57 |
| 5.2 | (a) Árvore binária invertida de 15 tarefas, (b) escalonamento produzido por HAL para $(F, L) = (1, 1)$ , (c) mapeamento produzido por AMC, (d) escalonamento produzido por LDBS. . . . .                                    | 58 |
| 5.3 | Uma comparação dos escalonamentos gerados por CPH com HEFT e LDBS, em relação aos makespans, em termos de resultados piores ( $>$ ), iguais ( $=$ ) ou melhores ( $<$ ) para GADs <i>árvore binária</i> . . . . .           | 60 |
| 5.4 | (a) Árvore binária 15 tarefas, (b) escalonamento produzido por HAL para $(F, L) = (1, 1)$ . . . . .                                                                                                                         | 61 |
| 5.5 | (a) Escalonamento CPH para a árvore binária da figura anterior e (b) resultado produzido por LDBS. . . . .                                                                                                                  | 62 |
| 5.6 | Uma comparação dos escalonamentos gerados por CPH com HEFT e LDBS, em relação aos makespans, em termos de resultados piores ( $>$ ), iguais ( $=$ ) ou melhores ( $<$ ) para GADs <i>diamante</i> . . . . .                 | 63 |
| 5.7 | Uma comparação dos escalonamentos gerados por CPH com HEFT e LDBS, em relação aos makespans, em termos de resultados piores ( $>$ ), iguais ( $=$ ) ou melhores ( $<$ ) para GADs <i>randômicos</i> . . . . .               | 65 |
| 5.8 | Uma comparação dos escalonamentos gerados por CPH com HEFT e LDBS, em relação aos makespans, em termos de resultados piores ( $>$ ), iguais ( $=$ ) ou melhores ( $<$ ) para GADs <i>eliminação de gauss</i> . . . . .      | 67 |

- 5.9 (a) GAD que representa o código da eliminação de gauss de 27 tarefas, (b) o escalonamento HAL correspondente para  $(F, L) = (1, 10)$ , (c) mapeamento resultante do segundo estágio e (d) escalonamento obtido por LDBS. . . . . 68
- 5.10 (a) Trecho do GAD que representa o código da eliminação de Gauss de 54 tarefas e (b) escalonamento parcial obtido por LDBS. . . . . 69
- 5.11 Uma comparação dos escalonamentos gerados por CPH com HEFT e LDBS, em relação aos makespans, em termos de resultados piores ( $>$ ), iguais ( $=$ ) ou melhores ( $<$ ) para GADs *dinâmica molecular*. . . . 70
- 5.12 GAD UET-UDT com 41 tarefas que representa o código de um problema de dinâmica molecular da física. . . . . 71
- 5.13 Uma comparação dos escalonamentos gerados por CPH com HEFT e LDBS, em relação aos makespans, em termos de resultados piores ( $>$ ), iguais ( $=$ ) ou melhores ( $<$ ) para GADs *PSG*. . . . . 72



# Lista de Tabelas

|     |                                                                                                                                                                                                                |    |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1 | Atributos relativos à computação paralela representada pelo GAD da Figura 2.1, considerando o modelo de comunicação onde $L = 1$ . . . . .                                                                     | 12 |
| 4.1 | Atributos relativos ao GAD escalonado da Figura 4.2. . . . .                                                                                                                                                   | 36 |
| 5.1 | Comparação percentual entre os resultados de escalonamento de todos os GADs <i>árvore binária invertida</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1), (5, 1), (1, 1), (1, 5), (1, 10)$ . . . . . | 57 |
| 5.2 | Comparação percentual entre os resultados de escalonamento de todos os GADs <i>árvore binária</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1), (5, 1), (1, 1), (1, 5), (1, 10)$ . . . . .           | 62 |
| 5.3 | Comparação percentual entre os resultados de escalonamento de todos os GADs <i>diamante</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1), (5, 1), (1, 1), (1, 5), (1, 10)$ . . . . .                 | 64 |
| 5.4 | Comparação percentual entre os resultados de escalonamento de todos os GADs <i>randômicos</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1), (5, 1), (1, 1), (1, 5), (1, 10)$ . . . . .               | 65 |
| 5.5 | Comparação percentual entre os resultados de escalonamento de todos os GADs <i>eliminação de gauss</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1), (5, 1), (1, 1), (1, 5), (1, 10)$ . . . . .      | 67 |

|     |                                                                                                                                                                                                                    |    |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 5.6 | Comparação percentual entre os resultados de escalonamento de todos os GADs <i>dinâmica molecular</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1), (5, 1), (1, 1), (1, 5), (1, 10)$ . . . . .           | 72 |
| 5.7 | Comparação percentual entre os resultados de escalonamento de cada GAD <i>PSG</i> obtidos pelos algoritmos HEFT, LDBS e CPH para $(F, L) = (1, 1)$ . . . . .                                                       | 73 |
| 5.8 | Razão entre o número de processadores necessários para o escalonamento CPH e os escalonamentos HEFT e LDBS, para diversas classes de GADs. . . . .                                                                 | 74 |
| 5.9 | Comparação percentual entre os resultados de escalonamento de todos os GADs produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1), (5, 1), (1, 1), (1, 5), (1, 10)$ , em ambientes restritos com $ P  = 20$ . . . | 76 |
| A.1 | Resultados de escalonamento do GAD <i>in-tree-31</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes duais. . . . .                                                                           | 81 |
| A.2 | Resultados de escalonamento do GAD <i>in-tree-31</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (5, 1)$ em ambientes duais. . . . .                                                                            | 82 |
| A.3 | Resultados de escalonamento do GAD <i>in-tree-31</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . . .                                                                            | 82 |
| A.4 | Resultados de escalonamento do GAD <i>in-tree-31</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes duais. . . . .                                                                            | 82 |
| A.5 | Resultados de escalonamento do GAD <i>in-tree-31</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes duais. . . . .                                                                           | 83 |
| A.6 | Resultados de escalonamento do GAD <i>in-tree-31</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes aleatórios. . . . .                                                                      | 83 |
| A.7 | Resultados de escalonamento do GAD <i>in-tree-31</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (5, 1)$ em ambientes aleatórios. . . . .                                                                       | 83 |

|      |                                                                                                                                               |    |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------|----|
| A.8  | Resultados de escalonamento do GAD <i>in-tree-31</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. . . . .  | 83 |
| A.9  | Resultados de escalonamento do GAD <i>in-tree-31</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes aleatórios. . . . .  | 84 |
| A.10 | Resultados de escalonamento do GAD <i>in-tree-31</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes aleatórios. . . . . | 84 |
| B.1  | Resultados de escalonamento do GAD <i>out-tree-127</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes duais. . . .      | 85 |
| B.2  | Resultados de escalonamento do GAD <i>out-tree-127</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (5, 1)$ em ambientes duais. . . .       | 85 |
| B.3  | Resultados de escalonamento do GAD <i>out-tree-127</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . .       | 86 |
| B.4  | Resultados de escalonamento do GAD <i>out-tree-127</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes duais. . . .       | 86 |
| B.5  | Resultados de escalonamento do GAD <i>out-tree-127</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes duais. . . .      | 86 |
| B.6  | Resultados de escalonamento do GAD <i>out-tree-127</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes aleatórios. .     | 87 |
| B.7  | Resultados de escalonamento do GAD <i>out-tree-127</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (5, 1)$ em ambientes aleatórios. .      | 87 |
| B.8  | Resultados de escalonamento do GAD <i>out-tree-127</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. .      | 87 |
| B.9  | Resultados de escalonamento do GAD <i>out-tree-127</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes aleatórios. .      | 87 |
| B.10 | Resultados de escalonamento do GAD <i>out-tree-127</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes aleatórios. .     | 87 |

|      |                                                                                                                                           |    |
|------|-------------------------------------------------------------------------------------------------------------------------------------------|----|
| C.1  | Resultados de escalonamento do GAD <i>diamante-256</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes duais. . . .  | 88 |
| C.2  | Resultados de escalonamento do GAD <i>diamante-256</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (5, 1)$ em ambientes duais. . . .   | 88 |
| C.3  | Resultados de escalonamento do GAD <i>diamante-256</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . .   | 89 |
| C.4  | Resultados de escalonamento do GAD <i>diamante-256</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes duais. . . .   | 89 |
| C.5  | Resultados de escalonamento do GAD <i>diamante-256</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes duais. . . .  | 89 |
| C.6  | Resultados de escalonamento do GAD <i>diamante-256</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes aleatórios. . | 90 |
| C.7  | Resultados de escalonamento do GAD <i>diamante-256</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (5, 1)$ em ambientes aleatórios. .  | 90 |
| C.8  | Resultados de escalonamento do GAD <i>diamante-256</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. .  | 90 |
| C.9  | Resultados de escalonamento do GAD <i>diamante-256</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes aleatórios. .  | 90 |
| C.10 | Resultados de escalonamento do GAD <i>diamante-256</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes aleatórios. . | 90 |
| D.1  | Resultados de escalonamento do GAD <i>randômico-15</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes duais. . . .  | 91 |
| D.2  | Resultados de escalonamento do GAD <i>randômico-15</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (5, 1)$ em ambientes duais. . . .   | 91 |
| D.3  | Resultados de escalonamento do GAD <i>randômico-15</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . .   | 92 |

|      |                                                                                                                                                 |    |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------|----|
| D.4  | Resultados de escalonamento do GAD <i>randômico-15</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes duais. . . . .       | 92 |
| D.5  | Resultados de escalonamento do GAD <i>randômico-15</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes duais. . . . .      | 92 |
| D.6  | Resultados de escalonamento do GAD <i>randômico-15</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes aleatórios. . . . . | 93 |
| D.7  | Resultados de escalonamento do GAD <i>randômico-15</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (5, 1)$ em ambientes aleatórios. . . . .  | 93 |
| D.8  | Resultados de escalonamento do GAD <i>randômico-15</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. . . . .  | 93 |
| D.9  | Resultados de escalonamento do GAD <i>randômico-15</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes aleatórios. . . . .  | 93 |
| D.10 | Resultados de escalonamento do GAD <i>randômico-15</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes aleatórios. . . . . | 93 |
| E.1  | Resultados de escalonamento do GAD <i>gauss-35</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . . .           | 94 |
| E.2  | Resultados de escalonamento do GAD <i>gauss-35</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes duais. . . . .           | 95 |
| E.3  | Resultados de escalonamento do GAD <i>gauss-35</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes duais. . . . .          | 95 |
| E.4  | Resultados de escalonamento do GAD <i>gauss-35</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. . . . .      | 96 |
| E.5  | Resultados de escalonamento do GAD <i>gauss-35</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes aleatórios. . . . .      | 96 |
| E.6  | Resultados de escalonamento do GAD <i>gauss-35</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes aleatórios. . . . .     | 96 |

|      |                                                                                                                                               |     |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------|-----|
| F.1  | Resultados de escalonamento do GAD <i>dinâmica molecular</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes duais. .    | 97  |
| F.2  | Resultados de escalonamento do GAD <i>dinâmica molecular</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (5, 1)$ em ambientes duais. . .   | 98  |
| F.3  | Resultados de escalonamento do GAD <i>dinâmica molecular</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . .   | 98  |
| F.4  | Resultados de escalonamento do GAD <i>dinâmica molecular</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes duais. . .   | 98  |
| F.5  | Resultados de escalonamento do GAD <i>dinâmica molecular</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes duais. .    | 99  |
| F.6  | Resultados de escalonamento do GAD <i>dinâmica molecular</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes aleatórios. | 99  |
| F.7  | Resultados de escalonamento do GAD <i>dinâmica molecular</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (5, 1)$ em ambientes aleatórios.  | 99  |
| F.8  | Resultados de escalonamento do GAD <i>dinâmica molecular</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios.  | 99  |
| F.9  | Resultados de escalonamento do GAD <i>dinâmica molecular</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes aleatórios.  | 100 |
| F.10 | Resultados de escalonamento do GAD <i>dinâmica molecular</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes aleatórios. | 100 |
| G.1  | Resultados de escalonamento do GAD PSG “al-maasarani” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . . .      | 101 |
| G.2  | Resultados de escalonamento do GAD PSG “al-mouhamed” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . . .       | 101 |

|                                                                                                                                                    |     |
|----------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| G.3 Resultados de escalonamento do GAD PSG “g4” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . . .                 | 102 |
| G.4 Resultados de escalonamento do GAD PSG “kruatrachue-fig8” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . . .   | 102 |
| G.5 Resultados de escalonamento do GAD PSG “lctd-eg1” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . . .           | 102 |
| G.6 Resultados de escalonamento do GAD PSG “lwb” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . . .                | 103 |
| G.7 Resultados de escalonamento do GAD PSG “mccreary1” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . . .          | 103 |
| G.8 Resultados de escalonamento do GAD PSG “pbsa-ipp95” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . . .         | 103 |
| G.9 Resultados de escalonamento do GAD PSG “ranka” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . . .              | 104 |
| G.10 Resultados de escalonamento do GAD PSG “tyang1” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . . .            | 104 |
| G.11 Resultados de escalonamento do GAD PSG “tyang4” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . . .            | 104 |
| G.12 Resultados de escalonamento do GAD PSG “al-maasarani” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. . . . . | 105 |
| G.13 Resultados de escalonamento do GAD PSG “al-mouhamed” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. . . . .  | 105 |
| G.14 Resultados de escalonamento do GAD PSG “g4” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. . . . .           | 105 |

|                                                                                                                                                            |     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| G.15 Resultados de escalonamento do GAD PSG “kruatrachue-fig8” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. . . . .     | 105 |
| G.16 Resultados de escalonamento do GAD PSG “lctd-eg1” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. .                   | 105 |
| G.17 Resultados de escalonamento do GAD PSG “lwb” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. .                        | 106 |
| G.18 Resultados de escalonamento do GAD PSG “mccreary1” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios.                    | 106 |
| G.19 Resultados de escalonamento do GAD PSG “pbsa-ipp95” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios.                   | 106 |
| G.20 Resultados de escalonamento do GAD PSG “ranka” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. .                      | 106 |
| G.21 Resultados de escalonamento do GAD PSG “tyang1” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. .                     | 106 |
| G.22 Resultados de escalonamento do GAD PSG “tyang4” produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. .                     | 107 |
| H.1 Resultados de escalonamento do GAD <i>árvore binária invertida</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes duais. . . . . | 108 |
| H.2 Resultados de escalonamento do GAD <i>árvore binária invertida</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (5, 1)$ em ambientes duais. . . . .  | 108 |
| H.3 Resultados de escalonamento do GAD <i>árvore binária invertida</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . . .  | 109 |



|      |                                                                                                                                                             |     |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| H.4  | Resultados de escalonamento do GAD <i>árvore binária invertida</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes duais. . . . .       | 109 |
| H.5  | Resultados de escalonamento do GAD <i>árvore binária invertida</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes duais. . . . .      | 109 |
| H.6  | Resultados de escalonamento do GAD <i>árvore binária invertida</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes aleatórios. . . . . | 110 |
| H.7  | Resultados de escalonamento do GAD <i>árvore binária invertida</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (5, 1)$ em ambientes aleatórios. . . . .  | 110 |
| H.8  | Resultados de escalonamento do GAD <i>árvore binária invertida</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. . . . .  | 110 |
| H.9  | Resultados de escalonamento do GAD <i>árvore binária invertida</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes aleatórios. . . . .  | 110 |
| H.10 | Resultados de escalonamento do GAD <i>árvore binária invertida</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes aleatórios. . . . . | 110 |
| H.11 | Resultados de escalonamento do GAD <i>árvore binária</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes duais. . . .                  | 111 |
| H.12 | Resultados de escalonamento do GAD <i>árvore binária</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (5, 1)$ em ambientes duais. . . .                   | 111 |
| H.13 | Resultados de escalonamento do GAD <i>árvore binária</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . .                   | 111 |

|                                                                                                                                                        |     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| H.14 Resultados de escalonamento do GAD <i>árvore binária</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes duais. . . . .       | 112 |
| H.15 Resultados de escalonamento do GAD <i>árvore binária</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes duais. . . . .      | 112 |
| H.16 Resultados de escalonamento do GAD <i>árvore binária</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes aleatórios. . . . . | 112 |
| H.17 Resultados de escalonamento do GAD <i>árvore binária</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (5, 1)$ em ambientes aleatórios. . . . .  | 112 |
| H.18 Resultados de escalonamento do GAD <i>árvore binária</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. . . . .  | 113 |
| H.19 Resultados de escalonamento do GAD <i>árvore binária</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes aleatórios. . . . .  | 113 |
| H.20 Resultados de escalonamento do GAD <i>árvore binária</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes aleatórios. . . . . | 113 |
| H.21 Resultados de escalonamento do GAD <i>diamante</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes duais. . . . .            | 114 |
| H.22 Resultados de escalonamento do GAD <i>diamante</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (5, 1)$ em ambientes duais. . . . .             | 114 |
| H.23 Resultados de escalonamento do GAD <i>diamante</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . . .             | 114 |
| H.24 Resultados de escalonamento do GAD <i>diamante</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes duais. . . . .             | 115 |
| H.25 Resultados de escalonamento do GAD <i>diamante</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes duais. . . . .            | 115 |
| H.26 Resultados de escalonamento do GAD <i>diamante</i> produzidos por HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes aleatórios. . . . .       | 115 |

|                                                                                                                                                     |     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| H.27 Resultados de escalonamento do GAD <i>diamante</i> produzidos por HEFT,<br>LDBS e CPH para $(F, L) = (5, 1)$ em ambientes aleatórios. . . . .  | 115 |
| H.28 Resultados de escalonamento do GAD <i>diamante</i> produzidos por HEFT,<br>LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. . . . .  | 116 |
| H.29 Resultados de escalonamento do GAD <i>diamante</i> produzidos por HEFT,<br>LDBS e CPH para $(F, L) = (1, 5)$ em ambientes aleatórios. . . . .  | 116 |
| H.30 Resultados de escalonamento do GAD <i>diamante</i> produzidos por HEFT,<br>LDBS e CPH para $(F, L) = (1, 10)$ em ambientes aleatórios. . . . . | 116 |
| H.31 Resultados de escalonamento do GAD <i>randômico</i> produzidos por<br>HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes duais. . . .       | 117 |
| H.32 Resultados de escalonamento do GAD <i>randômico</i> produzidos por<br>HEFT, LDBS e CPH para $(F, L) = (5, 1)$ em ambientes duais. . . .        | 117 |
| H.33 Resultados de escalonamento do GAD <i>randômico</i> produzidos por<br>HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes duais. . . .        | 117 |
| H.34 Resultados de escalonamento do GAD <i>randômico</i> produzidos por<br>HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes duais. . . .        | 118 |
| H.35 Resultados de escalonamento do GAD <i>randômico</i> produzidos por<br>HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes duais. . . .       | 118 |
| H.36 Resultados de escalonamento do GAD <i>randômico</i> produzidos por<br>HEFT, LDBS e CPH para $(F, L) = (10, 1)$ em ambientes aleatórios. .      | 118 |
| H.37 Resultados de escalonamento do GAD <i>randômico</i> produzidos por<br>HEFT, LDBS e CPH para $(F, L) = (5, 1)$ em ambientes aleatórios. .       | 118 |
| H.38 Resultados de escalonamento do GAD <i>randômico</i> produzidos por<br>HEFT, LDBS e CPH para $(F, L) = (1, 1)$ em ambientes aleatórios. .       | 119 |
| H.39 Resultados de escalonamento do GAD <i>randômico</i> produzidos por<br>HEFT, LDBS e CPH para $(F, L) = (1, 5)$ em ambientes aleatórios. .       | 119 |

|                                                                                                                                                |     |
|------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| H.40 Resultados de escalonamento do GAD <i>randômico</i> produzidos por<br>HEFT, LDBS e CPH para $(F, L) = (1, 10)$ em ambientes aleatórios. . | 119 |
|------------------------------------------------------------------------------------------------------------------------------------------------|-----|

# Capítulo 1

## Introdução

A evolução tecnológica vem permitindo que novos computadores cada vez mais velozes sejam produzidos continuamente. Mesmo assim, em diversos campos de pesquisa, a demanda por recursos computacionais ainda mais rápidos é bastante elevada. A simulação numérica de fenômenos experimentais pode exigir tempos de computação de elevada ordem de grandeza mesmo utilizando-se os computadores mais rápidos disponíveis no mercado. Por exemplo, no estudo de recursos para combater o vírus causador da Aids, pesquisadores do Centro de Supercomputação de Pittsburgh necessitam de 328 horas de tempo de computação em um Cray T3E para obter resultados confiáveis em uma simulação de 1ns de movimento de um modelo simplificado da molécula da enzima responsável pela multiplicação do vírus [37].

Em meio a esse panorama de constante busca por maiores velocidades de computação, o que tem sido observado nos últimos anos é que o ganho de velocidade que pode ser obtido com melhoria da tecnologia de projeto e construção de processadores e computadores pode ser expresso em uma taxa bem definida. A capacidade computacional vem dobrando em um período de 18 meses. Por outro lado, os avanços obtidos em comunicação permitem que a velocidade das redes que conectam computadores dobre a cada 9 meses, uma taxa bem maior [23]. Essas tendências apontam para o desenvolvimento de **sistemas computacionais paralelos e distribuídos** como um caminho promissor na obtenção de maiores velocidades de processamento.

Os primeiros computadores paralelos foram projetados e construídos no início dos anos 60, mas só no final dos anos 80 se tornaram comercialmente viáveis [50]. Sistemas computacionais paralelos podem ter diferentes implementações. **Super-computadores** paralelos de alto custo baseados em microprocessadores podem reunir centenas de processadores e fornecer recursos computacionais dedicados para a resolução de problemas complexos [39]. Os avanços na tecnologia de redes de comunicação e interconexão de computadores permitiram a criação de **clusters de computadores** [12]. Estes nada mais são do que agrupamentos de máquinas disponíveis no mercado conectadas por uma rede de comunicação de alta velocidade, dedicados para compartilhamento de recursos de processamento e armazenamento de dados. Desta forma, procura-se obter um melhor desempenho a um custo mais baixo, ao mesmo tempo em que se torna mais acessível a utilização de máquinas paralelas. Recentemente surgiram as **grades computacionais** [24, 25], sistemas de alto desempenho capazes de agregar um grande número de ambientes computacionais distribuídos geograficamente. Supercomputadores, clusters, intranets, bancos de dados, sensores, instrumentos científicos, ou simples computadores pessoais, interconectados pela Internet e redes de alta velocidade, podem fazer parte de uma grade. Esse novo paradigma populariza ainda mais a utilização de processamento paralelo, permitindo que capacidade computacional ociosa seja compartilhada entre diversas instituições, e que um grande número de recursos possam ser reunidos para a execução de tarefas específicas, permitindo um desempenho computacional superior a qualquer sistema já existente.

Entretanto, para que os recursos disponibilizados por esses sistemas distribuídos possam ser aproveitados ao máximo, faz-se necessário que o paralelismo existente nas aplicações seja convenientemente explorado. Para tanto, antes de mais nada deve-se identificar o paralelismo inerente a cada aplicação. Neste trabalho, aplicações podem ser particionadas em tarefas que podem apresentar uma relação de dependência de controle e de dados entre si [18]. Após a identificação dessas tarefas e suas dependências, deve-se decidir como designá-las entre os diversos recursos disponíveis no sistema distribuído em questão, da melhor maneira possível para que se tire proveito de todo o sistema. Essa distribuição não é trivial, e corresponde ao problema

de escalonamento de aplicações em recursos distribuídos.

Devido à sua grande importância, o **escalonamento de tarefas** é objeto de um grande número de estudos [35], e diversas técnicas já foram apresentadas para a solução deste problema que é NP-completo [26, 48, 52] em sua forma geral. Os métodos de escalonamento propostos são desenvolvidos geralmente levando em conta determinadas simplificações do problema, considerando a aplicação paralela ou o ambiente alvo. Por exemplo, muitas heurísticas admitem, para o cálculo do escalonamento, que o ambiente de execução é constituído por um número infinito de processadores homogêneos [32, 47, 59]. Essa simplificação pode não conduzir a resultados práticos eficientes, pois muitos ambientes computacionais disponíveis hoje em dia são heterogêneos. Laboratórios de universidades, por exemplo, tradicionalmente contêm uma variedade de computadores com diferentes configurações conectados através de uma rede de alta velocidade. Até mesmo nos sistemas homogêneos, em que processos disputam o tempo de uso da CPU, provavelmente a capacidade computacional disponível para uma aplicação paralela em cada máquina será diferente [41]. Em consequência disso, nos últimos anos tem sido proposta uma grande variedade de heurísticas de escalonamento que levam em consideração ambientes heterogêneos [4, 15, 53]. A maior parte dessas heurísticas, entretanto, pertence a classe dos algoritmos de *list scheduling* [20, 44, 54, 58]. Isso se deve, provavelmente, à relativa simplicidade e baixa complexidade dos algoritmos dessa classe quando comparados a outros tipos de heurísticas. Além disso, algoritmos do tipo *list scheduling* podem ser adotados facilmente quando o número de processadores no ambiente alvo é limitado.

Algoritmos do tipo *list scheduling* que levam em consideração a heterogeneidade do ambiente de execução podem parecer, a princípio, mais propícios para o escalonamento de tarefas em ambientes heterogêneos como as grades computacionais. Mas por outro lado, sabe-se que heurísticas de aglomeração [45] produzem resultados de escalonamento bem melhores para o caso dos ambientes homogêneos [4, 10], em particular quando os custos de comunicação são mais elevados que a média dos custos de computação. Alguns algoritmos baseados na técnica de replicação de tarefas foram propostos em [9] baseados na metodologia de projeto de algoritmos de aglomeração

apresentada em [8]. Esses algoritmos escalonam aplicações representadas por grafos de tarefas em ambientes com um número ilimitado de processadores, de acordo com o modelo de comunicação LogP [19]. Seus resultados são melhores que aqueles obtidos por outros algoritmos de escalonamento bastante conhecidos tanto para LogP quanto para o modelo de latência [8, 9]. Uma análise teórica em [10] demonstrou que esses algoritmos geram escalonamentos ótimos para o modelo de latência em condições de baixa comunicação [16]. Em particular, a Heurística de Aglomeração para LogP (HAL) [42] apresentou soluções bem melhores para um grande número de grafos de entrada, quando comparadas às soluções dos algoritmos DSC [59] e PLW [47]. O parâmetro considerado nesta comparação é o tempo total de execução paralelo da computação escalonada, ou seja, o *makespan* do escalonamento.

Tradicionalmente, algoritmos de aglomeração são divididos em dois estágios. Durante o primeiro estágio, denominado estágio de aglomeração [45], são criadas coleções onde tarefas são incluídas juntamente com seus predecessores para diminuir o custo de comunicação. O segundo estágio, chamado de estágio de mapeamento, é responsável por alocar as coleções de tarefas resultantes do primeiro estágio nos processadores disponíveis no ambiente alvo. Algoritmos de aglomeração tendem a ser bem-sucedidos porque têm uma visão mais geral da aplicação paralela, enquanto algoritmos de *list scheduling* tomam decisões de acordo com configurações locais.

O objetivo deste trabalho é estudar a aplicabilidade de HAL [42], um dos algoritmos baseados na metodologia de projeto apresentada em [8], para o escalonamento de tarefas em grades computacionais, a fim de tirar proveito dos excelentes resultados obtidos por este algoritmo no escalonamento de tarefas em ambientes homogêneos. Para isso, é necessário desenvolver uma estratégia de escalonamento de aplicações em um número limitado de processadores heterogêneos, que nesse trabalho será realizada em dois estágios. No primeiro estágio, HAL constrói as coleções de tarefas com o objetivo de amenizar os efeitos dos custos de comunicação na execução da aplicação. O segundo estágio, que é proposto neste trabalho de dissertação, implementa uma política de mapeamento de coleções nos processadores heterogêneos. A meta deste processo de mapeamento é distribuir as coleções produzidas pelo primeiro estágio de escalonamento nos processadores reais do sistema distribuído, de forma



a minimizar o *makespan* da aplicação. O algoritmo de mapeamento proposto é do tipo *list scheduling* e explora tanto características das coleções quanto do sistema alvo.

## 1.1 Organização

Este documento está organizado da seguinte forma. No próximo capítulo serão apresentadas as definições teóricas utilizadas neste estudo. No capítulo seguinte, será apresentada uma classificação geral das heurísticas de escalonamento e discutidos, em maior detalhe, os algoritmos de maior interesse para este trabalho. No Capítulo 4, será apresentado o segundo estágio de escalonamento. O Capítulo 5 é dedicado à apresentação e análise dos resultados experimentais obtidos para a validação deste trabalho. Finalmente, no Capítulo 6 serão apresentadas as conclusões decorrentes deste estudo e propostas para trabalhos futuros.

# Capítulo 2

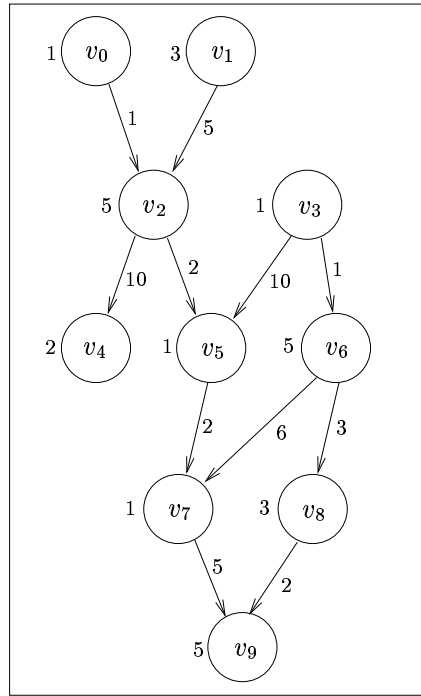
## Definição do Problema

Neste capítulo são apresentados os modelos adotados para a representação da aplicação paralela e do ambiente computacional distribuído. Em seguida, é definido o problema do escalonamento. Finalmente, são apresentados alguns atributos dos grafos acíclicos direcionados utilizados como entrada para as heurísticas de escalonamento deste trabalho.

### 2.1 Modelo da Aplicação

Esse trabalho considera a classe de aplicações paralelas que podem ser representadas por um Grafo Acíclico Direcionado (GAD) [52, 55]. Cada tarefa de uma aplicação é uma unidade de processamento indivisível que é executada após receber todos os dados de entrada de tarefas predecessoras, e após sua execução envia dados de saída para tarefas sucessoras.

Um GAD é denotado por  $G = (V, E, \epsilon, \omega)$ , onde  $V$  é o conjunto de  $n$  nós do grafo e  $E$  é o conjunto de arcos. Cada nó  $v \in V$  representa uma tarefa da aplicação com peso de computação  $\epsilon(v)$  e cada arco  $(u, v) \in E$  representa a restrição de precedência entre as tarefas  $u$  e  $v$ , ou seja, a tarefa  $u$  deve completar sua execução antes que a tarefa  $v$  comece, e ainda, os dados enviados de  $u$  para  $v$  devem estar disponíveis no processador alocado para  $v$  antes de iniciar sua execução. Um peso  $\omega(u, v)$  pode estar



**Figura 2.1: GAD de uma aplicação paralela com 10 tarefas.**

associado ao arco  $(u, v)$  representando a quantidade de dados que são enviados de  $u$  para  $v$ . A Figura 2.1 mostra um exemplo de GAD para uma aplicação hipotética, onde os números ao lado dos nós representam pesos de computação e os números ao lado dos arcos representam pesos de comunicação.

## 2.2 Modelo Arquitetural

O modelo arquitetural define as características da arquitetura paralela a ser considerada. Nesse trabalho, o ambiente computacional é representado por um grafo não direcionado  $Q = (P, R, h, l)$ , onde  $P$  é o conjunto de  $r$  processadores heterogêneos e  $R$  é o conjunto de  $s$  interconexões heterogêneas entre esses processadores [43]. O tempo de execução de uma dada tarefa qualquer  $v$  em um processador  $p \in P$  é dado pelo produto do peso de computação  $\epsilon(v)$  da tarefa pelo *fator de heterogeneidade*  $h(p)$  associado àquele processador, ou seja,  $\epsilon(v) \times h(p)$ . Cada elemento  $(p, q) \in R$  representa a interconexão entre os processadores  $p$  e  $q \in P$ , ou seja, se  $\exists (p, q) \in R$  há uma ligação que permite a troca de dados entre esses dois processadores. Quando

duas tarefas  $u$  e  $v$  adjacentes são alocadas em dois processadores distintos  $p$  e  $q$ , o custo associado com a comunicação de  $\omega(u, v)$  unidades de dados entre esses processadores é definido de acordo com o modelo de comunicação adotado.

### 2.2.1 Modelo de Comunicação

O modelo LogP [19] considera que as características relevantes que afetam o desempenho da aplicação são os parâmetros: custo de transmissão de uma unidade de dados entre dois processadores interconectados (latência  $L$ ); sobrecarga imposta ao processador na preparação para o envio e recebimento de mensagens (*overhead*  $o$ ); tempo mínimo entre duas transmissões subsequentes ou recepções de mensagens em um mesmo processador (*gap*  $g$ ); e número de processadores disponíveis ( $P$ ). Este modelo tende a representar clusters de processadores mais realisticamente [19]. O modelo HLogP [41] define canais e processadores heterogêneos com parâmetros de comunicação semelhantes ao LogP, mas captura a variância dos custos associados a sistemas como as grades computacionais.

Todavia, apesar da existência de modelos arquiteturais que se propõem a representar máquinas atuais, a grande maioria dos algoritmos de escalonamento propostos até hoje considera o modelo de latência [48] como modelo de comunicação padrão [14, 20, 54, 58]. Este modelo tem como único parâmetro de comunicação a latência ( $L$ ), ou seja, o tempo gasto para transmitir uma unidade de dados por um canal de comunicação.

Em um ambiente heterogêneo, não somente os processadores são heterogêneos (modelados através do fator de heterogeneidade já descrito) mas também os canais de comunicação. Desta forma, de acordo com o modelo de latência, um valor de latência  $l(p, q)$  está associado ao segmento  $(p, q)$  representando a taxa de transmissão de dados entre os processadores  $p$  e  $q$ . Duas tarefas  $u$  e  $v$  adjacentes alocadas em dois processadores distintos  $p$  e  $q$ , respectivamente, terão o custo associado com a comunicação de dados igual a  $\omega(u, v) \times l(p, q)$ .

Neste trabalho adotou-se o modelo de latência. Como em outros trabalhos [14],

para simplificação, considerou-se que todos os processadores estão conectados entre si e que a velocidade de comunicação é homogênea, ou seja,  $\forall p, \forall q, l(p, q) = L$ . Esta condição pode representar uma rede local de comunicação entre processadores distintos, em que a latência tende a ser uniforme.

## 2.3 Granularidade

O conceito de granularidade de uma aplicação paralela representada por um GAD pode ser definido genericamente como a razão entre a quantidade de computação e a quantidade de comunicação do grafo [34, 55, 59]. Desta forma, quando uma aplicação tem um elevado custo de computação em relação ao custo de comunicação, diz-se que tem granularidade grossa. Por outro lado, quando os custos de comunicação são maiores do que os de computação, diz-se que tem granularidade fina. Este conceito é importante pois a relação entre comunicação e computação estabelece até que ponto é vantajoso paralelizar ou não uma aplicação [5]. Há diferentes formulações para o cálculo da granularidade [27, 30, 52]. É importante salientar que as definições de granularidade foram propostas até então para ambientes homogêneos. Neste trabalho, definiremos uma formulação híbrida para granularidade, como uma tentativa de classificar as relações entre computação e comunicação antes de escalonar o grafo. A formulação utilizada é similar à definição proposta em [34] para o  $ccr$ , que é a taxa entre o custo de comunicação e computação de um GAD:

$$ccr = \frac{\sum_{\forall (u,v) \in E} \omega(u, v) \times L}{\sum_{\forall v \in V} \epsilon(v) \times \bar{h}}, \quad (2.1)$$

onde  $L$  é a latência da rede de comunicação e  $\bar{h}$  é o fator de heterogeneidade médio, ou seja, a média dos fatores de heterogeneidade dos processadores disponíveis neste ambiente. Este parâmetro é usado para estimar o custo de computação de uma dada tarefa  $v \in V$  no ambiente, como  $\epsilon(v) \times \bar{h}$ . A partir dessa definição, diz-se que um grafo é de granularidade fina se  $ccr > 1$ , e de granularidade grossa se  $ccr \leq 1$ .

## 2.4 Escalonamento de Tarefas

Conhecendo-se a aplicação paralela e os processadores disponíveis no ambiente alvo para sua execução, deve-se decidir como as tarefas que constituem essa aplicação serão distribuídas entre os diversos processadores que fazem parte deste ambiente. Define-se um escalonamento  $S$  como um conjunto finito de tuplas  $(v_i, p_j, t_l)$ , cada uma especificando que uma tarefa  $v_i \in V$  será executada no processador  $p_j \in P$  iniciando no tempo  $t_l$ . Cada tarefa  $v_i \in V$  deve ser escalonada em pelo menos um processador  $p_j \in P$ , e ela só pode ser executada depois que todos os seus predecessores imediatos tenham completado a sua execução e que os respectivos dados necessários para a sua execução estejam disponíveis em  $p_j$ .

Os processadores que estão alocados a uma determinada tarefa  $v_i$  são denotados por  $proc(v_i)$ . Denota-se o tempo de início de execução de uma tarefa  $v_i$  em um processador  $p_j$  por  $t_I(v_i, p_j)$ . Consequentemente, o tempo de fim da tarefa  $v_i$  em  $p_j$ , denotado por  $t_F$ , é igual a  $t_I(v_i, p_j) + \epsilon(v_i) \times h(p_j)$ . O tempo total de execução da aplicação paralela  $G$  no ambiente  $P$  de acordo com o escalonamento  $S$ , também chamado de comprimento do escalonamento, tempo de execução do escalonamento ou *makespan* do escalonamento, é denotado por  $mspan(S)$  e pode ser definido como

$$mspan(S) = \max_{\forall v \in V, \forall p \in P} \{t_I(v, p) + \epsilon(v) \times h(p)\}. \quad (2.2)$$

## 2.5 Atributos de um GAD

Para auxiliar o processo de escalonamento de um GAD, podem ser definidos uma série de atributos que são normalmente utilizados por diversos algoritmos de escalonamento para ambientes homogêneos. Esses atributos podem ser úteis também no escalonamento em ambientes heterogêneos, uma vez que é difícil estabelecer atributos análogos específicos para estes ambientes. Como esses atributos levam em consideração os custos e computação e comunicação, seu cálculo utiliza parâmetros ambientais relativos à velocidade de processamento e de comunicação de dados. Em nosso modelo de ambiente, em que os processadores são heterogêneos, faz-se

necessário definir o fator de heterogeneidade médio  $\bar{h}$  como a média dos fatores de heterogeneidade dos processadores disponíveis neste ambiente. Este parâmetro é usado para estimar o custo de computação de uma dada tarefa  $v \in V$  no ambiente, como  $\epsilon(v) \times \bar{h}$ .

O conjunto de tarefas predecessoras imediatas de uma tarefa  $v \in V$ , é denotado por  $pred(v)$ , onde  $pred(v) = \{u | (u, v) \in E\}$ . De forma semelhante, o conjunto das tarefas sucessoras imediatas de  $v$  é denotado por  $succ(v)$ , onde  $succ(v) = \{w | (v, w) \in E\}$ . Tarefas origem são os nós de entrada de um GAD, aqueles que não possuem nenhum predecessor, ou seja,  $pred(v) = \emptyset$ . Tarefas destino são os nós de saída de um GAD, aqueles que não possuem nenhum sucessor, ou seja,  $succ(v) = \emptyset$ .

O *nível topológico* de um nó  $v$ , denotado por  $level(v)$ , representa o número de tarefas no maior caminho de um nó origem até o nó  $v$ , ou seja,

$$level(v) = \max_{\forall u \in pred(v)} \{1 + level(u)\}, \quad (2.3)$$

onde  $level(v) = 0$ , se  $pred(v) = \emptyset$ , isto é, se  $v$  é um nó origem. O *tlevel* (*co-nível* ou *top level*) de um nó  $v$  é o tamanho do maior caminho de um nó origem até o nó  $v$ , excetuando-se do cálculo o custo deste nó, ou seja,

$$tlevel(v) = \max_{\forall u \in pred(v)} \{tlevel(u) + \epsilon(u) \times \bar{h} + \omega(u, v) \times L\}, \quad (2.4)$$

onde  $tlevel(v) = 0$ , se  $pred(v) = \emptyset$ , isto é, se  $v$  é um nó origem. O *blevel* (*nível* ou *bottom level*) de um nó  $v$  é o tamanho do maior caminho desde um nó  $v$  até um nó destino, considerando-se os custos de computação e comunicação associados com o grafo de entrada, ou seja,

$$blevel(v) = \max_{\forall w \in succ(v)} \{blevel(w) + \omega(v, w) \times L\} + \epsilon(v) \times \bar{h}, \quad (2.5)$$

onde  $blevel(v) = \epsilon(v)$ , se  $succ(v) = \emptyset$ , isto é,  $v$  é um nó de saída. O *blevel* e o *tlevel* são atributos normalmente utilizados para estabelecimento de prioridades de seleção de tarefas em algoritmos de escalonamento do tipo *list scheduling* [35, 56, 58].

O predecessor crítico de uma tarefa  $v$ , aqui denotado por  $cri(v)$ , é a tarefa predecessora imediata de  $v$  que faz parte do maior caminho de um nó origem até o nó  $v$ , ou seja,  $cri(v) \in pred(v)$  e  $\forall u \in pred(v)$  temos:

$$tlevel(cri(v)) + \epsilon(cri(v)) \times \bar{h} + \omega(cri(v), v) \times L \geq tlevel(u) + \epsilon(u) \times \bar{h} + \omega(u, v) \times L \quad (2.6)$$

onde  $cri(v) = \emptyset$ , se  $pred(v) = \emptyset$ , isto é, se  $v$  é um nó origem. O caminho crítico do grafo, denotado por  $CP$ , é formado pelo conjunto de tarefas que descrevem o caminho mais longo que se pode percorrer dentro do grafo, ou seja,

$$u \in CP, \text{ sse } \exists v \in CP \mid u = cri(v)$$

O custo do caminho crítico do GAD  $G$ , denotado por  $ccrit(G)$ , é  $\max_{v \in V} \{blevel(v)\}$ .

A Tabela 2.1, apresentada abaixo, mostra os valores de alguns atributos correspondentes ao GAD da Figura 2.1.

| $v_i$ | $\epsilon(v_i)$ | $succ(v_i)$ | $pred(v_i)$ | $cri(v_i)$  | $level(v_i)$ | $tlevel(v_i)$ | $blevel(v_i)$ |
|-------|-----------------|-------------|-------------|-------------|--------------|---------------|---------------|
| 0     | 1               | $\{2\}$     | $\emptyset$ | $\emptyset$ | 0            | 0             | 22            |
| 1     | 3               | $\{2\}$     | $\emptyset$ | $\emptyset$ | 0            | 0             | 29            |
| 2     | 5               | $\{4, 5\}$  | $\{0, 1\}$  | $v_1$       | 1            | 8             | 21            |
| 3     | 1               | $\{5, 6\}$  | $\emptyset$ | $\emptyset$ | 0            | 0             | 25            |
| 4     | 2               | $\emptyset$ | $\{2\}$     | $v_2$       | 2            | 23            | 2             |
| 5     | 1               | $\{7\}$     | $\{2, 3\}$  | $v_2$       | 2            | 15            | 14            |
| 6     | 5               | $\{7, 8\}$  | $\{3\}$     | $v_3$       | 1            | 2             | 22            |
| 7     | 1               | $\{9\}$     | $\{5, 6\}$  | $v_5$       | 3            | 18            | 11            |
| 8     | 3               | $\{9\}$     | $\{6\}$     | $v_6$       | 2            | 10            | 10            |
| 9     | 5               | $\emptyset$ | $\{7, 8\}$  | $v_7$       | 4            | 24            | 5             |

Tabela 2.1: Atributos relativos à computação paralela representada pelo GAD da Figura 2.1, considerando o modelo de comunicação onde  $L = 1$ .

Seja um escalonamento parcial  $S'$  de  $G$  em  $P$  de tal forma que  $V'$  é o subconjunto de  $k$  elementos de  $V$  que contém as tarefas já escalonadas. Podem ser definidos mais alguns atributos relativos às tarefas da GAD e a esse escalonamento parcial  $S'$ . O conjunto das tarefas livres, denotado por  $V_R$ , é aquele que contém todas as tarefas



ainda não escalonadas cujos predecessores imediatos já foram escalonados, ou seja,

$$v \in V_R, \text{ sse } (v \notin V') \wedge (\forall u \in \text{pred}(v), u \in V') \quad (2.7)$$

O *tempo de início-mais-cedo* de uma dada tarefa  $v \in V_R$  no processador  $p_j \in P$ , aqui denotado por  $t_{imc}(v, p_j)$ , é o menor tempo em que a tarefa  $v$  pode ter início no processador  $p_j$  levando em consideração que todos os seus predecessores imediatos tenham completado a sua execução e que os respectivos dados necessários para a sua execução estejam disponíveis em  $p_j$ . Além disso, a tarefa só pode iniciar sua execução em  $p_j$  a partir do momento  $t_{free}(p_j)$  em que esse processador se torna ocioso. Assim sendo, o  $t_{imc}(v, p_j)$  pode ser definido como:

$$t_{imc}(v, p_j) = \max_{\forall u \in \text{pred}(v)} \{t_F(u, \text{proc}(u)) + \text{com}(u, v), t_{free}(p_j)\}, \quad (2.8)$$

onde  $\text{com}(u, v) = 0$ , se  $\text{proc}(u) = p_j$ , e  $\text{com}(u, v) = \omega(u, v) \times L$  caso contrário. O *tempo de fim-mais-cedo* de uma dada tarefa  $v \in V_R$  no processador  $p_j \in P$ , aqui denotado por  $t_{fmc}(v, p_j)$ , é calculado a partir do  $t_{imc}(v, p_j)$ :

$$t_{fmc}(v, p_j) = t_{imc}(v, p_j) + \epsilon(v) \times h(p_j) \quad (2.9)$$

O predecessor crítico de uma tarefa  $v \in V_R$  também pode ser definido para um escalonamento parcial considerando um dado processador  $p_j \in P$ . Neste caso é denotado por  $pcri(v, p_j)$  e calculado levando em consideração que todos os predecessores imediatos de  $v$  tenham completado a sua execução e que os respectivos dados necessários para a sua execução estejam disponíveis em  $p_j$ . A tarefa  $u \in \text{pred}(v)$  que maximiza a expressão  $\{\text{com}(u, v) + t_F(u, \text{proc}(u))\}$ , é  $pcri(v, p_j)$ .

## 2.6 Resumo

Neste capítulo foram apresentados o modelo da aplicação, o modelo arquitetural e o modelo de comunicação, e definido formalmente o escalonamento de tarefas.

---

Em seguida, foram apresentados alguns atributos úteis para algoritmos de escalonamento. No próximo capítulo será apresentada uma classificação de heurísticas de escalonamento disponíveis na literatura. Serão descritos os algoritmos que podem ser empregadas em ambientes heterogêneos e têm maior interesse para este trabalho.

# Capítulo 3

## Heurísticas de Escalonamento

Devido à sua grande importância para o bom desempenho das aplicações paralelas, o escalonamento de tarefas é objeto de inúmeros estudos e diversas técnicas bastante distintas [5, 8, 35] já foram apresentadas para a solução deste problema NP-completo. Essas técnicas diferem tanto em função das suposições em relação à estrutura das aplicações paralelas quanto em relação ao ambiente alvo. Neste capítulo é apresentada uma classificação das heurísticas de escalonamento e, em seguida, são descritos dois algoritmos que foram propostos na literatura para o problema do escalonamento em ambientes heterogêneos. Estes algoritmos foram escolhidos pois parecem ser os melhores até o presente momento.

### 3.1 Heurísticas de Escalonamento de Tarefas

De uma forma geral, o escalonamento de tarefas pode ser aplicado estaticamente ou dinamicamente. O escalonamento estático é aquele realizado completamente antes da execução da aplicação, considerando que as características relevantes da aplicação paralela são conhecidas antes de sua execução, ou, pelo menos, podem ser estimadas com certa precisão. O escalonamento dinâmico é aplicado quando apenas alguns pressupostos sobre a aplicação podem ser considerados antes do início da execução e as estimativas não podem ser muito exatas. Assim, o escalonamento é realizado

concomitantemente com a execução das tarefas da aplicação paralela [7, 38, 46]. O interesse deste trabalho concentra-se no escalonamento de tarefas estático e para entender os mecanismos fundamentais propostos na literatura para a implementação de heurísticas de construção, uma das várias classificações existentes que enquadram as diversas heurísticas [13, 30, 35, 42, 58] será discutida a seguir. Uma forma de classificar essas heurísticas é de acordo com os mecanismos adotados durante o processo de escalonamento. Segundo essa abordagem, as heurísticas podem ser divididas em duas grandes classes [58]: busca aleatória ou construção. Aquelas baseadas em busca aleatória são chamadas de meta-heurísticas, como, por exemplo, Algoritmos Genéticos [17], *Simulated Annealing* [53] e Busca Tabu [49]. Este grupo não será objeto de estudo neste trabalho, pois o algoritmo proposto não pertence a esta categoria. Por sua vez, as heurísticas de construção, onde uma heurística define somente uma solução do problema, podem ser divididas em quatro grupos: heurísticas de *list scheduling* [36, 44, 28, 54, 58, 60], heurísticas de aglomeração [14, 42, 47, 59], heurísticas de análise do caminho crítico [32, 33] e heurísticas de particionamento de grafos [30, 40]. As heurísticas de aglomeração e *list scheduling* são mais importantes para este trabalho, uma vez que os algoritmos aqui utilizados para escalonamento em dois estágios pertencem a essas classes.

**Algoritmo 1 : *List Scheduling***

```

1  Ordena(TaskList) ;
2  Enquanto (TaskList  $\neq \emptyset$ ) faça {
3    Seleciona próxima tarefa  $v_i$  em TaskList;
4    Seleciona processador destino  $p_j$  em ProcList;
5    Aloca  $v_i$  em  $p_j$ ;
6     $TaskList = TaskList - v_i$ ;
   }
```

Nas heurísticas do tipo ***list scheduling***, cujo *framework* é apresentado no Algoritmo 1, os nós do GAD a ser escalonado são dispostos em uma lista ordenada de acordo com uma dada prioridade (linha 1). A cada passo do algoritmo, a tarefa livre (aquela cujos predecessores já foram escalonados) de maior prioridade é selecionada (linha 3). A seguir, é selecionado o processador no qual essa tarefa será executada,

obedecendo uma função de custo pré-definida (linha 4). A tarefa selecionada é alocada no processador escolhido (linha 5) e o algoritmo prossegue até que todas as tarefas sejam escalonadas.

Algoritmos deste tipo têm geralmente baixa complexidade, são apropriados para um ambiente com número limitado de processadores e podem ser facilmente adaptados a ambientes heterogêneos. O *Heterogeneous-Earliest-Finish-Time* (HEFT) [58], o *Dynamic Level Scheduling* (DLS) [54] e o *Best Imaginary Level* (BIL) [44], são alguns exemplos deste grupo.

Nas heurísticas baseadas em **aglomeração de tarefas**, são criadas coleções de tarefas de tal forma que o tempo de execução paralelo da aplicação seja minimizado, através da redução do custo de comunicação e, em alguns casos, do tempo total de computação de cada coleção. A cada iteração uma tarefa pode ser adicionada à uma coleção, que normalmente contém predecessores da respectiva tarefa. Algoritmos deste tipo tendem a ter maior complexidade e pressupõem, inicialmente, um número infinito de processadores homogêneos. Por esse motivo, geralmente, fazem parte dessa classe de heurísticas estratégias de dois estágios [31, 45, 52], nas quais, durante o primeiro estágio, as coleções de tarefas são formadas e cada uma alocada a um processador virtual. A seguir, em um segundo estágio, as coleções resultantes são mapeadas nos processadores reais disponíveis. O *Dominant Sequence Clustering* (DSC) [59], o PLW [47], o HAL [42] e o HTschedule [14] são exemplos deste grupo.

As heurísticas baseadas na análise do caminho crítico tentam diminuir o caminho mais longo do grafo alocando em um mesmo processador as tarefas que fazem parte do caminho crítico. Com isso o custo de comunicação entre essas tarefas é eliminado. O *Linear Clustering* (LC) [32] e o *Dynamical Critical-Path* (DCP) [33] são exemplos deste grupo, que, da mesma forma que os algoritmos de aglomeração de tarefas, são geralmente empregadas para um número infinito de processadores homogêneos.

Finalmente, as heurísticas de particionamento de grafos decompõem o grafo da aplicação em subconjuntos, denominados clãs [21], de acordo com sua topologia. Um clã é composto por nós ou clãs que guardam a mesma relação com nós ou clãs que não fazem parte de seu subconjunto. Clãs independentes são compostos por nós ou

clãs que podem ser executados paralelamente, enquanto clãs lineares são compostos por nós ou clãs que guardam relação de precedência e devem ser executados em determinada ordem. Para tentar balancear o custo de comunicação e o grau de paralelismo, com a finalidade de obter um tempo de execução vantajoso, após a decomposição do grafo, os algoritmos desta classe levam em consideração os custos de computação e comunicação relativos aos nós e arestas do grafo para decidir pela aglomeração de clãs em um mesmo processador ou sua paralelização, isto é, alocação em processadores distintos [30, 40].

## 3.2 Replicação de Tarefas

Para minimizar o custo de comunicação entre processadores em um escalonamento, algumas heurísticas fazem uso da técnica de replicação de tarefas [1, 48]. Neste caso, cópias de uma mesma tarefa podem ser alocadas em diferentes processadores juntamente com sucessores imediatos, reduzindo o custo de comunicação associado à troca de mensagens entre processadores distintos. A aplicação dessa técnica é particularmente vantajosa em cenários com elevado custo de comunicação, e quando as tarefas da aplicação têm um grande número de sucessores imediatos [8]. A Figura 3.1 (a) mostra um exemplo de GAD, na qual os números ao lado dos vértices representam os custos de computação e os números ao lado dos arcos representam o custo de comunicação. O escalonamento ótimo deste GAD em dois processadores homogêneos (Figura 3.1 (b)) pode ser obtido apenas com o emprego da replicação da tarefa  $v_0$ .

O critério de seleção das tarefas a serem replicadas em um escalonamento varia de acordo com cada algoritmo. O PLW [47], HAL [42] e HTschedule [14] são exemplos de algoritmos de aglomeração de tarefas que fazem uso desta técnica. O *Levelized Duplication Based Scheduling* (LDBS) [20] é um exemplo de algoritmo do tipo *list scheduling* com replicação.

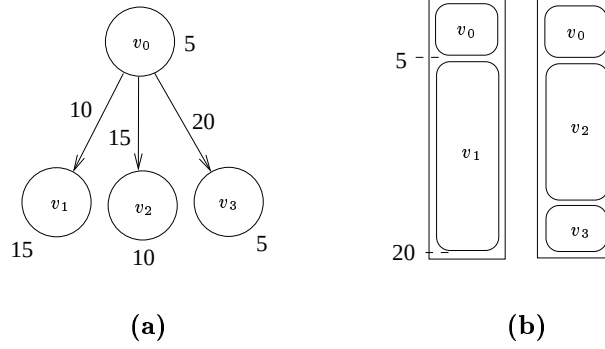


Figura 3.1: (a) GAD de uma aplicação paralela com 4 tarefas e (b) o respectivo escalonamento em 2 processadores homogêneos com replicação de tarefas.

### 3.3 Escalonamento em Ambientes Heterogêneos

Desde o advento da Computação em Cluster [12], vários algoritmos de escalonamento de aplicações para ambientes heterogêneos têm sido propostos. Alguns foram elaborados visando o mesmo modelo de arquitetura adotado neste trabalho (Seção 2.2), como em [14], e outros aplicando modelos mais genéricos, com variação da velocidade de comunicação entre processadores ou topologia da rede [2, 20, 54, 57, 58]. Dentre diversas estratégias pertencentes a esses dois grupos de heurísticas para escalonamento em ambientes heterogêneos, aquelas que apresentam resultados de boa qualidade foram estudadas para que se pudesse definir um ambiente composto de heurísticas representativas para análise experimental e poder, assim, validar a proposta deste trabalho.

O *Dynamic Level Scheduling* (DLS), proposto em [54], trata-se de um algoritmo de *list scheduling* em que a prioridade na seleção de tarefas prontas para escalonamento é baseada em um *nível dinâmico*. Sendo uma prioridade dinâmica, o nível dinâmico é recalculado, a cada iteração, para cada tarefa livre em cada processador disponível no ambiente. Esse índice leva em consideração o tempo de execução de uma tarefa  $v$  em um dado processador  $p$ , o custo de computação do caminho crítico associado à  $v$ , e o custo da alocação de uma tarefa sucessora de  $v$  com maior custo de comunicação em um processador  $p' \neq p$ . Em [44], foi apresentado o *Best Imaginary Level* (BIL), um algoritmo bastante semelhante ao DLS, pois também

se baseia em um nível dinâmico, o chamado *melhor nível imaginário*. A principal diferença é que no cálculo do melhor nível imaginário, todos os sucessores de uma dada tarefa são levados em consideração, para analisar a influência no escalonamento desta tarefa ser executada em um processador, e seus sucessores em processadores diferentes deste. Segundo os resultados apresentados em [44], registrados a partir de 100 experimentos, BIL obteve um ganho de 17% em relação ao *makespan* quando comparado ao DLS.

Os algoritmos *Heterogeneous-Earliest-Finish-Time* (HEFT) e *Critical-Path-On-Processor* (CPOP) foram propostos em [58] e seguem os mecanismos do *framework* de *list scheduling*. O HEFT, que será detalhado adiante, utiliza o *blevel* como prioridade de seleção da tarefa para o escalonamento, que é alocada no processador que minimiza o tempo de fim da tarefa. No CPOP, a seleção de tarefas para escalonamento ocorre na ordem não crescente da soma do *blevel* e *tlevel* de cada tarefa (que nada mais é que o custo do caminho crítico associado à tarefa). Além disso, as tarefas pertencentes ao caminho crítico do grafo são alocadas obrigatoriamente no processador que minimiza seu tempo de execução. Em [58] o desempenho do HEFT foi comparado a três outros algoritmos bem conhecidos, DLS [54], MH [22] e LMT [29], além de CPOP (*Critical-Path-On-a-Processor*). O algoritmo *Mapping Heuristics* (MH) é um *list scheduling* bastante semelhante ao HEFT, mas a prioridade de escolha de tarefas é dada pelo *blevel* calculado sem levar em consideração o custo de comunicação. O *Levelized Min-Time* (LMT) também é um algoritmo do tipo *list scheduling*, que primeiramente agrupa as tarefas por nível topológico (calculado conforme a Equação 2.3), e depois, a partir do nível 0, aloca cada tarefa, em ordem não crescente de peso de computação, no processador que executa a tarefa mais rapidamente. O HEFT teve desempenho melhor que todos os outros em 86% dos casos.

Em [11] foi apresentado o algoritmo *Iso-Level Heterogeneous Allocation* (ILHA), que durante o escalonamento procura realizar um balanceamento de carga entre todos os processadores disponíveis, dando maior prioridade à alocação de tarefas adjacentes no mesmo processador do que à minimização do tempo de fim das tarefas. Assim, os objetivos principais são reduzir a troca de mensagens entre processadores



e distribuir carga computacional equilibrada entre os processadores. Nos testes conduzidos em [11], o ILHA foi comparado com HEFT, CPOP, DLS, BIL e PCT. O algoritmo *Minimum Partial Completion Time Static Priority* (PCT) [38] é um *list scheduling* bastante semelhante ao HEFT, mas sem política de inserção de tarefas em espaços vagos no processador. Os melhores resultados foram obtidos por ILHA e HEFT. Nota-se, entretanto, que ILHA só é adequado a ambientes com um número de processadores bem menor que o número de tarefas, fugindo do interesse deste trabalho.

Em [20] foi apresentado o algoritmo *Levelized Duplication Based Scheduling* (LDBS), também do tipo *list scheduling*, mas que faz uso da replicação de tarefas. Foi demonstrado, através de diversas simulações, que embora para grafos de granularidade grossa o HEFT produza escalonamentos em média 2% menores que o LDBS, para grafos de granularidade unitária o LDBS é 19% melhor que o HEFT, e 30% no caso de granularidade fina.

O *Triplet* [15] é um dos poucos algoritmos de aglomeração de tarefas desenvolvidos para o escalonamento de tarefas em ambientes completamente heterogêneos. Ele propõe não apenas a reunião de tarefas em coleções para diminuir a comunicação, mas que as máquinas disponíveis em um dado ambiente heterogêneo também sejam agrupadas de acordo com características comuns de velocidade de processamento e comunicação. Posteriormente, coleções de tarefas são designadas a grupos de máquinas similares, que formam conjuntos aproximadamente homogêneos, levando-se em consideração a distribuição de carga. De acordo com os resultados experimentais descritos em [15], o desempenho do algoritmo *Triplet* demonstrou-se superior ao HEFT.

O *HTschedule* [14] é outro algoritmo de aglomeração de tarefas para ambientes heterogêneos. Ele foi desenvolvido adotando-se o mesmo modelo de ambiente deste trabalho e conta com o recurso de replicação de tarefas. Em uma primeira fase são formadas coleções  $c(v)$  para cada tarefa  $v \in V$ . A coleção correspondente a uma dada tarefa  $v$  é igual a coleção de sua tarefa predecessora crítica  $cri(v)$  acrescida da própria tarefa, ou seja,  $c(v) = c(cri(v)) \cup \{v\}$ . Ou seja, as coleções correspondem

a cada caminho crítico do GAD com destino em  $v$ . Em uma segunda fase, após a construção de todas as coleções, o algoritmo elimina aquelas que estão contidas em outras maiores, ou seja, são redundantes. Em seguida, monta uma lista de coleções em ordem não crescente do maior tempo de recebimento das mensagens enviadas pela tarefa dona da coleção, começando pela coleção que contém a tarefa destino do grafo. Uma lista com os processadores em ordem decrescente de velocidade também é criada. No mapeamento cada coleção é designada ao processador mais rápido e assim sucessivamente, até que todas as coleções tenham sido mapeadas. Observa-se como uma clara desvantagem do *HTschedule* que este exige um número de processadores que deve ser maior ou igual ao número de coleções produzidas. Em [14], o *HTschedule* foi comparado ao *STDS* [51] para um único caso, obtendo um resultado 7,3% melhor. Pela sua simplicidade e especificação de uma rede de processadores heterogêneos maior que o número de coleções, essa estratégia não foi implementada neste trabalho.

Nos resultados observados em [11] e [58], se conclui que o algoritmo *HEFT* produz, em geral, melhores resultados, destacando-se como o algoritmo do tipo *list scheduling* sem replicação mais adequado para o cálculo do escalonamento de tarefas no ambiente heterogêneos. Em razão disso, ele foi selecionado para fazer parte do ambiente de comparação proposto neste trabalho. Por esse motivo, maiores detalhes sobre esta estratégia serão apresentados a seguir. Ainda neste ambiente, devido ao seu excelente desempenho perante o *HEFT*, e como o algoritmo em dois estágios proposto neste trabalho também faz uso da técnica de replicação de tarefas, o *LDBS* também foi selecionado para comparação experimental e também será detalhado a seguir.

### 3.3.1 HEFT

O *Heterogeneous-Earliest-Finish-Time* (*HEFT*) [58] é um algoritmo de escalonamento do tipo *list scheduling* para ambientes heterogêneos com um número limitado de processadores. Antes de iniciar o processo iterativo de escalonamento propriamente dito, o *HEFT* cria uma lista ordenada de tarefas em ordem não crescente

de *blevel* (Algoritmo 2, linha 1). O *blevel* de cada tarefa é calculado de forma semelhante à Equação 2.5, mas considerando o custo médio de execução da tarefa nos diversos processadores disponíveis no ambiente. Para o ambiente descrito na Seção 2.2, o cálculo do *blevel* tem a seguinte formulação:

$$blevel(v) = \max_{\forall w \in succ(v)} \{blevel(w) + \omega(v, w) \times L\} + \epsilon(v) \times \bar{h}, \quad (3.1)$$

onde  $blevel(v) = \epsilon(v) \times \bar{h}$ , se  $succ(v) = \emptyset$ , isto é,  $v$  é um nó de saída, e  $\bar{h}$  é a média dos fatores de heterogeneidade dos processadores disponíveis no ambiente. A média aqui se faz necessária devido à heterogeneidade dos processadores. Como o processo de escalonamento ainda se realizará, o custo de computação exato ainda não pode ser calculado. A seguir, em cada iteração da fase de escalonamento, o algoritmo seleciona a primeira tarefa da lista ordenada ainda não escalonada, ou seja, aquela de maior prioridade (linha 3), e verifica em que processador essa tarefa termina de executar mais cedo.

**Algoritmo 2 : HEFT**

```

1  Ready = Ordena(V);
2  enquanto (Ready ≠ ∅) faça {
3    v = primeiro(Ready);
4    t* = ∞;
5    ∀pj ∈ P faça {
6      determina timc(v, pj);
7      tfmc(v, pj) = timc(v, pj) + ε(v) × h(pj);
8      se (tfmc(v, pj) < t*) então {
9        t* = tfmc(v, pj);
10       p* = pj;
11     }
12   }
13   S = S ∪ {(v, p*, t*)};
14   Ready = Ready − {v};
15 }
```

Para isso, o algoritmo determina o melhor tempo possível para o início de  $v$  em  $p_j$ ,  $t_{imc}(v, p_j)$  (linha 6), adotando uma política de inserção em espaços ociosos nos

processadores, que reduz o *makespan* final do escalonamento. Uma dada tarefa  $v$  só pode ser executada em um dado processador  $p_j$ , a partir do instante em que recebe todas as mensagens de seus predecessores  $u \in \text{pred}(v)$ . O algoritmo verifica se é possível escalonar, a partir deste instante, a tarefa  $v$  em algum intervalo ocioso entre tarefas já escalonadas em  $p_j$ , decorrente da alocação de tarefas adjacentes em processadores distintos. Para que isso ocorra, o intervalo ocioso deve ter tamanho suficiente para a execução de  $v$  em  $p_j$ . Caso não seja possível, a tarefa  $v$  será escalonada após a execução da última tarefa já alocada naquele processador. O tempo de fim mais cedo  $t_{fmc}(v, p_i)$  de  $v$  em  $p_j$  é assim calculado (linha 7). Se este é o melhor resultado até o momento, é armazenado em  $t^*$ , e o processador correspondente em  $p^*$  (linhas 9 e 10). Depois que todos os processadores são levados em consideração,  $v$  é alocada naquele que fornece o melhor resultado (linha 11). A tarefa escalonada é retirada da lista (linha 12) e o algoritmo prossegue até que todas as tarefas tenham sido alocadas.

O HEFT possui a vantagem de ser de simples implementação e baixa complexidade, quando comparados com algoritmos que fazem uso do recurso de replicação. Tal fato permite que escalonamentos de aplicações com um número grande de tarefas sejam calculados com bastante rapidez. Além disso apresenta bons resultados no escalonamento de aplicações de granularidade grossa, isto é, que têm baixo custo de comunicação, quando comparada ao tempo de computação.

### 3.3.2 LDBS

O algoritmo *Levelized Duplication Based Scheduling* (LDBS) [20] é uma heurística que pertence à classe *list scheduling*, mas, diferentemente do HEFT, faz uso de replicação de tarefas. Como o HEFT, o LDBS considera ambientes heterogêneos, ou seja, tanto os processadores têm características distintas, como também os canais de comunicação. Inicialmente, o algoritmo classifica as tarefas de acordo com seu nível topológico no GAD (Algoritmo 3, linha 1), calculado segundo a Equação 2.3. As tarefas de mesmo nível topológico  $l$  são ordenadas de forma não crescente por *blevel* (conforme Equação 3.1) na lista  $TList_l$ . Seja  $t(v, p_j)$  o tempo de execução da tarefa

$v$  no processador  $p_j$ . Para cada uma das tarefas  $v \in TList_0$  (tarefas origem) e cada um dos processadores  $p_j$  disponíveis no ambiente,  $t(v, p_j)$  é calculado e o par  $(v^*, p^*)$  que minimiza esse tempo é determinado. A tarefa  $v^*$  é então escalonada em  $p^*$ , retirada da lista inicial, e as tarefas restantes serão mais uma vez testadas em todos os processadores. Este procedimento é repetido até que todas as tarefas de  $TList_0$  tenham sido escalonadas (linhas 4 a 11). Seja  $D$  o maior nível topológico do grafo de entrada  $V$ . Após o escalonamento das tarefas de nível topológico 0, o algoritmo prossegue escalonando grupos de tarefas do nível 1 até o nível  $D$  (linha 12).

**Algoritmo 3 : *LDBS***

```

1   $\forall v \in V$ , calcula  $level(v)$ ;
2   $D = \max_{v \in V} \{level(v)\}$ ;
3  determina  $TList_l$ , para  $l = 0, 1, \dots, D - 1$ ;
4  enquanto  $(TList_0 \neq \emptyset)$  faça {
5       $t^* = \infty$ ;
6       $\forall v \in TList_0$  e  $\forall p_j \in P$  faça {
7          calcula  $t(v, p_j)$ ;
8          Se  $(t(v, p_j) < t^*)$  então {
9               $t^* = t(v, p_j)$ ;  $v^* = v$ ;  $p^* = p_j$ ;
10             }
11         }
12      $S = S \cup \{(v^*, p^*, t^*)\}$ ;
13      $TList_0 = TList_0 - \{v^*\}$ ;
14 }
15  $\forall l = 1$  até  $D - 1$  Escalona  $(TList_l)$ ;
```

Neste processo, descrito mais detalhadamente no Algoritmo 4, o algoritmo faz uso do recurso de replicação de tarefas. Seja  $TList_l$  a lista ordenada contendo todas as tarefas de nível  $l$ . No momento em que o procedimento  $Escalona(TList_l)$  é chamado, essas tarefas estão livres para escalonamento, pois o conjunto  $TList_{l-1}$  foi escalonado na iteração anterior. Para cada tarefa  $v \in TList_l$  e cada processador  $p_j \in P$  (linha 3), o algoritmo determina o tempo de início mais cedo da tarefa  $v$  (linha 4), fazendo uso da política de inserção em espaços ociosos do processador. Além disso, verifica se a replicação de  $u$ , o predecessor crítico de  $v$ , em algum processador  $p_k \in P$ , permite que  $v$  inicie sua execução mais cedo.

**Algoritmo 4 :** *Escalona*( $TList_l$ )

```

1  enquanto ( $TList_l \neq \emptyset$ ) faça {
2       $t^* = \infty$ ;
3       $\forall v \in TList_l$  e  $\forall p_j \in P$  faça {
4          determina  $t_{imc}(v, p_j)$ ;
5           $t_{fmc}(v, p_j) = t_{imc}(v, p_j) + t(v, p_j)$ ;
6          se ( $t_{fmc}(v, p_j) < t^*$ ) então {
7               $t^* = t_{fmc}(v, p_j)$ ;
8               $v^* = v$ ;  $p^* = p_j$ ;
9               $u^* = \emptyset$ ;  $p_R^* = \emptyset$ ;
10             }
11          $u = cri(v)$ ;
12          $\forall p_k \in P$  faça {
13              $t_{imc_R} = AnalisaReplica(u, p_k, v, p_j, t_{imc}(v, p_j))$ ;
14             se ( $(t_{imc_R} + t(v, p_j)) < t^*$ ) então {
15                  $t^* = t_{imc_R} + t(v, p_j)$ ;
16                  $v^* = v$ ;  $p^* = p_j$ ;
17                  $u^* = u$ ;  $p_R^* = p_k$ ;
18             }
19         }
20     }
21      $S = S \cup \{ \langle v^*, p^*, t^* \rangle \}$ ;
22      $Replica(u^*, p_R^*, v^*, p^*, \infty)$ ;
23      $TList_l = TList_l - \{v^*\}$ ;
24 }
```

O procedimento *AnalisaReplica* (linha 12), responsável por esta operação, pode ser chamado para os outros predecessores imediatos de  $v$  que também se tornem críticos. Se houver replicação, a tupla  $\langle v, p_j, u, p_k \rangle$ , onde  $u$  é o predecessor crítico de  $v$  a ser replicado em  $p_k$ , tal que o tempo de fim de  $v$  seja minimizado, é definida nesta iteração. Note que o grau de replicação de tarefas é restrito aos predecessores imediatos de  $v$ . Ao final da análise em todos os processadores, a tarefa  $v^*$  é realmente escalonada em  $p^*$  (linha 17) e ainda, caso adequado, cópias de predecessores críticos  $u^*$  são realmente escalonados em  $p_k^*$  pelo procedimento *Replica* (linha 18). Este procedimento é repetido até que todas as tarefas de  $TList_l$  tenham sido escalonadas (linha 1).

Em [20] é apresentada também uma segunda versão do algoritmo LDBS. Na nova versão, a ordenação das tarefas em  $TList$  (Algoritmo 3, linha 1) ainda é em ordem não decrescente de nível topológico, mas as tarefas de um mesmo nível topológico são colocadas em ordem não crescente de  $blevel$ . Além disso, há uma simplificação durante o escalonamento das tarefas em cada nível (Algoritmo 4). Na primeira versão, para a escolha de uma tupla  $\langle v^*, p^*, u^*, p_R^* \rangle$  todas as tarefas livres  $v \in TList_l$  são testadas em todos os processadores  $p_j \in P$ . Na segunda versão, as tarefas  $v \in TList_l$  são escalonadas na ordem, ou seja, primeiramente são escalonadas as tarefas de maior  $blevel$ . O objetivo disso é evitar que tarefas que possivelmente têm menor importância para o escalonamento sejam escalonadas antes de outras mais críticas, e assim diminuir a complexidade do algoritmo.

Os resultados experimentais obtidos em [20] demonstraram, que embora para grafos de granularidade grossa ( $CCR = 0, 1$ ) o HEFT produza escalonamentos em média 2% menores que o LDBS, para grafos de granularidade unitária ( $CCR = 1, 0$ ) o LDBS é 19% melhor que o HEFT, e 30% no caso de granularidade fina ( $CCR = 10, 0$ ).

## 3.4 Resumo

Neste capítulo foram apresentadas a classificação das principais heurísticas de escalonamento disponíveis na literatura e maiores detalhes sobre as heurísticas que podem ser empregadas em ambientes heterogêneos e, portanto, são de maior interesse para este trabalho. No próximo capítulo serão apresentados o escalonamento em dois estágios, as principais características do primeiro estágio do escalonamento, o algoritmo HAL, e do segundo estágio, o algoritmo de mapeamento proposto.

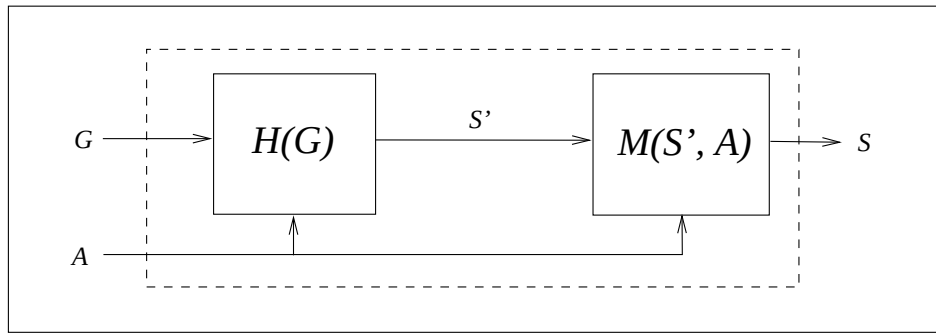
# Capítulo 4

## Estratégia de Escalonamento em Dois Estágios

Algoritmos de escalonamento do tipo aglomeração de tarefas são geralmente desenvolvidos para processadores homogêneos e são tradicionalmente divididos em dois estágios [31, 32, 45]. Durante o primeiro estágio, são criadas coleções de tarefas que visam diminuir o custo de comunicação, levando em consideração um número ilimitado de processadores (virtuais). Em um segundo estágio, é realizado o mapeamento dessas coleções nos processadores reais. A suposição de um número ilimitado de processadores, no primeiro estágio, tem a finalidade de analisar os benefícios de zerar os arcos, ou seja, aglomerar tarefas independentes, face ao custo de comunicação decorrente da alocação de tarefas adjacentes em processadores distintos. O esquema realizado no primeiro estágio tem assim o objetivo de minimizar ao máximo o tempo de fim das tarefas, sem que um número limitado de processadores interfira nesta procura.

Ao tratarmos do problema de escalonamento para processadores heterogêneos, usar estratégias de aglomeração como aquelas da metodologia apresentada em [42] já considerando a diversidade/diferenças entre processadores, torna-se quase impraticável, devido à dificuldade de calcular o tempo mais cedo de fim em um processador supondo que já possam existir tarefas alocadas a esse processador. Assim sendo, ao desejar investir em algoritmos de aglomeração para processadores heterogêneos,





**Figura 4.1: Diagrama de blocos do escalonamento de tarefas para ambientes heterogêneos em dois estágios.**

esse trabalho almeja investigar o uso de algoritmos de aglomeração considerando processadores virtuais homogêneos no primeiro estágio para tornar a granularidade da aplicação mais grossa e tentar equiparar com o custo de comunicação médio da rede.

A Figura 4.1 mostra um diagrama que representa o escalonamento em dois estágios. O primeiro estágio é uma heurística de escalonamento  $H(G)$  que realiza a aglomeração do GAD  $G$  considerando um ambiente virtual constituído por um número ilimitado de processadores homogêneos. Esse resultado constitui-se em um escalonamento intermediário  $S'$  que serve de entrada para o segundo estágio. Este, por sua vez, implementa uma heurística de mapeamento  $M(S', A)$ , que recebe informações mais precisas sobre o ambiente alvo  $A$ , que pode abranger processadores heterogêneos e em quantidade inferior ao total de coleções produzidas no primeiro estágio.

O primeiro estágio é responsável pela aglomeração de tarefas em coleções que correspondem a processadores virtuais. Seja  $A$  o ambiente alvo do escalonamento que compreende um conjunto limitado de  $n_p$  processadores  $P = \{p_0, p_1, \dots, p_{n_p-1}\}$  heterogêneos, totalmente conectados entre si por uma rede de canais de latência  $L$ . Considera-se o ambiente virtual  $A'$  composto por um conjunto  $P'$  de processadores virtuais homogêneos, com velocidade de processamento igual a do processador  $p \in P$  de menor velocidade, totalmente conectados entre si por uma rede de canais de latência  $L$ . O número de processadores  $p' \in P'$  é considerado ilimitado de tal forma

que há tantos processadores virtuais quantos necessários para conter cada uma das coleções de tarefas formadas no primeiro estágio.

O escalonamento  $S'$  resultante do primeiro estágio poderia ser definido como um conjunto finito de uplas  $(v, p'_j, t'_l)$ , cada uma especificando que uma tarefa  $v \in V$  será executada no processador virtual  $p'_j \in P'$  iniciando no tempo  $t'_l$ . Cada tarefa  $v \in V$  deve ser escalonada em pelo menos um processador virtual  $p'_j \in P'$ , e ela só pode ser executada depois que todos os seus predecessores imediatos tenham completado a sua execução e que os respectivos dados estejam disponíveis em  $p'_j$ . Denota-se o tempo de início escalonado de uma tarefa  $v$  em um processador  $p'_j$  por  $t'_I(v, p'_j)$ . Consequentemente, o tempo de fim escalonado da tarefa  $v$  em  $p'_j$  é igual a  $t'_I(v, p'_j) + \epsilon(v) \times h(p'_j)$ . O tempo total de execução da aplicação paralela  $G$  no ambiente  $P'$  de acordo com o escalonamento intermediário  $S'$ , ou *makespan* do escalonamento, é denotado por  $mspan(S')$  e pode ser definido como  $mspan(S') = \max_{v \in V, \forall p'_j \in P'} t'_I(v, p'_j) + \epsilon(v) \times h(p'_j)$ . O conjunto de tarefas definido pelo primeiro estágio para a execução em determinado processador virtual  $p'_j$ , constitui uma coleção de tarefas  $C(v)$ , onde  $v$ , a tarefa dona da coleção, é tal que  $\forall (u, p'_j, t'_k) \in S'$  e  $u \neq v$  temos  $t'_I(v, p'_j) \geq t'_I(u, p'_j) + \epsilon(u) \times h(p'_j)$ .

A função de mapeamento  $M(S', A)$  que implementa o segundo estágio completa o escalonamento associando cada coleção  $C(v)$ , formada na primeira fase, a um processador real  $p_j \in P$  em que as tarefas que compõem esta coleção deverão ser executadas. Ou seja, o mapeamento redefine o conjunto finito de tuplas  $(v, p_j, t_l)$ , cada uma especificando que uma tarefa  $v \in V$  será executada no processador real  $p_j \in P$  a partir do tempo  $t_l$ .

Este trabalho propõe a utilização de dois algoritmos em sequência, constituindo uma estratégia de aglomeração em dois estágios para processadores heterogêneos chamada **Clusterização para Processadores Heterogêneos (CPH)**. O trabalho proposto em [42] foi selecionado como primeiro estágio, a fim de tirar proveito dos excelentes resultados que este algoritmo tem demonstrado [6, 8, 42]. Para o segundo estágio é proposto um **Algoritmo de Mapeamento de Coleções (AMC)** do tipo *list scheduling* capaz de atribuir as coleções produzidas no primeiro estágio

a processadores heterogêneos disponíveis em um dado ambiente.

## 4.1 Primeiro Estágio: HAL

Neste trabalho, a Heurística de Aglomeração para LogP (HAL) [42] foi selecionada para ser utilizada no primeiro estágio de CPH. Trata-se de um algoritmo de escalonamento baseado na metodologia de projeto de algoritmos de aglomeração proposta em [8] que exploram a técnica de replicação de tarefas. A solução obtida por HAL para diversos grafos de entrada é bastante vantajosa quando comparada à solução de outros algoritmos de escalonamento [6, 8, 42]. Seus resultados mostraram sua eficiência não somente em aplicações de granularidade grossa, mas principalmente em aplicações de granularidade fina.

Num primeiro estágio, algoritmos desta classe procuram achar um escalonamento ótimo para um GAD construindo uma coleção  $C(v) = \langle u_1, u_2, \dots, v \rangle$  para cada tarefa  $v \in V$ , onde  $v$  é dita dona da coleção e as tarefas  $u_i$  são predecessoras de  $v$  incluídas na coleção de tal forma que  $v$  seja executada no tempo mais cedo possível  $s(v)$ . Ao final, são identificadas as coleções realmente necessárias para a execução de todas as tarefas da aplicação.

HAL constrói as coleções  $C(v)$ , de acordo com a metodologia proposta em [8]. Em linhas gerais, inicialmente cada coleção  $C(v)$  contém apenas a tarefa  $v \in V$  dona da coleção. Em cada iteração, a cópia de uma tarefa predecessora  $w$ , tal que  $w \notin C(v)$ , pode ser acrescentada à coleção, com o objetivo de minimizar o tempo de início  $s(v)$  da dona da coleção. A tarefa  $w$  é efetivamente incluída em  $C(v)$  se a nova coleção satisfaz à condição de inclusão. O processo continua até que a condição de parada seja atingida.

Seja  $C(v) = \langle u_1, u_2, \dots, u_j, \dots, u_k, v \rangle$  uma coleção de tarefas aglomeradas em uma dada iteração do algoritmo. Uma tarefa  $w_i$  é uma possível candidata a ser incluída na coleção  $C(v)$  sempre que existir um arco  $(w_i, u_j)$  incidente à  $C(v)$ , onde  $w_i \notin C(v)$  e  $u_j \in C(v)$ . O conjunto de tarefas  $w_i$  incidentes a uma coleção  $C(v)$  é denotado

por  $iancs(C(v))$ , e pode ser definido como:

$$iancs(C(v)) = \{w_i | (\exists (w_i, u_j) \in E) \wedge (w_i \notin C(v)) \wedge (u_j \in C(v))\}. \quad (4.1)$$

Cada tarefa  $w_i$  incidente a  $C(v)$ , está associada a um *caminho-predecessor*, ou seja, o caminho percorrido do nó  $w_i$  (ainda fora da coleção) até  $v$ , e que contém o arco  $(w_i, u_j)$ . De acordo com o modelo de latência, o custo deste *caminho-predecessor*, denotado por  $c(w_i, u_j, v)$ , leva em consideração  $s(w_i)$  (o tempo mais cedo de início de  $w_i$ ), o custo de  $w_i$ , o custo de comunicação entre  $w_i$  e  $u_j$ , e o custo das tarefas da coleção no caminho entre  $u_j$  e  $v$ , denotado por  $ccam(u_j, v)$ :

$$c(w_i, u_j, v) = s(w_i) + \epsilon(w_i) + (\tau \times \omega(w_i, u_j)) + ccam(u_j, v). \quad (4.2)$$

O *caminho-predecessor-crítico* é aquele que tem o maior custo  $c(w_i, u_j, v)$ , denotado por  $maxc$ . Este valor indica um limite para o tempo mais cedo  $s(v)$  em que a tarefa  $v$  poderá começar a executar, considerando todos os arcos incidentes à coleção  $C(v)$ , ou seja, as dependências de dados da coleção  $C(v)$ .

O *caminho-coleção* compreende todas as tarefas predecessoras de  $v$  já incluídas na coleção  $C(v)$ . Um custo  $m(C(v))$  está associado ao *caminho-coleção*, e representa o limite inferior para o tempo  $s(v)$  em que a tarefa  $v$  dona da coleção pode ter início, pois equivale ao tempo que precisa ser gasto pelo processador na execução de todas essas tarefas, sem contabilizar espaços ociosos devido à comunicação. Desta forma, o tempo mais cedo possível  $s(v)$  em que a tarefa  $v$  poderá ter início será o valor máximo entre  $maxc$ , o custo do *caminho-predecessor-crítico*, e  $m(C(v))$ , o custo *caminho-coleção*.

O Algoritmo 5 mostra, em maiores detalhes, o funcionamento da fase de construção da coleção  $C(v)$ , para todo  $v \in V$ . Em uma fase posterior, HAL seleciona somente as coleções  $C(v)$  necessárias para a execução de  $G$ . Inicialmente, cada coleção é constituída pela tarefa dona da coleção  $v$  (linha 1). O valor  $s(v)$  é zero se  $v$  é uma tarefa origem (linha 3). Caso a tarefa  $v$  não seja origem,  $s(v)$  é determinado depois que é formada a coleção  $C(v)$ , de acordo com os valores  $s(u_i)$  associados às

coleções  $C(u_i)$ ,  $\forall u_i \in \text{pred}(v)$ . A fim de reduzir o valor do tempo de início  $s(v)$  da tarefa dona da coleção, a tarefa  $w$  que determina o *caminho-predecessor-crítico* é selecionada como candidata à inclusão em  $C(v)$  (linha 5). Em caso de empate, seleciona-se a tarefa de menor *b-level*, desta forma prioriza-se a tarefa mais próxima da dona da coleção.

**Algoritmo 5 :** *Construção da Coleção(v)*

```

1   $C(v) = \{v\};$ 
2  se  $\text{pred}(v) = \emptyset$  então {
3     $s(v) = 0;$ 
4  }
5  senão {
6    seleciona  $w \in \text{iancs}(C(v));$ 
7     $\text{maxc} = c(w, u_j, v);$ 
8     $m(C(v)) = 0;$   $s(v) = \text{maxc};$ 
9    enquanto  $(m(C(v)) < \text{maxc})$  faça {
10      $C(v) = C(v) \cup \{w\};$ 
11      $m'(C(v)) = m(C(v));$ 
12      $m(C(v)) = m(C(v) \cup \{w\});$ 
13     se  $(m(C(v)) \leq \text{maxc})$  então {
14       seleciona  $w \in \text{iancs}(C(v));$ 
15        $\text{maxc} = c(w, u_j, v);$ 
16        $s(v) = \max\{m(C(v)), \text{maxc}\};$ 
17     }
18     senão {
19        $m(C(v)) = m'(C(v));$ 
20        $C(v) = C(v) - \{w\};$ 
21     }
22   }
23 }
```

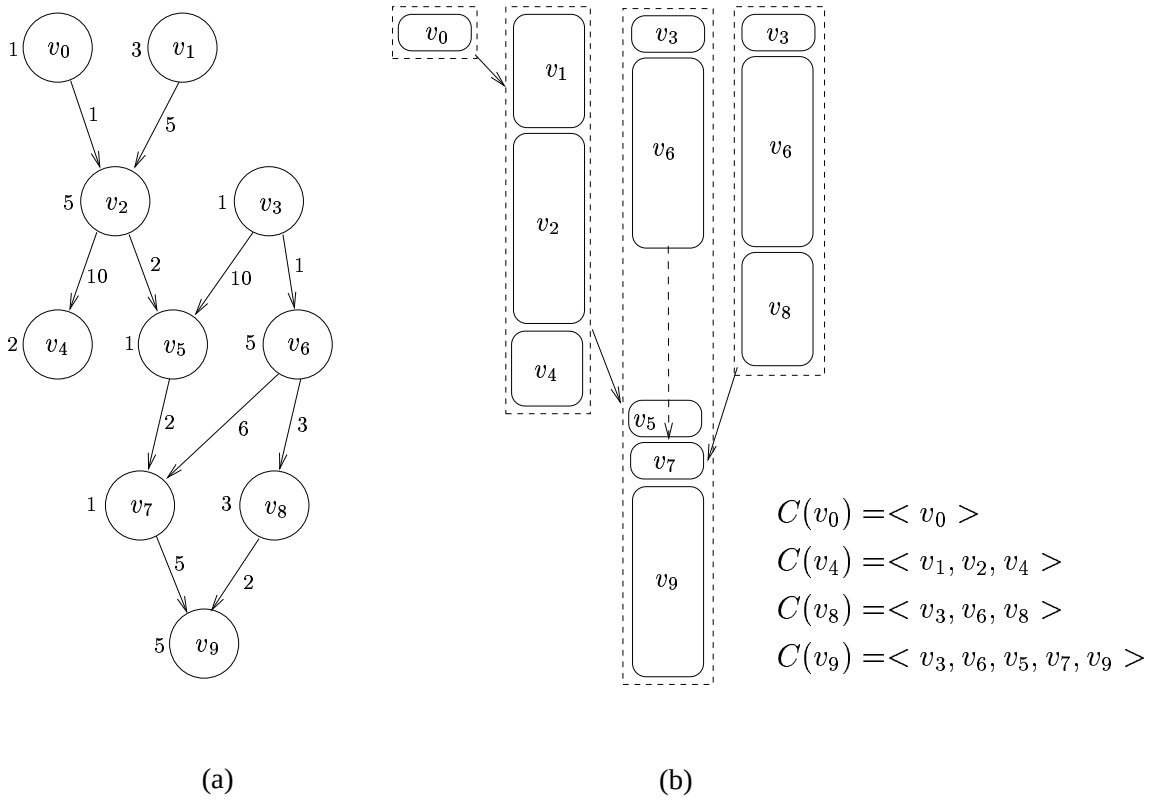
O custo do *caminho-coleção* é zero, inicialmente, e o *caminho-crítico* é o próprio *caminho-predecessor-crítico* (linha 7). Com a inclusão de cada nova tarefa candidata na coleção (linha 9), determina-se o novo custo do *caminho-coleção* (linha 11). Para saber se  $w$  deve permanecer ou não em  $C(v)$ , compara-se o custo do *caminho-predecessor-crítico* com o custo do *caminho-coleção* de  $C(v)$  após a inclusão de  $w$

(linha 12). Se  $m(C(v))$  se torna maior que  $maxc$ , então não vale a pena incluir a tarefa, pois o tempo paralelo da coleção não se tornará menor com a inclusão de  $w$ . Caso contrário, a tarefa  $w$  é incluída e o processo de aglomeração continua. Observa-se que o algoritmo pode eventualmente incluir tarefas na coleção  $C(v)$  mesmo que o tempo paralelo da aplicação aumente. Isto pode ocorrer se o novo *caminho-predecessor-crítico* for maior que o *caminho-crítico* anterior. Assim é possível escapar de alguns mínimos locais enquanto o *caminho-coleção* não for crítico (linha 8).

É importante observar que o critério de ordenação de tarefas adotado durante a construção das coleções é crucial, pois ele contribui diretamente para o resultado do escalonamento. Este também é um problema NP-completo [60]. HAL pode ordenar as tarefas de uma coleção de duas maneiras diferentes. A primeira opção é chamada ordenação por *valor-s*. Neste caso, todas as tarefas  $u_i$  contidas na coleção  $C(v)$  estão ordenadas de forma não-decrescente de acordo com o valor  $s(u_i)$  que é o tempo mais cedo de início de  $u_i$  calculado quando  $C(u_i)$  foi formada. A segunda opção é chamada ordenação *sucessor*. Neste caso a ordem de uma dada tarefa  $w_i$  é definida no momento de sua inclusão na coleção. Ela sempre é posicionada imediatamente antes de seu sucessor imediato  $u_j$  já pertencente à coleção  $C(v)$ . Desta forma é possível escalonar  $w_i$  em tempo suficiente para receber mensagens de seus predecessores, sem atrasar outras tarefas que não tenham dependência com  $w_i$ . Os resultados experimentais apresentados em [42] demonstram que a ordenação *sucessor* oferece melhores resultados para grafos com granularidade fina e com muitas tarefas paralelas.

Depois que todas as coleções são criadas, em uma fase final do primeiro estágio, são identificadas e mantidas apenas as coleções indispensáveis para a execução de todas as tarefas da aplicação. Muitas coleções podem ser redundantes, uma vez que há replicação de tarefas e foi criada uma coleção para cada tarefa  $v \in C(v)$ . Essa operação reduz o número de coleções que deverão ser mapeadas pelo segundo estágio.

A Figura 4.2 (b) mostra um exemplo das coleções produzidas por HAL para o GAD da Figura 4.2 (a). As nove tarefas do GAD foram aglomeradas em quatro



**Figura 4.2:** (a) GAD de uma aplicação hipotética com 10 tarefas e (b) o conjunto resultante de coleções necessárias definidas pelo primeiro estágio (HAL).

coleções distintas  $C(v_0)$ ,  $C(v_4)$ ,  $C(v_8)$  e  $C(v_9)$ . As tarefas  $v_3$  e  $v_6$  possuem cópias tanto em  $C(v_8)$  quanto em  $C(v_9)$ . A seta tracejada representa um arco que foi zerado pela aglomeração das tarefas adjacentes  $v_6$  e  $v_7$  em uma mesma coleção, isto é, o custo de comunicação entre estas tarefas foi eliminado.

## 4.2 Atributos de um GAD Escalonado

Após a primeira fase de escalonamento de um GAD  $G$ , as tarefas  $v \in V$  foram agrupadas em coleções que minimizam a comunicação e permitem que cada tarefa seja executada no tempo mais cedo possível  $s(v)$ , considerando um ambiente de processadores virtuais homogêneos. De maneira análoga ao que foi apresentado na Seção 2.4, alguns atributos que levam em consideração esse escalonamento podem ser associados às tarefas desse GAD. Esses atributos são úteis para a implementação

| $v$ | $C(v)$                                    | $tlevel_e(C(v))$ | $blevel_e(C(v))$ |
|-----|-------------------------------------------|------------------|------------------|
| 0   | $\langle v_0 \rangle$                     | 0                | 16               |
| 4   | $\langle v_1, v_2, v_4 \rangle$           | 8                | 17               |
| 8   | $\langle v_3, v_6, v_8 \rangle$           | 6                | 16               |
| 9   | $\langle v_3, v_6, v_5, v_7, v_9 \rangle$ | 12               | 12               |

Tabela 4.1: Atributos relativos ao GAD escalonado da Figura 4.2.

da heurística de mapeamento da segunda fase do escalonamento.

Os valores de  $blevel$  e  $tlevel$  de cada tarefa podem ser calculados para a GAD  $G$  levando em consideração o escalonamento intermediário  $S'$ . Uma vez que coleções definem grupos de tarefas que devem ser executadas num mesmo processador virtual, as comunicações entre tarefas alocadas em um mesmo processador podem ser consideradas com custo nulo, mesmo quando há dependência de dados. Seja a coleção de tarefas  $C(v) = \langle u_1, u_2, \dots, u_i, v \rangle$  alocada ao processador  $p_j$ . Definimos então o  $tlevel$  escalonado da tarefa  $v$  em um escalonamento  $S'$ , como o equivalente ao  $tlevel$  desta tarefa calculado considerando-se nulo o custo das comunicações entre tarefas pertencentes a uma mesma coleção:

$$tlevel_e(v, p_j) = \max_{u \in pred(v)} \{tlevel_e(u, p_u) + \epsilon(u) \times h(p_u) + com(u, v)\}, \quad (4.3)$$

onde o custo de comunicação  $com(u, v)$  é dado por:

$$com(u, v) = \begin{cases} \omega(u, v) \times L, & \text{se } p_u \neq p_j \\ 0, & \text{caso contrário} \end{cases}$$

O  $blevel$  escalonado da tarefa  $v$  em um escalonamento  $S$  equivale ao  $blevel$  desta tarefa calculado considerando-se nulo o custo das comunicações entre tarefas pertencentes a uma mesma coleção:

$$blevel_e(v, p_j) = \epsilon(v) \times h(p_j) + \max_{w \in succ(v)} \{blevel_e(w, p_w) + com(v, w)\}, \quad (4.4)$$

O  $tlevel_e(C(v))$  de uma coleção é definido como o menor  $tlevel_e(u_i, p_j)$  dentre todas as tarefas  $u_i$  pertencentes à coleção, ou seja,  $tlevel_e(C(v)) = \min_{u_i \in C(v)} tlevel_e(u_i, p_j)$ . O  $blevel_e(C(v))$ , por sua vez, é definido como o maior  $blevel_e(u_i, p_j)$  dentre todas as



tarefas  $u_i \in C(v)$ , ou seja,  $blevel_e(C(v)) = \max_{u_i \in C(v)} blevel_e(u_i, p_j)$ .

Essas propriedades podem ser úteis para o algoritmo que executa a segunda fase de escalonamento para permitir a atribuição de prioridades às tarefas e coleções. A Tabela 4.1 apresenta os valores correspondentes aos atributos da GAD  $G$  para o escalonamento representado na Figura 4.2.

### 4.3 Segundo Estágio: AMC

A necessidade de um segundo estágio de escalonamento é uma característica comum a certas classes de algoritmos desenvolvidos para um número ilimitado de processadores homogêneos [32, 45, 59]. No caso de HAL, por exemplo, mesmo com a eliminação de coleções de tarefas redundantes ao final do escalonamento, o número de coleções resultantes, que a princípio necessitariam de um processador exclusivo para sua execução, ainda pode ser muito maior que o número de processadores disponíveis [6]. Esta é uma desvantagem deste algoritmo, pois dificulta sua aplicabilidade direta em ambientes reais onde o número de processadores muitas vezes é reduzido. Além disso, o algoritmo não leva em consideração a heterogeneidade em um ambiente real disponível. O segundo estágio do escalonamento se faz necessário justamente para permitir que essas coleções de tarefas, produzidas pelo primeiro estágio para um ambiente virtual ilimitado e homogêneo, possam ser executadas em um ambiente real heterogêneo, e com um número de processadores menor que o total de coleções.

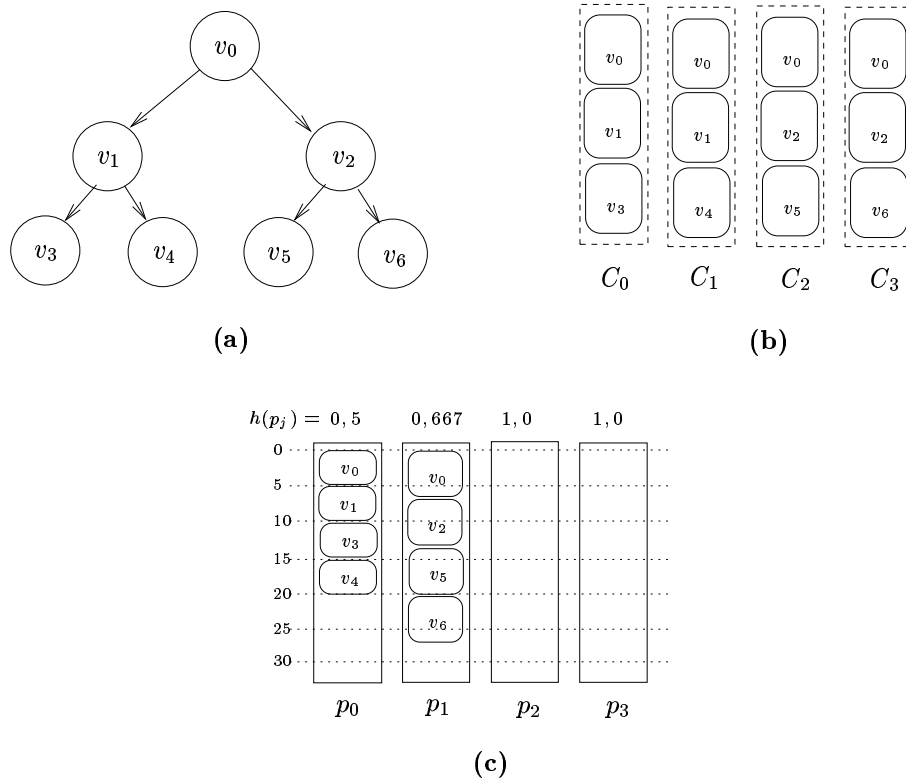
Seja  $n_c$  o número de coleções produzidas pelo primeiro estágio de CPH, e  $n_p = |P|$  o número de processadores heterogêneos disponíveis no ambiente. Na prática, o algoritmo de mapeamento deve ser capaz de combinar as  $n_c$  coleções resultantes do primeiro estágio, em um número  $n'_c \leq n_c$  de forma a explorar da melhor maneira possível a heterogeneidade do ambiente. A aglomeração de tarefas no primeiro estágio reduziu o custo de comunicação levando em consideração um ambiente virtual onde a latência da rede de dados é conhecida, e a velocidade dos processadores estimada pelo processador mais lento disponível no ambiente. Um ambiente real com

processadores mais rápidos permite que algumas tarefas sejam executadas em menos tempo que no escalonamento virtual do primeiro estágio, e através da combinação de coleções é possível obter uma melhor distribuição de cargas no escalonamento, eventualmente aglomerando mais coleções (e tarefas) em processadores mais velozes. Assim, o algoritmo de mapeamento deve buscar uma boa distribuição das  $n'_c$  coleções resultantes nos  $n_p$  processadores disponíveis. Além disso, se  $n_c > n_p$ , o algoritmo deve ser capaz de combinar e reduzir o número coleções com o menor prejuízo para o escalonamento. Finalmente, o algoritmo deve ser capaz de determinar uma boa ordenação para as tarefas nas coleções resultantes e ainda eliminar as cópias desnecessárias de tarefas no escalonamento final.

O algoritmo proposto neste trabalho para a execução do segundo estágio de escalonamento é denominado *Algoritmo de Mapeamento de Coleções (AMC)*. Optou-se por um algoritmo do tipo *list scheduling* devido à sua natural adequação quando levamos em consideração um ambiente heterogêneo e com número limitado de processadores.

A Figura 4.3 (a) mostra o GAD de uma aplicação em forma de árvore binária com sete tarefas, no qual todas as tarefas e comunicações têm peso unitário. Na Figura 4.3 (b) é apresentado o escalonamento fornecido por HAL, que considera um ambiente homogêneo. Seja um ambiente heterogêneo composto por 4 processadores em que  $h(p_0) = 0,5$ ,  $h(p_1) = 0,667$ ,  $h(p_2) = 1,0$  e  $h(p_3) = 1,0$ , e com latência unitária. A Figura 4.3 (c) exemplifica o melhor mapeamento que pode ser obtido neste ambiente para o escalonamento HAL apresentado. Observa-se que apesar de haver inicialmente quatro coleções e quatro processadores disponíveis, a utilização apenas dos dois processadores mais rápidos oferece um melhor resultado.

O Algoritmo de Mapeamento de Coleções (AMC) proposto faz parte da classe de heurísticas do tipo *list scheduling*. No caso dos algoritmos tradicionais desta classe [56, 58], tarefas são selecionadas dentro de uma ordem de prioridade e designadas a um processador, obedecendo uma função de custo pré-definida (como especificado no Algoritmo 1). Neste trabalho, entretanto, estamos interessados em designar coleções inteiras de tarefas produzidas pelo primeiro estágio a processadores



**Figura 4.3:** (a) Árvore binária com 7 tarefas, (b) o escalonamento HAL correspondente, e (c) mapeamento em um ambiente heterogêneo.

do ambiente heterogêneo. A finalidade em ambos os casos é minimizar o *makespan* do escalonamento final, ou seja, o tempo total de execução da aplicação de entrada em um dado número de processadores.

Para facilitar a notação, o conjunto de coleções necessárias produzidas pelo primeiro estágio será denotado por  $LC = \{c_0, c_1, c_2, \dots, c_{n_c-1}\}$ , onde  $c_i$  é uma coleção necessária. O resultado do primeiro estágio de CPH é visto pelo segundo estágio como um mapeamento virtual  $M_v$  em que cada coleção  $c_i$  está alocada a um processador virtual  $p'_i$  de velocidade unitária, ou seja, um processador equivalente ao mais lento da rede de processadores. O mapeamento final  $M$  definirá a alocação de cada coleção de tarefas  $c_i$  nos processadores reais. A cada iteração do algoritmo proposto, uma coleção  $c_i$ , ainda não mapeada para um processador real, deve ser selecionada de uma lista de coleções candidatas e alocada a um processador  $p_j$ .

O Algoritmo 6 mostra o *framework* de AMC. Inicialmente, a lista de coleções candidatas ao mapeamento  $LC$  contém todas as  $n_c$  coleções produzidas pelo primeiro

estágio de CPH. O conjunto de coleções mapeadas  $M$  é inicializado como vazio (linha 1) e o mapeamento virtual  $M_v$  compreende todas as  $n_c$  coleções (linha 2).

**Algoritmo 6 :** *Mapeamento de Coleções (LC)*

```

1   $M = \emptyset$ ;
2   $\forall c_i \in LC$  faça  $M_v = M_v \cup \{\langle c_i, p'_i \rangle\}$ ;
3  enquanto  $LC \neq \emptyset$  faça {
4       $c_i = \text{Seleciona Coleção } (LC)$ ;
5       $p_j = \text{Seleciona Processador } (c_i, P)$ ;
6       $LC = LC - \{c_i\}$ ;
7       $M = Aloca(c_i, p_j)$ ;
      }
8  Elimina Réplicas;
```

O conjunto de tuplas  $M_v$  é definido, neste momento, como um escalonamento virtual, para que os tempos de início e fim de cada coleção (e suas tarefas) sejam calculados. Esses tempos são também parâmetros para as decisões tomadas pelo algoritmo de mapeamento, juntamente com as características das coleções já mapeadas e as características dos processadores reais (sua carga já alocada e seu fator de heterogeneidade). AMC propõe-se, então, a tentar minimizar ainda mais o *makespan* da respectiva aplicação ao realocar as coleções para processadores mais ou tão rápidos quanto os processadores virtuais. Na linha 3 o processo de mapeamento (realocação nos processadores reais) se inicia, sendo executado enquanto houver coleções em  $LC$  para serem mapeadas. A cada iteração, uma coleção  $c_i \in LC$  é selecionada de acordo com uma prioridade (linha 4). A partir de um determinado critério relacionando essa coleção candidata aos processadores disponíveis no ambiente, é escolhido o processador  $p_j \in P$  no qual a coleção  $c_i$  será alocada (linha 5). A coleção  $c_i$  é então removida da lista de candidatas  $LC$  (linha 6), a respectiva tupla  $\langle c_i, p'_i \rangle$  retirada do mapeamento virtual  $M_v$ , e a nova tupla  $\langle c_i, p_j \rangle$  é inserida no conjunto das coleções mapeadas  $M$  (linha 7).

### 4.3.1 Escolha da Próxima Coleção

A cada iteração do programa, é selecionada uma coleção para mapeamento (Algoritmo 6, linha 4). O algoritmo tenta avaliar a importância das coleções ainda não alocadas em processadores reais do ambiente em relação ao mapeamento final a partir das propriedades das tarefas no mapeamento parcial em uma dada iteração. Conforme a definição apresentada na Equação 4.4, o  $blevel_e(c)$  é capaz de indicar as tarefas que fazem parte do caminho crítico do grafo parcialmente escalonado e, conseqüentemente, as coleções que as contêm. Com a utilização deste critério para a seleção de coleções, é possível fazer uma alocação das coleções desde aquelas que contêm tarefas origem no caminho crítico, até aquelas que contêm tarefas destino no caminho crítico, tentando reduzir o *makespan* da aplicação escalonada como um todo.

**Algoritmo 7 : *Seleciona Coleção (LC)***

```

1   $max_{bl} = 0; max_{\delta} = 0;$ 
2   $\forall c \in LC$  faça {
3      se  $(blevel_e(c) > max_{bl})$  então {
4           $max_{bl} = blevel_e(c); max_{\delta} = \delta(c); c^* = c;$ 
5          }
6      senão se  $(blevel_e(c) == max_{bl})$  então {
7          se  $\delta(c) > max_{\delta}$  então {
8               $max_{\delta} = \delta(c); c^* = c;$ 
9          }
10     }
11 }
12 retorna  $(c^*);$ 

```

O Algoritmo 7 mostra o procedimento *Seleciona Coleções* detalhadamente. O primeiro critério para a seleção é a escolha de uma coleção de maior *blevel escalonado* (linha 3). A coleção  $c$  que tem o maior  $blevel_e(c)$  será aquela que contém a primeira tarefa  $u_i$  de maior  $blevel_e(u_i, p'_j)$ , ou seja, aquela tarefa do grafo escalonado que é origem do maior caminho até um nó de saída. Essa tarefa é a origem do caminho-crítico não escalonado. Sendo as prioridades aqui utilizadas conside-

radas “dinâmicas”, o  $blevel_e$  de cada tarefa e, conseqüentemente, de cada coleção, são recalculados a cada iteração.

Como um critério de desempate na escolha de coleções, o conceito de **discrepância** de uma coleção  $c \in LC$  ainda não mapeada é aqui proposto. A discrepância de  $c$  representa uma estimativa do custo de execução das tarefas pertencentes a  $c$  que não estão alocadas nas coleções já mapeadas nos processadores já utilizados. Formalmente, a discrepância  $\delta(c)$  de  $c$  é definida como:

$$\delta(c) = \min_{\forall p \in P} \left\{ \sum_{\forall u \in c \cap u \notin p} \epsilon(u) \times h(p) \right\}, \quad (4.5)$$

Assim, quanto maior a discrepância  $\delta(c)$  de uma coleção  $c$  em relação às coleções já remapeadas, maior será o custo provável decorrente do mapeamento desta coleção, devido ao peso de computação que faz parte de  $c$  e não é comum às coleções já mapeadas no conjunto de processadores reais disponíveis. Selecionando-se a coleção  $c$  com a maior discrepância (linha 6), dentre as coleções com empate de maior  $blevel$  escalonado, é dada maior prioridade para a coleção que tem o maior peso de computação correspondente a tarefas ainda não escalonadas.

Seja o GAD da Figura 4.4 (a), cujas coleções resultantes do escalonamento do primeiro estágio (HAL) podem ser vistas em (b). Considere-se o ambiente heterogêneo composto por 4 processadores em que  $h(p_0) = 0,5$ ,  $h(p_1) = 0,667$ ,  $h(p_2) = 1,0$  e  $h(p_3) = 1,0$ , e com latência  $L = 1$  unitária. A primeira coleção a ser escolhida para o mapeamento é a coleção  $c_1$ , que tem o maior  $blevel_e(c) = 17$ . Suponha que esta coleção é mapeada no processador  $p_0$ . Esta situação é representada na Figura 4.4 (c). Para seleção da próxima coleção verifica-se que  $c_2$  e  $c_3$  têm mesmo valor de  $blevel_e$ , que é 16, mas  $\delta(c_2) = 6,5$  que é maior que  $\delta(c_3) = 4,5$ . Então a coleção  $c_2$  é escolhida como próxima a ser mapeada.

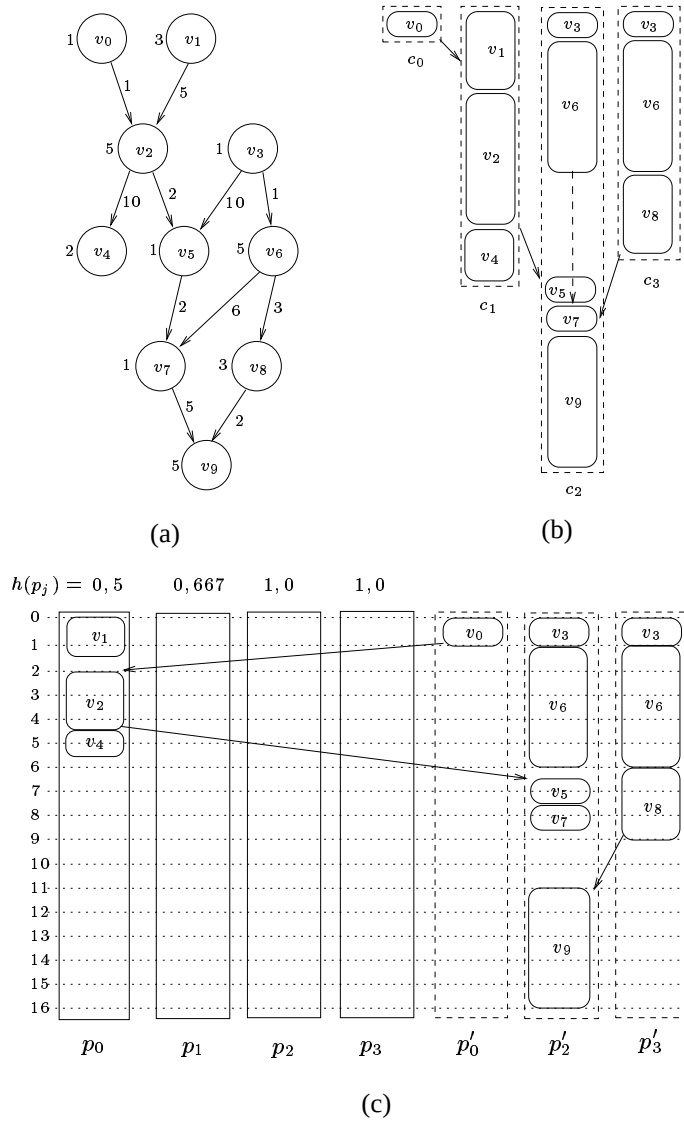


Figura 4.4: (a) GAD de uma aplicação, (b) o escalonamento HAL e (c) a escolha da coleção  $c_i$  durante a primeira iteração de AMC.

### 4.3.2 Escolha do Processador

A princípio, a maioria dos *list scheduling* tradicionais tem como sua função de custo a minimização do *makespan* da aplicação considerando o escalonamento parcial. No entanto, na estratégia aqui proposta, foi observado que durante várias iterações esse *makespan* permanece o mesmo. Isso se deve ao fato que esse cálculo leva em consideração coleções destino (que possuem tarefas destino) que ainda não foram selecionadas para serem mapeadas em processadores reais. Assim, é importante considerar primeiramente o impacto de alocar a coleção selecionada em um

processador real considerando somente as coleções já mapeadas, que pode ser representado pelo *makespan* parcial denotado por  $mspan(M \cup \{\langle c, p_j \rangle\})$ , onde  $\langle c, p_j \rangle$  é o mapeamento da coleção  $c$  em  $p_j$ . Portanto, o algoritmo proposto seleciona o processador  $p_j$  que minimiza  $mspan(M \cup \{\langle c, p_j \rangle\})$ . Em caso de empate, seleciona aquele  $p_j$  que minimiza o *makespan* parcial denotado por  $mspan(M_{all})$ , que considera tanto as coleções já mapeadas quanto as ainda não mapeadas, ou seja,  $M_{all} = M_v \cup M \cup \{\langle c, p_j \rangle\}$ .

Se ainda houver empate, foi proposta mais uma condição a ser testada, definida a partir da observação dos processadores disponíveis no ambiente e da evolução do processo de mapeamento. Seja  $P$  o conjunto dos  $n_p$  processadores disponíveis em um ambiente heterogêneo. Define-se a lista  $P_e = \langle p_0, p_1, p_2, \dots, p_{n_e-1} \rangle$  de  $n_e$  processadores, como aquela que contém todos os processadores  $p \in P$  ainda sem carga, ou seja, ainda sem nenhuma coleção mapeada, ordenada de forma não decrescente por fator de heterogeneidade, ou seja,  $h(p_j) \leq h(p_{j+1})$ ,  $\forall j = 0, 1, 2, \dots, n_e - 1$ . Esta lista inicialmente contém todos os processadores  $p \in P$ . Seja  $P_u$  a lista dos processadores utilizados, ou seja, aqueles processadores que têm coleções de tarefas alocadas, em uma dada iteração do processo de mapeamento, ordenada de acordo com o mesmo critério. Esta lista inicialmente é vazia. À medida em que o mapeamento progride, cada vez que uma coleção  $c$  é designada a um processador ainda sem carga, este é retirado de  $P_e$  e incluído em  $P_u$ . Em cada iteração, a coleção escolhida para ser mapeada é avaliada em cada processador da lista  $P_u$  e também no primeiro processador da lista de processadores ainda não utilizados  $P_e$ , chamado  $p_{next}$ , que é o próximo processador mais rápido ainda sem carga designada. Quando um processador é escolhido, se para todo  $p_j \in (P_u \cup \{p_{next}\})$  houver empate nos critérios até então descritos, a coleção  $c$  poderia ser alocada em  $p_{next}$ , acrescentando mais um processador ao conjunto de processadores utilizados  $P_u$ , que é mais ou tão lento quanto o processador mais lento até então utilizado. No entanto, ser guloso no uso de processadores pode ser desvantajoso nas iterações seguintes do processo de mapeamento. Assim, na tentativa de capturar a idéia de “carga máxima de execução”, calcula-se o quanto os processadores já utilizados ainda podem ser associados a coleções considerando-se a configuração de mapeamento da iteração corrente.



Seja  $T(p_j)$  a lista das tarefas já mapeadas para um processador  $p_j \in P_u$ . Seja  $c$  a coleção selecionada. A carga média associada aos processadores de  $P_u$  é definida como:

$$\bar{\lambda}(P_u) = \frac{\sum_{\forall p_j \in P_u} \sum_{\forall v \in T(p_j)} \epsilon(v) \times h(p_j)}{|P_u|} \quad (4.6)$$

Seja a carga a executar em um novo processador  $p_{next}$  igual a

$$\lambda(c, p_{next}) = \sum_{\forall v \in c} \epsilon(v) \times h(p_{next}). \quad (4.7)$$

O processador  $p_{next}$  deve ser selecionado se  $\lambda(c, p_{next}) < \bar{\lambda}(P_u) + \delta(c)$ , pois neste caso, na verdade,  $p_{next}$  deve ser rápido o suficiente para executar as tarefas de  $c$ . Caso contrário, o conjunto de processadores  $P_u$  tem maiores chances de executar mais rapidamente suas coleções já mapeadas e a coleção  $c$  selecionada.

O Algoritmo 8 mostra o procedimento *Seleciona Processador* detalhadamente. Em cada iteração, a coleção escolhida para ser mapeada é avaliada em cada processador  $p_j$  da lista  $P_u$  e também em  $p_{next}$  (linha 2), o primeiro processador da lista de processadores ainda não utilizados  $P_e$ . No início do mapeamento  $P_e$  contém todos os  $n_p$  processadores disponíveis e  $P_u = \emptyset$ . Neste caso, a primeira coleção selecionada é necessariamente alocada em  $p_{next}$ , que é o processador mais rápido do ambiente. A avaliação da coleção  $c$  em um dado processador  $p_j$  ocorre da seguinte forma. A coleção  $c$  é removida do conjunto de coleções não mapeadas  $M_v$  (linha 1). São calculados os valores de tempo de início e tempo de fim de cada tarefa  $v_j \in V$  considerando o mapeamento parcial definido por  $M_{all} = M_v \cup M \cup \{\langle c, p_j \rangle\}$ . Seja  $mspan(M \cup \{\langle c, p_j \rangle\})$ , o *makespan* do conjunto  $M \cup \{\langle c, p_j \rangle\}$  de coleções, ou seja, o maior tempo de fim das tarefas  $v \in (M \cup \{\langle c, p_j \rangle\})$ . Analogamente,  $mspan(M_{all})$  é o *makepan* de toda a aplicação considerando o mapeamento parcial na respectiva iteração. O algoritmo seleciona o processador  $p_j$  que minimiza  $mspan(M \cup \{\langle c, p_j \rangle\})$  (linhas 4 a 7). Em caso de empate, é selecionado então o processador  $p_j$  que minimiza o *makespan* da aplicação  $mspan(M_{all})$  (linhas 8 a 11). Se ainda assim houver empate, o critério de seleção leva em consideração a comparação da carga média associada aos processadores de  $P_u$ , a carga a executar em  $p_{next}$  e a discrepância,

conforme definidos nas Equações 4.5, 4.6 e 4.7. Se  $\lambda(c, p_{next}) < \bar{\lambda}(P_u) + \delta(c)$ , sempre que há empate nas duas primeiras condições, o mapeador troca o melhor processador até o momento armazenando em  $proc^*$  pelo um processador  $p_j$  mais lento que o anterior (linhas 13), aumentando a distribuição de carga pelos processadores mais lentos. Se  $\lambda(c, p_{next}) > \bar{\lambda}(P_u) + \delta(c)$  e há empate nas duas primeiras condições, o mapeador mantém sua escolha prévia, favorecendo a concentração de carga nos processadores mais rápidos. Ao final, o processador escolhido  $proc^*$ , se ainda não estava sendo utilizado, deixa de fazer parte  $P_e$  e é incluído em  $P_u$  (linha 15).

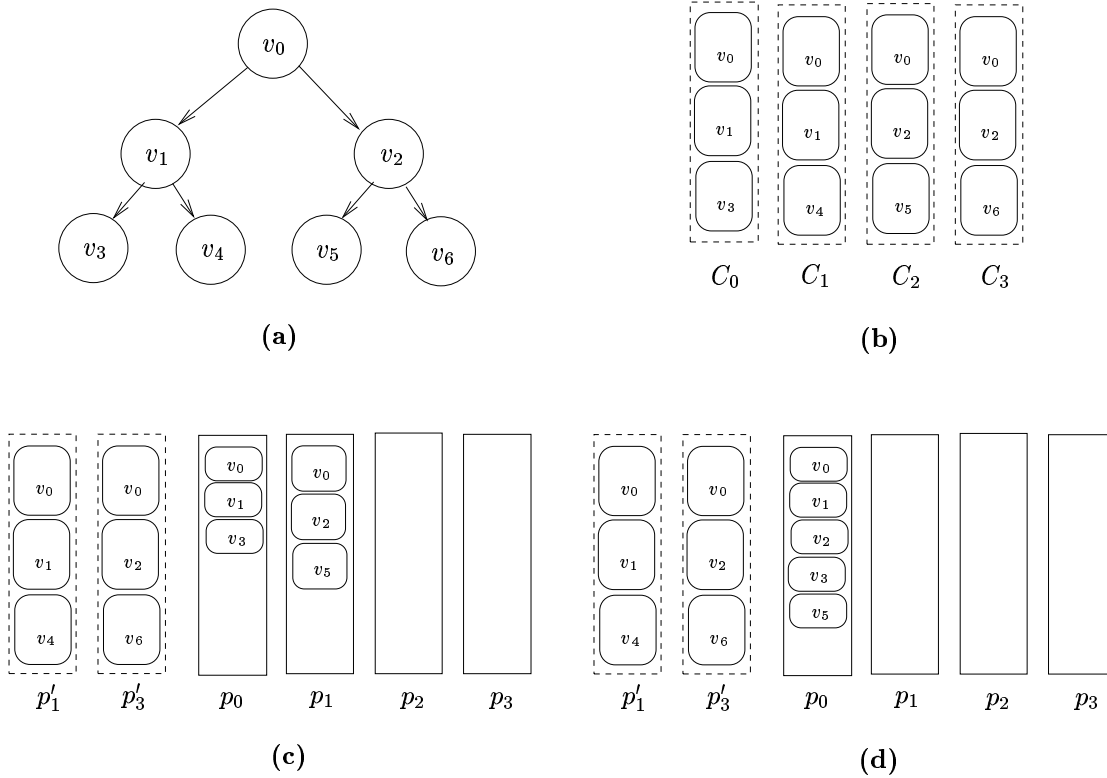
**Algoritmo 8 :** *Seleciona Processador* ( $c, P$ )

```

1   $M_v = M_v - \langle c, p'_c \rangle$ ;  $mspan^* = \infty$ ;
2   $\forall p_j \in (P_u \cup \{p_{next}\})$  faça {
3     $M_{all} = M \cup M_v \cup \langle c, p_j \rangle$ ;
4    se ( $mspan(M \cup \{\langle c, p_j \rangle\}) < mspan^*$ ) então {
5       $mspan^* = mspan(M \cup \{\langle c, p_j \rangle\})$ ;
6       $mspan^*_{all} = mspan(M_{all})$ ;
7       $proc^* = p_j$ ;
8    }
9    se ( $mspan(M_{all}) < mspan^*_{all}$ ) então {
10      $mspan^*_{all} = mspan(M_{all})$ ;
11      $proc^* = p_j$ ;
12   }
13   se ( $mspan(M_{all}) == mspan^*_{all}$ ) então {
14     se ( $\lambda(c, p_{next}) < \bar{\lambda}(P_u) + \delta(c)$ ) então  $proc^* = p_j$ ;
15   }
16 }
17 se ( $proc^* == p_{next}$ ) então {
18    $P_u = P_u \cup \{proc^*\}$ ;  $P_e = P_e - \{proc^*\}$ ;
19 }
20 retorna ( $proc^*$ );
```

A Figura 4.5 mostra um exemplo de aplicação do Algoritmo *Seleciona Processador* para o GAD descrito em (a) e o escalonamento HAL correspondente representado em (b), durante uma dada iteração de AMC. A coleção  $c_0 = \langle v_0, v_1, v_3 \rangle$  já foi mapeada para o processador  $p_0$ , e a próxima coleção a ser mapeada é a

$c_2 = \langle v_0, v_2, v_5 \rangle$ . Na Figura 4.5 (c) é mostrada a tentativa de mapeamento de  $c_2$  em  $p_1$ , e em (d)  $c_2$  é mapeado em  $p_0$ . Em ambos os casos,  $mspan(M_{all})$ , o *makespan* do conjunto, é igual a 3. Mas para  $M = \{\langle c_0, p_0 \rangle, \langle c_2, p_0 \rangle\}$ , o valor de *makespan* escalonado é menor do que para  $M = \{\langle c_0, p_0 \rangle, \langle c_2, p_1 \rangle\}$ . Assim,  $p_1$  é o processador selecionado.



**Figura 4.5:** (a) Árvore binária com 7 tarefas, (b) o escalonamento HAL correspondente, (c) e (d) passos intermediários do mapeamento do grafo em que  $c_2$  é testada em  $p_1$  e  $p_0$ .

### 4.3.3 Alocação de uma Coleção $c$ ao Processador $p_j$

Ao mapearmos uma coleção  $c_i$  no processador  $p_j$  (Algoritmo 6, linha 7), as cópias de tarefas que já tenham sido alocadas em  $p_j$  devem ser desconsideradas. Além disso, o conjunto de tarefas final, resultante da aglomeração de coleções em um processador, deve ser ordenado da melhor forma possível. A questão de ordenação constitui um problema NP-completo [60].

A primeira fase de CPH já cria um conjunto de coleções em que as tarefas são

ordenadas dentro de critérios específicos. Neste trabalho, em que HAL é utilizado na primeira fase, as tarefas podem estar ordenadas por *valor-s* ou por *ordem sucessor* [42], conforme discutido na Seção 4.1. A segunda fase mantém a ordem das tarefas em  $c$  especificada por HAL. No entanto, quando coleções diferentes são mapeadas em um mesmo processador, após a eliminação das cópias desnecessárias das tarefas, é preciso estabelecer uma ordenação entre as tarefas pertencentes às diferentes coleções (tarefas que, obviamente, não são independentes).

**Algoritmo 9 :** *Aloca* ( $c, p_j$ )

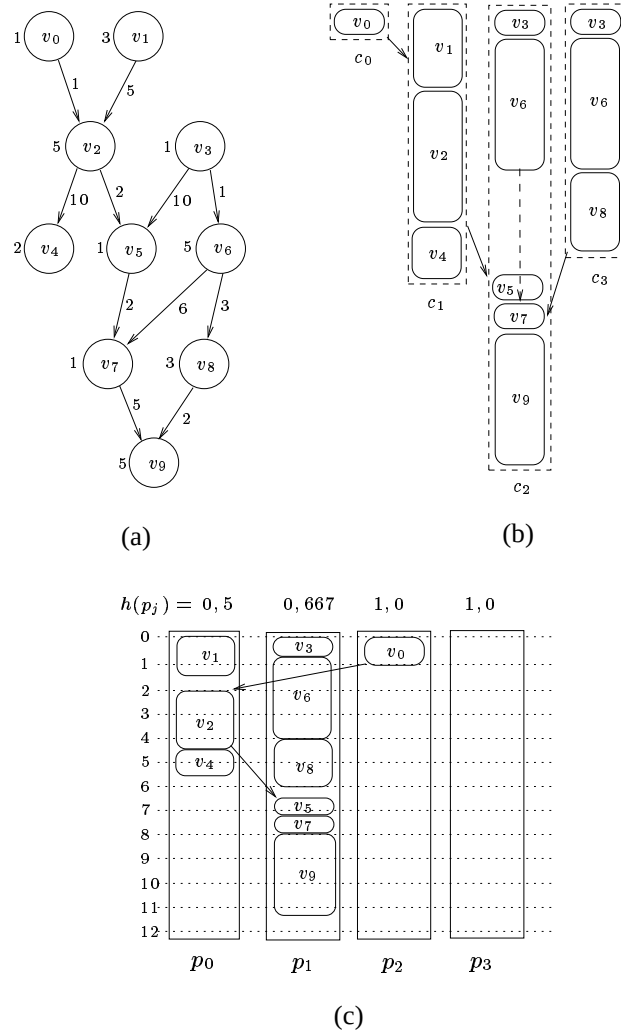
```

1   $L_1 = T(p_j); L_2 = c; T(p_j) = \emptyset;$ 
2   $v = \text{primeiro}(L_1); u = \text{primeiro}(L_2);$ 
3  enquanto  $((L_1 \neq \emptyset) \text{ e } (L_2 \neq \emptyset))$  faça {
4      se  $(v == u)$  então {
5           $T(p_j) = T(p_j) + \{v\}; L_1 = L_1 - \{v\}; L_2 = L_2 - \{u\};$ 
6           $v = \text{primeiro}(L_1); u = \text{primeiro}(L_2);$ 
7      }
8      se  $(t_{imc}(v, p_j) < t_{imc}(u, p_j))$  então {
9           $T(p_j) = T(p_j) + \{v\}; L_1 = L_1 - \{v\}; v = \text{primeiro}(L_1);$ 
10     }
11     se  $(t_{imc}(v, p_j) == t_{imc}(u, p_j))$  então {
12         se  $(blevel_e(v) \geq blevel_e(u))$  então {
13              $T(p_j) = T(p_j) + \{v\}; L_1 = L_1 - \{v\}; v = \text{primeiro}(L_1);$ 
14         }
15         se  $(t_{imc}(v, p_j) == t_{imc}(u, p_j))$  então {
16              $T(p_j) = T(p_j) + \{u\}; L_2 = L_2 - \{u\}; u = \text{primeiro}(L_2);$ 
17         }
18     }
19 }

```

Seja  $T(p_j)$  a lista da tarefas já mapeadas para um processador  $p_j \in P_u$  e  $c$  a coleção que será alocada em  $p_j$  em uma dada iteração do algoritmo. Seja  $t_{imc}(v, p_j)$

o tempo de início mais cedo da tarefa  $v$  em  $p_j$ , conforme definido na Equação 2.8, calculado para todo  $v \in T(p_j)$  e  $u \in c$ , considerando o processador  $p_j$  e o mapeamento temporário  $M_{all}$ . A nova lista de tarefas alocadas a  $T(p_j)$  é ordenada de acordo com o menor valor entre  $t_{imc}(v, p_j)$  e  $t_{imc}(u, p_j)$ , e, quando há empate, de acordo com o maior valor entre  $blevel_e(v)$  e  $blevel_e(u)$ , conforme descrito no Algoritmo 9.



**Figura 4.6:** (a) GAD de uma aplicação, (b) escalonamento fornecido por HAL e (c) mapeamento produzido por AMC.

O primeiro critério garante que cada tarefa terá início o mais cedo possível, reduzindo o tempo ocioso de  $p_j$ . De acordo com o segundo critério, tarefas que fazem parte de um caminho mais longo até uma tarefa de saída, e que provavelmente enviam mensagens para outras coleções, receberão prioridade maior na ordenação.

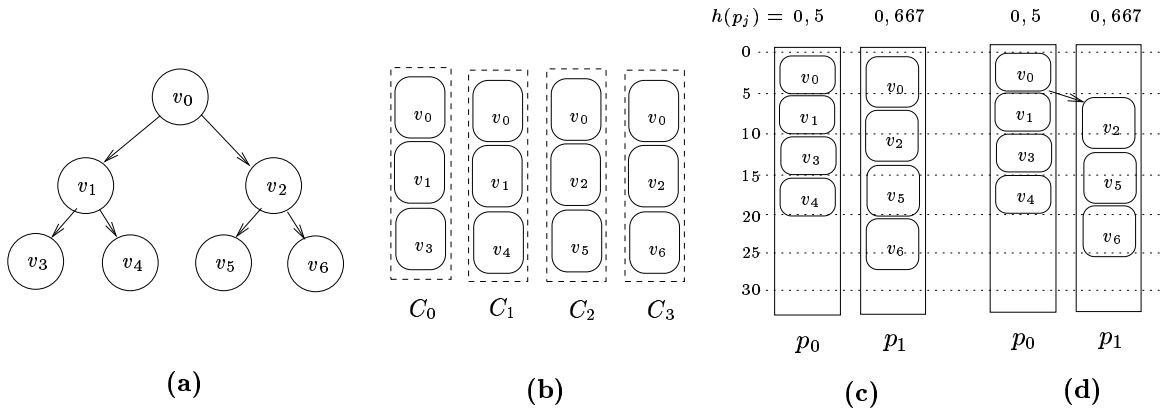
Seja o GAD representado na Figura 4.6 (a), cujo escalonamento HAL pode ser

visto na Figura 4.6 (b). Considere-se o ambiente heterogêneo composto por 4 processadores em que  $h(p_0) = 0,5$ ,  $h(p_1) = 0,667$ ,  $h(p_2) = 1,0$  e  $h(p_3) = 1,0$ , e com latência unitária. A Figura 4.6 (c) representa o mapeamento obtido pelo algoritmo proposto para este ambiente, com um *makespan* de 11,17. A ordenação das tarefas do processador  $p_1$  é dada por  $T(p_1) = \langle v_3, v_6, v_8, v_5, v_7, v_9 \rangle$ . Devido à restrição  $\omega(v_2, v_5)$ , a tarefa  $v_5$  só pode iniciar em  $t_{imc}(v_5) = 6,5$ . As tarefas  $v_6$  e  $v_8$  podem iniciar antes, por isso são alocadas primeiro.

#### 4.3.4 Otimização

Finalmente, após todas as coleções terem sido designadas aos processadores disponíveis no ambiente, existe ainda a possibilidade de minimizar o *makespan* resultante. O primeiro estágio do escalonamento criou coleções que permitiam que cada tarefa fosse executada mais cedo criando réplicas de seus predecessores para reduzir a comunicação entre as coleções, e, conseqüentemente, o *makespan*. Após o mapeamento, as coleções originais foram designadas a um número igual ou menor de processadores. No segundo estágio, AMC, na verdade, reaglomera tarefas, e durante esse processo algumas cópias de uma mesma tarefa pertencentes a coleções distintas podem ter sido eliminadas quando mapeadas para um mesmo processador real (Algoritmo 9 linhas 4 a 6). Conseqüentemente, com o remapeamento, algumas mensagens definidas entre coleções distintas também podem ter sido eliminadas quando essas coleções são mapeadas para o mesmo processador.

Uma subsequente eliminação de cópias de uma mesma tarefa poderia ainda ser realizada considerando, obviamente, o modelo de latência. Na verdade, a eliminação se torna possível devido à diferença de velocidades entre processadores. Quando uma cópia de uma tarefa  $v$  é remapeada para um processador  $p_k$  mais rápido, sua respectiva mensagem poderá ser enviada (e conseqüentemente recebida) antes do que havia sido previsto pelo primeiro estágio (HAL). Assim, essa cópia  $v$  poderia enviar mensagens para outros sucessores de  $v$  sem atrasar o início desses sucessores. Muitas vezes, o início desses outros sucessores podem ser adiantados. Por exemplo, seja o GAD na Figura 4.7 (a) onde todas as tarefas têm peso  $\epsilon = 10$ , as comunicações



**Figura 4.7:** (a) GAD de uma árvore binária, (b) o respectivo escalonamento produzido pelo primeiro estágio, (c) o mapeamento produzido por AMC antes da eliminação de réplicas, (d) e depois da eliminação.

têm peso  $\omega = 1$  e a latência de comunicação é  $L = 1$ . As coleções produzidas por HAL podem ser vistas em (b). AMC remapeia as coleções para os processadores  $p_0$  e  $p_1$ , onde  $h(p_0) = 0,5$  e  $h(p_1) = 0,667$ , da forma descrita em (c), com *makespan* resultante igual a 26,67. Nota-se que a cópia de  $v_0$  em  $p_0$  pode enviar mensagem para  $v_2$  em  $p_1$  de tal modo que  $v_2$  iniciará antes. Assim, vale a pena eliminar a cópia de  $v_0$  em  $p_1$  e estabelecer a mensagem entre  $v_0$  em  $p_0$  para  $v_2$  em  $p_1$ , como ilustrado em (d), e o novo *makespan* passa a ser 26. Note que a tupla  $\langle v_0, p_1, 0 \rangle$  fazia parte do caminho crítico.

Essa eliminação é realizada visitando-se o caminho crítico descrito no escalonamento final  $S$ , definido da seguinte forma. A tarefa  $v$  em  $p_j$  faz parte de  $CP$  se  $v$  é tarefa destino e  $t_F(v, p_j) = mspan(S)$  ou se  $v \in pred(w)$ ,  $w \in CP$  e  $t_F(v, p_j) = t_{imc}(w, p_w)$ . Uma tarefa crítica  $v$  em  $p_j$ , que possua uma cópia em outro processador cujas mensagens cheguem mais cedo que  $t_F(v, p_j)$  em  $p_j$ , é candidata à eliminação. Como pelo menos um sucessor de uma tarefa crítica é também crítico, a eliminação poderá levar a um menor tempo de início deste seu sucessor e reduzir o comprimento do escalonamento.

O algoritmo de *Eliminação de Réplicas* (Algoritmo 10) determina o caminho crítico do escalonamento  $S$  resultante (linha 2). Em seguida, para cada tarefa  $v \in CP$ , desde os nós de saída até os nós de entrada (visita *bottom-up*), o algo-

ritmo verifica se existem cópias em outros processadores  $p_k \neq p_j$  (linha 5). Se for possível reduzir o *makespan* do escalonamento retirando-se essa tarefa e fazendo seus sucessores receberem mensagem de outra cópia sua (linha 6), a tarefa é eliminada. A operação é repetida até que o caminho crítico não seja mais modificado (linha 1).

**Algoritmo 10 :** *Elimina Réplicas (S)*

```

1  faça {
2    determina CP;
3    eliminou = 0 ;
4     $\forall \langle v, p_j, t_l \rangle \in CP$  faça {
5       $\forall p_k \in \text{proc}(v) \mid p_k \neq p_j$  faça {
6        se  $mspan(S - \{\langle v, p_j, t_l \rangle\}) < mspan(S)$  então {
7           $T(p_j) = T(p_j) - \{v\}$  ;
8          eliminou = 1 ;
9           $mspan(S) = mspan(S - \{\langle v, p_j, t_l \rangle\})$  ;
        }
      }
    }
  }
  enquanto eliminou }
```

## 4.4 Resumo

Neste capítulo foram apresentadas as características da estratégia de escalonamento em dois estágios. Foi explicado o funcionamento de HAL, o algoritmo selecionado para a primeira fase do escalonamento. Foram apresentados alguns atributos que estão associados às tarefas e coleções de um grafo escalonado. AMC, o algoritmo proposto para o segundo estágio de escalonamento, foi apresentado detalhadamente. No próximo capítulo serão apresentados os resultados experimentais obtidos a partir dos testes realizados.



# Capítulo 5

## Resultados: Uma Análise Experimental

Este capítulo apresenta a análise experimental da abordagem de dois estágios proposta neste trabalho, a Clusterização para Processadores Heterogêneos (CPH), para validação de sua aplicabilidade e benefícios. Como parte desta análise experimental, são realizadas comparações com duas outras estratégias, ambas algoritmos do tipo *list scheduling*, que realizam o escalonamento em um único estágio: HEFT [58], que não usa replicação de tarefas, e LDBS [20], que dispõe deste recurso.

### 5.1 Análise experimental

O estudo do problema de escalonamento deste trabalho, devido a sua complexidade, se concentra em classes específicas de GADs que podem ser divididas de acordo com a topologia ou com a granularidade, ou ambos. De uma forma geral, enquanto a topologia dos grafos escolhidos pode representar classes específicas de aplicações, a variação da granularidade pode corresponder a diferentes sistemas destino. Essa abordagem permite obter resultados teóricos para instâncias específicas do problema, identificando, por exemplo, condições e resultados ótimos, e faixas de parâmetros para o qual um algoritmo proposto é mais eficiente.

Neste estudo, parte da análise experimental está baseada em um conjunto de GADs com pesos de computação e de comunicação unitários, tradicionalmente conhecidos na literatura de escalonamento para processadores homogêneos como *unit execution transfer and unit data transfer* (UET-UDT), e que inclui grafos regulares (vários tamanhos de árvores binárias, árvores binárias invertidas, diamantes) [5] e irregulares (gerados aleatoriamente e uma aplicação da dinâmica molecular [32]). Também foram considerados os grafos PSG, GADs irregulares com pesos arbitrários, extraídos da literatura e propostos por Kwok e Ahmad para serem usados como parte de um conjunto de *benchmark* para a comparação entre algoritmos de escalonamento [34]. Além disso, uma família de grafos de diversos tamanhos representando o cálculo da eliminação de Gauss [18], juntamente com o grafo representando um problema de dinâmica molecular da física [32], permitiram analisar o desempenho de CPH perante aplicações reais.

Para poder variar a granularidade das GADs de entrada e assim determinar mais precisamente aplicações de granularidade grossa ou fina, foi definido um fator multiplicativo da computação  $F$ , de tal forma que o peso de execução de qualquer tarefa  $v \in G$  passa a ser  $F \times \epsilon(v)$ . Da mesma forma, variou-se a latência para diferentes experimentos de tal modo que os custos de comunicação associados a cada aresta  $(u, v) \in E$  se tornam iguais a  $L \times \omega(u, v)$ . A variação desses fatores foi implementada praticamente para todos os GADs UET-UDT.

É importante ressaltar que ambientes heterogêneos acrescentam uma maior complexidade ao problema do escalonamento [3] ao, por exemplo, tornar conceitos como granularidade (Seção 2.3) ou prioridade (usado extensivamente em heurísticas de *list scheduling* para o problema de processadores homogêneos) difíceis de calcular ou aplicar. Além disso, ao contrário do que ocorre com relação às aplicações paralelas representadas pelos GADs, há pouca tradição na modelagem de ambientes heterogêneos para o estudo de escalonamento estático de tarefas.

## 5.2 Ambientes Não Restritos

Em uma primeira análise, foram criados ambientes computacionais de tamanhos diferentes de tal forma que o número de processadores disponíveis fosse sempre suficiente para o mapeamento de todas as coleções produzidas pelo primeiro estágio de CPH. Esse tipo de experimento é importante para analisar o impacto das heurísticas na minimização do *makespan*, assim como na redução eficiente do número de processadores realmente necessários. Assim, foram necessárias quatro dimensões distintas de ambiente, contendo um total de 20, 30, 50 ou 100 processadores, respectivamente. Considerou-se também que todos os processadores são interconectados por uma rede homogênea. Esta condição, também adotada em estudos anteriores [14], pode ser vista como a modelagem de um conjunto de diferentes PCs ou estações de trabalho interconectados através de um switch.

Dois tipos de ambientes computacionais simulados foram propostos. No primeiro deles, chamado de **dual**, seus processadores podem ser divididos em dois subconjuntos, cada subconjunto associado ao mesmo fator de heterogeneidade, classificando assim os processadores em rápidos e lentos. O segundo tipo, chamado de **aleatório**, tem processadores com fatores de heterogeneidade determinados aleatoriamente, para uma dada média e um dado desvio padrão. O ambiente dual poderia representar, por exemplo, um cluster de computadores composto por máquinas com dois tipos de processadores diferentes, um deles de última geração e mais rápido que o outro. Já o ambiente aleatório poderia representar um cluster onde as máquinas são semelhantes, mas têm cargas diferentes.

Para cada grafo e para cada par de parâmetros  $(F, L)$  utilizados, foram considerados 11 ambientes computacionais do tipo dual. Em cada ambiente, havia  $x$  processadores rápidos, com  $h() = 0, 5$ , e  $|P| - x$  processadores lentos, com  $h() = 1, 0$ , onde  $x$  varia de 0 até  $|P|$  em incrementos de 10% de  $|P|$ , sendo  $|P| = 20, 30, 50$  e 100. No caso dos ambientes do tipo aleatório, foi avaliado o escalonamento de cada grafo para 4 ambientes com  $|P|$  processadores em que a média dos fatores de heterogeneidade era sempre unitária, e o desvio padrão variava entre 0 (caso homogêneo), 0,125, 0,25 ou 0,5, com  $|P| = 20, 30, 50$  e 100.

Nos Apêndices A a G são apresentados os resultados obtidos para o escalonamento de todos os grafos, tabelados para cada par de parâmetros  $(F, L)$ , separadamente para os ambientes duais e aleatórios. Em cada tabela do ambiente dual, a primeira coluna contém  $x$ , o incremento do número de processadores mais rápidos. A primeira coluna das tabelas referentes aos ambientes aleatórios é  $\bar{\sigma}$ , o desvio padrão do ambiente. Nas colunas seguintes são apresentados o *makespan* do escalonamento e o número de processadores utilizados, para HEFT, LDBS, CHP com HAL ordenando por valor-s (o0), e CHP com HAL com ordenação por sucessor (o1), respectivamente. As duas últimas colunas mostram a vantagem percentual entre o melhor escalonamento obtido por CHP e HEFT, e CHP e LDBS (melhor CHP com HAL por ordenação o0 e o1).

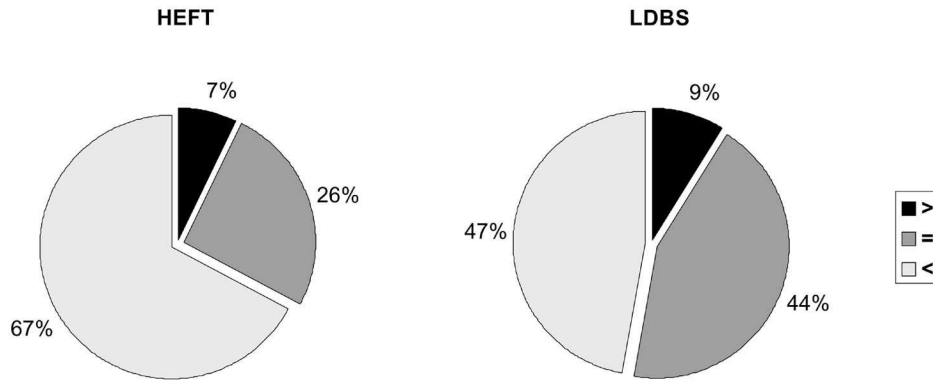
### 5.2.1 GADs Unitários

Para a análise dos GADs UET-UDT, foi conduzida uma série de experimentos com GADS do tipo árvore binária (ou *out-tree*), árvore binária invertida (ou *in-tree*), diamante e randômicos, considerando os pares de parâmetros  $(F, L)$  igual a  $(1, 1)$ ,  $(5, 1)$  e  $(10, 1)$  para cada GAD de entrada, caracterizando GADs de granularidade grossa, e  $(1, 5)$  e  $(1, 10)$ , caracterizando GADs de granularidade fina.

#### Árvores Binárias Invertidas

Foram analisados grafos do tipo árvore binária invertida (ou *in-tree*) com 15, 31, 63 e 127 tarefas, para os cinco pares de parâmetros  $(F, L)$  e todas as variações de ambientes duais e aleatórios enumeradas anteriormente, num total de 280 casos. A Figura 5.1 mostra os percentuais de todos os resultados melhores ( $<$ ), iguais ( $=$ ) ou piores ( $>$ ) obtidos por CPH, quando comparados com HEFT e LDBS. Observa-se que em relação ao HEFT, CPH produz escalonamento melhor ou igual em 93% dos casos. Em comparação ao LDBS, CPH tem sucesso ou empata em 91% dos casos. A Tabela 5.1 apresenta a vantagem percentual dos resultados obtidos por CPH, em relação ao *makespan*, quando comparados a HEFT e LDBS, para cada par de

parâmetros  $(F, L)$ , para os ambientes dual e aleatório.



**Figura 5.1:** Uma comparação dos escalonamentos gerados por CPH com HEFT e LDBS, em relação aos makespans, em termos de resultados piores ( $>$ ), iguais ( $=$ ) ou melhores ( $<$ ) para GADs *árvore binária invertida*.

O resultado do escalonamento fornecido por CPH é, em média, 27,6% melhor que o do HEFT e 9,6% melhor que aqueles gerados por LDBS para os ambientes duais, e 24,7% e 6,9% melhor para os ambientes aleatórios. Observa-se, entretanto, que há uma grande variação deste percentual com a granularidade do grafo de entrada.

| (F, L)  | Dual       |            | Aleatório  |            |
|---------|------------|------------|------------|------------|
|         | CPH x HEFT | CPH x LDBS | CPH x HEFT | CPH x LDBS |
| (10, 1) | 0.04%      | 0.04%      | -0.69%     | -0.87%     |
| (5, 1)  | -0.09%     | -0.09%     | 0.28%      | -0.10%     |
| (1, 1)  | 22.28%     | 0.98%      | 11.35%     | 1.65%      |
| (1, 5)  | 49.67%     | 10.79%     | 49.00%     | 7.51%      |
| (1, 10) | 66.30%     | 36.44%     | 63.47%     | 26.24%     |
| Média   | 27.64%     | 9.63%      | 24.68%     | 6.89%      |

**Tabela 5.1:** Comparação percentual entre os resultados de escalonamento de todos os GADs *árvore binária invertida* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1), (5, 1), (1, 1), (1, 5), (1, 10)$ .

Em ambientes duais, para os pares de parâmetros  $(10, 1)$  e  $(5, 1)$  (granularidade grossa), o CPH produz, em média, escalonamentos com *makespan* bastante próximos a HEFT e o LDBS. Para  $(1, 1)$  o desempenho do CPH é, em média, ligeiramente superior ao LDBS (1,0%), e bem melhor que o HEFT (22,3%). Observando-se cada instância detalhadamente, verifica-se que na maioria dos casos CPH produz escalonamento equivalente a HEFT e LDBS. A exceção é geralmente em ambientes com

10% de processadores mais rápidos, situação em que o algoritmo proposto não consegue recombina as coleções fornecidas pelo primeiro estágio de forma a produzir um escalonamento eficiente capaz de tirar proveito do número limitado de processadores mais rápidos. Isso se deve provavelmente ao fato de que essas coleções de tarefas geradas no primeiro estágio de CPH são inapropriadas para um ambiente com um número tão pequeno de processadores mais rápidos. A Figura 5.2 mostra um exemplo desta situação. Em (b) podem ser vistas as as coleções formadas pelo primeiro estágio para a árvore binária invertida de 15 tarefas em (a), com  $(F, L) = (1, 1)$ , para o ambiente dual com 2 processadores mais rápidos. O melhor mapeamento possível para esse conjunto de coleções, encontrado pelo segundo estágio e demonstrado em (c), é 11,1% pior que o resultado fornecido por LDBS, visto em (d), que consegue um escalonamento com menor *makespan* nos processadores mais rápidos.

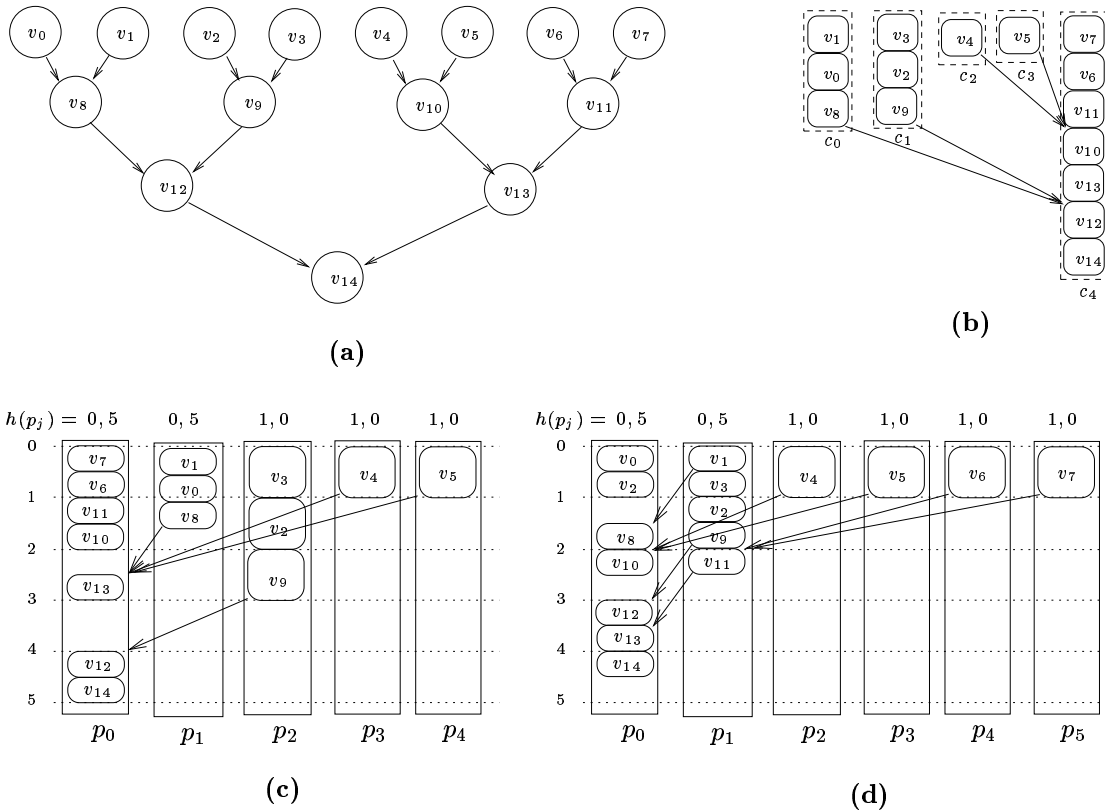


Figura 5.2: (a) Árvore binária invertida de 15 tarefas, (b) escalonamento produzido por HAL para  $(F, L) = (1, 1)$ , (c) mapeamento produzido por AMC, (d) escalonamento produzido por LDBS.

No entanto, para este exemplo e para vários outros experimentos analisados

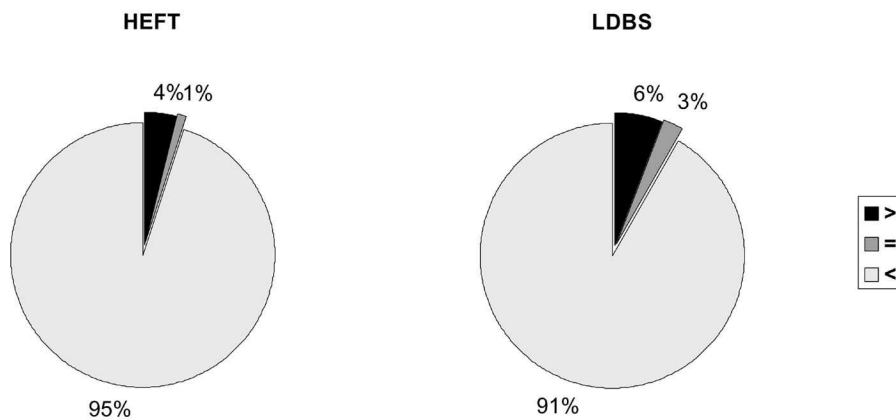
(mas que obviamente não podem ser apresentados em detalhe), LDBS escalona as tarefas se preocupando unicamente em minimizar o *makespan*, sem observar se as tarefas adjacentes estão sendo alocadas para um mesmo processador, especificando um escalonamento em que um número excessivo de mensagens são trocadas entre os processadores. Esse aspecto é levado em consideração pela metodologia de aglomeração implementada em HAL, levando à redução deste número de mensagens no escalonamento resultante de CPH.

Por outro lado, para os pares (1, 5) e (1, 10) (granularidade fina), o resultado do escalonamento de CPH é muito melhor que o do HEFT e LDBS. O escalonamento para GADs do tipo árvore binária invertida, a princípio, não se beneficia com a replicação de tarefas, já que cada tarefa tem apenas um sucessor imediato. Apesar disso, a técnica de replicação traz vantagens tanto para CPH quanto LDBS (a metodologia traz uma vantagem maior). A replicação permite ao algoritmo LDBS desfazer decisões ruins de escalonamento feitas em iterações anteriores e aglomerar subárvores de forma a minimizar a comunicação. Já para CPH, o bom resultado para granularidade fina é obtido através de uma boa aglomeração de tarefas pelo primeiro estágio, que forma coleções que minimizam de modo eficiente a comunicação, e através de uma boa escolha de processadores, realizada pelo segundo estágio, processadores essas capazes de executar mais rapidamente o maior número de coleções (e tarefas). Além disso, os algoritmos HEFT e LDBS não produzem bons resultados para grafos deste tipo com granularidade fina, pois são heurísticas muito gulosas.

Nos ambientes aleatórios o comportamento dos resultados é semelhante ao observado para os ambientes duais, com o CPH desempenhando-se sempre melhor que HEFT e LDBS para os grafos de granularidade fina ( $(F, L) = (1, 5), (1, 10)$ ). Por outro lado, para (10, 1) e (5, 1), CPH é inferior tanto ao HEFT quanto ao LDBS. Observando-se cada instância detalhadamente, nota-se que para os grafos de granularidade grossa, o escalonamento produzido por CPH piora com o aumento do desvio padrão do ambiente, isto é, com o aumento da diferença de velocidades entre os processadores do ambiente alvo. O algoritmo proposto demonstra dificuldade em mapear as coleções originais em um número muito menor de processadores mais rápido de forma eficiente.

### Árvores Binárias

Foram analisados grafos do tipo árvore binária (ou *out-tree*) com 15, 31, 63 e 127 tarefas, para os cinco pares de parâmetros  $(F, L)$ , como já reportado, e todos os ambientes duais e aleatórios, num total de 280 casos. A Figura 5.3 mostra os percentuais de todos os resultados melhores, iguais ou piores obtidos por CPH, quando comparados com HEFT e LDBS. Observa-se que em relação ao HEFT, CPH nunca é pior em 96% dos casos. Em comparação ao LDBS, CPH é tão ou mais bem sucedido em 94% do número total de casos.

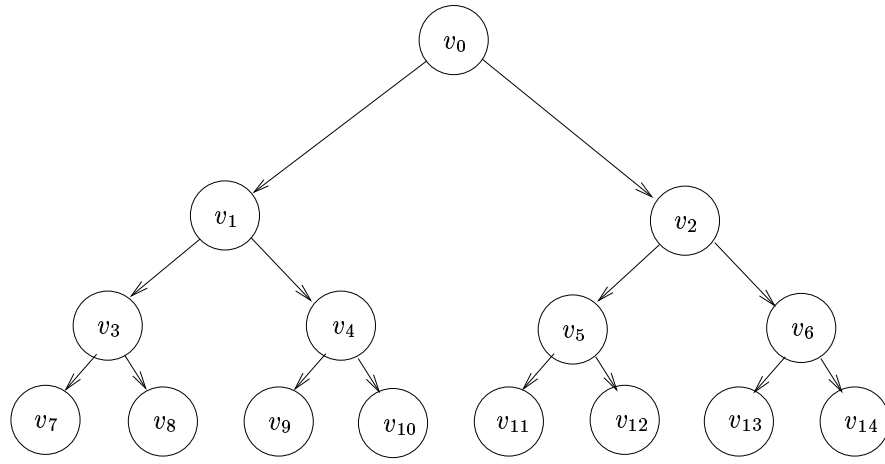


**Figura 5.3:** Uma comparação dos escalonamentos gerados por CPH com HEFT e LDBS, em relação aos makespans, em termos de resultados piores ( $>$ ), iguais ( $=$ ) ou melhores ( $<$ ) para GADs *árvore binária*.

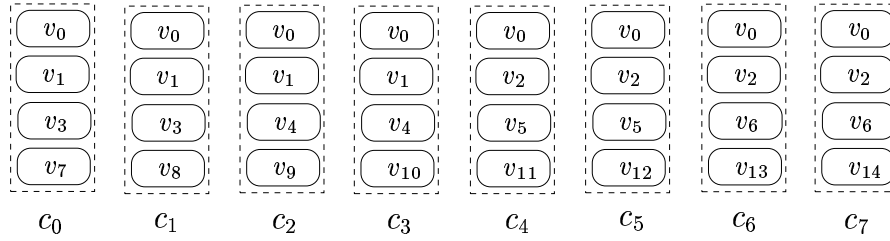
Já foi demonstrado que o primeiro estágio do CPH produz escalonamentos ótimos para GADs do tipo árvore binária em processadores homogêneos [8]. As tarefas em cada caminho da árvore são aglomeradas conjuntamente, tornando desnecessária qualquer comunicação entre as coleções de tarefas. A Figura 5.4 (b) mostra as coleções formadas pelo primeiro estágio para a árvore binária de 15 tarefas em (a), com  $(F, L) = (1, 1)$ .

A cada execução do procedimento *Seleciona Coleção* do segundo estágio de CPH, todas as coleções ainda não mapeadas têm os mesmos valores de  $blevel_e$ . Portanto, CPH usa a *discrepância*  $\delta(c)$  para definir a prioridade para selecionar coleções  $c$  que têm menos tarefas em comum com aquelas já mapeadas. Inicialmente, essas coleções serão alocadas nos processadores mais rápidos. Quando todos os processadores rápi-





(a)



(b)

**Figura 5.4:** (a) Árvore binária 15 tarefas, (b) escalonamento produzido por HAL para  $(F, L) = (1, 1)$ .

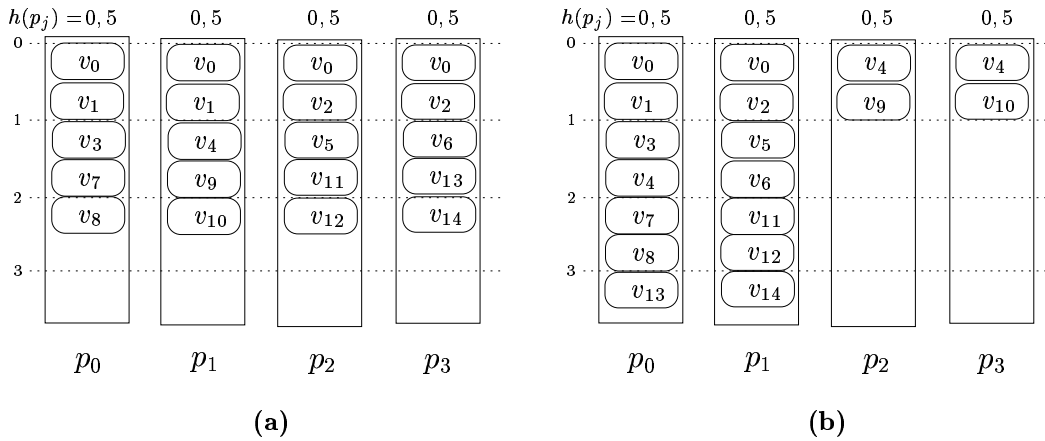
dos já tiverem recebido coleções, o algoritmo tenta minimizar o *makespan* mapeando a próxima coleção ou em um processador lento desocupado ou em um processador rápido com o qual a coleção tenha maior afinidade (tarefas em comum), de acordo com o balanceamento de carga definido nas Equações 4.6 e 4.7.

A Tabela 5.2 apresenta a vantagem percentual média, em relação ao *makespan*, dos resultados obtidos por CPH quando comparado ao HEFT e LDBS, para cada par de parâmetros  $(F, L)$ , para os ambientes dual e aleatório. Para árvores binárias com todos os pares de parâmetros, CPH produz escalonamentos que em média nunca são piores ao HEFT e LDBS, tanto para os ambientes duais quanto aleatórios. Para  $(10, 1)$ ,  $(5, 1)$  e  $(1, 1)$  (granularidade grossa) CPH é bem melhor que o HEFT e o LDBS. A vantagem aumenta ainda mais para  $(1, 5)$  e  $(1, 10)$  (granularidade fina), o que já

|         | Dual       |            | Aleatório  |            |
|---------|------------|------------|------------|------------|
| (F, L)  | CPH x HEFT | CPH x LDBS | CPH x HEFT | CPH x LDBS |
| (10, 1) | 21.10%     | 16.20%     | 18.59%     | 15.19%     |
| (5, 1)  | 29.06%     | 19.93%     | 23.77%     | 17.31%     |
| (1, 1)  | 58.99%     | 41.10%     | 51.41%     | 34.16%     |
| (1, 5)  | 80.19%     | 64.72%     | 77.54%     | 60.08%     |
| (1, 10) | 83.42%     | 68.74%     | 82.33%     | 67.21%     |
| Média   | 54.55%     | 42.14%     | 50.73%     | 38.79%     |

**Tabela 5.2:** Comparação percentual entre os resultados de escalonamento de todos os GADs *árvore binária* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1), (5, 1), (1, 1), (1, 5), (1, 10)$ .

era esperado, pois o primeiro estágio eliminou a comunicação e o segundo estágio, apesar dos vários empates de prioridade, faz um bom mapeamento agrupando um maior número de tarefas nos processadores mais rápidos.

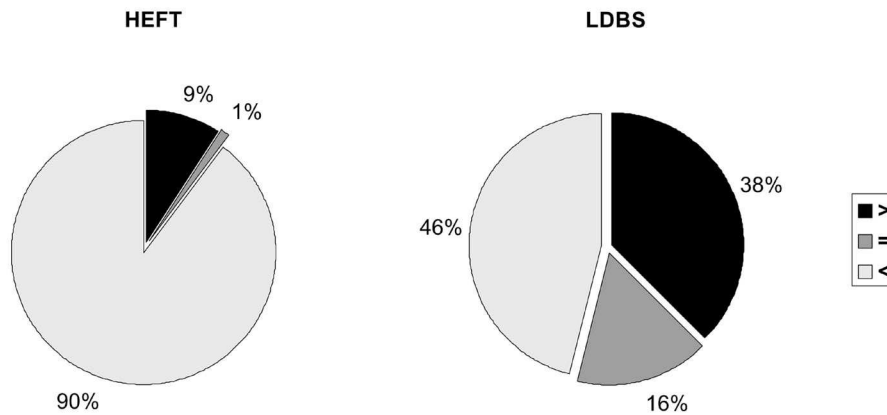


**Figura 5.5:** (a) Escalonamento CPH para a árvore binária da figura anterior e (b) resultado produzido por LDBS.

Conforme demonstrado na Figura 5.5 (a), o segundo estágio foi capaz de combinar as oito coleções produzidas pelo primeiro estágio (Figura 5.4) para mapeá-las nos processadores mais rápidos de um ambiente dual com apenas quatro processadores mais velozes, obtendo um resultado ótimo. Em (b) observa-se que o escalonamento de LDBS não foi tão eficiente. A média final resulta em 54,6% melhor que o HEFT e 42,1% melhor que o LDBS, em ambientes duais, e 50,7% e 38,8%, em ambientes aleatórios.

### Diamantes

Foram analisados grafos do tipo diamante com 64, 100, 144 e 256 tarefas, para os cinco pares de parâmetros  $(F, L)$  e todas as variações de ambientes duais e aleatórios enumeradas anteriormente, totalizando 280 casos. A Figura 5.6 mostra os percentuais de todos os resultados piores, iguais e melhores obtidos por CPH, quando comparados com HEFT e LDBS. Observa-se que em relação ao HEFT, CPH é melhor ou equivalente em 94% dos casos. Em comparação ao LDBS, CPH tem desempenho superior ou igual em 63% do número total de casos.



**Figura 5.6:** Uma comparação dos escalonamentos gerados por CPH com HEFT e LDBS, em relação aos makespans, em termos de resultados piores ( $>$ ), iguais ( $=$ ) ou melhores ( $<$ ) para GADs *diamante*.

No caso de grafos diamante com granularidade grossa, o primeiro estágio produz coleções lineares [10, 32], isto é, coleções sem tarefas independentes, e o grau de replicação é bastante elevado, geralmente maior que dos escalonamentos produzidos pelo LDBS. Isso se deve ao fato de que as coleções resultantes são construídas considerando-se um número ilimitado de processadores, o que acarreta um número de coleções também grande. A consequência disso é que CPH termina produzindo, em média, escalonamentos que utilizam muito mais processadores que o LDBS e o HEFT. Em ambientes duais com poucos processadores rápidos ou em ambientes randômicos com grande variação de velocidade entre os processadores disponíveis, o mapeamento do segundo estágio tende a reduzir muito o número de coleções finais com um certo prejuízo em relação ao *makespan* resultante.

| (F, L)  | Dual       |            | Aleatório  |            |
|---------|------------|------------|------------|------------|
|         | CPH x HEFT | CPH x LDBS | CPH x HEFT | CPH x LDBS |
| (10, 1) | 0.50%      | -2.77%     | 0.17%      | -1.30%     |
| (5, 1)  | 6.21%      | -1.03%     | 4.08%      | -0.93%     |
| (1, 1)  | 16.51%     | -11.22%    | 16.60%     | -1.52%     |
| (1, 5)  | 29.13%     | 4.86%      | 37.18%     | 11.23%     |
| (1, 10) | 18.95%     | 11.75%     | 35.04%     | 20.96%     |
| Média   | 14.26%     | 0.32%      | 18.62%     | 5.69%      |

**Tabela 5.3: Comparação percentual entre os resultados de escalonamento de todos os GADs *diamante* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1), (5, 1), (1, 1), (1, 5), (1, 10)$ .**

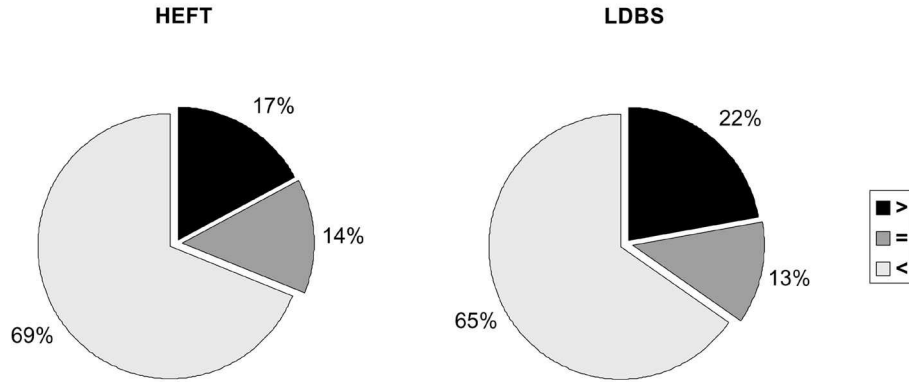
Por outro lado, para GADs de granularidade fina, a replicação esconde os altos custos de comunicação e beneficia os escalonamentos produzidos por CPH. Esse fato pode ser verificado na Tabela 5.3, que apresenta a vantagem percentual média dos resultados obtidos pelo CPH comparados ao HEFT e LDBS, para cada par de parâmetros  $(F, L)$ , para os ambientes dual e aleatório.

Observa-se que para granularidade fina, com os parâmetros  $(1, 5)$  e  $(1, 10)$ , tanto nos ambientes duais quanto aleatórios, o CPH tem, em média, desempenho melhor que HEFT e LDBS. Para  $(10, 1)$ ,  $(5, 1)$  e  $(1, 1)$ , CPH perde para LDBS mas ainda é melhor que HEFT tanto em ambientes duais quanto aleatórios. O problema, mais uma vez, acontece para ambientes têm 10%, 20% e até 30% de processadores mais rápidos, denotando a dificuldade do segundo estágio em reduzir significativamente o número de coleções finais do escalonamento. Ainda assim, na média o CPH apresenta resultados mais vantajosos 14,3% que HEFT e 0,3% que LDBS, em ambientes duais, e 18,6% e 5,7%, respectivamente, em ambientes randômicos.

### GADs Randômicos

Foram analisados quatro grafos gerados aleatoriamente [5] com 124, 140, 152 e 153 tarefas, respectivamente, para os cinco pares de parâmetros  $(F, L)$  e todas as variações de ambientes duais e aleatórios enumeradas anteriormente, totalizando 280 casos. A Figura 5.7 mostra os percentuais de todos os resultados piores, iguais e melhores obtidos por CPH, quando comparados com HEFT e LDBS. Observa-se

que em relação ao HEFT, CPH foi melhor ou equivalente em 83% dos casos. Em comparação ao LDBS, CPH foi tão ou mais vantajoso em 78% do número total de casos.



**Figura 5.7:** Uma comparação dos escalonamentos gerados por CPH com HEFT e LDBS, em relação aos makespans, em termos de resultados piores ( $>$ ), iguais ( $=$ ) ou melhores ( $<$ ) para GADs *randômicos*.

De forma semelhante aos GADs diamante, para os grafos randômicos, o primeiro estágio produz um grande número de coleções com elevado grau de replicação. E como é de se esperar, essa aglomeração é bastante eficiente para os grafos de granularidade fina, reduzindo a comunicação entre as coleções resultantes. Já para os grafos de granularidade grossa, o segundo estágio tem dificuldade em mapear essas coleções em um número pequeno de processadores mais rápidos.

|         | Dual       |            | Aleatório  |            |
|---------|------------|------------|------------|------------|
| (F, L)  | CPH x HEFT | CPH x LDBS | CPH x HEFT | CPH x LDBS |
| (10, 1) | -9.55%     | -6.01%     | -4.47%     | -4.37%     |
| (5, 1)  | -7.20%     | -6.14%     | -4.44%     | -4.57%     |
| (1, 1)  | 20.61%     | 3.55%      | 15.65%     | 3.82%      |
| (1, 5)  | 33.11%     | 10.42%     | 35.64%     | 14.25%     |
| (1, 10) | 44.87%     | 17.03%     | 44.48%     | 20.04%     |
| Média   | 16.37%     | 3.77%      | 17.37%     | 5.83%      |

**Tabela 5.4:** Comparação percentual entre os resultados de escalonamento de todos os GADs *randômicos* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1), (5, 1), (1, 1), (1, 5), (1, 10)$ .

A Tabela 5.4 apresenta a vantagem percentual média dos resultados obtidos pelo CPH quando comparado ao HEFT e LDBS, para cada par de parâmetros  $(F,$

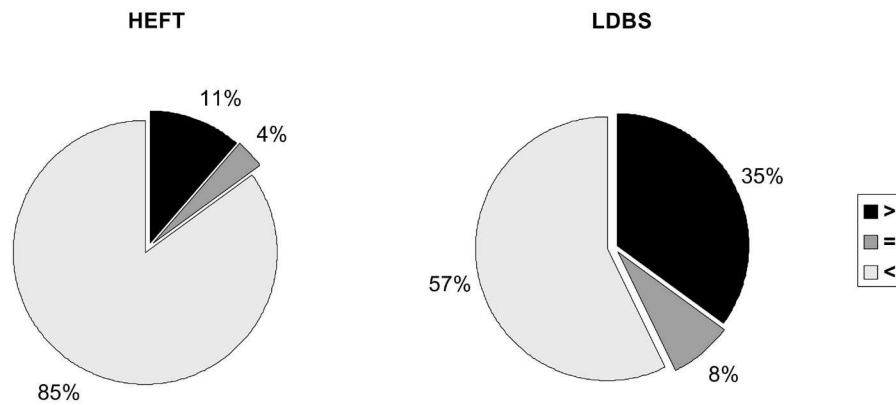
$L$ ), para os ambientes dual e aleatório. Para os pares de parâmetros  $(1, 5)$  e  $(1, 10)$ , CPH produz escalonamentos, em média, melhores que o HEFT e LDBS, tanto para os ambientes duais quanto aleatórios. Para o par  $(1, 1)$  o CPH ainda produz, no maior número de casos, escalonamentos melhores que os outros algoritmos com apenas alguns casos de derrota, para ambiente com 10% de processadores ou 0,5 de desvio padrão. Por conta disso, sua vantagem, média, em relação ao HEFT e LDBS se reduz. Para  $(10, 1)$  e  $(5, 1)$ , CPH se torna, em média, pior que as outras heurísticas. No geral, o CPH é 16,4% melhor que o HEFT e 3,4% melhor que o LDBS, em ambientes duais, e 17,4% e 5,8% melhor que o LDBS.

### 5.2.2 Grafos representantes de aplicações reais

Para analisar o desempenho de CPH perante aplicações reais, foram utilizados uma classe de grafos de diversos tamanhos representando o cálculo da eliminação de Gauss, conforme modelado em [18], e o grafo representante de um problema de dinâmica molecular da física [32]. A eliminação de Gauss é um algoritmo de decomposição triangular de matrizes, que é aplicado na solução de um sistema de  $n$  equações e  $n$  incógnitas, e requer  $O(n)$  operações aritméticas. O algoritmo sequencial pode ser paralelizado resultando em um GAD com  $\frac{(n^2+n)}{2} - 1$  tarefas. Um exemplo pode ser visto na Figura 5.9 (a), onde  $n = 7$ , e, conseqüentemente, o número de tarefas é 27. Os números ao lado dos nós representam pesos de computação e os números ao lado dos arcos representam pesos de comunicação.

#### Eliminação de Gauss

Os fatores  $F$  e  $L$  também foram utilizados para variar a granularidade dos GADs Gauss. No entanto, como esses GADs são de granularidade inerentemente grossa, os pares de parâmetros  $(F, L)$  para cada GAD de entrada foram variados apenas entre  $(1, 1)$ ,  $(1, 5)$  e  $(1, 10)$ . Para cada um desses pares e todas as variações de ambientes duais e aleatórios enumeradas anteriormente, foram analisados grafos do tipo gauss com 27, 35, 44 e 54 tarefas, num total de 168 casos.



**Figura 5.8:** Uma comparação dos escalonamentos gerados por CPH com HEFT e LDBS, em relação aos makespans, em termos de resultados piores ( $>$ ), iguais ( $=$ ) ou melhores ( $<$ ) para GADs *eliminação de gauss*.

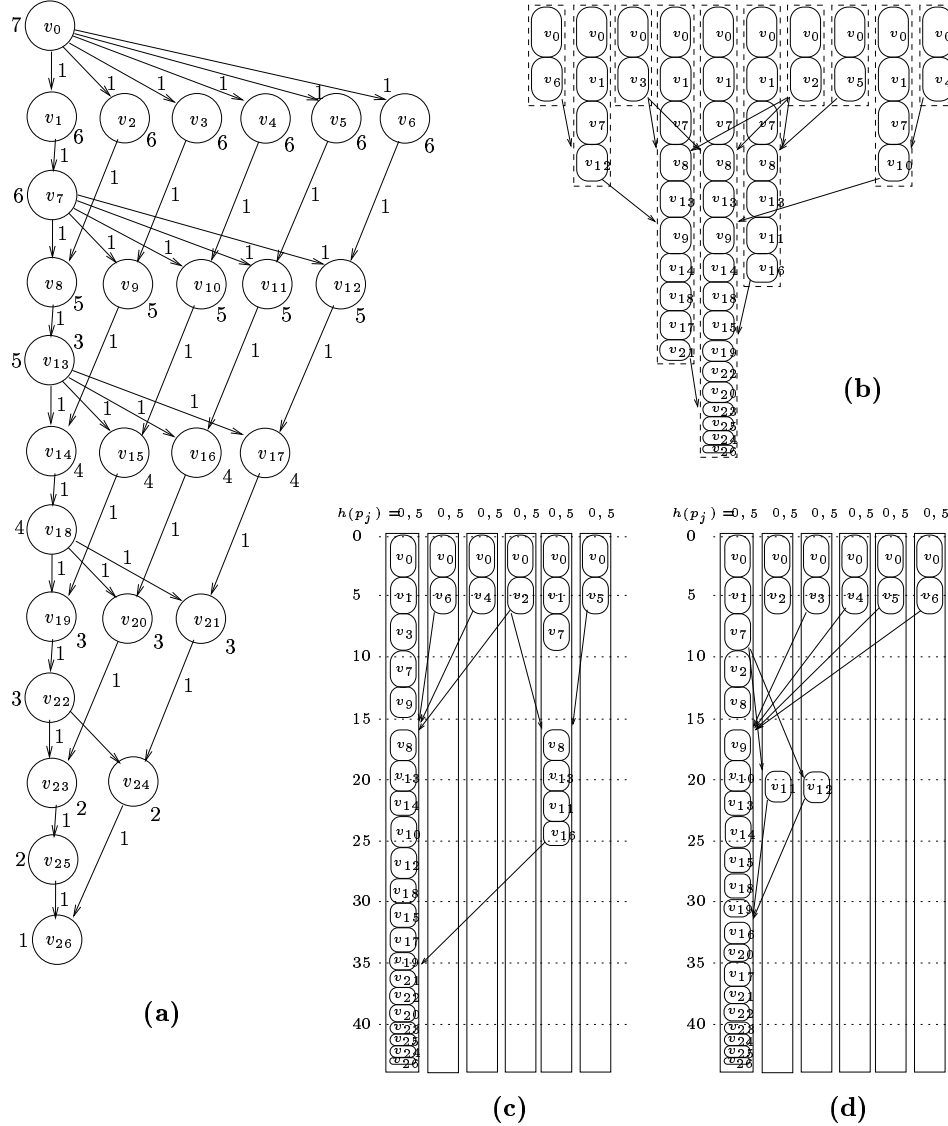
A Figura 5.8, através da contabilidade de todos os resultados melhores, iguais ou piores obtidos por CPH, quando comparados com HEFT e LDBS, mostra o benefício de executar tal aplicação de acordo com o escalonamento produzido por CPH. Observa-se que em relação ao HEFT, CPH foi melhor ou igual em 89% dos casos. CPH foi melhor ou igual a LDBS em 65% do número total de casos.

|         | Dual       |            | Aleatório  |            |
|---------|------------|------------|------------|------------|
| (F, L)  | CPH x HEFT | CPH x LDBS | CPH x HEFT | CPH x LDBS |
| (1, 1)  | -1.19%     | 0.18%      | -0.27%     | -0.27%     |
| (1, 5)  | 15.84%     | 2.72%      | 13.21%     | 7.09%      |
| (1, 10) | 14.20%     | -3.35%     | 18.02%     | 5.62%      |
| Média   | 9.62%      | -0.15%     | 10.32%     | 4.15%      |

**Tabela 5.5:** Comparação percentual entre os resultados de escalonamento de todos os GADs *eliminação de gauss* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1), (5, 1), (1, 1), (1, 5), (1, 10)$ .

Ao analisar os resultados em relação às granularidades especificadas, o comportamento é semelhante ao já observado nas outras classes de GAD. A Tabela 5.5 apresenta a vantagem percentual média dos resultados obtidos pelo CPH quando comparado ao HEFT e LDBS, para cada par de parâmetros  $(F, L)$ , para os ambientes dual e aleatório. Em média, o desempenho de CPH é bastante próximo ao de HEFT e LDBS para  $(1, 1)$ . Para o ambiente dual o CPH é pior que o HEFT em 1,2%, e melhor 0,2% que LDBS. Para o ambiente aleatório, CPH é 0,3% pior

se comparado a ambos os algoritmos. Para (1, 5), em média, o resultado é 15,8% melhor que o HEFT e 2,7% melhor que o LDBS, para ambientes duais, e com 13,2% e 7,1%, para ambientes aleatórios. Finalmente, para (1, 10) CPH produz resultados melhores que HEFT em 14,2% em ambientes duais e 18,0% melhor que o HEFT e 5,6% melhor que o LDBS para ambientes aleatórios.



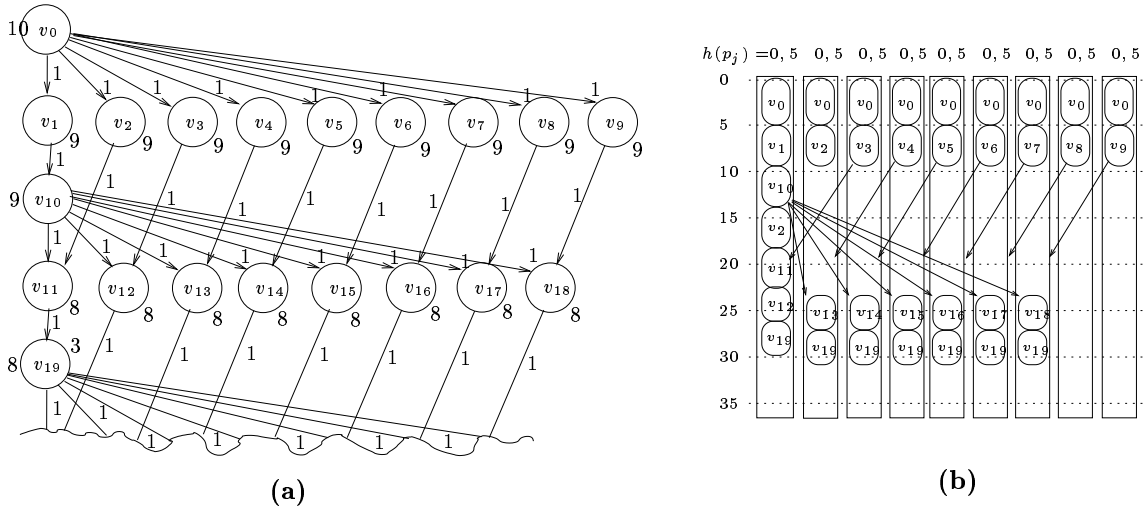
**Figura 5.9:** (a) GAD que representa o código da eliminação de gauss de 27 tarefas, (b) o escalonamento HAL correspondente para  $(F, L) = (1, 10)$ , (c) mapeamento resultante do segundo estágio e (d) escalonamento obtido por LDBS.

Como exceção, LDBS é melhor 3,4% que CPH em ambientes duais. Isto acontece porque, devido às características do GAD, LDBS obtém um escalonamento



mais eficiente quando o custo de comunicação e o número de tarefas aumentam. A Figura 5.9(b) mostra o conjunto de coleções produzidas pelo primeiro estágio para o GAD em (a) para  $(F, L) = (1, 10)$ , em um ambiente dual com 6 processadores mais rápidos. Em (c) é apresentado o resultado do mapeamento do segundo estágio e em (d) pode ser visto o escalonamento fornecido por LDBS para o mesmo GAD. Ambos obtêm  $makespan = 44$ , mas observa-se que LDBS, como é típico de um algoritmo do tipo *list scheduling*, não procura agrupar caminhos nos processadores, mas escalona o GAD eficientemente fazendo replicação de tarefas nos intervalos criados pela comunicação de dados ( $v_2$  em  $p_0$ ).

A Figura 5.10(a) mostra um trecho do GAD que representa o código da eliminação de gauss para  $n = 10$  (54 tarefas). Em (b) pode ser visto o escalonamento parcial de LDBS para as tarefas de menor nível do GAD, com  $(F, L) = (1, 10)$ , em um ambiente dual com 10 processadores mais rápidos. Observa-se que LDBS, apesar de ser uma heurística gulosa, escalona eficientemente a aplicação em um número de processadores nunca maior que a largura do grafo. O processador  $p_0$  recebe o caminho crítico do grafo ( $\langle v_0, v_1, v_{10}, v_{11}, v_{19} \rangle$ ), juntamente com cópias de tarefas predecessoras das tarefas  $v \in CP$  que minimizam o custo de comunicação ( $v_2$  e  $v_{12}$ ).



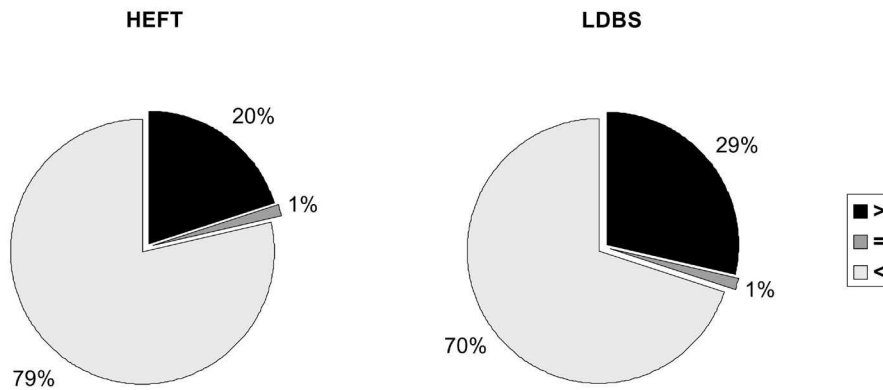
**Figura 5.10:** (a) Trecho do GAD que representa o código da eliminação de Gauss de 54 tarefas e (b) escalonamento parcial obtido por LDBS.

A replicação das tarefas que têm muitos sucessores ( $v_0$  e  $v_{19}$ ) diminui o custo de comunicação para os outros processadores. Mesmo assim, o algoritmo LDBS não

se preocupa em diminuir o número de mensagens quando o custo de comunicação não atrasa a execução das tarefas sucessoras (mensagens enviadas por  $v_3$ ,  $v_4$ ,  $v_5$ ,  $v_6$ ,  $v_7$ ,  $v_8$  e  $v_9$ ). O *makespan* resultante é 83, em 9 processadores. Para o mesmo ambiente dual com 10 processadores mais rápidos e  $(F, L) = (1, 10)$ , o primeiro estágio de CPH produz 30 coleções e o segundo estágio usa 11 processadores para o mapeamento dessas coleções, obtendo um *makespan* 13,9% pior. Na média geral, o CPH é melhor 12,5% que o HEFT, em ambientes duais, e 9,7% em ambientes aleatórios, e 3,5% melhor que o LDBS em ambos.

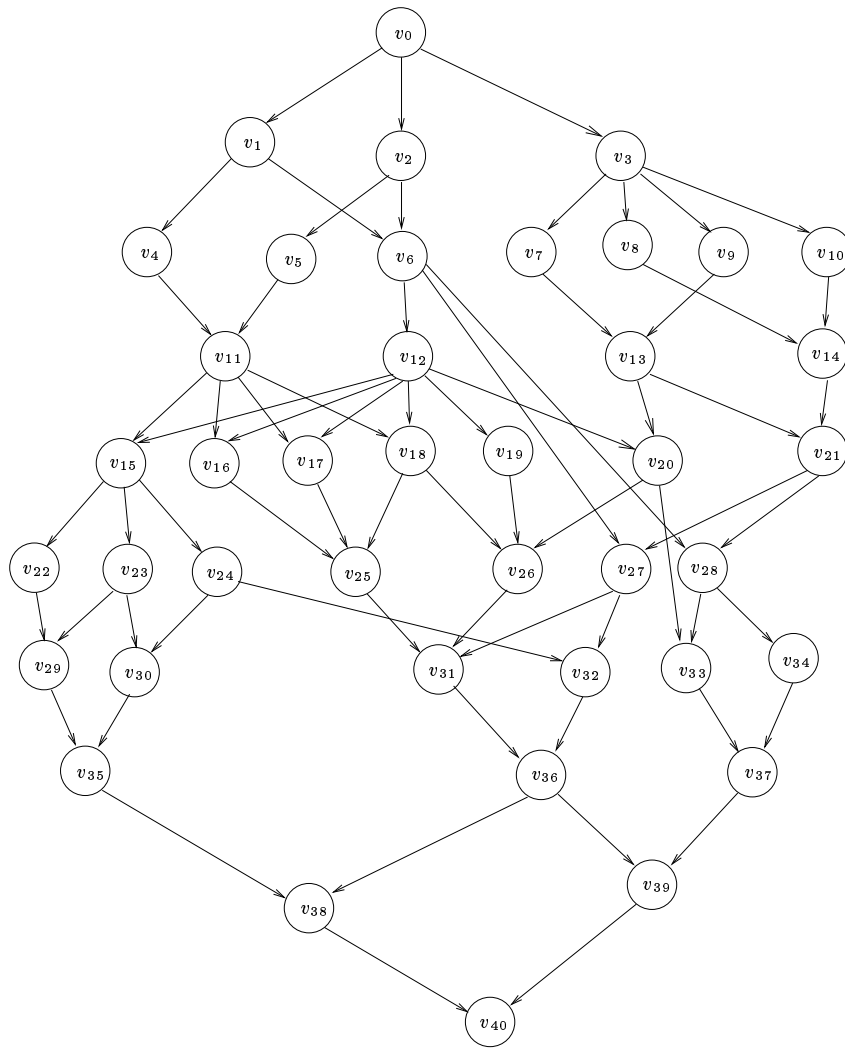
### Dinâmica Molecular

O código de dinâmica molecular pode ser representado pelo grafo unitário do tipo UET-UDT com 41 tarefas, conforme pode ser visto na Figura 5.12. Assim os valores para o par de parâmetros  $(F, L)$  utilizados foram, mais uma vez,  $(1, 1)$ ,  $(5, 1)$ ,  $(10, 1)$ ,  $(1, 5)$  e  $(1, 10)$ , para todas as variações de ambientes duais e aleatórios enumeradas anteriormente, totalizando 70 casos.



**Figura 5.11:** Uma comparação dos escalonamentos gerados por CPH com HEFT e LDBS, em relação aos *makespans*, em termos de resultados piores (>), iguais (=) ou melhores (<) para GADs *dinâmica molecular*.

A Figura 5.11 mostra os percentuais de todos os resultados obtidos que são favoráveis, indiferentes ou desfavoráveis ao CPH, comparados ao HEFT e LDBS. Observa-se que em relação ao HEFT, CPH foi melhor ou igual em 80% dos casos. Em comparação ao LDBS, CPH foi melhor ou igual em 71% das vezes.



**Figura 5.12: GAD UET-UDT com 41 tarefas que representa o código de um problema de dinâmica molecular da física.**

A Tabela 5.6 resume a vantagem percentual média dos resultados obtidos pelo CPH quando comparado ao HEFT e LDBS, para cada par de parâmetros  $(F, L)$ , para os ambientes dual e aleatório. Para este grafo e os pares de parâmetros  $(1, 5)$  e  $(1, 10)$ , CPH produziu escalonamentos, em média, melhores que o HEFT e LDBS, tanto para os ambientes duais quanto aleatórios.

Para o par  $(1, 1)$  o CPH ainda produz escalonamentos melhores que o HEFT, mas é praticamente equivalente a LDBS. Para  $(5, 1)$ , CPH ainda é melhor que o HEFT apenas em ambientes duais. De resto perde para as outras heurísticas. Para  $(10, 1)$  CPH é pior, em média, que todos os outros, principalmente nos ambientes randômicos Na média geral, o CPH é 11,0% melhor que o HEFT e 3,9% melhor que

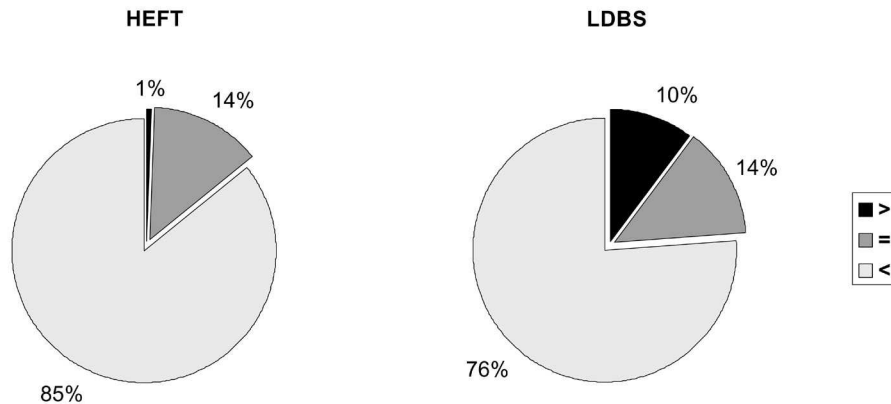
|         | Dual       |            | Aleatório  |            |
|---------|------------|------------|------------|------------|
| (F, L)  | CPH x HEFT | CPH x LDBS | CPH x HEFT | CPH x LDBS |
| (10, 1) | -3.12%     | -3.29%     | -6.38%     | -7.97%     |
| (5, 1)  | 1.52%      | -1.33%     | -1.44%     | -3.83%     |
| (1, 1)  | 19.67%     | 0.18%      | 15.24%     | 1.75%      |
| (1, 5)  | 27.48%     | 13.56%     | 36.80%     | 12.44%     |
| (1, 10) | 9.53%      | 10.48%     | 23.77%     | 14.32%     |
| Média   | 11.02%     | 3.92%      | 13.60%     | 3.34%      |

**Tabela 5.6:** Comparação percentual entre os resultados de escalonamento de todos os GADs *dinâmica molecular* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1), (5, 1), (1, 1), (1, 5), (1, 10)$ .

o LDBS, em ambientes duais, e 13,6% e 3,3% melhor que o LDBS, em ambientes aleatórios.

### 5.2.3 Grafos PSG

Os grafos PSG são um conjunto de 11 grafos não unitários e de topologia irregular retirados da literatura e pertencentes ao *benchmark* de Kwok [34]. Nesses experimentos, sua granularidade não foi variada através dos parâmetros  $(F, L)$ , que foram mantidos como  $(1, 1)$  para todos os grafos, preservando a granularidade inerente de cada um. Os escalonamentos foram executados para todas as variações de ambientes duais e aleatórios, num total de 154 casos.



**Figura 5.13:** Uma comparação dos escalonamentos gerados por CPH com HEFT e LDBS, em relação aos makespans, em termos de resultados piores ( $>$ ), iguais ( $=$ ) ou melhores ( $<$ ) para GADs *PSG*.

A Figura 5.13 mostra os percentuais de todos os resultados obtidos que são piores, iguais e melhores obtidos por CPH, comparados a HEFT e LDBS. Observa-se que em relação a HEFT, CPH foi melhor ou igual em 99% dos casos. CPH foi melhor ou igual a LDBS em 90% do número total de casos.

De um modo geral, o CPH também produziu bons escalonamentos para os grafos irregulares em PSG, conforme pode ser verificado na Tabela 5.7, que apresenta a granularidade e a vantagem percentual média dos resultados de escalonamento obtidos pelo CPH quando comparado ao HEFT e LDBS, para cada grafo do conjunto PSG, considerando todas as variações de ambiente para o caso dual e aleatório. Na média geral, o escalonamento produzido por CPH foi 22,0% melhor que o HEFT e 15,4% melhor que o LDBS para ambientes duais, e 20,2% e 14,8% para ambientes aleatórios.

|                  |       | Dual       |            | Aleatório  |            |
|------------------|-------|------------|------------|------------|------------|
| GAD              | ccr   | CPH x HEFT | CPH x LDBS | CPH x HEFT | CPH x LDBS |
| al-maasarani     | 1.098 | 14.80%     | 5.60%      | 13.46%     | 10.19%     |
| al-mouhamed      | 0.632 | 22.84%     | 7.81%      | 17.11%     | 6.68%      |
| g4               | 1.552 | 18.59%     | -0.35%     | 20.44%     | 10.20%     |
| kruatrachue-fig8 | 4.600 | 0.83%      | 36.87%     | 3.40%      | 22.44%     |
| lctd-eg1         | 2.821 | 34.14%     | 34.53%     | 24.71%     | 32.81%     |
| lwb              | 0.572 | 5.28%      | -1.14%     | 11.31%     | -0.20%     |
| mccreary1        | 3.198 | 18.11%     | 5.86%      | 24.74%     | 6.35%      |
| pbsa-ipps95      | 4.516 | 63.88%     | 42.92%     | 56.26%     | 37.78%     |
| ranka            | 2.391 | 28.89%     | 29.24%     | 20.50%     | 27.63%     |
| tyang1           | 1.192 | 0.76%      | 0.00%      | 0.10%      | -0.60%     |
| tyang4           | 1.279 | 33.33%     | 7.81%      | 30.30%     | 9.99%      |
| Média            | —     | 21.95%     | 15.38%     | 20.21%     | 14.84%     |

**Tabela 5.7:** Comparação percentual entre os resultados de escalonamento de cada GAD *PSG* obtidos pelos algoritmos HEFT, LDBS e CPH para  $(F, L) = (1, 1)$ .

Para a maior parte dos grafos, CPH produziu resultados, em média, melhores que os demais. As exceções no ambiente dual foram para os grafos “g4” e “lwb”. Para o ambiente aleatório, tanto o “lwb” como o “tyang1”, GADs de granularidade grossa, tiveram alguns resultados de CPH piores que LDBS em ambientes com elevado desvio-padrão. Esses resultados se devem à própria característica das coleções formadas pelo primeiro estágio. Para que CPH pudesse obter melhores resultados nesses casos, aproveitando a heterogeneidade do ambiente e alocando mais tarefas

nos processadores mais rápidos, seria necessário que fosse capaz de retirar tarefas de certas coleções para incluí-las em outras.

### 5.2.4 Número de Processadores

O primeiro estágio de CPH cria todas as coleções necessárias para minimizar o custo de comunicação entre as tarefas da aplicação. Como consequência, dependendo da topologia dos GADs de entrada, o número de coleções geradas pode ser bastante elevado. O segundo estágio procura explorar da melhor forma possível a heterogeneidade do ambiente alvo, e pode reaglomerar coleções nos processadores mais rápidos disponíveis, reduzindo o número de coleções originais. Em média, considerando todas as GADs de entrada, AMC obtém uma redução de 14,9% do número inicial de coleções produzidas por HAL. Ainda assim, o escalonamento CPH exige, em média, uma quantidade de processadores aproximadamente três vezes maior quando comparado a HEFT e duas vezes maior que LDBS.

|                    | IN  | OUT | DIA | RAN | EG  | DM  | PSG |
|--------------------|-----|-----|-----|-----|-----|-----|-----|
| $n_{CPH}/n_{HEFT}$ | 0.8 | 5.1 | 5.1 | 1.8 | 2.6 | 3.8 | 2.0 |
| $n_{CPH}/n_{LDBS}$ | 0.8 | 2.7 | 3.0 | 1.7 | 1.9 | 1.9 | 1.7 |

**Tabela 5.8: Razão entre o número de processadores necessários para o escalonamento CPH e os escalonamentos HEFT e LDBS, para diversas classes de GADs.**

A quantidade de processadores necessários está fortemente relacionada com a topologia dos GADs de entrada. Os GADs do tipo *árvore binária invertida* (IN), por exemplo, são uma exceção, e CPH produz um escalonamento com 21,4% menos processadores que HEFT ou LDBS. Nesse caso HAL não faz replicação de tarefas e HEFT e LDBS agem como heurísticas bastante gulosas. Para os GADs *árvore binária* (OUT), por outro lado, HAL elimina toda comunicação entre as coleções, resultando em um grande número de processadores necessários ao final do segundo estágio. Neste caso, CPH necessita, em média, de 5,1 vezes mais processadores que HEFT e 2,7 que LDBS. Para os GADs *diamantes* (DIA), o resultado final de CPH também exige um grande número de processadores, 5,1 vezes mais processadores que

HEFT e três vezes mais que LDBS. Os resultados de CPH para GADs *randômicos* (RAN) necessitam, em média, de 81,2% mais processadores que HEFT e 74,2% mais que LDBS. A Tabela 5.8 apresenta, além dos resultados já citados, a relação entre o número de processadores necessários para o escalonamento produzido por CPH ( $n_{CPH}$ ) e o número de processadores resultante do escalonamento HEFT ( $n_{HEFT}$ ) e LDBS ( $n_{LDBS}$ ), para os GADs da *eliminação de Gauss* (EG), *dinâmica molecular* (DM) e *PSG*.

### 5.3 Ambientes Restritos

Outra bateria de experimentos foi realizada com o objetivo de analisar o desempenho de CHP para um número restrito de processadores em relação ao número de coleções produzidas pelo primeiro estágio, ou seja, um número de processadores menor que o número de coleções a serem mapeadas. Nesse caso, o segundo estágio deve ser capaz de combinar essas coleções com o menor prejuízo possível. Para isso foi utilizado um ambiente computacional contendo um total de 20 processadores interconectados por uma rede homogênea. Para cada grafo e para cada par de parâmetros  $(F, L)$  utilizados, foram considerados 11 ambientes computacionais do tipo dual. Em cada ambiente, havia  $x$  processadores rápidos, com  $h() = 0,5$ , e  $20 - x$  processadores lentos, com  $h() = 1,0$ , onde  $x$  varia de 0 até 20 em incrementos de 2. No caso dos ambientes do tipo aleatório, foi avaliado o escalonamento de cada grafo para 4 ambientes com 20 processadores em que a média dos fatores de heterogeneidade era sempre unitária, e o desvio padrão variava entre 0 (caso homogêneo), 0,125, 0,25 ou 0,5.

Para a análise do escalonamento de GADs UET-UDT em ambientes restritos, foi conduzida uma série de experimentos com árvore binária de 127 tarefas, árvore binária invertida de 127 tarefas, diamante de 400 tarefas e um GAD aleatório de 140 tarefas. Foram especificados os pares de parâmetros  $(F, L)$  igual a  $(1, 1)$ ,  $(5, 1)$  e  $(10, 1)$  para cada GAD de entrada, caracterizando GADs de granularidade grossa, e  $(1, 5)$  e  $(1, 10)$ , caracterizando GADs de granularidade fina.

|         | Dual       |            | Aleatório  |            |
|---------|------------|------------|------------|------------|
| (F, L)  | CPH x HEFT | CPH x LDBS | CPH x HEFT | CPH x LDBS |
| (10, 1) | -19.57%    | -15.02%    | -22.33%    | -17.24%    |
| (5, 1)  | -16.56%    | -14.52%    | -20.61%    | -15.97%    |
| (1, 1)  | 9.94%      | -4.25%     | 0.52%      | -8.34%     |
| (1, 5)  | 40.89%     | 18.19%     | 35.97%     | 11.89%     |
| (1, 10) | 54.80%     | 31.71%     | 51.52%     | 26.96%     |
| Média   | 13.90%     | 3.22%      | 9.01%      | -0.54%     |

Tabela 5.9: Comparação percentual entre os resultados de escalonamento de todos os GADs produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1), (5, 1), (1, 1), (1, 5), (1, 10)$ , em ambientes restritos com  $|P| = 20$ .

A Tabela 5.9 resume a vantagem percentual média dos resultados obtidos por CPH comparados a HEFT e LDBS, para cada par de parâmetros  $(F, L)$ , para os ambientes dual e aleatório restritos. Observa-se que, em média, para granularidade fina, CPH produz melhores resultados que LDBS e, principalmente, que HEFT. Mas para granularidades grossa chega a ser 22,2% pior que HEFT e 17,2% pior que LDBS. Isso se deve principalmente à dificuldade do segundo estágio em combinar um grande número de coleções para alocação em um número reduzido de processadores. O grande número de tarefas replicadas também dificulta a obtenção de um bom resultado, uma vez que a otimização só é realizada ao final do processo de mapeamento. A modificação das coleções originais, permitindo que tarefas possam ser retiradas de uma coleção para ser inseridas em outra, também favoreceria uma melhor distribuição de cargas entre os processadores de um ambiente restrito.

No Apêndice H são apresentados os resultados obtidos para o escalonamento de todos os grafos mencionados em ambiente restritos, tabelados para cada de parâmetros  $(F, L)$ , separadamente para os ambientes duais e aleatórios.

## 5.4 Resumo

Neste capítulo foi apresentada a análise experimental da abordagem híbrida proposta neste trabalho. No próximo capítulo serão expostas as conclusões e considerações finais.



# Capítulo 6

## Conclusões

O objetivo deste trabalho foi estudar a aplicabilidade de HAL [42] para o escalonamento de tarefas em grades computacionais, a fim de tirar proveito dos excelentes resultados obtidos por este algoritmo no escalonamento de tarefas em ambientes homogêneos. Foi proposta uma estratégia de escalonamento de aplicações em um número limitado de processadores heterogêneos, em dois estágios denominada CPH. No primeiro estágio, HAL constrói coleções de tarefas com o objetivo de amenizar os efeitos dos custos de comunicação na execução da aplicação, considerando um ambiente de latência conhecida e de processadores virtuais homogêneos, com velocidade igual ao mais lento disponível no ambiente real. O segundo estágio, AMC, proposto neste trabalho, é um algoritmo do tipo *list scheduling* de coleções que implementa uma política de mapeamento de coleções nos processadores heterogêneos disponíveis em um ambiente real de forma a minimizar o *makespan* da aplicação.

Para validação da aplicabilidade do algoritmo proposto, foi utilizado um conjunto de GADs com pesos de computação e de comunicação unitários, incluindo grafos regulares (vários tamanhos de árvores binárias, árvores binárias invertidas, diamantes) [5] e irregulares (gerados aleatoriamente). Também foram considerados os grafos PSG [34] e, para permitir analisar o desempenho de CPH perante aplicações reais, foi utilizada uma família de grafos de diversos tamanhos representando o cálculo da eliminação de Gauss [18], juntamente com o grafo representando um problema de dinâmica molecular da física [32]. A granularidade dos GADs de en-

trada foi variada através de um fator multiplicativo da computação  $F$  e um fator multiplicativo representando a latência  $L$  do ambiente.

Dois tipos de ambientes computacionais simulados foram propostos. No ambiente **dual** os processadores são divididos nos subconjuntos de processadores rápidos e lentos, variando-se o tamanho desses subconjuntos para um dado número total de processadores. No ambiente **aleatório**, os processadores têm fatores de heterogeneidade determinados aleatoriamente, para uma dada média e um dado desvio padrão.

Em uma primeira análise, foram criados ambientes computacionais em que o número de processadores disponíveis foi sempre suficiente para o mapeamento de todas as coleções produzidas pelo primeiro estágio de CPH, os chamados ambientes **não restritos**. Os resultados produzidos por CPH foram comparados aos resultados produzidos por dois outros algoritmos amplamente estudados na literatura, específicos para ambiente heterogêneos e número limitado de processadores, HEFT e LDBS. A análise dos resultados, de um modo geral, foi bastante favorável ao CPH. De todas os 1512 experimentos, CPH obteve resultado melhor ou igual ao HEFT em 90,8% dos casos, e resultado melhor ou igual ao LDBS em 80,0% dos casos. Em particular, se considerarmos apenas os resultados obtidos para o escalonamento de grafos de granularidade fina, esses percentuais sobem para 99,9% e 90,5%, respectivamente. Este fato denota a boa adequação de CPH para escalonamento de aplicações com dependência de dados e custo de comunicação elevado perante o custo de computação, que a princípio pareceriam inapropriadas para a execução em grades computacionais, ambientes em que a latência é razoavelmente alta [41].

Outro fato importante observado nas análises detalhadas de cada escalonamento foi em relação ao número de mensagens trocadas entre processadores no escalonamento final. Os escalonamentos produzidos por CPH definem um número de mensagens trocadas entre processadores bem menor que LDBS e HEFT. Isso se deve principalmente às definições de caminho-predecessor-crítico implementadas no primeiro estágio, que não só minimizam o tempo de escalonamento de cada tarefa, mas dão prioridade à aglomeração de tarefas adjacentes na mesma coleção. Durante

o segundo estágio, o algoritmo de mapeamento proposto aqui, investe ainda na junção de coleções com certa afinidade, ao utilizar a discrepância  $\delta(\cdot)$  como uma das prioridades decisórias. Já HEFT e LDBS se preocupam unicamente em minimizar o tempo de fim de cada tarefa, através da implementação de política de inserção em espaços ociosos mesmo tendo que pagar caro por especificar um número elevado de mensagens a serem trocadas. Esse aspecto mostra claramente a aplicabilidade de CPH para sistemas distribuídos como as grades computacionais.

Já havia sido demonstrado que o desempenho do HAL é excelente quando comparado a outros algoritmos com e sem replicação [8, 42], para o escalonamento de aplicações em ambientes homogêneos. A implementação da estratégia de escalonamento em dois estágios permitiu a aplicação de HAL para o escalonamento em ambientes heterogêneos de forma vantajosa. Além do excelente resultado obtido pelo conjunto para granularidade fina, o segundo estágio reduziu o número de coleções finais do escalonamento para 14,9% do total produzido pelo primeiro estágio, oferecendo, além de uma melhora no desempenho, economia de recursos.

Observou-se que, em geral, a maioria dos casos em que o algoritmo CPH teve desempenho pior se comparado ao HEFT e LDBS, ocorreram quando o número de processadores mais rápidos era pequeno (10 a 20% do total de processadores). Neste caso, para um bom aproveitamento do ambiente disponível, é necessário que um maior número de tarefas seja alocado nos processadores mais rápidos. A formação de coleções do primeiro estágio pode não favorecer uma reaglomeração muito grande. Enquanto isso, os algoritmos do tipo *list scheduling* como o LDBS e HEFT estão mais focados no ambiente em si e buscam uma minimização local na alocação de tarefa a tarefa, sendo mais sensíveis, portanto, a esta configuração com poucos processadores mais rápidos. A pesquisa de critérios de escolha do processador para a alocação de uma dada coleção baseado na distribuição de carga mostrou-se uma forma de melhorar o desempenho de CHP nesse tipo de ambientes.

Outra bateria de experimentos foi realizada utilizando-se ambientes com um número de processadores bem menor comparado ao número de coleções produzidas pelo primeiro estágio, chamados ambientes **restritos**. Para esses ambientes foram

utilizados apenas alguns GADs com grande número de tarefas, escolhidos devido ao grande número de coleções produzidas pelo primeiro estágio. O resultado observado não foi tão bom como no caso anterior e mostrou que a utilização de CPH em ambientes restritos é vantajosa apenas para aplicações de granularidade fina. Para granularidade grossa, os *makespans* obtidos por CPH se mostraram em média piores que os obtidos por HEFT e LDBS. O segundo estágio não foi capaz de mapear um grande número de coleções (com uma grande quantidade de tarefas replicadas) em uma quantidade de processadores muito menor de forma tão eficiente quanto para ambientes não restritos. A pesquisa de prioridades específicas para ambientes restritos para serem usadas na escolha da próxima coleção e do processador é um caminho para obtenção de melhores resultados. Outra possibilidade é o desenvolvimento de um algoritmo de aglomeração capaz de levar em consideração um número limitado de processadores e a heterogeneidade do ambiente.

As conclusões deste trabalho de Dissertação demonstram a vantagem da utilização da metodologia de aglomeração com replicação para redes de processadores heterogêneos, cujos custos de comunicação são altos. Para dar continuidade a este estudo, trabalhos futuros que podem ser investigados são:

- analisar a aplicabilidade do esquema de dois estágios para redes de comunicação heterogênea;
- especificar uma abordagem de um estágio baseada na metodologia de aglomeração com replicação e as prioridades implementadas no segundo estágio;
- derivar uma estratégia em dois estágios para o escalonamento dinâmico de tarefas em grades computacionais.

# Apêndice A

## Resultados para Árvores Binárias Invertidas

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 29.0 | 16 | 29.0 | 16 | 29.0   | 21 | 29.0   | 21 | 0.0%       | 0.0%       |
| 9   | 29.0 | 16 | 29.0 | 16 | 29.0   | 21 | 29.0   | 21 | 0.0%       | 0.0%       |
| 8   | 29.0 | 16 | 29.0 | 16 | 29.0   | 21 | 29.0   | 21 | 0.0%       | 0.0%       |
| 7   | 29.0 | 16 | 29.0 | 16 | 29.0   | 21 | 29.0   | 21 | 0.0%       | 0.0%       |
| 6   | 29.0 | 16 | 29.0 | 16 | 29.0   | 21 | 29.0   | 21 | 0.0%       | 0.0%       |
| 5   | 29.0 | 16 | 29.0 | 16 | 29.0   | 21 | 29.0   | 21 | 0.0%       | 0.0%       |
| 4   | 30.0 | 15 | 30.0 | 15 | 29.0   | 20 | 29.0   | 20 | 3.3%       | 3.3%       |
| 3   | 30.0 | 15 | 30.0 | 15 | 31.0   | 16 | 31.0   | 16 | -3.3%      | -3.3%      |
| 2   | 32.0 | 10 | 32.0 | 10 | 32.0   | 14 | 32.0   | 14 | 0.0%       | 0.0%       |
| 1   | 36.0 | 11 | 36.0 | 11 | 36.0   | 12 | 36.0   | 12 | 0.0%       | 0.0%       |
| 0   | 54.0 | 16 | 54.0 | 16 | 54.0   | 21 | 54.0   | 21 | 0.0%       | 0.0%       |

**Tabela A.1:** Resultados de escalonamento do GAD *in-tree-31* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 16.5 | 16 | 16.5 | 16 | 16.5   | 21 | 16.5   | 21 | 0.0%       | 0.0%       |
| 9   | 16.5 | 16 | 16.5 | 16 | 16.5   | 21 | 16.5   | 21 | 0.0%       | 0.0%       |
| 8   | 16.5 | 16 | 16.5 | 16 | 16.5   | 21 | 16.5   | 21 | 0.0%       | 0.0%       |
| 7   | 16.5 | 16 | 16.5 | 16 | 16.5   | 21 | 16.5   | 21 | 0.0%       | 0.0%       |
| 6   | 16.5 | 16 | 16.5 | 16 | 16.5   | 21 | 16.5   | 21 | 0.0%       | 0.0%       |
| 5   | 16.5 | 16 | 16.5 | 16 | 16.5   | 21 | 16.5   | 21 | 0.0%       | 0.0%       |
| 4   | 16.5 | 15 | 16.5 | 15 | 16.5   | 20 | 16.5   | 20 | 0.0%       | 0.0%       |
| 3   | 16.5 | 15 | 16.5 | 15 | 16.5   | 16 | 16.5   | 16 | 0.0%       | 0.0%       |
| 2   | 17.0 | 10 | 17.0 | 10 | 17.0   | 14 | 17.0   | 14 | 0.0%       | 0.0%       |
| 1   | 18.5 | 11 | 18.5 | 11 | 18.5   | 12 | 18.5   | 12 | 0.0%       | 0.0%       |
| 0   | 29.0 | 16 | 29.0 | 16 | 29.0   | 21 | 29.0   | 21 | 0.0%       | 0.0%       |

Tabela A.2: Resultados de escalonamento do GAD *in-tree-31* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 6.5  | 16 | 4.5  | 16 | 4.5    | 11 | 4.5    | 11 | 30.8%      | 0.0%       |
| 9   | 6.5  | 16 | 4.5  | 16 | 4.5    | 11 | 4.5    | 11 | 30.8%      | 0.0%       |
| 8   | 6.5  | 16 | 4.5  | 16 | 4.5    | 11 | 4.5    | 11 | 30.8%      | 0.0%       |
| 7   | 6.5  | 16 | 4.5  | 16 | 4.5    | 11 | 4.5    | 11 | 30.8%      | 0.0%       |
| 6   | 6.5  | 16 | 4.5  | 16 | 4.5    | 11 | 4.5    | 11 | 30.8%      | 0.0%       |
| 5   | 6.5  | 16 | 4.5  | 16 | 4.5    | 11 | 4.5    | 11 | 30.8%      | 0.0%       |
| 4   | 6.0  | 15 | 4.5  | 15 | 4.5    | 11 | 4.5    | 11 | 25.0%      | 0.0%       |
| 3   | 6.0  | 15 | 4.5  | 15 | 4.5    | 11 | 4.5    | 11 | 25.0%      | 0.0%       |
| 2   | 6.0  | 10 | 5.0  | 10 | 5.0    | 11 | 5.0    | 11 | 16.7%      | 0.0%       |
| 1   | 6.0  | 11 | 5.5  | 11 | 5.0    | 11 | 5.0    | 11 | 16.7%      | 9.1%       |
| 0   | 9.0  | 16 | 9.0  | 16 | 9.0    | 11 | 9.0    | 11 | 0.0%       | 0.0%       |

Tabela A.3: Resultados de escalonamento do GAD *in-tree-31* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|---|--------|---|------------|------------|
| 10  | 22.5 | 16 | 12.5 | 16 | 9.5    | 7 | 9.5    | 7 | 57.8%      | 24.0%      |
| 9   | 22.5 | 16 | 12.5 | 16 | 9.5    | 7 | 9.5    | 7 | 57.8%      | 24.0%      |
| 8   | 22.5 | 16 | 12.5 | 16 | 9.5    | 7 | 9.5    | 7 | 57.8%      | 24.0%      |
| 7   | 22.5 | 16 | 12.5 | 16 | 9.5    | 7 | 9.5    | 7 | 57.8%      | 24.0%      |
| 6   | 22.5 | 16 | 12.5 | 16 | 9.5    | 7 | 9.5    | 7 | 57.8%      | 24.0%      |
| 5   | 22.5 | 16 | 12.5 | 16 | 9.5    | 7 | 9.5    | 7 | 57.8%      | 24.0%      |
| 4   | 18.0 | 15 | 12.5 | 15 | 9.5    | 7 | 9.5    | 7 | 47.2%      | 24.0%      |
| 3   | 18.0 | 15 | 12.5 | 15 | 9.5    | 7 | 9.5    | 7 | 47.2%      | 24.0%      |
| 2   | 18.0 | 10 | 12.5 | 10 | 9.5    | 7 | 9.5    | 7 | 47.2%      | 24.0%      |
| 1   | 18.0 | 11 | 13.0 | 11 | 10.0   | 6 | 9.5    | 6 | 47.2%      | 26.9%      |
| 0   | 25.0 | 16 | 15.0 | 16 | 14.0   | 7 | 14.0   | 7 | 44.0%      | 6.7%       |

Tabela A.4: Resultados de escalonamento do GAD *in-tree-31* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|---|--------|---|------------|------------|
| 10  | 42.5 | 16 | 22.5 | 16 | 13.0   | 3 | 13.0   | 3 | 69.4%      | 42.2%      |
| 9   | 42.5 | 16 | 22.5 | 16 | 13.0   | 3 | 13.0   | 3 | 69.4%      | 42.2%      |
| 8   | 42.5 | 16 | 22.5 | 16 | 13.0   | 3 | 13.0   | 3 | 69.4%      | 42.2%      |
| 7   | 42.5 | 16 | 22.5 | 16 | 13.0   | 3 | 13.0   | 3 | 69.4%      | 42.2%      |
| 6   | 42.5 | 16 | 22.5 | 16 | 13.0   | 3 | 13.0   | 3 | 69.4%      | 42.2%      |
| 5   | 42.5 | 16 | 22.5 | 16 | 13.0   | 3 | 13.0   | 3 | 69.4%      | 42.2%      |
| 4   | 33.0 | 15 | 22.5 | 15 | 13.0   | 3 | 13.0   | 3 | 60.6%      | 42.2%      |
| 3   | 33.0 | 15 | 22.5 | 15 | 13.0   | 3 | 13.0   | 3 | 60.6%      | 42.2%      |
| 2   | 33.0 | 10 | 22.5 | 10 | 13.0   | 3 | 13.0   | 3 | 60.6%      | 42.2%      |
| 1   | 33.0 | 11 | 23.0 | 11 | 13.0   | 3 | 13.0   | 3 | 60.6%      | 43.5%      |
| 0   | 45.0 | 16 | 25.0 | 16 | 19.0   | 5 | 18.0   | 5 | 60.0%      | 28.0%      |

Tabela A.5: Resultados de escalonamento do GAD *in-tree-31* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes duais.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 54.0 | 16 | 54.0 | 16 | 54.0   | 21 | 54.0   | 21 | 0.0%       | 0.0%       |
| 0.125          | 46.7 | 16 | 46.7 | 16 | 46.6   | 16 | 46.6   | 16 | 0.2%       | 0.1%       |
| 0.250          | 41.1 | 16 | 41.0 | 16 | 41.0   | 16 | 41.0   | 16 | 0.4%       | 0.1%       |
| 0.500          | 33.6 | 16 | 33.3 | 16 | 33.3   | 16 | 33.3   | 16 | 1.0%       | 0.1%       |

Tabela A.6: Resultados de escalonamento do GAD *in-tree-31* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 29.0 | 16 | 29.0 | 16 | 29.0   | 21 | 29.0   | 21 | 0.0%       | 0.0%       |
| 0.125          | 25.3 | 16 | 25.3 | 16 | 25.3   | 16 | 25.3   | 16 | 0.2%       | 0.1%       |
| 0.250          | 22.6 | 16 | 22.5 | 16 | 22.5   | 16 | 22.5   | 16 | 0.3%       | 0.0%       |
| 0.500          | 18.8 | 16 | 18.7 | 16 | 18.7   | 16 | 18.7   | 16 | 0.9%       | 0.0%       |

Tabela A.7: Resultados de escalonamento do GAD *in-tree-31* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 9.0  | 16 | 9.0  | 16 | 9.0    | 11 | 9.0    | 11 | 0.0%       | 0.0%       |
| 0.125          | 8.3  | 16 | 7.7  | 16 | 7.7    | 11 | 7.7    | 11 | 7.3%       | 0.0%       |
| 0.250          | 7.7  | 16 | 6.7  | 16 | 6.7    | 11 | 6.7    | 11 | 13.7%      | 0.0%       |
| 0.500          | 7.0  | 16 | 5.5  | 16 | 5.3    | 11 | 5.3    | 11 | 24.3%      | 4.2%       |

Tabela A.8: Resultados de escalonamento do GAD *in-tree-31* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|---|--------|---|------------|------------|
| 0.000          | 25.0 | 16 | 15.0 | 16 | 14.0   | 7 | 14.0   | 7 | 44.0%      | 6.7%       |
| 0.125          | 24.3 | 16 | 14.3 | 16 | 12.7   | 6 | 12.7   | 6 | 47.8%      | 11.7%      |
| 0.250          | 23.7 | 16 | 13.8 | 16 | 11.7   | 6 | 11.7   | 6 | 50.8%      | 15.4%      |
| 0.500          | 23.0 | 16 | 13.0 | 16 | 10.3   | 6 | 10.3   | 6 | 55.3%      | 21.2%      |

**Tabela A.9:** Resultados de escalonamento do GAD *in-tree-31* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|---|--------|---|------------|------------|
| 0.000          | 45.0 | 16 | 25.0 | 16 | 19.0   | 5 | 18.0   | 5 | 60.0%      | 28.0%      |
| 0.125          | 44.3 | 16 | 24.3 | 16 | 16.8   | 5 | 16.2   | 5 | 63.4%      | 33.5%      |
| 0.250          | 43.7 | 16 | 23.8 | 16 | 16.3   | 4 | 15.9   | 4 | 63.6%      | 33.1%      |
| 0.500          | 43.0 | 16 | 23.0 | 16 | 14.1   | 4 | 14.1   | 4 | 67.2%      | 38.8%      |

**Tabela A.10:** Resultados de escalonamento do GAD *in-tree-31* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes aleatórios.



# Apêndice B

## Resultados para Árvore Binárias

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 41.0 | 62 | 38.0 | 64 | 35.0   | 64 | 35.0   | 64 | 14.6%      | 7.9%       |
| 9   | 41.0 | 62 | 38.0 | 64 | 35.0   | 64 | 35.0   | 64 | 14.6%      | 7.9%       |
| 8   | 41.0 | 62 | 38.0 | 64 | 35.0   | 64 | 35.0   | 64 | 14.6%      | 7.9%       |
| 7   | 41.0 | 62 | 38.0 | 64 | 35.0   | 64 | 35.0   | 64 | 14.6%      | 7.9%       |
| 6   | 42.0 | 60 | 41.0 | 60 | 40.0   | 60 | 40.0   | 60 | 4.8%       | 2.4%       |
| 5   | 44.0 | 50 | 41.0 | 50 | 40.0   | 50 | 40.0   | 50 | 9.1%       | 2.4%       |
| 4   | 45.0 | 40 | 42.0 | 40 | 40.0   | 40 | 40.0   | 40 | 11.1%      | 4.8%       |
| 3   | 47.0 | 32 | 46.0 | 32 | 55.0   | 30 | 55.0   | 30 | -17.0%     | -19.6%     |
| 2   | 50.0 | 35 | 47.0 | 36 | 55.0   | 20 | 55.0   | 20 | -10.0%     | -17.0%     |
| 1   | 55.0 | 46 | 52.0 | 46 | 65.0   | 42 | 65.0   | 42 | -18.2%     | -25.0%     |
| 0   | 76.0 | 64 | 73.0 | 64 | 70.0   | 64 | 70.0   | 64 | 7.9%       | 4.1%       |

Tabela B.1: Resultados de escalonamento do GAD *out-tree-127* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 23.5 | 56 | 20.5 | 62 | 17.5   | 64 | 17.5   | 64 | 25.5%      | 14.6%      |
| 9   | 23.5 | 56 | 20.5 | 62 | 17.5   | 64 | 17.5   | 64 | 25.5%      | 14.6%      |
| 8   | 23.5 | 56 | 20.5 | 62 | 17.5   | 64 | 17.5   | 64 | 25.5%      | 14.6%      |
| 7   | 23.5 | 56 | 20.5 | 62 | 17.5   | 64 | 17.5   | 64 | 25.5%      | 14.6%      |
| 6   | 23.5 | 56 | 21.0 | 60 | 20.0   | 60 | 20.0   | 60 | 14.9%      | 4.8%       |
| 5   | 24.0 | 50 | 21.0 | 50 | 20.0   | 50 | 20.0   | 50 | 16.7%      | 4.8%       |
| 4   | 25.0 | 40 | 22.0 | 40 | 20.0   | 40 | 20.0   | 40 | 20.0%      | 9.1%       |
| 3   | 26.0 | 30 | 23.5 | 30 | 27.5   | 30 | 27.5   | 30 | -5.8%      | -17.0%     |
| 2   | 27.5 | 32 | 24.5 | 36 | 27.5   | 20 | 27.5   | 20 | 0.0%       | -12.2%     |
| 1   | 30.0 | 42 | 27.0 | 46 | 30.0   | 42 | 30.0   | 42 | 0.0%       | -11.1%     |
| 0   | 41.0 | 62 | 38.0 | 64 | 35.0   | 64 | 35.0   | 64 | 14.6%      | 7.9%       |

Tabela B.2: Resultados de escalonamento do GAD *out-tree-127* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 9.0  | 32 | 6.0  | 50 | 3.5    | 64 | 3.5    | 64 | 61.1%      | 41.7%      |
| 9   | 9.0  | 32 | 6.0  | 50 | 3.5    | 64 | 3.5    | 64 | 61.1%      | 41.7%      |
| 8   | 9.0  | 32 | 6.0  | 50 | 3.5    | 64 | 3.5    | 64 | 61.1%      | 41.7%      |
| 7   | 9.0  | 32 | 6.0  | 50 | 3.5    | 64 | 3.5    | 64 | 61.1%      | 41.7%      |
| 6   | 9.0  | 32 | 6.0  | 50 | 4.0    | 60 | 4.0    | 60 | 55.6%      | 33.3%      |
| 5   | 9.0  | 32 | 6.0  | 50 | 4.0    | 50 | 4.0    | 50 | 55.6%      | 33.3%      |
| 4   | 9.0  | 32 | 6.5  | 40 | 4.0    | 40 | 4.0    | 40 | 55.6%      | 38.5%      |
| 3   | 9.5  | 30 | 6.5  | 30 | 5.5    | 30 | 5.5    | 30 | 42.1%      | 15.4%      |
| 2   | 9.5  | 20 | 6.5  | 20 | 5.5    | 20 | 5.5    | 20 | 42.1%      | 15.4%      |
| 1   | 10.0 | 26 | 7.5  | 40 | 6.5    | 42 | 6.5    | 42 | 35.0%      | 13.3%      |
| 0   | 13.0 | 32 | 10.0 | 52 | 7.0    | 64 | 7.0    | 64 | 46.2%      | 30.0%      |

Tabela B.3: Resultados de escalonamento do GAD *out-tree-127* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 24.0 | 8  | 13.0 | 16 | 3.5    | 64 | 3.5    | 64 | 85.4%      | 73.1%      |
| 9   | 24.0 | 8  | 13.0 | 16 | 3.5    | 64 | 3.5    | 64 | 85.4%      | 73.1%      |
| 8   | 24.0 | 8  | 13.0 | 16 | 3.5    | 64 | 3.5    | 64 | 85.4%      | 73.1%      |
| 7   | 24.0 | 8  | 13.0 | 16 | 3.5    | 64 | 3.5    | 64 | 85.4%      | 73.1%      |
| 6   | 24.0 | 8  | 13.0 | 16 | 4.0    | 60 | 4.0    | 60 | 83.3%      | 69.2%      |
| 5   | 24.0 | 8  | 13.0 | 16 | 4.0    | 50 | 4.0    | 50 | 83.3%      | 69.2%      |
| 4   | 24.0 | 8  | 13.0 | 16 | 4.0    | 40 | 4.0    | 40 | 83.3%      | 69.2%      |
| 3   | 24.0 | 8  | 13.0 | 16 | 5.5    | 30 | 5.5    | 30 | 77.1%      | 57.7%      |
| 2   | 24.0 | 8  | 13.0 | 16 | 5.5    | 20 | 5.5    | 20 | 77.1%      | 57.7%      |
| 1   | 24.0 | 8  | 13.5 | 14 | 7.0    | 42 | 7.0    | 42 | 70.8%      | 48.1%      |
| 0   | 31.0 | 16 | 18.0 | 18 | 7.0    | 64 | 7.0    | 64 | 77.4%      | 61.1%      |

Tabela B.4: Resultados de escalonamento do GAD *out-tree-127* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes duais.

| $x$ | HEFT |   | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|---|------|----|--------|----|--------|----|------------|------------|
| 10  | 36.5 | 4 | 18.5 | 18 | 3.5    | 64 | 3.5    | 64 | 90.4%      | 81.1%      |
| 9   | 36.5 | 4 | 18.5 | 18 | 3.5    | 64 | 3.5    | 64 | 90.4%      | 81.1%      |
| 8   | 36.5 | 4 | 18.5 | 18 | 3.5    | 64 | 3.5    | 64 | 90.4%      | 81.1%      |
| 7   | 36.5 | 4 | 18.5 | 18 | 3.5    | 64 | 3.5    | 64 | 90.4%      | 81.1%      |
| 6   | 36.5 | 4 | 18.5 | 18 | 4.0    | 60 | 4.0    | 60 | 89.0%      | 78.4%      |
| 5   | 36.5 | 4 | 18.5 | 18 | 4.0    | 50 | 4.0    | 50 | 89.0%      | 78.4%      |
| 4   | 36.5 | 4 | 18.5 | 18 | 4.0    | 40 | 4.0    | 40 | 89.0%      | 78.4%      |
| 3   | 36.5 | 4 | 18.5 | 18 | 5.5    | 30 | 5.5    | 30 | 84.9%      | 70.3%      |
| 2   | 36.5 | 4 | 18.5 | 18 | 5.5    | 20 | 5.5    | 20 | 84.9%      | 70.3%      |
| 1   | 36.5 | 4 | 19.5 | 20 | 7.0    | 42 | 7.0    | 42 | 80.8%      | 64.1%      |
| 0   | 48.0 | 8 | 26.0 | 16 | 7.0    | 64 | 7.0    | 64 | 85.4%      | 73.1%      |

Tabela B.5: Resultados de escalonamento do GAD *out-tree-127* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes duais.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 76.0 | 64 | 73.0 | 64 | 70.0   | 64 | 70.0   | 64 | 7.9%       | 4.1%       |
| 0.125          | 68.9 | 63 | 66.3 | 64 | 71.3   | 53 | 71.3   | 53 | -3.5%      | -7.6%      |
| 0.250          | 62.4 | 56 | 60.2 | 58 | 63.5   | 46 | 63.5   | 46 | -1.8%      | -5.5%      |
| 0.500          | 52.6 | 49 | 49.8 | 48 | 52.8   | 40 | 52.8   | 40 | -0.5%      | -6.1%      |

Tabela B.6: Resultados de escalonamento do GAD *out-tree-127* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 41.0 | 62 | 38.0 | 64 | 35.0   | 64 | 35.0   | 64 | 14.6%      | 7.9%       |
| 0.125          | 37.2 | 56 | 34.6 | 62 | 36.0   | 53 | 36.0   | 53 | 3.3%       | -4.0%      |
| 0.250          | 33.8 | 51 | 31.0 | 51 | 32.3   | 46 | 32.3   | 46 | 4.5%       | -4.2%      |
| 0.500          | 28.6 | 46 | 26.3 | 46 | 27.1   | 40 | 27.1   | 40 | 5.4%       | -3.1%      |

Tabela B.7: Resultados de escalonamento do GAD *out-tree-127* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 13.0 | 32 | 10.0 | 52 | 7.0    | 64 | 7.0    | 64 | 46.2%      | 30.0%      |
| 0.125          | 11.9 | 37 | 9.0  | 55 | 7.2    | 53 | 7.2    | 53 | 39.3%      | 19.9%      |
| 0.250          | 11.1 | 34 | 8.1  | 51 | 6.7    | 46 | 6.7    | 46 | 39.4%      | 17.6%      |
| 0.500          | 9.8  | 30 | 7.1  | 44 | 5.7    | 40 | 5.7    | 40 | 42.2%      | 20.3%      |

Tabela B.8: Resultados de escalonamento do GAD *out-tree-127* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 31.0 | 16 | 18.0 | 18 | 7.0    | 64 | 7.0    | 64 | 77.4%      | 61.1%      |
| 0.125          | 29.4 | 17 | 16.1 | 18 | 7.2    | 53 | 7.2    | 53 | 75.4%      | 55.0%      |
| 0.250          | 28.1 | 12 | 15.4 | 18 | 6.7    | 46 | 6.7    | 46 | 76.0%      | 56.1%      |
| 0.500          | 25.6 | 12 | 12.9 | 18 | 5.7    | 40 | 5.7    | 40 | 77.7%      | 55.9%      |

Tabela B.9: Resultados de escalonamento do GAD *out-tree-127* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 48.0 | 8  | 26.0 | 16 | 7.0    | 64 | 7.0    | 64 | 85.4%      | 73.1%      |
| 0.125          | 45.3 | 11 | 23.7 | 24 | 7.2    | 53 | 7.2    | 53 | 84.0%      | 69.5%      |
| 0.250          | 43.3 | 8  | 21.8 | 26 | 6.7    | 46 | 6.7    | 46 | 84.4%      | 69.1%      |
| 0.500          | 39.3 | 5  | 19.4 | 24 | 5.7    | 40 | 5.7    | 40 | 85.5%      | 70.8%      |

Tabela B.10: Resultados de escalonamento do GAD *out-tree-127* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes aleatórios.

# Apêndice C

## Resultados para GADs Diamantes

| $x$ | HEFT  |    | LDBS  |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|-------|----|-------|----|--------|----|--------|----|------------|------------|
| 10  | 180.0 | 15 | 171.0 | 16 | 170.0  | 30 | 170.0  | 30 | 5.6%       | 0.6%       |
| 9   | 180.0 | 15 | 171.0 | 16 | 170.0  | 30 | 170.0  | 30 | 5.6%       | 0.6%       |
| 8   | 180.0 | 15 | 171.0 | 16 | 170.0  | 30 | 170.0  | 30 | 5.6%       | 0.6%       |
| 7   | 180.0 | 15 | 171.0 | 16 | 170.0  | 30 | 170.0  | 30 | 5.6%       | 0.6%       |
| 6   | 180.0 | 15 | 171.0 | 16 | 170.0  | 30 | 170.0  | 30 | 5.6%       | 0.6%       |
| 5   | 180.0 | 15 | 171.0 | 16 | 170.0  | 25 | 170.0  | 25 | 5.6%       | 0.6%       |
| 4   | 180.0 | 15 | 172.0 | 15 | 183.0  | 15 | 183.0  | 15 | -1.7%      | -6.4%      |
| 3   | 180.0 | 15 | 172.0 | 15 | 183.0  | 15 | 183.0  | 15 | -1.7%      | -6.4%      |
| 2   | 190.0 | 10 | 183.0 | 10 | 217.0  | 10 | 217.0  | 10 | -14.2%     | -18.6%     |
| 1   | 255.0 | 10 | 247.0 | 11 | 284.0  | 10 | 284.0  | 10 | -11.4%     | -15.0%     |
| 0   | 330.0 | 15 | 325.0 | 16 | 325.0  | 30 | 325.0  | 30 | 1.5%       | 0.0%       |

Tabela C.1: Resultados de escalonamento do GAD *diamante-256* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes duais.

| $x$ | HEFT  |    | LDBS  |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|-------|----|-------|----|--------|----|--------|----|------------|------------|
| 10  | 104.5 | 15 | 96.5  | 15 | 92.5   | 30 | 92.5   | 30 | 11.5%      | 4.1%       |
| 9   | 104.5 | 15 | 96.5  | 15 | 92.5   | 30 | 92.5   | 30 | 11.5%      | 4.1%       |
| 8   | 104.5 | 15 | 96.5  | 15 | 92.5   | 30 | 92.5   | 30 | 11.5%      | 4.1%       |
| 7   | 104.5 | 15 | 96.5  | 15 | 92.5   | 30 | 92.5   | 30 | 11.5%      | 4.1%       |
| 6   | 104.5 | 15 | 96.5  | 15 | 92.5   | 30 | 92.5   | 30 | 11.5%      | 4.1%       |
| 5   | 104.5 | 15 | 96.5  | 15 | 92.5   | 25 | 92.5   | 25 | 11.5%      | 4.1%       |
| 4   | 104.5 | 15 | 96.5  | 15 | 93.5   | 15 | 93.5   | 15 | 10.5%      | 3.1%       |
| 3   | 104.5 | 15 | 96.5  | 15 | 93.5   | 15 | 93.5   | 15 | 10.5%      | 3.1%       |
| 2   | 108.5 | 10 | 101.0 | 11 | 105.0  | 10 | 105.0  | 10 | 3.2%       | -4.0%      |
| 1   | 138.0 | 9  | 130.0 | 10 | 144.0  | 10 | 144.0  | 10 | -4.3%      | -10.8%     |
| 0   | 180.0 | 15 | 171.0 | 16 | 170.0  | 30 | 170.0  | 30 | 5.6%       | 0.6%       |

Tabela C.2: Resultados de escalonamento do GAD *diamante-256* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes duais.

| $x$ | HEFT |   | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|---|------|----|--------|----|--------|----|------------|------------|
| 10  | 43.5 | 6 | 31.5 | 11 | 39.5   | 17 | 38.5   | 17 | 11.5%      | -22.2%     |
| 9   | 43.5 | 6 | 31.5 | 11 | 39.5   | 17 | 38.5   | 17 | 11.5%      | -22.2%     |
| 8   | 43.5 | 6 | 31.5 | 11 | 39.5   | 17 | 38.5   | 17 | 11.5%      | -22.2%     |
| 7   | 43.5 | 6 | 31.5 | 11 | 39.5   | 17 | 38.5   | 17 | 11.5%      | -22.2%     |
| 6   | 43.5 | 6 | 31.5 | 11 | 39.5   | 17 | 38.5   | 17 | 11.5%      | -22.2%     |
| 5   | 43.5 | 6 | 31.5 | 11 | 39.5   | 17 | 38.5   | 17 | 11.5%      | -22.2%     |
| 4   | 43.5 | 6 | 31.5 | 11 | 39.0   | 15 | 39.0   | 15 | 10.3%      | -23.8%     |
| 3   | 43.5 | 6 | 31.5 | 11 | 39.0   | 15 | 39.0   | 15 | 10.3%      | -23.8%     |
| 2   | 43.5 | 6 | 32.0 | 11 | 44.0   | 11 | 44.0   | 11 | -1.1%      | -37.5%     |
| 1   | 43.5 | 6 | 36.5 | 7  | 44.0   | 15 | 44.0   | 15 | -1.1%      | -20.5%     |
| 0   | 60.0 | 8 | 51.0 | 12 | 46.0   | 29 | 46.0   | 29 | 23.3%      | 9.8%       |

Tabela C.3: Resultados de escalonamento do GAD *diamante-256* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT  |   | LDBS |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|-------|---|------|---|--------|----|--------|----|------------|------------|
| 10  | 115.5 | 2 | 72.5 | 3 | 73.0   | 17 | 73.0   | 17 | 36.8%      | -0.7%      |
| 9   | 115.5 | 2 | 72.5 | 3 | 73.0   | 17 | 73.0   | 17 | 36.8%      | -0.7%      |
| 8   | 115.5 | 2 | 72.5 | 3 | 73.0   | 17 | 73.0   | 17 | 36.8%      | -0.7%      |
| 7   | 115.5 | 2 | 72.5 | 3 | 73.0   | 17 | 73.0   | 17 | 36.8%      | -0.7%      |
| 6   | 115.5 | 2 | 72.5 | 3 | 73.0   | 17 | 73.0   | 17 | 36.8%      | -0.7%      |
| 5   | 115.5 | 2 | 72.5 | 3 | 73.0   | 17 | 73.0   | 17 | 36.8%      | -0.7%      |
| 4   | 115.5 | 2 | 72.5 | 3 | 73.0   | 17 | 73.0   | 17 | 36.8%      | -0.7%      |
| 3   | 115.5 | 2 | 72.5 | 3 | 73.0   | 17 | 73.0   | 17 | 36.8%      | -0.7%      |
| 2   | 115.5 | 2 | 72.5 | 3 | 73.0   | 16 | 73.0   | 16 | 36.8%      | -0.7%      |
| 1   | 115.5 | 2 | 72.5 | 3 | 74.5   | 15 | 74.5   | 15 | 35.5%      | -2.8%      |
| 0   | 156.0 | 3 | 90.0 | 7 | 80.0   | 24 | 80.0   | 24 | 48.7%      | 11.1%      |

Tabela C.4: Resultados de escalonamento do GAD *diamante-256* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes duais.

| $x$ | HEFT  |   | LDBS  |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|-------|---|-------|---|--------|----|--------|----|------------|------------|
| 10  | 128.0 | 1 | 110.5 | 2 | 89.5   | 25 | 89.5   | 25 | 30.1%      | 19.0%      |
| 9   | 128.0 | 1 | 110.5 | 2 | 89.5   | 25 | 89.5   | 25 | 30.1%      | 19.0%      |
| 8   | 128.0 | 1 | 110.5 | 2 | 89.5   | 25 | 89.5   | 25 | 30.1%      | 19.0%      |
| 7   | 128.0 | 1 | 110.5 | 2 | 89.5   | 25 | 89.5   | 25 | 30.1%      | 19.0%      |
| 6   | 128.0 | 1 | 110.5 | 2 | 89.5   | 25 | 89.5   | 25 | 30.1%      | 19.0%      |
| 5   | 128.0 | 1 | 110.5 | 2 | 89.5   | 25 | 89.5   | 25 | 30.1%      | 19.0%      |
| 4   | 128.0 | 1 | 110.5 | 2 | 89.5   | 16 | 89.5   | 16 | 30.1%      | 19.0%      |
| 3   | 128.0 | 1 | 110.5 | 2 | 89.5   | 16 | 89.5   | 16 | 30.1%      | 19.0%      |
| 2   | 128.0 | 1 | 110.5 | 2 | 89.5   | 13 | 89.5   | 13 | 30.1%      | 19.0%      |
| 1   | 128.0 | 1 | 110.5 | 2 | 94.0   | 15 | 94.0   | 15 | 26.6%      | 14.9%      |
| 0   | 231.0 | 2 | 145.0 | 3 | 104.0  | 31 | 104.0  | 31 | 55.0%      | 28.3%      |

Tabela C.5: Resultados de escalonamento do GAD *diamante-256* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes duais.

| $\bar{\sigma}$ | HEFT  |    | LDBS  |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|-------|----|-------|----|--------|----|--------|----|------------|------------|
| 0.000          | 330.0 | 15 | 325.0 | 16 | 325.0  | 30 | 325.0  | 30 | 1.5%       | 0.0%       |
| 0.125          | 290.2 | 15 | 286.1 | 16 | 287.0  | 17 | 287.0  | 17 | 1.1%       | -0.3%      |
| 0.250          | 257.8 | 15 | 253.4 | 15 | 259.4  | 16 | 259.4  | 16 | -0.6%      | -2.4%      |
| 0.500          | 215.7 | 14 | 206.1 | 15 | 210.4  | 15 | 210.4  | 15 | 2.5%       | -2.1%      |

Tabela C.6: Resultados de escalonamento do GAD *diamante-256* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT  |    | LDBS  |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|-------|----|-------|----|--------|----|--------|----|------------|------------|
| 0.000          | 180.0 | 15 | 171.0 | 16 | 170.0  | 30 | 170.0  | 30 | 5.6%       | 0.6%       |
| 0.125          | 159.4 | 14 | 150.8 | 16 | 151.1  | 16 | 151.1  | 16 | 5.2%       | -0.2%      |
| 0.250          | 142.8 | 14 | 134.6 | 15 | 137.2  | 16 | 137.2  | 16 | 3.9%       | -1.9%      |
| 0.500          | 121.3 | 14 | 111.1 | 15 | 112.4  | 14 | 112.4  | 14 | 7.3%       | -1.2%      |

Tabela C.7: Resultados de escalonamento do GAD *diamante-256* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |   | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|---|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 60.0 | 8 | 51.0 | 12 | 46.0   | 29 | 46.0   | 29 | 23.3%      | 9.8%       |
| 0.125          | 54.6 | 8 | 46.8 | 12 | 45.6   | 26 | 45.6   | 26 | 16.4%      | 2.6%       |
| 0.250          | 51.2 | 8 | 42.4 | 14 | 45.4   | 25 | 45.3   | 25 | 11.6%      | -6.7%      |
| 0.500          | 46.4 | 8 | 35.7 | 11 | 45.9   | 22 | 45.9   | 22 | 1.0%       | -28.5%     |

Tabela C.8: Resultados de escalonamento do GAD *diamante-256* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT  |   | LDBS |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|-------|---|------|---|--------|----|--------|----|------------|------------|
| 0.000          | 156.0 | 3 | 90.0 | 7 | 80.0   | 24 | 80.0   | 24 | 48.7%      | 11.1%      |
| 0.125          | 118.1 | 4 | 82.1 | 7 | 74.5   | 24 | 74.5   | 24 | 37.0%      | 9.3%       |
| 0.250          | 112.4 | 5 | 78.2 | 6 | 72.8   | 22 | 72.8   | 22 | 35.2%      | 6.9%       |
| 0.500          | 115.0 | 5 | 73.7 | 5 | 74.0   | 18 | 74.0   | 18 | 35.6%      | -0.4%      |

Tabela C.9: Resultados de escalonamento do GAD *diamante-256* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT  |   | LDBS  |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|-------|---|-------|---|--------|----|--------|----|------------|------------|
| 0.000          | 231.0 | 2 | 145.0 | 3 | 104.0  | 31 | 104.0  | 31 | 55.0%      | 28.3%      |
| 0.125          | 184.6 | 2 | 126.1 | 5 | 97.8   | 26 | 97.8   | 26 | 47.0%      | 22.4%      |
| 0.250          | 178.6 | 2 | 114.0 | 4 | 95.0   | 23 | 95.0   | 23 | 46.8%      | 16.7%      |
| 0.500          | 150.0 | 1 | 105.0 | 3 | 91.1   | 19 | 91.1   | 19 | 39.3%      | 13.2%      |

Tabela C.10: Resultados de escalonamento do GAD *diamante-256* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes aleatórios.

# Apêndice D

## Resultados para GADs Randômicos

| $x$ | HEFT  |    | LDBS  |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|-------|----|-------|----|--------|----|--------|----|------------|------------|
| 10  | 62.0  | 31 | 61.0  | 34 | 60.0   | 76 | 60.0   | 76 | 3.2%       | 1.6%       |
| 9   | 62.0  | 31 | 61.0  | 34 | 60.0   | 76 | 60.0   | 76 | 3.2%       | 1.6%       |
| 8   | 62.0  | 31 | 61.0  | 34 | 60.0   | 76 | 60.0   | 76 | 3.2%       | 1.6%       |
| 7   | 62.0  | 31 | 61.0  | 34 | 60.0   | 76 | 60.0   | 76 | 3.2%       | 1.6%       |
| 6   | 62.0  | 31 | 61.0  | 34 | 60.0   | 76 | 60.0   | 76 | 3.2%       | 1.6%       |
| 5   | 62.0  | 31 | 61.0  | 34 | 60.0   | 76 | 60.0   | 76 | 3.2%       | 1.6%       |
| 4   | 62.0  | 31 | 61.0  | 34 | 60.0   | 72 | 60.0   | 72 | 3.2%       | 1.6%       |
| 3   | 62.0  | 30 | 62.0  | 30 | 68.0   | 70 | 68.0   | 70 | -9.7%      | -9.7%      |
| 2   | 62.0  | 30 | 62.0  | 20 | 73.0   | 70 | 73.0   | 70 | -17.7%     | -17.7%     |
| 1   | 62.0  | 28 | 77.0  | 23 | 91.0   | 68 | 91.0   | 68 | -46.8%     | -18.2%     |
| 0   | 122.0 | 31 | 121.0 | 34 | 120.0  | 74 | 120.0  | 74 | 1.6%       | 0.8%       |

Tabela D.1: Resultados de escalonamento do GAD *randômico-15* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 32.0 | 31 | 31.0 | 34 | 30.0   | 76 | 30.0   | 76 | 6.2%       | 3.2%       |
| 9   | 32.0 | 31 | 31.0 | 34 | 30.0   | 76 | 30.0   | 76 | 6.2%       | 3.2%       |
| 8   | 32.0 | 31 | 31.0 | 34 | 30.0   | 76 | 30.0   | 76 | 6.2%       | 3.2%       |
| 7   | 32.0 | 31 | 31.0 | 34 | 30.0   | 76 | 30.0   | 76 | 6.2%       | 3.2%       |
| 6   | 32.0 | 31 | 31.0 | 34 | 30.0   | 76 | 30.0   | 76 | 6.2%       | 3.2%       |
| 5   | 32.0 | 31 | 31.0 | 34 | 30.0   | 76 | 30.0   | 76 | 6.2%       | 3.2%       |
| 4   | 32.0 | 31 | 31.0 | 34 | 31.5   | 72 | 31.5   | 72 | 1.6%       | -1.6%      |
| 3   | 32.0 | 30 | 32.0 | 30 | 39.5   | 70 | 39.5   | 70 | -23.4%     | -23.4%     |
| 2   | 32.0 | 30 | 32.0 | 20 | 38.0   | 69 | 38.0   | 69 | -18.8%     | -18.8%     |
| 1   | 32.0 | 28 | 39.5 | 23 | 46.0   | 68 | 46.0   | 68 | -43.8%     | -16.5%     |
| 0   | 62.0 | 31 | 61.0 | 34 | 60.0   | 74 | 60.0   | 74 | 3.2%       | 1.6%       |

Tabela D.2: Resultados de escalonamento do GAD *randômico-15* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 10.5 | 31 | 9.0  | 36 | 9.0    | 78 | 8.5    | 78 | 19.0%      | 5.6%       |
| 9   | 10.5 | 31 | 9.0  | 36 | 9.0    | 78 | 8.5    | 78 | 19.0%      | 5.6%       |
| 8   | 10.5 | 31 | 9.0  | 36 | 9.0    | 78 | 8.5    | 78 | 19.0%      | 5.6%       |
| 7   | 10.5 | 31 | 9.0  | 36 | 9.0    | 74 | 8.5    | 74 | 19.0%      | 5.6%       |
| 6   | 10.5 | 31 | 9.0  | 36 | 9.0    | 71 | 8.5    | 71 | 19.0%      | 5.6%       |
| 5   | 10.5 | 31 | 9.0  | 36 | 9.0    | 68 | 8.5    | 68 | 19.0%      | 5.6%       |
| 4   | 10.5 | 31 | 9.0  | 36 | 10.0   | 67 | 8.5    | 67 | 19.0%      | 5.6%       |
| 3   | 10.5 | 30 | 10.0 | 30 | 10.5   | 64 | 10.5   | 64 | 0.0%       | -5.0%      |
| 2   | 10.5 | 25 | 10.0 | 20 | 11.0   | 65 | 10.5   | 65 | 0.0%       | -5.0%      |
| 1   | 10.5 | 26 | 11.5 | 22 | 11.0   | 64 | 10.5   | 64 | 0.0%       | 8.7%       |
| 0   | 16.0 | 31 | 16.0 | 34 | 13.0   | 77 | 13.0   | 77 | 18.8%      | 18.8%      |

Tabela D.3: Resultados de escalonamento do GAD *randômico-15* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 26.0 | 31 | 20.5 | 36 | 18.0   | 75 | 18.0   | 75 | 30.8%      | 12.2%      |
| 9   | 26.0 | 31 | 20.5 | 36 | 18.0   | 75 | 18.0   | 75 | 30.8%      | 12.2%      |
| 8   | 26.0 | 31 | 20.5 | 36 | 18.0   | 75 | 18.0   | 75 | 30.8%      | 12.2%      |
| 7   | 26.0 | 31 | 20.5 | 36 | 18.0   | 74 | 18.0   | 74 | 30.8%      | 12.2%      |
| 6   | 26.0 | 31 | 20.5 | 36 | 18.0   | 70 | 18.0   | 70 | 30.8%      | 12.2%      |
| 5   | 26.0 | 31 | 20.5 | 36 | 18.0   | 67 | 18.0   | 67 | 30.8%      | 12.2%      |
| 4   | 26.0 | 31 | 20.5 | 36 | 18.0   | 64 | 18.0   | 64 | 30.8%      | 12.2%      |
| 3   | 26.0 | 30 | 22.5 | 30 | 21.5   | 62 | 18.5   | 62 | 28.8%      | 17.8%      |
| 2   | 26.0 | 25 | 24.0 | 20 | 19.0   | 62 | 19.0   | 62 | 26.9%      | 20.8%      |
| 1   | 26.0 | 26 | 21.5 | 22 | 19.5   | 66 | 19.0   | 66 | 26.9%      | 11.6%      |
| 0   | 31.0 | 31 | 28.0 | 36 | 22.0   | 79 | 22.0   | 79 | 29.0%      | 21.4%      |

Tabela D.4: Resultados de escalonamento do GAD *randômico-15* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 46.0 | 31 | 33.5 | 36 | 26.5   | 69 | 26.5   | 69 | 42.4%      | 20.9%      |
| 9   | 46.0 | 31 | 33.5 | 36 | 26.5   | 69 | 26.5   | 69 | 42.4%      | 20.9%      |
| 8   | 46.0 | 31 | 33.5 | 36 | 26.5   | 69 | 26.5   | 69 | 42.4%      | 20.9%      |
| 7   | 46.0 | 31 | 33.5 | 36 | 26.5   | 69 | 26.5   | 69 | 42.4%      | 20.9%      |
| 6   | 46.0 | 31 | 33.5 | 36 | 26.5   | 68 | 26.5   | 68 | 42.4%      | 20.9%      |
| 5   | 46.0 | 31 | 33.5 | 36 | 27.0   | 62 | 27.0   | 62 | 41.3%      | 19.4%      |
| 4   | 46.0 | 31 | 33.5 | 36 | 26.5   | 61 | 26.5   | 61 | 42.4%      | 20.9%      |
| 3   | 46.0 | 30 | 31.0 | 30 | 27.0   | 58 | 27.0   | 58 | 41.3%      | 12.9%      |
| 2   | 46.0 | 25 | 33.0 | 20 | 27.5   | 56 | 27.5   | 56 | 40.2%      | 16.7%      |
| 1   | 46.0 | 26 | 34.0 | 22 | 28.0   | 54 | 27.5   | 54 | 40.2%      | 19.1%      |
| 0   | 52.0 | 31 | 41.0 | 36 | 30.0   | 79 | 30.0   | 79 | 42.3%      | 26.8%      |

Tabela D.5: Resultados de escalonamento do GAD *randômico-15* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes duais.



| $\bar{\sigma}$ | HEFT  |    | LDBS  |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|-------|----|-------|----|--------|----|--------|----|------------|------------|
| 0.000          | 122.0 | 31 | 121.0 | 34 | 120.0  | 74 | 120.0  | 74 | 1.6%       | 0.8%       |
| 0.125          | 104.0 | 31 | 103.1 | 33 | 102.3  | 76 | 102.3  | 76 | 1.7%       | 0.8%       |
| 0.250          | 90.7  | 31 | 89.8  | 31 | 96.1   | 72 | 96.1   | 72 | -6.0%      | -7.0%      |
| 0.500          | 72.4  | 31 | 71.5  | 31 | 82.0   | 71 | 82.0   | 71 | -13.4%     | -14.8%     |

Tabela D.6: Resultados de escalonamento do GAD *randômico-15* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 62.0 | 31 | 61.0 | 34 | 60.0   | 74 | 60.0   | 74 | 3.2%       | 1.6%       |
| 0.125          | 53.0 | 31 | 52.1 | 34 | 51.1   | 76 | 51.1   | 76 | 3.5%       | 1.8%       |
| 0.250          | 46.3 | 31 | 45.4 | 32 | 44.4   | 75 | 44.4   | 75 | 4.2%       | 2.2%       |
| 0.500          | 37.2 | 31 | 36.2 | 32 | 41.0   | 71 | 41.0   | 71 | -10.3%     | -13.2%     |

Tabela D.7: Resultados de escalonamento do GAD *randômico-15* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 16.0 | 31 | 16.0 | 34 | 13.0   | 77 | 13.0   | 77 | 18.8%      | 18.8%      |
| 0.125          | 13.4 | 31 | 14.0 | 34 | 11.7   | 69 | 11.6   | 69 | 13.2%      | 17.0%      |
| 0.250          | 12.1 | 31 | 12.4 | 34 | 11.3   | 70 | 11.3   | 70 | 6.6%       | 8.6%       |
| 0.500          | 10.9 | 31 | 10.2 | 35 | 11.0   | 65 | 10.2   | 65 | 5.7%       | -0.3%      |

Tabela D.8: Resultados de escalonamento do GAD *randômico-15* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 31.0 | 31 | 28.0 | 36 | 22.0   | 79 | 22.0   | 79 | 29.0%      | 21.4%      |
| 0.125          | 32.7 | 31 | 26.2 | 36 | 22.1   | 70 | 21.2   | 70 | 35.2%      | 19.2%      |
| 0.250          | 28.9 | 31 | 24.2 | 36 | 22.0   | 69 | 19.7   | 69 | 31.8%      | 18.6%      |
| 0.500          | 27.1 | 31 | 21.8 | 36 | 20.3   | 61 | 20.3   | 61 | 25.3%      | 7.1%       |

Tabela D.9: Resultados de escalonamento do GAD *randômico-15* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 52.0 | 31 | 41.0 | 36 | 30.0   | 79 | 30.0   | 79 | 42.3%      | 26.8%      |
| 0.125          | 49.4 | 31 | 43.6 | 36 | 28.7   | 72 | 28.7   | 72 | 41.8%      | 34.1%      |
| 0.250          | 48.2 | 31 | 37.1 | 36 | 30.0   | 65 | 28.5   | 65 | 40.9%      | 23.2%      |
| 0.500          | 46.5 | 31 | 34.3 | 36 | 39.8   | 61 | 27.1   | 61 | 41.8%      | 21.1%      |

Tabela D.10: Resultados de escalonamento do GAD *randômico-15* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes aleatórios.

# Apêndice E

## Resultados para GADs representantes da Eliminação de Gauss

| $x$ | HEFT |   | LDBS |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|---|------|---|--------|----|--------|----|------------|------------|
| 10  | 32.5 | 7 | 32.5 | 7 | 31.5   | 22 | 31.5   | 22 | 3.1%       | 3.1%       |
| 9   | 32.5 | 7 | 32.5 | 7 | 31.5   | 22 | 31.5   | 22 | 3.1%       | 3.1%       |
| 8   | 32.5 | 7 | 32.5 | 7 | 31.5   | 22 | 31.5   | 22 | 3.1%       | 3.1%       |
| 7   | 32.5 | 7 | 32.5 | 7 | 31.5   | 22 | 31.5   | 22 | 3.1%       | 3.1%       |
| 6   | 32.5 | 7 | 32.5 | 7 | 31.5   | 22 | 31.5   | 22 | 3.1%       | 3.1%       |
| 5   | 32.5 | 7 | 32.5 | 7 | 31.5   | 22 | 31.5   | 22 | 3.1%       | 3.1%       |
| 4   | 32.5 | 7 | 32.5 | 7 | 31.5   | 22 | 31.5   | 22 | 3.1%       | 3.1%       |
| 3   | 32.5 | 7 | 32.5 | 7 | 31.5   | 19 | 31.5   | 19 | 3.1%       | 3.1%       |
| 2   | 32.5 | 7 | 32.5 | 7 | 34.5   | 18 | 34.5   | 18 | -6.2%      | -6.2%      |
| 1   | 32.5 | 5 | 34.5 | 5 | 35.5   | 15 | 35.5   | 15 | -9.2%      | -2.9%      |
| 0   | 63.0 | 7 | 63.0 | 7 | 63.0   | 22 | 63.0   | 22 | 0.0%       | 0.0%       |

Tabela E.1: Resultados de escalonamento do GAD *gauss-35* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT |   | LDBS |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|---|------|---|--------|----|--------|----|------------|------------|
| 10  | 53.5 | 6 | 45.0 | 7 | 43.0   | 15 | 43.0   | 15 | 19.6%      | 4.4%       |
| 9   | 53.5 | 6 | 45.0 | 7 | 43.0   | 15 | 43.0   | 15 | 19.6%      | 4.4%       |
| 8   | 53.5 | 6 | 45.0 | 7 | 43.0   | 15 | 43.0   | 15 | 19.6%      | 4.4%       |
| 7   | 53.5 | 6 | 45.0 | 7 | 43.0   | 15 | 43.0   | 15 | 19.6%      | 4.4%       |
| 6   | 53.5 | 6 | 45.0 | 7 | 43.0   | 15 | 43.0   | 15 | 19.6%      | 4.4%       |
| 5   | 53.5 | 6 | 45.0 | 7 | 43.0   | 15 | 43.0   | 15 | 19.6%      | 4.4%       |
| 4   | 53.5 | 6 | 45.0 | 7 | 43.0   | 15 | 43.0   | 15 | 19.6%      | 4.4%       |
| 3   | 53.5 | 6 | 45.0 | 7 | 43.0   | 15 | 43.0   | 15 | 19.6%      | 4.4%       |
| 2   | 53.5 | 6 | 45.0 | 7 | 48.5   | 13 | 48.5   | 13 | 9.3%       | -7.8%      |
| 1   | 53.5 | 5 | 48.5 | 5 | 49.0   | 10 | 49.0   | 10 | 8.4%       | -1.0%      |
| 0   | 78.0 | 7 | 77.0 | 7 | 68.0   | 21 | 68.0   | 21 | 12.8%      | 11.7%      |

**Tabela E.2:** Resultados de escalonamento do GAD *gauss-35* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes duais.

| $x$ | HEFT  |   | LDBS |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|-------|---|------|---|--------|----|--------|----|------------|------------|
| 10  | 71.5  | 6 | 56.5 | 7 | 60.0   | 10 | 58.5   | 10 | 18.2%      | -3.5%      |
| 9   | 71.5  | 6 | 56.5 | 7 | 60.0   | 10 | 58.5   | 10 | 18.2%      | -3.5%      |
| 8   | 71.5  | 6 | 56.5 | 7 | 60.0   | 10 | 58.5   | 10 | 18.2%      | -3.5%      |
| 7   | 71.5  | 6 | 56.5 | 7 | 60.0   | 10 | 58.5   | 10 | 18.2%      | -3.5%      |
| 6   | 71.5  | 6 | 56.5 | 7 | 60.0   | 10 | 58.5   | 10 | 18.2%      | -3.5%      |
| 5   | 71.5  | 6 | 56.5 | 7 | 60.0   | 10 | 58.5   | 10 | 18.2%      | -3.5%      |
| 4   | 71.5  | 6 | 56.5 | 7 | 60.0   | 10 | 58.5   | 10 | 18.2%      | -3.5%      |
| 3   | 71.5  | 6 | 56.5 | 7 | 60.0   | 10 | 58.5   | 10 | 18.2%      | -3.5%      |
| 2   | 71.5  | 6 | 56.5 | 7 | 60.0   | 10 | 58.5   | 10 | 18.2%      | -3.5%      |
| 1   | 61.5  | 5 | 60.0 | 5 | 58.5   | 5  | 58.5   | 5  | 4.9%       | 2.5%       |
| 0   | 107.0 | 6 | 90.0 | 7 | 86.0   | 13 | 84.0   | 13 | 21.5%      | 6.7%       |

**Tabela E.3:** Resultados de escalonamento do GAD *gauss-35* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes duais.

| $\bar{\sigma}$ | HEFT |   | LDBS |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|---|------|---|--------|----|--------|----|------------|------------|
| 0.000          | 63.0 | 7 | 63.0 | 7 | 63.0   | 22 | 63.0   | 22 | 0.0%       | 0.0%       |
| 0.125          | 54.0 | 7 | 54.0 | 7 | 53.7   | 22 | 53.7   | 22 | 0.6%       | 0.6%       |
| 0.250          | 47.1 | 7 | 47.1 | 7 | 46.6   | 22 | 46.6   | 22 | 1.1%       | 1.1%       |
| 0.500          | 37.7 | 7 | 37.7 | 7 | 36.9   | 22 | 36.9   | 22 | 2.2%       | 2.2%       |

Tabela E.4: Resultados de escalonamento do GAD *gauss-35* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |   | LDBS |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|---|------|---|--------|----|--------|----|------------|------------|
| 0.000          | 78.0 | 7 | 77.0 | 7 | 68.0   | 21 | 68.0   | 21 | 12.8%      | 11.7%      |
| 0.125          | 70.9 | 7 | 67.1 | 7 | 61.0   | 19 | 60.6   | 19 | 14.5%      | 9.6%       |
| 0.250          | 65.1 | 7 | 60.7 | 7 | 56.8   | 18 | 55.8   | 18 | 14.3%      | 8.0%       |
| 0.500          | 58.4 | 6 | 51.3 | 7 | 48.0   | 15 | 48.0   | 15 | 17.8%      | 6.4%       |

Tabela E.5: Resultados de escalonamento do GAD *gauss-35* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT  |   | LDBS |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|-------|---|------|---|--------|----|--------|----|------------|------------|
| 0.000          | 107.0 | 6 | 90.0 | 7 | 86.0   | 13 | 84.0   | 13 | 21.5%      | 6.7%       |
| 0.125          | 98.3  | 6 | 82.8 | 7 | 79.6   | 11 | 76.1   | 11 | 22.6%      | 8.1%       |
| 0.250          | 83.3  | 6 | 74.4 | 7 | 76.6   | 10 | 68.8   | 10 | 17.4%      | 7.5%       |
| 0.500          | 69.9  | 6 | 62.8 | 7 | 66.9   | 10 | 61.1   | 10 | 12.5%      | 2.7%       |

Tabela E.6: Resultados de escalonamento do GAD *gauss-35* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes aleatórios.

# Apêndice F

## Resultados para o GAD representante do Problema da Dinâmica Molecular

| $x$ | HEFT  |   | LDBS  |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|-------|---|-------|---|--------|----|--------|----|------------|------------|
| 10  | 56.0  | 7 | 56.0  | 9 | 55.0   | 20 | 55.0   | 20 | 1.8%       | 1.8%       |
| 9   | 56.0  | 7 | 56.0  | 9 | 55.0   | 20 | 55.0   | 20 | 1.8%       | 1.8%       |
| 8   | 56.0  | 7 | 56.0  | 9 | 55.0   | 20 | 55.0   | 20 | 1.8%       | 1.8%       |
| 7   | 56.0  | 7 | 56.0  | 9 | 55.0   | 20 | 55.0   | 20 | 1.8%       | 1.8%       |
| 6   | 56.0  | 7 | 56.0  | 9 | 55.0   | 18 | 55.0   | 18 | 1.8%       | 1.8%       |
| 5   | 56.0  | 7 | 56.0  | 9 | 60.0   | 16 | 60.0   | 16 | -7.1%      | -7.1%      |
| 4   | 56.0  | 7 | 56.0  | 9 | 64.0   | 14 | 64.0   | 14 | -14.3%     | -14.3%     |
| 3   | 56.0  | 7 | 56.0  | 9 | 64.0   | 13 | 64.0   | 13 | -14.3%     | -14.3%     |
| 2   | 59.0  | 6 | 58.0  | 6 | 64.0   | 10 | 64.0   | 10 | -8.5%      | -10.3%     |
| 1   | 75.0  | 5 | 75.0  | 5 | 75.0   | 12 | 75.0   | 12 | 0.0%       | 0.0%       |
| 0   | 106.0 | 7 | 106.0 | 9 | 105.0  | 22 | 105.0  | 22 | 0.9%       | 0.9%       |

**Tabela F.1:** Resultados de escalonamento do GAD *dinâmica molecular* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes duais.

| $x$ | HEFT |   | LDBS |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|---|------|---|--------|----|--------|----|------------|------------|
| 10  | 32.0 | 7 | 31.0 | 9 | 30.0   | 20 | 30.0   | 20 | 6.2%       | 3.2%       |
| 9   | 32.0 | 7 | 31.0 | 9 | 30.0   | 20 | 30.0   | 20 | 6.2%       | 3.2%       |
| 8   | 32.0 | 7 | 31.0 | 9 | 30.0   | 20 | 30.0   | 20 | 6.2%       | 3.2%       |
| 7   | 32.0 | 7 | 31.0 | 9 | 30.0   | 20 | 30.0   | 20 | 6.2%       | 3.2%       |
| 6   | 32.0 | 7 | 31.0 | 9 | 30.0   | 18 | 30.0   | 18 | 6.2%       | 3.2%       |
| 5   | 32.0 | 7 | 31.0 | 9 | 32.5   | 16 | 32.5   | 16 | -1.6%      | -4.8%      |
| 4   | 32.0 | 7 | 31.0 | 9 | 32.5   | 14 | 32.5   | 14 | -1.6%      | -4.8%      |
| 3   | 32.0 | 7 | 31.0 | 9 | 34.0   | 13 | 34.0   | 13 | -6.2%      | -9.7%      |
| 2   | 33.0 | 6 | 31.5 | 6 | 34.0   | 10 | 34.0   | 10 | -3.0%      | -7.9%      |
| 1   | 38.5 | 5 | 38.0 | 5 | 40.0   | 12 | 40.0   | 12 | -3.9%      | -5.3%      |
| 0   | 56.0 | 7 | 56.0 | 9 | 55.0   | 22 | 55.0   | 22 | 1.8%       | 1.8%       |

**Tabela F.2:** Resultados de escalonamento do GAD *dinâmica molecular* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes duais.

| $x$ | HEFT |   | LDBS |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|---|------|---|--------|----|--------|----|------------|------------|
| 10  | 12.0 | 4 | 9.5  | 8 | 9.0    | 14 | 9.0    | 14 | 25.0%      | 5.3%       |
| 9   | 12.0 | 4 | 9.5  | 8 | 9.0    | 14 | 9.0    | 14 | 25.0%      | 5.3%       |
| 8   | 12.0 | 4 | 9.5  | 8 | 9.0    | 14 | 9.0    | 14 | 25.0%      | 5.3%       |
| 7   | 12.0 | 4 | 9.5  | 8 | 9.0    | 14 | 9.0    | 14 | 25.0%      | 5.3%       |
| 6   | 12.0 | 4 | 9.5  | 8 | 9.0    | 14 | 9.0    | 14 | 25.0%      | 5.3%       |
| 5   | 12.0 | 4 | 9.5  | 8 | 9.0    | 14 | 9.0    | 14 | 25.0%      | 5.3%       |
| 4   | 12.0 | 4 | 9.5  | 8 | 10.0   | 14 | 10.0   | 14 | 16.7%      | -5.3%      |
| 3   | 12.0 | 4 | 9.5  | 8 | 10.0   | 11 | 10.0   | 11 | 16.7%      | -5.3%      |
| 2   | 12.0 | 4 | 9.5  | 6 | 11.0   | 12 | 11.0   | 12 | 8.3%       | -15.8%     |
| 1   | 12.5 | 3 | 10.5 | 4 | 11.5   | 10 | 11.5   | 10 | 8.0%       | -9.5%      |
| 0   | 18.0 | 5 | 16.0 | 8 | 15.0   | 16 | 15.0   | 16 | 16.7%      | 6.2%       |

**Tabela F.3:** Resultados de escalonamento do GAD *dinâmica molecular* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|-----|------|---|------|---|--------|---|--------|---|------------|------------|
| 10  | 20.5 | 1 | 17.5 | 3 | 15.0   | 7 | 15.0   | 7 | 26.8%      | 14.3%      |
| 9   | 20.5 | 1 | 17.5 | 3 | 15.0   | 7 | 15.0   | 7 | 26.8%      | 14.3%      |
| 8   | 20.5 | 1 | 17.5 | 3 | 15.0   | 7 | 15.0   | 7 | 26.8%      | 14.3%      |
| 7   | 20.5 | 1 | 17.5 | 3 | 15.0   | 7 | 15.0   | 7 | 26.8%      | 14.3%      |
| 6   | 20.5 | 1 | 17.5 | 3 | 15.0   | 7 | 15.0   | 7 | 26.8%      | 14.3%      |
| 5   | 20.5 | 1 | 17.5 | 3 | 15.0   | 7 | 15.0   | 7 | 26.8%      | 14.3%      |
| 4   | 20.5 | 1 | 17.5 | 3 | 15.0   | 7 | 15.0   | 7 | 26.8%      | 14.3%      |
| 3   | 20.5 | 1 | 17.5 | 3 | 15.0   | 7 | 15.0   | 7 | 26.8%      | 14.3%      |
| 2   | 20.5 | 1 | 17.5 | 3 | 15.5   | 7 | 15.5   | 7 | 24.4%      | 11.4%      |
| 1   | 20.5 | 1 | 17.5 | 3 | 15.5   | 6 | 15.5   | 6 | 24.4%      | 11.4%      |
| 0   | 36.0 | 3 | 25.0 | 4 | 22.0   | 8 | 22.0   | 8 | 38.9%      | 12.0%      |

**Tabela F.4:** Resultados de escalonamento do GAD *dinâmica molecular* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes duais.

| $x$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|-----|------|---|------|---|--------|---|--------|---|------------|------------|
| 10  | 20.5 | 1 | 21.0 | 3 | 19.0   | 5 | 19.0   | 5 | 7.3%       | 9.5%       |
| 9   | 20.5 | 1 | 21.0 | 3 | 19.0   | 5 | 19.0   | 5 | 7.3%       | 9.5%       |
| 8   | 20.5 | 1 | 21.0 | 3 | 19.0   | 5 | 19.0   | 5 | 7.3%       | 9.5%       |
| 7   | 20.5 | 1 | 21.0 | 3 | 19.0   | 5 | 19.0   | 5 | 7.3%       | 9.5%       |
| 6   | 20.5 | 1 | 21.0 | 3 | 19.0   | 5 | 19.0   | 5 | 7.3%       | 9.5%       |
| 5   | 20.5 | 1 | 21.0 | 3 | 19.0   | 5 | 19.0   | 5 | 7.3%       | 9.5%       |
| 4   | 20.5 | 1 | 21.0 | 3 | 19.0   | 5 | 19.0   | 5 | 7.3%       | 9.5%       |
| 3   | 20.5 | 1 | 21.0 | 3 | 19.0   | 5 | 19.0   | 5 | 7.3%       | 9.5%       |
| 2   | 20.5 | 1 | 21.0 | 3 | 19.0   | 5 | 19.0   | 5 | 7.3%       | 9.5%       |
| 1   | 20.5 | 1 | 21.0 | 3 | 20.0   | 3 | 19.0   | 3 | 7.3%       | 9.5%       |
| 0   | 41.0 | 1 | 35.0 | 3 | 28.0   | 6 | 28.0   | 6 | 31.7%      | 20.0%      |

Tabela F.5: Resultados de escalonamento do GAD *dinâmica molecular* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes duais.

| $\bar{\sigma}$ | HEFT  |   | LDBS  |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|-------|---|-------|---|--------|----|--------|----|------------|------------|
| 0.000          | 106.0 | 7 | 106.0 | 9 | 105.0  | 22 | 105.0  | 22 | 0.9%       | 0.9%       |
| 0.125          | 93.6  | 7 | 92.1  | 9 | 101.0  | 14 | 101.0  | 14 | -7.8%      | -9.7%      |
| 0.250          | 83.5  | 7 | 81.8  | 8 | 91.3   | 14 | 91.3   | 14 | -9.4%      | -11.6%     |
| 0.500          | 69.9  | 7 | 68.4  | 8 | 76.3   | 14 | 76.3   | 14 | -9.2%      | -11.5%     |

Tabela F.6: Resultados de escalonamento do GAD *dinâmica molecular* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |   | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|---|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 56.0 | 7 | 56.0 | 9  | 55.0   | 22 | 55.0   | 22 | 1.8%       | 1.8%       |
| 0.125          | 50.3 | 7 | 49.0 | 10 | 48.6   | 14 | 48.6   | 14 | 3.4%       | 0.8%       |
| 0.250          | 45.2 | 7 | 43.8 | 9  | 47.8   | 13 | 47.8   | 13 | -5.6%      | -9.2%      |
| 0.500          | 38.2 | 7 | 37.0 | 9  | 40.2   | 12 | 40.2   | 12 | -5.3%      | -8.7%      |

Tabela F.7: Resultados de escalonamento do GAD *dinâmica molecular* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |   | LDBS |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|---|------|---|--------|----|--------|----|------------|------------|
| 0.000          | 18.0 | 5 | 16.0 | 8 | 15.0   | 16 | 15.0   | 16 | 16.7%      | 6.2%       |
| 0.125          | 16.1 | 7 | 13.9 | 9 | 13.6   | 15 | 13.6   | 15 | 15.8%      | 2.7%       |
| 0.250          | 14.2 | 7 | 12.5 | 9 | 12.6   | 14 | 12.6   | 14 | 11.6%      | -1.0%      |
| 0.500          | 13.2 | 5 | 10.9 | 9 | 11.9   | 15 | 11.0   | 15 | 17.0%      | -1.0%      |

Tabela F.8: Resultados de escalonamento do GAD *dinâmica molecular* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|----------------|------|---|------|---|--------|---|--------|---|------------|------------|
| 0.000          | 36.0 | 3 | 25.0 | 4 | 22.0   | 8 | 22.0   | 8 | 38.9%      | 12.0%      |
| 0.125          | 32.1 | 3 | 22.9 | 4 | 20.5   | 7 | 20.3   | 7 | 36.7%      | 11.1%      |
| 0.250          | 30.1 | 3 | 21.3 | 5 | 19.4   | 7 | 19.0   | 7 | 37.0%      | 10.8%      |
| 0.500          | 24.6 | 1 | 19.1 | 4 | 16.1   | 7 | 16.1   | 7 | 34.6%      | 15.8%      |

**Tabela F.9:** Resultados de escalonamento do GAD *dinâmica molecular* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|----------------|------|---|------|---|--------|---|--------|---|------------|------------|
| 0.000          | 41.0 | 1 | 35.0 | 3 | 28.0   | 6 | 28.0   | 6 | 31.7%      | 20.0%      |
| 0.125          | 34.9 | 1 | 31.1 | 3 | 25.4   | 5 | 25.4   | 5 | 27.2%      | 18.3%      |
| 0.250          | 30.6 | 1 | 26.4 | 3 | 24.2   | 5 | 24.2   | 5 | 20.9%      | 8.6%       |
| 0.500          | 24.6 | 1 | 23.3 | 3 | 20.9   | 5 | 20.9   | 5 | 15.3%      | 10.4%      |

**Tabela F.10:** Resultados de escalonamento do GAD *dinâmica molecular* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes aleatórios.



# Apêndice G

## Resultados para os GADs PSG

| $x$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|-----|------|---|------|---|--------|---|--------|---|------------|------------|
| 10  | 29.0 | 4 | 26.0 | 4 | 24.0   | 6 | 24.0   | 6 | 17.2%      | 7.7%       |
| 9   | 29.0 | 4 | 26.0 | 4 | 24.0   | 6 | 24.0   | 6 | 17.2%      | 7.7%       |
| 8   | 29.0 | 4 | 26.0 | 4 | 24.0   | 6 | 24.0   | 6 | 17.2%      | 7.7%       |
| 7   | 29.0 | 4 | 26.0 | 4 | 24.0   | 6 | 24.0   | 6 | 17.2%      | 7.7%       |
| 6   | 29.0 | 4 | 26.0 | 4 | 24.0   | 6 | 24.0   | 6 | 17.2%      | 7.7%       |
| 5   | 29.0 | 4 | 26.0 | 4 | 24.0   | 6 | 24.0   | 6 | 17.2%      | 7.7%       |
| 4   | 29.0 | 4 | 26.0 | 4 | 24.0   | 6 | 24.0   | 6 | 17.2%      | 7.7%       |
| 3   | 29.0 | 4 | 26.0 | 4 | 24.0   | 6 | 24.0   | 6 | 17.2%      | 7.7%       |
| 2   | 29.0 | 4 | 26.0 | 4 | 29.5   | 5 | 28.0   | 5 | 3.4%       | -7.7%      |
| 1   | 34.0 | 3 | 29.5 | 2 | 31.0   | 3 | 30.5   | 3 | 10.3%      | -3.4%      |
| 0   | 45.0 | 4 | 45.0 | 5 | 40.0   | 6 | 40.0   | 6 | 11.1%      | 11.1%      |

Tabela G.1: Resultados de escalonamento do GAD PSG “al-maasarani” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|-----|------|---|------|---|--------|---|--------|---|------------|------------|
| 10  | 26.0 | 5 | 21.0 | 5 | 19.5   | 6 | 19.0   | 6 | 26.9%      | 9.5%       |
| 9   | 26.0 | 5 | 21.0 | 5 | 19.5   | 6 | 19.0   | 6 | 26.9%      | 9.5%       |
| 8   | 26.0 | 5 | 21.0 | 5 | 19.5   | 6 | 19.0   | 6 | 26.9%      | 9.5%       |
| 7   | 26.0 | 5 | 21.0 | 5 | 19.5   | 6 | 19.0   | 6 | 26.9%      | 9.5%       |
| 6   | 26.0 | 5 | 21.0 | 5 | 19.5   | 6 | 19.0   | 6 | 26.9%      | 9.5%       |
| 5   | 26.0 | 5 | 21.0 | 5 | 19.5   | 6 | 19.0   | 6 | 26.9%      | 9.5%       |
| 4   | 26.0 | 5 | 21.0 | 5 | 19.5   | 6 | 19.0   | 6 | 26.9%      | 9.5%       |
| 3   | 26.0 | 5 | 21.0 | 5 | 19.5   | 6 | 19.5   | 6 | 25.0%      | 7.1%       |
| 2   | 26.0 | 5 | 21.5 | 4 | 21.0   | 5 | 20.5   | 5 | 21.2%      | 4.7%       |
| 1   | 24.5 | 5 | 24.0 | 2 | 24.0   | 5 | 23.5   | 5 | 4.1%       | 2.1%       |
| 0   | 40.0 | 7 | 37.0 | 6 | 35.0   | 8 | 35.0   | 8 | 12.5%      | 5.4%       |

Tabela G.2: Resultados de escalonamento do GAD PSG “al-mouhamed” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT  |   | LDBS  |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|-----|-------|---|-------|---|--------|---|--------|---|------------|------------|
| 10  | 260.0 | 4 | 205.0 | 5 | 210.0  | 7 | 210.0  | 7 | 19.2%      | -2.4%      |
| 9   | 260.0 | 4 | 205.0 | 5 | 210.0  | 7 | 210.0  | 7 | 19.2%      | -2.4%      |
| 8   | 260.0 | 4 | 205.0 | 5 | 210.0  | 7 | 210.0  | 7 | 19.2%      | -2.4%      |
| 7   | 260.0 | 4 | 205.0 | 5 | 210.0  | 7 | 210.0  | 7 | 19.2%      | -2.4%      |
| 6   | 260.0 | 4 | 205.0 | 5 | 210.0  | 7 | 210.0  | 7 | 19.2%      | -2.4%      |
| 5   | 260.0 | 4 | 205.0 | 5 | 210.0  | 7 | 210.0  | 7 | 19.2%      | -2.4%      |
| 4   | 260.0 | 4 | 205.0 | 5 | 210.0  | 7 | 210.0  | 7 | 19.2%      | -2.4%      |
| 3   | 260.0 | 4 | 205.0 | 5 | 210.0  | 7 | 210.0  | 7 | 19.2%      | -2.4%      |
| 2   | 260.0 | 4 | 215.0 | 4 | 210.0  | 6 | 210.0  | 6 | 19.2%      | 2.3%       |
| 1   | 260.0 | 3 | 230.0 | 2 | 225.0  | 3 | 225.0  | 3 | 13.5%      | 2.2%       |
| 0   | 390.0 | 5 | 360.0 | 5 | 320.0  | 9 | 320.0  | 9 | 17.9%      | 11.1%      |

**Tabela G.3:** Resultados de escalonamento do GAD PSG “g4” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|-----|------|---|------|---|--------|---|--------|---|------------|------------|
| 10  | 5.5  | 1 | 9.0  | 3 | 5.5    | 1 | 5.5    | 1 | 0.0%       | 38.9%      |
| 9   | 5.5  | 1 | 9.0  | 3 | 5.5    | 1 | 5.5    | 1 | 0.0%       | 38.9%      |
| 8   | 5.5  | 1 | 9.0  | 3 | 5.5    | 1 | 5.5    | 1 | 0.0%       | 38.9%      |
| 7   | 5.5  | 1 | 9.0  | 3 | 5.5    | 1 | 5.5    | 1 | 0.0%       | 38.9%      |
| 6   | 5.5  | 1 | 9.0  | 3 | 5.5    | 1 | 5.5    | 1 | 0.0%       | 38.9%      |
| 5   | 5.5  | 1 | 9.0  | 3 | 5.5    | 1 | 5.5    | 1 | 0.0%       | 38.9%      |
| 4   | 5.5  | 1 | 9.0  | 3 | 5.5    | 1 | 5.5    | 1 | 0.0%       | 38.9%      |
| 3   | 5.5  | 1 | 9.0  | 3 | 5.5    | 1 | 5.5    | 1 | 0.0%       | 38.9%      |
| 2   | 5.5  | 1 | 9.0  | 3 | 5.5    | 1 | 5.5    | 1 | 0.0%       | 38.9%      |
| 1   | 5.5  | 1 | 9.0  | 2 | 5.5    | 1 | 5.5    | 1 | 0.0%       | 38.9%      |
| 0   | 11.0 | 1 | 12.0 | 3 | 11.0   | 1 | 10.0   | 1 | 9.1%       | 16.7%      |

**Tabela G.4:** Resultados de escalonamento do GAD PSG “kruatrachuefig8” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|-----|------|---|------|---|--------|---|--------|---|------------|------------|
| 10  | 19.5 | 1 | 19.5 | 1 | 12.5   | 4 | 12.5   | 4 | 35.9%      | 35.9%      |
| 9   | 19.5 | 1 | 19.5 | 1 | 12.5   | 4 | 12.5   | 4 | 35.9%      | 35.9%      |
| 8   | 19.5 | 1 | 19.5 | 1 | 12.5   | 4 | 12.5   | 4 | 35.9%      | 35.9%      |
| 7   | 19.5 | 1 | 19.5 | 1 | 12.5   | 4 | 12.5   | 4 | 35.9%      | 35.9%      |
| 6   | 19.5 | 1 | 19.5 | 1 | 12.5   | 4 | 12.5   | 4 | 35.9%      | 35.9%      |
| 5   | 19.5 | 1 | 19.5 | 1 | 12.5   | 4 | 12.5   | 4 | 35.9%      | 35.9%      |
| 4   | 19.5 | 1 | 19.5 | 1 | 12.5   | 4 | 12.5   | 4 | 35.9%      | 35.9%      |
| 3   | 19.5 | 1 | 19.5 | 1 | 12.5   | 4 | 12.5   | 4 | 35.9%      | 35.9%      |
| 2   | 19.5 | 1 | 19.5 | 1 | 12.5   | 4 | 12.5   | 4 | 35.9%      | 35.9%      |
| 1   | 19.5 | 1 | 19.5 | 1 | 14.0   | 3 | 14.0   | 3 | 28.2%      | 28.2%      |
| 0   | 33.0 | 2 | 35.0 | 2 | 25.0   | 4 | 25.0   | 4 | 24.2%      | 28.6%      |

**Tabela G.5:** Resultados de escalonamento do GAD PSG “lctd-eg1” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|-----|------|---|------|---|--------|---|--------|---|------------|------------|
| 10  | 8.0  | 3 | 7.5  | 3 | 7.5    | 4 | 7.5    | 4 | 6.2%       | 0.0%       |
| 9   | 8.0  | 3 | 7.5  | 3 | 7.5    | 4 | 7.5    | 4 | 6.2%       | 0.0%       |
| 8   | 8.0  | 3 | 7.5  | 3 | 7.5    | 4 | 7.5    | 4 | 6.2%       | 0.0%       |
| 7   | 8.0  | 3 | 7.5  | 3 | 7.5    | 4 | 7.5    | 4 | 6.2%       | 0.0%       |
| 6   | 8.0  | 3 | 7.5  | 3 | 7.5    | 4 | 7.5    | 4 | 6.2%       | 0.0%       |
| 5   | 8.0  | 3 | 7.5  | 3 | 7.5    | 4 | 7.5    | 4 | 6.2%       | 0.0%       |
| 4   | 8.0  | 3 | 7.5  | 3 | 7.5    | 4 | 7.5    | 4 | 6.2%       | 0.0%       |
| 3   | 8.0  | 3 | 7.5  | 3 | 7.5    | 4 | 7.5    | 4 | 6.2%       | 0.0%       |
| 2   | 8.0  | 3 | 7.5  | 3 | 7.5    | 4 | 7.5    | 4 | 6.2%       | 0.0%       |
| 1   | 8.0  | 2 | 8.0  | 2 | 9.0    | 2 | 9.0    | 2 | -12.5%     | -12.5%     |
| 0   | 14.0 | 3 | 12.0 | 3 | 12.0   | 4 | 12.0   | 4 | 14.3%      | 0.0%       |

**Tabela G.6:** Resultados de escalonamento do GAD PSG “lwb” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT  |   | LDBS  |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|-----|-------|---|-------|---|--------|---|--------|---|------------|------------|
| 10  | 79.5  | 1 | 70.0  | 2 | 66.0   | 2 | 66.0   | 2 | 17.0%      | 5.7%       |
| 9   | 79.5  | 1 | 70.0  | 2 | 66.0   | 2 | 66.0   | 2 | 17.0%      | 5.7%       |
| 8   | 79.5  | 1 | 70.0  | 2 | 66.0   | 2 | 66.0   | 2 | 17.0%      | 5.7%       |
| 7   | 79.5  | 1 | 70.0  | 2 | 66.0   | 2 | 66.0   | 2 | 17.0%      | 5.7%       |
| 6   | 79.5  | 1 | 70.0  | 2 | 66.0   | 2 | 66.0   | 2 | 17.0%      | 5.7%       |
| 5   | 79.5  | 1 | 70.0  | 2 | 66.0   | 2 | 66.0   | 2 | 17.0%      | 5.7%       |
| 4   | 79.5  | 1 | 70.0  | 2 | 66.0   | 2 | 66.0   | 2 | 17.0%      | 5.7%       |
| 3   | 79.5  | 1 | 70.0  | 2 | 66.0   | 2 | 66.0   | 2 | 17.0%      | 5.7%       |
| 2   | 79.5  | 1 | 70.0  | 2 | 66.0   | 2 | 66.0   | 2 | 17.0%      | 5.7%       |
| 1   | 79.5  | 1 | 70.0  | 2 | 66.0   | 2 | 66.0   | 2 | 17.0%      | 5.7%       |
| 0   | 180.0 | 2 | 137.0 | 3 | 130.0  | 3 | 127.0  | 3 | 29.4%      | 7.3%       |

**Tabela G.7:** Resultados de escalonamento do GAD PSG “mccreary1” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT  |   | LDBS  |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|-----|-------|---|-------|---|--------|---|--------|---|------------|------------|
| 10  | 346.5 | 2 | 213.0 | 3 | 118.5  | 8 | 118.5  | 8 | 65.8%      | 44.4%      |
| 9   | 346.5 | 2 | 213.0 | 3 | 118.5  | 8 | 118.5  | 8 | 65.8%      | 44.4%      |
| 8   | 346.5 | 2 | 213.0 | 3 | 118.5  | 8 | 118.5  | 8 | 65.8%      | 44.4%      |
| 7   | 346.5 | 2 | 213.0 | 3 | 118.5  | 8 | 118.5  | 8 | 65.8%      | 44.4%      |
| 6   | 346.5 | 2 | 213.0 | 3 | 118.5  | 8 | 118.5  | 8 | 65.8%      | 44.4%      |
| 5   | 346.5 | 2 | 213.0 | 3 | 118.5  | 8 | 118.5  | 8 | 65.8%      | 44.4%      |
| 4   | 346.5 | 2 | 213.0 | 3 | 118.5  | 8 | 118.5  | 8 | 65.8%      | 44.4%      |
| 3   | 346.5 | 2 | 213.0 | 3 | 118.5  | 6 | 118.5  | 6 | 65.8%      | 44.4%      |
| 2   | 346.5 | 2 | 213.0 | 3 | 119.0  | 5 | 118.5  | 5 | 65.8%      | 44.4%      |
| 1   | 346.5 | 2 | 241.5 | 2 | 141.5  | 6 | 141.5  | 6 | 59.2%      | 41.4%      |
| 0   | 454.0 | 2 | 322.0 | 3 | 221.0  | 8 | 221.0  | 8 | 51.3%      | 31.4%      |

**Tabela G.8:** Resultados de escalonamento do GAD PSG “pbsa-ipp95” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|-----|------|---|------|---|--------|---|--------|---|------------|------------|
| 10  | 23.0 | 1 | 23.0 | 1 | 16.0   | 4 | 16.0   | 4 | 30.4%      | 30.4%      |
| 9   | 23.0 | 1 | 23.0 | 1 | 16.0   | 4 | 16.0   | 4 | 30.4%      | 30.4%      |
| 8   | 23.0 | 1 | 23.0 | 1 | 16.0   | 4 | 16.0   | 4 | 30.4%      | 30.4%      |
| 7   | 23.0 | 1 | 23.0 | 1 | 16.0   | 4 | 16.0   | 4 | 30.4%      | 30.4%      |
| 6   | 23.0 | 1 | 23.0 | 1 | 16.0   | 4 | 16.0   | 4 | 30.4%      | 30.4%      |
| 5   | 23.0 | 1 | 23.0 | 1 | 16.0   | 4 | 16.0   | 4 | 30.4%      | 30.4%      |
| 4   | 23.0 | 1 | 23.0 | 1 | 16.0   | 4 | 16.0   | 4 | 30.4%      | 30.4%      |
| 3   | 23.0 | 1 | 23.0 | 1 | 16.0   | 4 | 16.0   | 4 | 30.4%      | 30.4%      |
| 2   | 23.0 | 1 | 23.0 | 1 | 16.0   | 4 | 16.0   | 4 | 30.4%      | 30.4%      |
| 1   | 23.0 | 1 | 23.0 | 1 | 17.5   | 2 | 17.5   | 2 | 23.9%      | 23.9%      |
| 0   | 40.0 | 2 | 42.0 | 2 | 32.0   | 4 | 32.0   | 4 | 20.0%      | 23.8%      |

Tabela G.9: Resultados de escalonamento do GAD PSG “ranka” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|-----|------|---|------|---|--------|---|--------|---|------------|------------|
| 10  | 11.0 | 3 | 11.0 | 4 | 11.0   | 3 | 11.0   | 3 | 0.0%       | 0.0%       |
| 9   | 11.0 | 3 | 11.0 | 4 | 11.0   | 3 | 11.0   | 3 | 0.0%       | 0.0%       |
| 8   | 11.0 | 3 | 11.0 | 4 | 11.0   | 3 | 11.0   | 3 | 0.0%       | 0.0%       |
| 7   | 11.0 | 3 | 11.0 | 4 | 11.0   | 3 | 11.0   | 3 | 0.0%       | 0.0%       |
| 6   | 11.0 | 3 | 11.0 | 4 | 11.0   | 3 | 11.0   | 3 | 0.0%       | 0.0%       |
| 5   | 11.0 | 3 | 11.0 | 4 | 11.0   | 3 | 11.0   | 3 | 0.0%       | 0.0%       |
| 4   | 11.0 | 3 | 11.0 | 4 | 11.0   | 3 | 11.0   | 3 | 0.0%       | 0.0%       |
| 3   | 11.0 | 3 | 11.0 | 4 | 11.0   | 3 | 11.0   | 3 | 0.0%       | 0.0%       |
| 2   | 11.0 | 3 | 11.0 | 4 | 11.0   | 3 | 11.0   | 3 | 0.0%       | 0.0%       |
| 1   | 12.0 | 3 | 11.0 | 2 | 11.0   | 2 | 11.0   | 2 | 8.3%       | 0.0%       |
| 0   | 16.0 | 3 | 16.0 | 4 | 16.0   | 3 | 16.0   | 3 | 0.0%       | 0.0%       |

Tabela G.10: Resultados de escalonamento do GAD PSG “tyang1” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|-----|------|---|------|---|--------|---|--------|---|------------|------------|
| 10  | 14.0 | 2 | 10.0 | 2 | 9.0    | 3 | 9.0    | 3 | 35.7%      | 10.0%      |
| 9   | 14.0 | 2 | 10.0 | 2 | 9.0    | 3 | 9.0    | 3 | 35.7%      | 10.0%      |
| 8   | 14.0 | 2 | 10.0 | 2 | 9.0    | 3 | 9.0    | 3 | 35.7%      | 10.0%      |
| 7   | 14.0 | 2 | 10.0 | 2 | 9.0    | 3 | 9.0    | 3 | 35.7%      | 10.0%      |
| 6   | 14.0 | 2 | 10.0 | 2 | 9.0    | 3 | 9.0    | 3 | 35.7%      | 10.0%      |
| 5   | 14.0 | 2 | 10.0 | 2 | 9.0    | 3 | 9.0    | 3 | 35.7%      | 10.0%      |
| 4   | 14.0 | 2 | 10.0 | 2 | 9.0    | 3 | 9.0    | 3 | 35.7%      | 10.0%      |
| 3   | 14.0 | 2 | 10.0 | 2 | 9.0    | 3 | 9.0    | 3 | 35.7%      | 10.0%      |
| 2   | 14.0 | 2 | 10.0 | 2 | 9.0    | 3 | 9.0    | 3 | 35.7%      | 10.0%      |
| 1   | 14.0 | 2 | 10.0 | 2 | 11.0   | 2 | 11.0   | 2 | 21.4%      | -10.0%     |
| 0   | 21.0 | 3 | 17.0 | 3 | 16.0   | 3 | 16.0   | 3 | 23.8%      | 5.9%       |

Tabela G.11: Resultados de escalonamento do GAD PSG “tyang4” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $\bar{\sigma}$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|----------------|------|---|------|---|--------|---|--------|---|------------|------------|
| 0.000          | 45.0 | 4 | 45.0 | 5 | 40.0   | 6 | 40.0   | 6 | 11.1%      | 11.1%      |
| 0.125          | 41.3 | 4 | 38.1 | 4 | 34.0   | 6 | 34.0   | 6 | 17.6%      | 10.6%      |
| 0.250          | 37.7 | 4 | 36.0 | 4 | 30.6   | 6 | 30.6   | 6 | 18.8%      | 15.0%      |
| 0.500          | 31.6 | 4 | 30.8 | 4 | 29.6   | 5 | 29.6   | 5 | 6.4%       | 4.1%       |

**Tabela G.12:** Resultados de escalonamento do GAD PSG “al-maasarani” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|----------------|------|---|------|---|--------|---|--------|---|------------|------------|
| 0.000          | 40.0 | 7 | 37.0 | 6 | 35.0   | 8 | 35.0   | 8 | 12.5%      | 5.4%       |
| 0.125          | 36.1 | 7 | 32.3 | 6 | 29.9   | 8 | 29.9   | 8 | 17.1%      | 7.4%       |
| 0.250          | 33.5 | 7 | 29.4 | 6 | 27.2   | 8 | 27.2   | 8 | 18.6%      | 7.5%       |
| 0.500          | 28.7 | 6 | 24.5 | 5 | 23.2   | 6 | 22.9   | 6 | 20.2%      | 6.4%       |

**Tabela G.13:** Resultados de escalonamento do GAD PSG “al-mouhamed” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT  |   | LDBS  |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|----------------|-------|---|-------|---|--------|---|--------|---|------------|------------|
| 0.000          | 390.0 | 5 | 360.0 | 5 | 320.0  | 9 | 320.0  | 9 | 17.9%      | 11.1%      |
| 0.125          | 346.7 | 5 | 306.2 | 5 | 278.2  | 9 | 278.2  | 9 | 19.8%      | 9.1%       |
| 0.250          | 320.4 | 5 | 288.4 | 5 | 254.4  | 8 | 254.4  | 8 | 20.6%      | 11.8%      |
| 0.500          | 290.4 | 4 | 243.6 | 5 | 222.3  | 8 | 222.3  | 8 | 23.4%      | 8.7%       |

**Tabela G.14:** Resultados de escalonamento do GAD PSG “g4” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|----------------|------|---|------|---|--------|---|--------|---|------------|------------|
| 0.000          | 11.0 | 1 | 12.0 | 3 | 11.0   | 1 | 10.0   | 1 | 9.1%       | 16.7%      |
| 0.125          | 9.4  | 1 | 11.1 | 3 | 9.4    | 1 | 9.0    | 1 | 4.3%       | 19.4%      |
| 0.250          | 8.3  | 1 | 10.6 | 3 | 8.3    | 1 | 8.3    | 1 | 0.2%       | 21.4%      |
| 0.500          | 6.5  | 1 | 9.7  | 3 | 6.5    | 1 | 6.5    | 1 | 0.0%       | 32.3%      |

**Tabela G.15:** Resultados de escalonamento do GAD PSG “kruatrachuefig8” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|----------------|------|---|------|---|--------|---|--------|---|------------|------------|
| 0.000          | 33.0 | 2 | 35.0 | 2 | 25.0   | 4 | 25.0   | 4 | 24.2%      | 28.6%      |
| 0.125          | 26.4 | 2 | 31.3 | 2 | 21.3   | 4 | 21.3   | 4 | 19.4%      | 32.0%      |
| 0.250          | 23.5 | 2 | 29.1 | 2 | 19.0   | 4 | 19.0   | 4 | 19.4%      | 34.8%      |
| 0.500          | 23.2 | 1 | 23.2 | 1 | 14.9   | 4 | 14.9   | 4 | 35.9%      | 35.9%      |

**Tabela G.16:** Resultados de escalonamento do GAD PSG “lctd-eg1” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|----------------|------|---|------|---|--------|---|--------|---|------------|------------|
| 0.000          | 14.0 | 3 | 12.0 | 3 | 12.0   | 4 | 12.0   | 4 | 14.3%      | 0.0%       |
| 0.125          | 12.2 | 3 | 10.7 | 3 | 10.7   | 3 | 10.7   | 3 | 12.7%      | -0.0%      |
| 0.250          | 11.1 | 3 | 9.9  | 3 | 9.9    | 3 | 9.9    | 3 | 10.8%      | -0.4%      |
| 0.500          | 9.2  | 3 | 8.4  | 3 | 8.5    | 3 | 8.5    | 3 | 7.5%       | -0.4%      |

**Tabela G.17:** Resultados de escalonamento do GAD PSG “lwb” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT  |   | LDBS  |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|----------------|-------|---|-------|---|--------|---|--------|---|------------|------------|
| 0.000          | 180.0 | 2 | 137.0 | 3 | 130.0  | 3 | 127.0  | 3 | 29.4%      | 7.3%       |
| 0.125          | 168.1 | 2 | 118.1 | 2 | 112.3  | 2 | 109.5  | 2 | 34.8%      | 7.2%       |
| 0.250          | 120.7 | 1 | 105.5 | 2 | 100.2  | 2 | 99.3   | 2 | 17.7%      | 5.8%       |
| 0.500          | 94.5  | 1 | 82.6  | 2 | 78.4   | 2 | 78.4   | 2 | 17.0%      | 5.0%       |

**Tabela G.18:** Resultados de escalonamento do GAD PSG “mccreary1” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT  |   | LDBS  |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|----------------|-------|---|-------|---|--------|---|--------|---|------------|------------|
| 0.000          | 454.0 | 2 | 322.0 | 3 | 221.0  | 8 | 221.0  | 8 | 51.3%      | 31.4%      |
| 0.125          | 417.4 | 2 | 316.5 | 3 | 188.0  | 8 | 188.0  | 8 | 55.0%      | 40.6%      |
| 0.250          | 376.6 | 2 | 254.2 | 3 | 167.8  | 8 | 167.8  | 8 | 55.5%      | 34.0%      |
| 0.500          | 357.8 | 2 | 239.3 | 3 | 131.3  | 8 | 131.3  | 8 | 63.3%      | 45.1%      |

**Tabela G.19:** Resultados de escalonamento do GAD PSG “pbsa-ipp95” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|----------------|------|---|------|---|--------|---|--------|---|------------|------------|
| 0.000          | 40.0 | 2 | 42.0 | 2 | 32.0   | 4 | 32.0   | 4 | 20.0%      | 23.8%      |
| 0.125          | 32.3 | 2 | 37.2 | 2 | 27.2   | 4 | 27.2   | 4 | 15.8%      | 26.9%      |
| 0.250          | 28.8 | 2 | 34.4 | 2 | 24.3   | 4 | 24.3   | 4 | 15.8%      | 29.4%      |
| 0.500          | 27.3 | 1 | 27.3 | 1 | 19.0   | 4 | 19.0   | 4 | 30.4%      | 30.4%      |

**Tabela G.20:** Resultados de escalonamento do GAD PSG “ranka” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|----------------|------|---|------|---|--------|---|--------|---|------------|------------|
| 0.000          | 16.0 | 3 | 16.0 | 4 | 16.0   | 3 | 16.0   | 3 | 0.0%       | 0.0%       |
| 0.125          | 13.8 | 3 | 13.8 | 3 | 13.8   | 3 | 13.8   | 3 | 0.0%       | 0.0%       |
| 0.250          | 13.1 | 3 | 13.1 | 3 | 13.1   | 3 | 13.1   | 3 | 0.0%       | 0.0%       |
| 0.500          | 11.9 | 3 | 11.6 | 4 | 11.9   | 3 | 11.9   | 3 | 0.4%       | -2.4%      |

**Tabela G.21:** Resultados de escalonamento do GAD PSG “tyang1” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |   | LDBS |   | CPH-o0 |   | CPH-o1 |   | CPH x HEFT | CPH x LDBS |
|----------------|------|---|------|---|--------|---|--------|---|------------|------------|
| 0.000          | 21.0 | 3 | 17.0 | 3 | 16.0   | 3 | 16.0   | 3 | 23.8%      | 5.9%       |
| 0.125          | 19.2 | 3 | 15.2 | 3 | 13.6   | 3 | 13.6   | 3 | 29.2%      | 10.5%      |
| 0.250          | 18.1 | 2 | 14.1 | 3 | 12.2   | 3 | 12.2   | 3 | 32.8%      | 13.8%      |
| 0.500          | 16.1 | 2 | 11.5 | 2 | 10.4   | 3 | 10.4   | 3 | 35.4%      | 9.8%       |

**Tabela G.22:** Resultados de escalonamento do GAD PSG “tyang4” produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

# Apêndice H

## Resultados para Ambientes Restritos

| $x$ | HEFT  |    | LDBS  |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|-------|----|-------|----|--------|----|--------|----|------------|------------|
| 10  | 54.0  | 20 | 54.0  | 20 | 56.0   | 20 | 56.0   | 20 | -3.7%      | -3.7%      |
| 9   | 56.0  | 20 | 56.0  | 20 | 57.0   | 20 | 57.0   | 20 | -1.8%      | -1.8%      |
| 8   | 58.0  | 20 | 58.0  | 20 | 58.0   | 20 | 58.0   | 20 | 0.0%       | 0.0%       |
| 7   | 58.0  | 20 | 58.0  | 20 | 62.0   | 20 | 62.0   | 20 | -6.9%      | -6.9%      |
| 6   | 62.0  | 20 | 62.0  | 20 | 67.0   | 20 | 67.0   | 20 | -8.1%      | -8.1%      |
| 5   | 66.0  | 20 | 66.0  | 20 | 68.0   | 20 | 68.0   | 20 | -3.0%      | -3.0%      |
| 4   | 67.0  | 20 | 67.0  | 20 | 77.0   | 20 | 77.0   | 20 | -14.9%     | -14.9%     |
| 3   | 72.0  | 20 | 72.0  | 20 | 77.0   | 20 | 77.0   | 20 | -6.9%      | -6.9%      |
| 2   | 77.0  | 20 | 77.0  | 20 | 82.0   | 20 | 82.0   | 20 | -6.5%      | -6.5%      |
| 1   | 85.0  | 20 | 85.0  | 20 | 92.0   | 20 | 92.0   | 20 | -8.2%      | -8.2%      |
| 0   | 104.0 | 20 | 104.0 | 20 | 111.0  | 20 | 111.0  | 20 | -6.7%      | -6.7%      |

Tabela H.1: Resultados de escalonamento do GAD *árvore binária invertida* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 29.0 | 20 | 29.0 | 20 | 30.0   | 20 | 30.0   | 20 | -3.4%      | -3.4%      |
| 9   | 29.0 | 20 | 29.0 | 20 | 30.0   | 20 | 30.0   | 20 | -3.4%      | -3.4%      |
| 8   | 30.5 | 20 | 30.5 | 20 | 30.5   | 20 | 30.5   | 20 | 0.0%       | 0.0%       |
| 7   | 30.5 | 20 | 30.5 | 20 | 33.0   | 20 | 33.0   | 20 | -8.2%      | -8.2%      |
| 6   | 32.0 | 20 | 32.0 | 20 | 34.5   | 20 | 34.5   | 20 | -7.8%      | -7.8%      |
| 5   | 33.5 | 20 | 33.5 | 20 | 35.5   | 20 | 35.5   | 20 | -6.0%      | -6.0%      |
| 4   | 35.0 | 20 | 35.0 | 20 | 39.5   | 20 | 39.5   | 20 | -12.9%     | -12.9%     |
| 3   | 37.0 | 20 | 37.0 | 20 | 39.5   | 20 | 39.5   | 20 | -6.8%      | -6.8%      |
| 2   | 39.5 | 20 | 39.5 | 20 | 41.0   | 20 | 41.0   | 20 | -3.8%      | -3.8%      |
| 1   | 44.0 | 20 | 44.0 | 20 | 47.0   | 20 | 47.0   | 20 | -6.8%      | -6.8%      |
| 0   | 54.0 | 20 | 54.0 | 20 | 56.0   | 20 | 56.0   | 20 | -3.7%      | -3.7%      |

Tabela H.2: Resultados de escalonamento do GAD *árvore binária invertida* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes duais.



| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 9.5  | 20 | 8.0  | 20 | 8.0    | 20 | 8.0    | 20 | 15.8%      | 0.0%       |
| 9   | 9.5  | 20 | 8.0  | 20 | 8.0    | 20 | 8.0    | 20 | 15.8%      | 0.0%       |
| 8   | 9.5  | 20 | 8.5  | 20 | 8.0    | 20 | 8.0    | 20 | 15.8%      | 5.9%       |
| 7   | 9.5  | 20 | 8.5  | 20 | 8.0    | 20 | 8.0    | 20 | 15.8%      | 5.9%       |
| 6   | 10.0 | 20 | 8.5  | 20 | 8.0    | 20 | 8.0    | 20 | 20.0%      | 5.9%       |
| 5   | 10.0 | 20 | 9.0  | 20 | 9.0    | 20 | 9.0    | 20 | 10.0%      | 0.0%       |
| 4   | 10.5 | 20 | 9.5  | 20 | 9.0    | 20 | 9.0    | 20 | 14.3%      | 5.3%       |
| 3   | 10.5 | 20 | 10.0 | 20 | 9.0    | 20 | 9.0    | 20 | 14.3%      | 10.0%      |
| 2   | 11.0 | 20 | 10.5 | 20 | 10.0   | 20 | 10.0   | 20 | 9.1%       | 4.8%       |
| 1   | 12.0 | 20 | 11.5 | 20 | 11.0   | 20 | 11.0   | 20 | 8.3%       | 4.3%       |
| 0   | 14.0 | 20 | 14.0 | 20 | 14.0   | 20 | 14.0   | 20 | 0.0%       | 0.0%       |

**Tabela H.3:** Resultados de escalonamento do GAD *árvore binária invertida* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 25.5 | 20 | 18.5 | 20 | 17.0   | 20 | 15.5   | 20 | 39.2%      | 16.2%      |
| 9   | 25.5 | 20 | 19.0 | 20 | 17.0   | 20 | 15.5   | 20 | 39.2%      | 18.4%      |
| 8   | 25.5 | 20 | 16.5 | 20 | 17.0   | 20 | 15.5   | 20 | 39.2%      | 6.1%       |
| 7   | 25.5 | 20 | 16.5 | 20 | 17.0   | 20 | 15.5   | 20 | 39.2%      | 6.1%       |
| 6   | 25.0 | 20 | 19.0 | 20 | 17.0   | 20 | 15.5   | 20 | 38.0%      | 18.4%      |
| 5   | 25.5 | 20 | 17.0 | 20 | 17.0   | 20 | 15.5   | 20 | 39.2%      | 8.8%       |
| 4   | 25.5 | 20 | 19.5 | 20 | 17.0   | 20 | 15.5   | 20 | 39.2%      | 20.5%      |
| 3   | 26.0 | 20 | 19.5 | 20 | 17.5   | 20 | 16.0   | 20 | 38.5%      | 17.9%      |
| 2   | 25.5 | 20 | 18.5 | 20 | 17.5   | 20 | 16.5   | 20 | 35.3%      | 10.8%      |
| 1   | 27.0 | 20 | 21.5 | 20 | 18.5   | 20 | 17.0   | 20 | 37.0%      | 20.9%      |
| 0   | 31.0 | 20 | 24.0 | 20 | 21.0   | 20 | 21.0   | 20 | 32.3%      | 12.5%      |

**Tabela H.4:** Resultados de escalonamento do GAD *árvore binária invertida* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 45.5 | 20 | 33.5 | 20 | 22.0   | 14 | 22.0   | 14 | 51.6%      | 34.3%      |
| 9   | 45.5 | 20 | 34.0 | 20 | 22.0   | 14 | 22.0   | 14 | 51.6%      | 35.3%      |
| 8   | 45.5 | 20 | 26.5 | 20 | 22.0   | 14 | 22.0   | 14 | 51.6%      | 17.0%      |
| 7   | 45.5 | 20 | 26.5 | 20 | 22.0   | 14 | 22.0   | 14 | 51.6%      | 17.0%      |
| 6   | 45.0 | 20 | 34.0 | 20 | 22.0   | 14 | 22.0   | 14 | 51.1%      | 35.3%      |
| 5   | 45.5 | 20 | 27.0 | 20 | 22.0   | 14 | 22.0   | 14 | 51.6%      | 18.5%      |
| 4   | 45.5 | 20 | 34.5 | 20 | 22.0   | 14 | 22.0   | 14 | 51.6%      | 36.2%      |
| 3   | 46.0 | 20 | 27.5 | 20 | 22.0   | 14 | 22.0   | 14 | 52.2%      | 20.0%      |
| 2   | 45.5 | 20 | 28.5 | 20 | 22.0   | 14 | 22.0   | 14 | 51.6%      | 22.8%      |
| 1   | 47.0 | 20 | 29.5 | 20 | 22.0   | 14 | 22.0   | 14 | 53.2%      | 25.4%      |
| 0   | 51.0 | 20 | 37.0 | 20 | 29.0   | 15 | 28.0   | 15 | 45.1%      | 24.3%      |

**Tabela H.5:** Resultados de escalonamento do GAD *árvore binária invertida* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes duais.

| $\bar{\sigma}$ | HEFT  |    | LDBS  |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|-------|----|-------|----|--------|----|--------|----|------------|------------|
| 0.000          | 104.0 | 20 | 104.0 | 20 | 111.0  | 20 | 111.0  | 20 | -6.7%      | -6.7%      |
| 0.125          | 98.8  | 20 | 98.8  | 20 | 99.9   | 20 | 99.9   | 20 | -1.1%      | -1.1%      |
| 0.250          | 95.0  | 20 | 95.0  | 20 | 98.7   | 20 | 98.7   | 20 | -3.8%      | -3.8%      |
| 0.500          | 89.6  | 20 | 89.6  | 20 | 94.5   | 20 | 94.5   | 20 | -5.6%      | -5.6%      |

Tabela H.6: Resultados de escalonamento do GAD *árvore binária invertida* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 54.0 | 20 | 54.0 | 20 | 56.0   | 20 | 56.0   | 20 | -3.7%      | -3.7%      |
| 0.125          | 50.0 | 20 | 50.0 | 20 | 51.6   | 20 | 51.6   | 20 | -3.2%      | -3.2%      |
| 0.250          | 48.9 | 20 | 48.9 | 20 | 51.4   | 20 | 51.4   | 20 | -5.2%      | -5.2%      |
| 0.500          | 45.7 | 20 | 45.7 | 20 | 47.8   | 20 | 47.8   | 20 | -4.5%      | -4.7%      |

Tabela H.7: Resultados de escalonamento do GAD *árvore binária invertida* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 14.0 | 20 | 14.0 | 20 | 14.0   | 20 | 14.0   | 20 | 0.0%       | 0.0%       |
| 0.125          | 13.5 | 20 | 12.9 | 20 | 14.3   | 20 | 14.3   | 20 | -6.1%      | -10.3%     |
| 0.250          | 13.1 | 20 | 12.4 | 20 | 12.4   | 20 | 12.4   | 20 | 5.2%       | 0.1%       |
| 0.500          | 12.2 | 20 | 11.5 | 20 | 11.3   | 20 | 11.3   | 20 | 7.5%       | 1.8%       |

Tabela H.8: Resultados de escalonamento do GAD *árvore binária invertida* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 31.0 | 20 | 24.0 | 20 | 21.0   | 20 | 21.0   | 20 | 32.3%      | 12.5%      |
| 0.125          | 36.1 | 20 | 22.4 | 20 | 21.0   | 20 | 20.0   | 20 | 44.7%      | 10.7%      |
| 0.250          | 32.0 | 20 | 22.2 | 20 | 19.7   | 20 | 18.8   | 20 | 41.3%      | 15.3%      |
| 0.500          | 31.0 | 20 | 19.9 | 20 | 19.8   | 19 | 17.3   | 19 | 44.2%      | 13.3%      |

Tabela H.9: Resultados de escalonamento do GAD *árvore binária invertida* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 51.0 | 20 | 37.0 | 20 | 29.0   | 15 | 28.0   | 15 | 45.1%      | 24.3%      |
| 0.125          | 66.1 | 20 | 37.1 | 20 | 26.2   | 15 | 26.0   | 15 | 60.6%      | 29.8%      |
| 0.250          | 57.0 | 20 | 36.5 | 20 | 25.0   | 14 | 25.0   | 14 | 56.2%      | 31.5%      |
| 0.500          | 56.1 | 20 | 29.9 | 20 | 26.1   | 12 | 24.4   | 12 | 56.5%      | 18.3%      |

Tabela H.10: Resultados de escalonamento do GAD *árvore binária invertida* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes aleatórios.

| $x$ | HEFT  |    | LDBS  |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|-------|----|-------|----|--------|----|--------|----|------------|------------|
| 10  | 54.0  | 20 | 51.0  | 20 | 55.0   | 20 | 55.0   | 20 | -1.9%      | -7.8%      |
| 9   | 56.0  | 20 | 55.0  | 20 | 55.0   | 18 | 55.0   | 18 | 1.8%       | 0.0%       |
| 8   | 58.0  | 20 | 56.0  | 20 | 55.0   | 16 | 55.0   | 16 | 5.2%       | 1.8%       |
| 7   | 60.0  | 20 | 57.0  | 20 | 75.0   | 20 | 75.0   | 20 | -25.0%     | -31.6%     |
| 6   | 63.0  | 20 | 61.0  | 20 | 90.0   | 20 | 90.0   | 20 | -42.9%     | -47.5%     |
| 5   | 66.0  | 20 | 65.0  | 20 | 90.0   | 20 | 90.0   | 20 | -36.4%     | -38.5%     |
| 4   | 69.0  | 20 | 67.0  | 20 | 87.0   | 20 | 87.0   | 20 | -26.1%     | -29.9%     |
| 3   | 74.0  | 20 | 71.0  | 20 | 95.0   | 20 | 95.0   | 20 | -28.4%     | -33.8%     |
| 2   | 78.0  | 20 | 77.0  | 20 | 95.0   | 20 | 95.0   | 20 | -21.8%     | -23.4%     |
| 1   | 88.0  | 20 | 86.0  | 20 | 117.0  | 20 | 117.0  | 20 | -33.0%     | -36.0%     |
| 0   | 104.0 | 20 | 101.0 | 20 | 110.0  | 20 | 110.0  | 20 | -5.8%      | -8.9%      |

Tabela H.11: Resultados de escalonamento do GAD *árvore binária* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 29.0 | 20 | 26.0 | 20 | 27.5   | 20 | 27.5   | 20 | 5.2%       | -5.8%      |
| 9   | 30.0 | 20 | 27.5 | 20 | 27.5   | 18 | 27.5   | 18 | 8.3%       | 0.0%       |
| 8   | 30.5 | 20 | 28.5 | 20 | 27.5   | 16 | 27.5   | 16 | 9.8%       | 3.5%       |
| 7   | 31.5 | 20 | 29.5 | 20 | 37.5   | 20 | 37.5   | 20 | -19.0%     | -27.1%     |
| 6   | 33.0 | 20 | 31.0 | 20 | 45.0   | 20 | 45.0   | 20 | -36.4%     | -45.2%     |
| 5   | 34.0 | 20 | 32.5 | 20 | 45.0   | 20 | 45.0   | 20 | -32.4%     | -38.5%     |
| 4   | 36.5 | 20 | 34.5 | 20 | 43.5   | 20 | 43.5   | 20 | -19.2%     | -26.1%     |
| 3   | 39.0 | 20 | 36.0 | 20 | 48.5   | 20 | 48.5   | 20 | -24.4%     | -34.7%     |
| 2   | 40.5 | 20 | 39.5 | 20 | 48.0   | 20 | 48.0   | 20 | -18.5%     | -21.5%     |
| 1   | 45.5 | 20 | 43.5 | 20 | 59.5   | 20 | 59.5   | 20 | -30.8%     | -36.8%     |
| 0   | 54.0 | 20 | 51.0 | 20 | 55.0   | 20 | 55.0   | 20 | -1.9%      | -7.8%      |

Tabela H.12: Resultados de escalonamento do GAD *árvore binária* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 9.5  | 20 | 6.5  | 20 | 5.5    | 20 | 5.5    | 20 | 42.1%      | 15.4%      |
| 9   | 9.5  | 18 | 7.0  | 20 | 5.5    | 18 | 5.5    | 18 | 42.1%      | 21.4%      |
| 8   | 9.5  | 16 | 7.0  | 20 | 5.5    | 16 | 5.5    | 16 | 42.1%      | 21.4%      |
| 7   | 10.0 | 20 | 7.0  | 20 | 7.5    | 20 | 7.5    | 20 | 25.0%      | -7.1%      |
| 6   | 10.0 | 20 | 7.5  | 20 | 9.0    | 20 | 9.0    | 20 | 10.0%      | -20.0%     |
| 5   | 10.5 | 20 | 8.0  | 20 | 9.0    | 20 | 9.0    | 20 | 14.3%      | -12.5%     |
| 4   | 10.5 | 20 | 8.5  | 20 | 10.0   | 20 | 10.0   | 20 | 4.8%       | -17.6%     |
| 3   | 11.0 | 20 | 8.5  | 20 | 10.0   | 20 | 10.0   | 20 | 9.1%       | -17.6%     |
| 2   | 11.0 | 20 | 9.0  | 20 | 11.0   | 20 | 11.0   | 20 | 0.0%       | -22.2%     |
| 1   | 12.0 | 20 | 10.0 | 20 | 13.0   | 20 | 13.0   | 20 | -8.3%      | -30.0%     |
| 0   | 15.0 | 20 | 12.0 | 20 | 11.0   | 20 | 11.0   | 20 | 26.7%      | 8.3%       |

Tabela H.13: Resultados de escalonamento do GAD *árvore binária* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 24.0 | 8  | 13.0 | 16 | 5.5    | 20 | 5.5    | 20 | 77.1%      | 57.7%      |
| 9   | 24.0 | 8  | 13.0 | 16 | 5.5    | 18 | 5.5    | 18 | 77.1%      | 57.7%      |
| 8   | 24.0 | 8  | 13.0 | 16 | 5.5    | 16 | 5.5    | 16 | 77.1%      | 57.7%      |
| 7   | 24.0 | 8  | 13.0 | 14 | 7.5    | 20 | 7.5    | 20 | 68.8%      | 42.3%      |
| 6   | 24.0 | 8  | 13.5 | 12 | 9.0    | 20 | 9.0    | 20 | 62.5%      | 33.3%      |
| 5   | 24.0 | 8  | 13.5 | 14 | 9.5    | 20 | 9.5    | 20 | 60.4%      | 29.6%      |
| 4   | 24.0 | 8  | 14.0 | 20 | 10.0   | 20 | 10.0   | 20 | 58.3%      | 28.6%      |
| 3   | 24.5 | 10 | 14.5 | 20 | 10.5   | 20 | 10.5   | 20 | 57.1%      | 27.6%      |
| 2   | 24.5 | 12 | 14.5 | 20 | 11.5   | 20 | 11.5   | 20 | 53.1%      | 20.7%      |
| 1   | 25.0 | 14 | 15.0 | 20 | 14.0   | 20 | 14.0   | 20 | 44.0%      | 6.7%       |
| 0   | 31.0 | 16 | 18.0 | 18 | 11.0   | 20 | 11.0   | 20 | 64.5%      | 38.9%      |

Tabela H.14: Resultados de escalonamento do GAD *árvore binária* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes duais.

| $x$ | HEFT |   | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|---|------|----|--------|----|--------|----|------------|------------|
| 10  | 36.5 | 4 | 18.5 | 18 | 5.5    | 20 | 5.5    | 20 | 84.9%      | 70.3%      |
| 9   | 36.5 | 4 | 18.5 | 18 | 5.5    | 18 | 5.5    | 18 | 84.9%      | 70.3%      |
| 8   | 36.5 | 4 | 18.5 | 16 | 5.5    | 16 | 5.5    | 16 | 84.9%      | 70.3%      |
| 7   | 36.5 | 4 | 19.0 | 14 | 7.5    | 20 | 7.5    | 20 | 79.5%      | 60.5%      |
| 6   | 36.5 | 4 | 19.5 | 14 | 9.0    | 20 | 9.0    | 20 | 75.3%      | 53.8%      |
| 5   | 36.5 | 4 | 19.5 | 20 | 9.5    | 20 | 9.5    | 20 | 74.0%      | 51.3%      |
| 4   | 36.5 | 4 | 19.5 | 20 | 10.0   | 20 | 10.0   | 20 | 72.6%      | 48.7%      |
| 3   | 36.5 | 4 | 20.0 | 20 | 10.5   | 20 | 10.5   | 20 | 71.2%      | 47.5%      |
| 2   | 36.5 | 4 | 20.5 | 20 | 11.5   | 20 | 11.5   | 20 | 68.5%      | 43.9%      |
| 1   | 37.0 | 6 | 21.0 | 20 | 14.0   | 20 | 14.0   | 20 | 62.2%      | 33.3%      |
| 0   | 48.0 | 8 | 26.0 | 16 | 11.0   | 20 | 11.0   | 20 | 77.1%      | 57.7%      |

Tabela H.15: Resultados de escalonamento do GAD *árvore binária* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes duais.

| $\bar{\sigma}$ | HEFT  |    | LDBS  |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|-------|----|-------|----|--------|----|--------|----|------------|------------|
| 0.000          | 104.0 | 20 | 101.0 | 20 | 110.0  | 20 | 110.0  | 20 | -5.8%      | -8.9%      |
| 0.125          | 98.8  | 20 | 97.9  | 20 | 121.1  | 20 | 121.1  | 20 | -22.7%     | -23.7%     |
| 0.250          | 97.0  | 20 | 95.1  | 20 | 124.9  | 20 | 124.9  | 20 | -28.8%     | -31.3%     |
| 0.500          | 92.2  | 19 | 89.8  | 19 | 124.1  | 16 | 124.1  | 16 | -34.6%     | -38.1%     |

Tabela H.16: Resultados de escalonamento do GAD *árvore binária* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 54.0 | 20 | 51.0 | 20 | 55.0   | 20 | 55.0   | 20 | -1.9%      | -7.8%      |
| 0.125          | 50.9 | 20 | 49.5 | 20 | 60.6   | 20 | 60.6   | 20 | -19.0%     | -22.3%     |
| 0.250          | 49.7 | 20 | 48.1 | 20 | 62.4   | 20 | 62.4   | 20 | -25.7%     | -29.8%     |
| 0.500          | 47.6 | 19 | 45.5 | 19 | 61.1   | 16 | 61.1   | 16 | -28.5%     | -34.4%     |

Tabela H.17: Resultados de escalonamento do GAD *árvore binária* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 15.0 | 20 | 12.0 | 20 | 11.0   | 20 | 11.0   | 20 | 26.7%      | 8.3%       |
| 0.125          | 13.8 | 20 | 11.4 | 20 | 12.6   | 20 | 12.6   | 20 | 8.5%       | -10.4%     |
| 0.250          | 13.3 | 20 | 10.9 | 20 | 11.8   | 20 | 11.8   | 20 | 10.9%      | -8.7%      |
| 0.500          | 12.5 | 18 | 10.2 | 19 | 13.7   | 16 | 13.7   | 16 | -9.3%      | -33.8%     |

**Tabela H.18:** Resultados de escalonamento do GAD *árvore binária* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 31.0 | 16 | 18.0 | 18 | 11.0   | 20 | 11.0   | 20 | 64.5%      | 38.9%      |
| 0.125          | 29.7 | 14 | 17.6 | 17 | 12.6   | 20 | 12.6   | 20 | 57.5%      | 28.6%      |
| 0.250          | 28.8 | 10 | 16.7 | 17 | 12.9   | 20 | 12.9   | 20 | 55.1%      | 22.8%      |
| 0.500          | 25.9 | 11 | 16.0 | 15 | 14.9   | 16 | 14.9   | 16 | 42.5%      | 6.9%       |

**Tabela H.19:** Resultados de escalonamento do GAD *árvore binária* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |   | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|---|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 48.0 | 8 | 26.0 | 16 | 11.0   | 20 | 11.0   | 20 | 77.1%      | 57.7%      |
| 0.125          | 45.4 | 9 | 24.5 | 20 | 12.6   | 20 | 12.6   | 20 | 72.3%      | 48.6%      |
| 0.250          | 43.8 | 9 | 23.5 | 20 | 12.9   | 20 | 12.9   | 20 | 70.5%      | 45.2%      |
| 0.500          | 39.6 | 6 | 21.9 | 16 | 15.4   | 16 | 15.4   | 16 | 61.0%      | 29.3%      |

**Tabela H.20:** Resultados de escalonamento do GAD *árvore binária* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes aleatórios.

| $x$ | HEFT  |    | LDBS  |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|-------|----|-------|----|--------|----|--------|----|------------|------------|
| 10  | 228.0 | 19 | 217.0 | 20 | 214.0  | 20 | 214.0  | 20 | 6.1%       | 1.4%       |
| 9   | 229.0 | 18 | 218.0 | 18 | 232.0  | 18 | 232.0  | 18 | -1.3%      | -6.4%      |
| 8   | 232.0 | 16 | 218.0 | 16 | 247.0  | 16 | 247.0  | 16 | -6.5%      | -13.3%     |
| 7   | 235.0 | 14 | 223.0 | 16 | 239.0  | 14 | 239.0  | 14 | -1.7%      | -7.2%      |
| 6   | 244.0 | 12 | 235.0 | 13 | 259.0  | 12 | 259.0  | 12 | -6.1%      | -10.2%     |
| 5   | 263.0 | 10 | 255.0 | 10 | 281.0  | 10 | 281.0  | 10 | -6.8%      | -10.2%     |
| 4   | 294.0 | 10 | 285.0 | 12 | 342.0  | 12 | 342.0  | 12 | -16.3%     | -20.0%     |
| 3   | 327.0 | 12 | 316.0 | 14 | 376.0  | 12 | 376.0  | 12 | -15.0%     | -19.0%     |
| 2   | 360.0 | 15 | 346.0 | 15 | 403.0  | 16 | 403.0  | 16 | -11.9%     | -16.5%     |
| 1   | 391.0 | 17 | 381.0 | 17 | 390.0  | 20 | 390.0  | 20 | 0.3%       | -2.4%      |
| 0   | 418.0 | 19 | 409.0 | 20 | 410.0  | 20 | 410.0  | 20 | 1.9%       | -0.2%      |

Tabela H.21: Resultados de escalonamento do GAD *diamante* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes duais.

| $x$ | HEFT  |    | LDBS  |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|-------|----|-------|----|--------|----|--------|----|------------|------------|
| 10  | 132.5 | 19 | 122.5 | 20 | 116.5  | 20 | 116.5  | 20 | 12.1%      | 4.9%       |
| 9   | 133.5 | 18 | 122.5 | 18 | 116.5  | 18 | 116.5  | 18 | 12.7%      | 4.9%       |
| 8   | 135.5 | 16 | 123.5 | 16 | 120.5  | 16 | 120.5  | 16 | 11.1%      | 2.4%       |
| 7   | 135.5 | 14 | 125.0 | 15 | 126.5  | 14 | 126.5  | 14 | 6.6%       | -1.2%      |
| 6   | 137.5 | 12 | 128.5 | 13 | 134.5  | 12 | 134.5  | 12 | 2.2%       | -4.7%      |
| 5   | 144.5 | 10 | 137.5 | 11 | 148.5  | 10 | 148.5  | 10 | -2.8%      | -8.0%      |
| 4   | 158.5 | 9  | 150.5 | 10 | 180.0  | 10 | 180.0  | 10 | -13.6%     | -19.6%     |
| 3   | 176.0 | 11 | 167.0 | 14 | 189.0  | 12 | 189.0  | 12 | -7.4%      | -13.2%     |
| 2   | 195.0 | 13 | 181.0 | 16 | 202.5  | 16 | 202.5  | 16 | -3.8%      | -11.9%     |
| 1   | 211.0 | 16 | 200.0 | 18 | 215.0  | 20 | 215.0  | 20 | -1.9%      | -7.5%      |
| 0   | 228.0 | 19 | 217.0 | 20 | 214.0  | 20 | 214.0  | 20 | 6.1%       | 1.4%       |

Tabela H.22: Resultados de escalonamento do GAD *diamante* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 55.5 | 7  | 40.0 | 14 | 53.0   | 20 | 53.0   | 20 | 4.5%       | -32.5%     |
| 9   | 55.5 | 7  | 40.0 | 14 | 55.5   | 18 | 55.5   | 18 | 0.0%       | -38.8%     |
| 8   | 55.5 | 7  | 40.0 | 14 | 56.0   | 16 | 56.0   | 16 | -0.9%      | -40.0%     |
| 7   | 55.5 | 7  | 40.0 | 14 | 55.0   | 14 | 55.0   | 14 | 0.9%       | -37.5%     |
| 6   | 55.5 | 7  | 40.5 | 12 | 56.0   | 14 | 56.0   | 14 | -0.9%      | -38.3%     |
| 5   | 55.5 | 7  | 42.0 | 11 | 55.5   | 15 | 55.5   | 15 | 0.0%       | -32.1%     |
| 4   | 55.5 | 7  | 44.0 | 10 | 55.5   | 15 | 55.5   | 15 | 0.0%       | -26.1%     |
| 3   | 55.5 | 8  | 47.0 | 9  | 57.0   | 20 | 55.5   | 20 | 0.0%       | -18.1%     |
| 2   | 56.5 | 8  | 51.5 | 10 | 58.5   | 20 | 58.5   | 20 | -3.5%      | -13.6%     |
| 1   | 67.0 | 9  | 59.5 | 12 | 66.0   | 20 | 66.0   | 20 | 1.5%       | -10.9%     |
| 0   | 76.0 | 10 | 65.0 | 15 | 81.0   | 20 | 81.0   | 20 | -6.6%      | -24.6%     |

Tabela H.23: Resultados de escalonamento do GAD *diamante* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT  |   | LDBS  |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|-------|---|-------|---|--------|----|--------|----|------------|------------|
| 10  | 159.5 | 2 | 94.5  | 4 | 91.0   | 20 | 91.0   | 20 | 42.9%      | 3.7%       |
| 9   | 159.5 | 2 | 94.5  | 4 | 96.5   | 20 | 96.5   | 20 | 39.5%      | -2.1%      |
| 8   | 159.5 | 2 | 94.5  | 4 | 95.0   | 18 | 95.0   | 18 | 40.4%      | -0.5%      |
| 7   | 159.5 | 2 | 94.5  | 4 | 95.5   | 16 | 95.5   | 16 | 40.1%      | -1.1%      |
| 6   | 159.5 | 2 | 94.5  | 4 | 96.5   | 18 | 96.5   | 18 | 39.5%      | -2.1%      |
| 5   | 159.5 | 2 | 94.5  | 4 | 96.0   | 17 | 96.0   | 17 | 39.8%      | -1.6%      |
| 4   | 159.5 | 2 | 94.5  | 4 | 96.5   | 18 | 96.5   | 18 | 39.5%      | -2.1%      |
| 3   | 159.5 | 2 | 94.5  | 4 | 97.5   | 20 | 97.5   | 20 | 38.9%      | -3.2%      |
| 2   | 159.5 | 2 | 94.5  | 4 | 99.5   | 20 | 99.5   | 20 | 37.6%      | -5.3%      |
| 1   | 159.5 | 2 | 94.5  | 5 | 100.0  | 20 | 100.0  | 20 | 37.3%      | -5.8%      |
| 0   | 204.0 | 4 | 116.0 | 7 | 132.0  | 20 | 132.0  | 20 | 35.3%      | -13.8%     |

**Tabela H.24:** Resultados de escalonamento do GAD *diamante* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes duais.

| $x$ | HEFT  |   | LDBS  |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|-------|---|-------|---|--------|----|--------|----|------------|------------|
| 10  | 200.0 | 1 | 152.5 | 2 | 115.5  | 20 | 115.5  | 20 | 42.2%      | 24.3%      |
| 9   | 200.0 | 1 | 152.5 | 2 | 117.0  | 20 | 117.0  | 20 | 41.5%      | 23.3%      |
| 8   | 200.0 | 1 | 152.5 | 2 | 117.5  | 18 | 117.5  | 18 | 41.2%      | 23.0%      |
| 7   | 200.0 | 1 | 152.5 | 2 | 116.5  | 19 | 116.5  | 19 | 41.8%      | 23.6%      |
| 6   | 200.0 | 1 | 152.5 | 2 | 116.0  | 17 | 116.0  | 17 | 42.0%      | 23.9%      |
| 5   | 200.0 | 1 | 152.5 | 2 | 119.5  | 20 | 119.5  | 20 | 40.2%      | 21.6%      |
| 4   | 200.0 | 1 | 152.5 | 2 | 121.5  | 20 | 121.5  | 20 | 39.2%      | 20.3%      |
| 3   | 200.0 | 1 | 152.5 | 2 | 122.0  | 20 | 122.0  | 20 | 39.0%      | 20.0%      |
| 2   | 200.0 | 1 | 152.5 | 2 | 126.0  | 20 | 126.0  | 20 | 37.0%      | 17.4%      |
| 1   | 200.0 | 1 | 152.5 | 2 | 138.5  | 20 | 138.5  | 20 | 30.8%      | 9.2%       |
| 0   | 319.0 | 2 | 189.0 | 4 | 160.0  | 20 | 160.0  | 20 | 49.8%      | 15.3%      |

**Tabela H.25:** Resultados de escalonamento do GAD *diamante* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes duais.

| $\bar{\sigma}$ | HEFT  |    | LDBS  |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|-------|----|-------|----|--------|----|--------|----|------------|------------|
| 0.000          | 418.0 | 19 | 409.0 | 20 | 410.0  | 20 | 410.0  | 20 | 1.9%       | -0.2%      |
| 0.125          | 397.1 | 17 | 390.9 | 17 | 424.3  | 16 | 424.3  | 16 | -6.8%      | -8.5%      |
| 0.250          | 387.5 | 14 | 380.5 | 14 | 448.7  | 12 | 448.7  | 12 | -15.8%     | -17.9%     |
| 0.500          | 339.3 | 11 | 336.2 | 11 | 406.9  | 10 | 406.9  | 10 | -19.9%     | -21.0%     |

**Tabela H.26:** Resultados de escalonamento do GAD *diamante* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT  |    | LDBS  |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|-------|----|-------|----|--------|----|--------|----|------------|------------|
| 0.000          | 228.0 | 19 | 217.0 | 20 | 214.0  | 20 | 214.0  | 20 | 6.1%       | 1.4%       |
| 0.125          | 214.7 | 17 | 207.1 | 17 | 233.9  | 14 | 233.9  | 14 | -8.9%      | -12.9%     |
| 0.250          | 205.8 | 13 | 197.2 | 15 | 239.0  | 11 | 239.0  | 11 | -16.1%     | -21.2%     |
| 0.500          | 180.0 | 11 | 173.6 | 12 | 207.7  | 10 | 207.7  | 10 | -15.4%     | -19.6%     |

**Tabela H.27:** Resultados de escalonamento do GAD *diamante* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 76.0 | 10 | 65.0 | 15 | 81.0   | 20 | 81.0   | 20 | -6.6%      | -24.6%     |
| 0.125          | 70.9 | 10 | 61.1 | 17 | 83.1   | 20 | 82.6   | 20 | -16.5%     | -35.1%     |
| 0.250          | 67.2 | 10 | 58.1 | 15 | 92.0   | 17 | 92.0   | 17 | -36.8%     | -58.4%     |
| 0.500          | 59.8 | 10 | 52.3 | 11 | 78.4   | 11 | 78.4   | 11 | -31.1%     | -50.0%     |

**Tabela H.28:** Resultados de escalonamento do GAD *diamante* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT  |   | LDBS  |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|-------|---|-------|---|--------|----|--------|----|------------|------------|
| 0.000          | 204.0 | 4 | 116.0 | 7 | 132.0  | 20 | 132.0  | 20 | 35.3%      | -13.8%     |
| 0.125          | 152.0 | 5 | 114.2 | 9 | 126.6  | 20 | 126.6  | 20 | 16.7%      | -10.9%     |
| 0.250          | 148.7 | 6 | 107.1 | 7 | 135.0  | 12 | 135.0  | 12 | 9.2%       | -26.1%     |
| 0.500          | 142.4 | 7 | 96.0  | 6 | 116.9  | 10 | 116.9  | 10 | 17.9%      | -21.8%     |

**Tabela H.29:** Resultados de escalonamento do GAD *diamante* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT  |   | LDBS  |   | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|-------|---|-------|---|--------|----|--------|----|------------|------------|
| 0.000          | 319.0 | 2 | 189.0 | 4 | 160.0  | 20 | 160.0  | 20 | 49.8%      | 15.3%      |
| 0.125          | 247.2 | 5 | 160.6 | 5 | 158.0  | 20 | 158.0  | 20 | 36.1%      | 1.6%       |
| 0.250          | 254.1 | 3 | 150.7 | 6 | 148.2  | 16 | 148.2  | 16 | 41.7%      | 1.7%       |
| 0.500          | 216.9 | 2 | 143.8 | 3 | 135.4  | 12 | 135.4  | 12 | 37.6%      | 5.9%       |

**Tabela H.30:** Resultados de escalonamento do GAD *diamante* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes aleatórios.



| $x$ | HEFT  |    | LDBS  |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|-------|----|-------|----|--------|----|--------|----|------------|------------|
| 10  | 58.0  | 20 | 62.0  | 20 | 72.0   | 20 | 72.0   | 20 | -24.1%     | -16.1%     |
| 9   | 58.0  | 20 | 63.0  | 18 | 73.0   | 20 | 73.0   | 20 | -25.9%     | -15.9%     |
| 8   | 58.0  | 20 | 63.0  | 18 | 77.0   | 20 | 77.0   | 20 | -32.8%     | -22.2%     |
| 7   | 58.0  | 20 | 67.0  | 20 | 77.0   | 20 | 77.0   | 20 | -32.8%     | -14.9%     |
| 6   | 58.0  | 20 | 68.0  | 20 | 81.0   | 20 | 81.0   | 20 | -39.7%     | -19.1%     |
| 5   | 59.0  | 20 | 72.0  | 20 | 83.0   | 20 | 83.0   | 20 | -40.7%     | -15.3%     |
| 4   | 62.0  | 20 | 77.0  | 20 | 92.0   | 20 | 92.0   | 20 | -48.4%     | -19.5%     |
| 3   | 70.0  | 20 | 82.0  | 20 | 101.0  | 20 | 101.0  | 20 | -44.3%     | -23.2%     |
| 2   | 82.0  | 20 | 101.0 | 20 | 110.0  | 20 | 110.0  | 20 | -34.1%     | -8.9%      |
| 1   | 97.0  | 20 | 111.0 | 20 | 121.0  | 20 | 121.0  | 20 | -24.7%     | -9.0%      |
| 0   | 113.0 | 20 | 122.0 | 20 | 133.0  | 20 | 133.0  | 20 | -17.7%     | -9.0%      |

Tabela H.31: Resultados de escalonamento do GAD *randômico* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 31.5 | 20 | 32.0 | 20 | 39.5   | 20 | 39.5   | 20 | -25.4%     | -23.4%     |
| 9   | 31.5 | 20 | 33.0 | 18 | 37.5   | 20 | 37.5   | 20 | -19.0%     | -13.6%     |
| 8   | 31.5 | 20 | 33.0 | 18 | 37.5   | 20 | 37.5   | 20 | -19.0%     | -13.6%     |
| 7   | 31.5 | 20 | 34.5 | 20 | 40.5   | 20 | 40.5   | 20 | -28.6%     | -17.4%     |
| 6   | 30.5 | 20 | 35.5 | 20 | 43.5   | 20 | 43.5   | 20 | -42.6%     | -22.5%     |
| 5   | 32.5 | 20 | 37.0 | 20 | 43.5   | 20 | 43.5   | 20 | -33.8%     | -17.6%     |
| 4   | 34.5 | 20 | 39.5 | 20 | 48.5   | 20 | 48.5   | 20 | -40.6%     | -22.8%     |
| 3   | 35.5 | 20 | 43.5 | 20 | 50.0   | 20 | 50.0   | 20 | -40.8%     | -14.9%     |
| 2   | 42.5 | 20 | 51.0 | 20 | 57.0   | 20 | 57.0   | 20 | -34.1%     | -11.8%     |
| 1   | 49.5 | 20 | 56.0 | 20 | 61.0   | 20 | 61.0   | 20 | -23.2%     | -8.9%      |
| 0   | 58.0 | 20 | 62.0 | 20 | 68.0   | 20 | 68.0   | 20 | -17.2%     | -9.7%      |

Tabela H.32: Resultados de escalonamento do GAD *randômico* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 10.5 | 20 | 10.0 | 20 | 13.0   | 20 | 11.5   | 20 | -9.5%      | -15.0%     |
| 9   | 10.5 | 20 | 9.5  | 18 | 13.0   | 20 | 11.5   | 20 | -9.5%      | -21.1%     |
| 8   | 11.0 | 20 | 9.5  | 19 | 13.0   | 20 | 11.5   | 20 | -4.5%      | -21.1%     |
| 7   | 12.0 | 20 | 10.0 | 20 | 13.0   | 20 | 11.5   | 20 | 4.2%       | -15.0%     |
| 6   | 11.5 | 20 | 10.0 | 20 | 13.0   | 20 | 11.5   | 20 | 0.0%       | -15.0%     |
| 5   | 12.0 | 20 | 10.5 | 20 | 13.0   | 20 | 12.0   | 20 | 0.0%       | -14.3%     |
| 4   | 12.0 | 20 | 11.0 | 20 | 13.0   | 20 | 12.0   | 20 | 0.0%       | -9.1%      |
| 3   | 12.0 | 20 | 11.5 | 20 | 13.0   | 20 | 13.0   | 20 | -8.3%      | -13.0%     |
| 2   | 13.0 | 20 | 13.0 | 20 | 14.0   | 20 | 14.0   | 20 | -7.7%      | -7.7%      |
| 1   | 14.0 | 20 | 14.5 | 20 | 14.0   | 20 | 14.0   | 20 | 0.0%       | 3.4%       |
| 0   | 18.0 | 20 | 16.0 | 20 | 15.0   | 20 | 15.0   | 20 | 16.7%      | 6.2%       |

Tabela H.33: Resultados de escalonamento do GAD *randômico* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 27.5 | 20 | 21.0 | 20 | 22.5   | 20 | 21.0   | 20 | 23.6%      | 0.0%       |
| 9   | 27.5 | 20 | 20.5 | 18 | 21.5   | 20 | 21.0   | 20 | 23.6%      | -2.4%      |
| 8   | 26.5 | 20 | 21.0 | 18 | 23.5   | 20 | 21.0   | 20 | 20.8%      | 0.0%       |
| 7   | 27.5 | 20 | 21.0 | 20 | 23.5   | 20 | 21.0   | 20 | 23.6%      | 0.0%       |
| 6   | 26.0 | 20 | 22.5 | 20 | 21.5   | 20 | 21.0   | 20 | 19.2%      | 6.7%       |
| 5   | 27.5 | 20 | 23.0 | 20 | 22.0   | 20 | 21.0   | 20 | 23.6%      | 8.7%       |
| 4   | 27.0 | 20 | 22.5 | 20 | 24.5   | 20 | 21.0   | 20 | 22.2%      | 6.7%       |
| 3   | 26.0 | 20 | 23.0 | 20 | 23.0   | 20 | 22.0   | 20 | 15.4%      | 4.3%       |
| 2   | 26.0 | 20 | 23.5 | 20 | 23.5   | 20 | 22.5   | 20 | 13.5%      | 4.3%       |
| 1   | 29.5 | 20 | 25.5 | 20 | 25.0   | 20 | 24.5   | 20 | 16.9%      | 3.9%       |
| 0   | 36.0 | 20 | 28.0 | 20 | 25.0   | 20 | 25.0   | 20 | 30.6%      | 10.7%      |

Tabela H.34: Resultados de escalonamento do GAD *randômico* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes duais.

| $x$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|-----|------|----|------|----|--------|----|--------|----|------------|------------|
| 10  | 46.0 | 20 | 30.5 | 20 | 32.0   | 20 | 27.5   | 20 | 40.2%      | 9.8%       |
| 9   | 46.0 | 20 | 30.5 | 18 | 33.0   | 20 | 27.5   | 20 | 40.2%      | 9.8%       |
| 8   | 46.0 | 20 | 36.0 | 18 | 33.0   | 20 | 27.5   | 20 | 40.2%      | 23.6%      |
| 7   | 46.5 | 20 | 36.5 | 20 | 34.5   | 20 | 27.5   | 20 | 40.9%      | 24.7%      |
| 6   | 46.0 | 20 | 33.0 | 20 | 33.5   | 20 | 27.5   | 20 | 40.2%      | 16.7%      |
| 5   | 46.5 | 20 | 33.0 | 20 | 34.5   | 20 | 28.0   | 20 | 39.8%      | 15.2%      |
| 4   | 46.5 | 20 | 32.5 | 20 | 34.5   | 20 | 28.5   | 20 | 38.7%      | 12.3%      |
| 3   | 46.0 | 20 | 34.0 | 20 | 34.5   | 20 | 29.0   | 20 | 37.0%      | 14.7%      |
| 2   | 45.5 | 20 | 34.5 | 20 | 34.0   | 20 | 30.0   | 20 | 34.1%      | 13.0%      |
| 1   | 45.0 | 20 | 33.5 | 20 | 35.5   | 20 | 34.0   | 20 | 24.4%      | -1.5%      |
| 0   | 55.0 | 20 | 42.0 | 20 | 39.0   | 20 | 36.0   | 20 | 34.5%      | 14.3%      |

Tabela H.35: Resultados de escalonamento do GAD *randômico* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes duais.

| $\bar{\sigma}$ | HEFT  |    | LDBS  |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|-------|----|-------|----|--------|----|--------|----|------------|------------|
| 0.000          | 113.0 | 20 | 122.0 | 20 | 133.0  | 20 | 133.0  | 20 | -17.7%     | -9.0%      |
| 0.125          | 97.0  | 20 | 107.0 | 20 | 133.6  | 20 | 133.6  | 20 | -37.8%     | -24.9%     |
| 0.250          | 87.7  | 20 | 103.9 | 20 | 127.3  | 20 | 127.3  | 20 | -45.3%     | -22.6%     |
| 0.500          | 80.3  | 20 | 96.8  | 18 | 127.0  | 20 | 127.0  | 20 | -58.2%     | -31.2%     |

Tabela H.36: Resultados de escalonamento do GAD *randômico* produzidos por HEFT, LDBS e CPH para  $(F, L) = (10, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 58.0 | 20 | 62.0 | 20 | 68.0   | 20 | 68.0   | 20 | -17.2%     | -9.7%      |
| 0.125          | 50.0 | 20 | 57.1 | 20 | 68.1   | 20 | 68.1   | 20 | -36.2%     | -19.1%     |
| 0.250          | 45.2 | 20 | 54.5 | 20 | 67.6   | 20 | 67.6   | 20 | -49.7%     | -24.2%     |
| 0.500          | 41.8 | 20 | 49.9 | 18 | 63.7   | 20 | 63.7   | 20 | -52.5%     | -27.7%     |

Tabela H.37: Resultados de escalonamento do GAD *randômico* produzidos por HEFT, LDBS e CPH para  $(F, L) = (5, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 18.0 | 20 | 16.0 | 20 | 15.0   | 20 | 15.0   | 20 | 16.7%      | 6.2%       |
| 0.125          | 14.3 | 20 | 14.1 | 20 | 15.2   | 20 | 15.2   | 20 | -6.8%      | -8.3%      |
| 0.250          | 14.7 | 20 | 14.2 | 20 | 17.7   | 20 | 17.7   | 20 | -20.5%     | -25.1%     |
| 0.500          | 12.2 | 20 | 12.9 | 16 | 15.4   | 20 | 15.4   | 20 | -26.5%     | -19.8%     |

**Tabela H.38:** Resultados de escalonamento do GAD *randômico* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 1)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 36.0 | 20 | 28.0 | 20 | 25.0   | 20 | 25.0   | 20 | 30.6%      | 10.7%      |
| 0.125          | 30.2 | 20 | 27.0 | 20 | 28.0   | 20 | 28.0   | 20 | 7.5%       | -3.7%      |
| 0.250          | 30.2 | 20 | 27.3 | 20 | 29.5   | 20 | 29.5   | 20 | 2.2%       | -7.9%      |
| 0.500          | 28.1 | 19 | 24.1 | 16 | 33.8   | 20 | 25.4   | 20 | 9.3%       | -5.5%      |

**Tabela H.39:** Resultados de escalonamento do GAD *randômico* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 5)$  em ambientes aleatórios.

| $\bar{\sigma}$ | HEFT |    | LDBS |    | CPH-o0 |    | CPH-o1 |    | CPH x HEFT | CPH x LDBS |
|----------------|------|----|------|----|--------|----|--------|----|------------|------------|
| 0.000          | 55.0 | 20 | 42.0 | 20 | 39.0   | 20 | 36.0   | 20 | 34.5%      | 14.3%      |
| 0.125          | 50.2 | 20 | 40.6 | 20 | 40.2   | 20 | 36.4   | 20 | 27.6%      | 10.4%      |
| 0.250          | 50.2 | 20 | 36.9 | 20 | 37.5   | 20 | 36.2   | 20 | 27.9%      | 1.9%       |
| 0.500          | 48.1 | 19 | 38.9 | 16 | 34.2   | 20 | 34.2   | 20 | 28.9%      | 12.0%      |

**Tabela H.40:** Resultados de escalonamento do GAD *randômico* produzidos por HEFT, LDBS e CPH para  $(F, L) = (1, 10)$  em ambientes aleatórios.

# Apêndice I

## Notação

|                |                                                                    |
|----------------|--------------------------------------------------------------------|
| $\epsilon(v)$  | peso de computação da tarefa $v$                                   |
| $\omega(u, v)$ | peso de comunicação do arco $(u, v)$                               |
| $ccr(G)$       | razão entre peso de comunicação e computação                       |
| $h(p)$         | fator de heterogeneidade do processador $p$                        |
| $\bar{h}$      | fator de heterogeneidade médio de um conjunto de processadores $P$ |
| $l(p, q)$      | latência de comunicação entre os processadores $p$ e $q$           |
| $com(u, v)$    | custo de comunicação no arco $(u, v)$                              |
| $S$ :          | escalonamento                                                      |
| $mspan(S)$     | <i>makespan</i> do escalonamento $S$                               |
| $level(v)$     | nível topológico da tarefa $v$                                     |
| $tlevel(v)$    | nível da tarefa $v$                                                |
| $blevel(v)$    | conível da tarefa $v$                                              |
| $cri(v)$       | predecessor crítico da tarefa $v$                                  |
| $pcri(v, p_j)$ | predecessor crítico da tarefa $v$ escalonada em $p_j$              |
| $CP$           | caminho crítico de um escalonamento                                |
| $t_I(v, p_j)$  | tempo de início da tarefa $v$ escalonada em $p_j$                  |
| $t_F(v, p_j)$  | tempo de fim da tarefa $v$ escalonada em $p_j$                     |

|                   |                                                  |
|-------------------|--------------------------------------------------|
| $t_{imc}(v, p_j)$ | tempo de início mais cedo da tarefa $v$ em $p_j$ |
| $t_{fmc}(v, p_j)$ | tempo de fim mais cedo da tarefa $v$ em $p_j$    |
| $t_{free}(p_j)$   | tempo em que o processador $p_j$ se torna ocioso |
| $tlevel_e(c)$     | tlevel escalonado da coleção $c$                 |
| $blevel_e(c)$     | blevel escalonado da coleção $c$                 |
| $\delta(c)$       | discrepância da coleção $c$                      |

# Bibliografia

- [1] AHMAD, I., E KWOK, Y. K. On exploiting task duplication in parallel program scheduling. *IEEE Transactions on Parallel and Distributed Systems* 9, 9 (1998), 872–900.
- [2] BANSAL, S., KUMAR, P., E SINGH, K. An improved duplication strategy for scheduling precedence constrained graphs in multiprocessor systems. *IEEE Transactions on Parallel and Distributed Systems* 14, 6 (Junho 2003), 533–544.
- [3] BEAUMONT, O., BOUDET, V., LEGRAND, A., RASTELLO, F., E ROBERT, Y. Heterogeneity considered harmful to algorithm designers. Relatório Técnico RR-2000-24, Laboratoire de l'Informatique du Parallélisme, École Normale Supérieure de Lyon, Junho 2000.
- [4] BEAUMONT, O., LEGRAND, A., E ROBERT, Y. Static scheduling strategies for heterogeneous systems. *Computing and Informatics* 21 (2002), 413–430.
- [5] BOERES, C. *Versatile Communication Cost Modelling for Multiprocessor Task Scheduling Heuristics*. PhD thesis, Department of Computer Science, University of Edinburgh, Maio 1997.
- [6] BOERES, C., CUNHA, G., E REBELLO, V. E. F. On minimising the processor requirements of LogP schedules. *Proceedings of the The 7th International Euro-Par Conference on Parallel Processing (Euro-Par'01)* (2001).
- [7] BOERES, C., LIMA, A., E REBELLO, V. E. F. Hybrid task scheduling: Integrating static and dynamic heuristics. *Journal of the 15th Symposium on*

- Computer Architecture and High Performance Computing - SBAC-PAD 2002* (Novembro 2003).
- [8] BOERES, C., NASCIMENTO, A. P., E REBELLO, V. E. F. Cluster-based task scheduling for the LogP model. *International Journal of Foundations of Computer Sciences* 10, 4 (1999), 405–424.
- [9] BOERES, C., E REBELLO, V. E. F. On the design of clustering-based scheduling algorithms for realistic machine models. In *The Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS'01)* (Abril 2001), IEEE Computer Society Press.
- [10] BOERES, C., E REBELLO, V. E. F. Towards optimal task scheduling for realistic machine models: Theory and practice. *The International Journal of High Performance Applications* 17, 2 (2003), 173–190.
- [11] BOUDET, V., E ROBERT, Y. Scheduling heuristics for heterogeneous processors. *2001 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'2001)* (2001), 2109–2115.
- [12] BUYYA, R. *High Performance Cluster Computing: Programming and Applications*. Prentice Hall PTR, New Jersey, 1999.
- [13] CASAVANT, T. L., E KUHL, J. G. A taxonomy of scheduling in general-purpose distributed computing systems. *IEEE Transactions on Software Engineering* 14, 2 (Fevereiro 1988), 141–154.
- [14] CHOE, T., E PARK, C. A task duplication based scheduling algorithm with optimality condition in heterogeneous systems. *Proceedings of the International Conference on Parallel Processing Workshops* (2002), 531–536.
- [15] CIROU, B., E JEANNOT, E. Triplet: a clustering scheduling algorithm for heterogeneous platforms. *Proceedings of the International Conference on Parallel Processing Workshops* (Setembro 2001), 231–236.
- [16] COLIN, J. Y., E CHRETIENNE, P. CPM scheduling with small communication time. *Operational Research* 39 (Março 1991), 680–684.

- [17] CORRÊA, R., FERREIRA, A., E REBREYEND, P. Scheduling multiprocessor tasks with genetic algorithms. *IEEE Transactions on Parallel and Distributed Systems* 10, 8 (Agosto 1999), 825–837.
- [18] COSNARD, M., E TRYSTRAM, D. *Parallel Algorithms and Architectures*. Int. Thomson Computer Press, 1995.
- [19] CULLER, D. E., KARP, R. M., PATTERSON, D. A., SAHAY, A., SCHAUER, K. E., SANTOS, E., SUBRAMONIAN, R., E VON EICKEN, T. LogP: Towards a realistic model of parallel computation. *4TH ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (Maio 1993), 1–12.
- [20] DOGAN, A., E ÖZGÜNER, F. LDBS: A duplication based scheduling algorithm for heterogeneous computing systems. *Proceedings of the International Conference on Parallel Processing (ICPP'02)* (2002).
- [21] EHRENFEUCHT, A., E ROZENBERG, G. Theory of 2-structures, part i: Clans, basic subclasses, and morphisms. *Theoretical Computer Science* 70, 3 (Fevereiro 1990), 277–303.
- [22] EL-REWINI, H., E LEWIS, T. G. Scheduling parallel program tasks onto arbitrary target machines. *Journal of Parallel and Distributed Computing* 9, 2 (1990), 138–153.
- [23] FOSTER, I. The grid: A new infrastructure for 21st century science. *Physics Today* (Fevereiro 2002).
- [24] FOSTER, I., E KESSELMAN, C. Computational grids. *Chapter 2 of The Grid: Blueprint for a New Computing Infrastructure* (1999).
- [25] FOSTER, I., KESSELMAN, C., E TUECKE, S. The anatomy of the grid. *Proceedings of International J. Supercomputer Applications* (2001).
- [26] GAREY, M. R., E JOHNSON, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.



- [27] GERASOULIS, A., VENUGOPAL, S., E YANG, T. Clustering task graphs for message passing architectures. *Proceedings 1990 International Conference on Supercomputing, ACM SIGARCH Computer Architecture News* 18, 3 (1990), 447–456.
- [28] HWANG, J. J., CHOW, Y. C., ANGER, F. D., E LEE, C. Y. Scheduling precedence graphs in systems with interprocessor communication times. *SIAM Journal on Computing* 18, 2 (Abril 1989), 244–257.
- [29] IVERSON, M., OZGUNER, F., E FOLLEN, G. Parallelizing existing applications in a distributed heterogeneous environment. *Proceedings of 4th Heterogeneous Computing Workshop (HCW '95)* (Abril 1995), 93–100.
- [30] KHAN, A. A., MCCREARY, C. L., E JONES, M. S. A comparison of multiprocessor scheduling heuristics. *Proceedings of International Conference on Parallel Processing* 2 (Abril 1994), 243–250.
- [31] KIANZAD, V., E BHATTACHARYYA, S. S. A comparison of clustering and scheduling techniques for embedded multiprocessor systems. Relatório Técnico UMIACS-TR-2003-114, Institute for Advanced Computer Studies, University of Maryland at College Park, Dezembro 2003.
- [32] KIM, S. J., E BROWNE, J. C. A general approach to mapping of parallel computations upon multiprocessor architectures. *International Conference on Parallel Processing* (1988).
- [33] KWOK, Y., E AHMAD, I. Dynamic critical-path scheduling: An effective technique for allocating task graphs to multiprocessors. *IEEE Transactions on Parallel and Distributed Systems* 7, 5 (1996), 506–521.
- [34] KWOK, Y., E AHMAD, I. Benchmarking and comparison of the task graph scheduling algorithms. *Journal of Parallel and Distributed Computing* 59, 3 (1999), 381–422.

- [35] KWOK, Y., E AHMAD, I. Static scheduling algorithms for allocating direct task graphs to multiprocessors. *ACM Computing Surveys* 31, 4 (Dezembro 1999), 367–432.
- [36] MACEY, B., E ZOMAYA, A. A performance evaluation of CP list scheduling heuristics for communication intensive task graphs. *Proceedings of IPPS/SPDP* (1998), 538–541.
- [37] MADRID, M., JACOBO-MOLINA, A., DING, J., E ARNOLD, E. Major sub-domain rearrangements in HIV-1 reverse transcriptase simulated by molecular dynamics. *Proteins: Structure, Function and Genetic* (1999), 332–337.
- [38] MAHESWARAN, M., E SIEGEL, H. J. A dynamic matching and scheduling algorithm for heterogeneous computing systems. *7th Heterogeneous Computing Workshop* (Março 1998), 57–69.
- [39] MATTSON, T. G., E HENRY, G. An overview of the intel TFLOPS supercomputer. *Intel Technology Journal Q1* (1998), 12.
- [40] MCCREARY, C. L., THOMPSON, J., GILL, H., SMITH, T., E ZHU, Y. Partitioning and scheduling using graph decomposition. Relatório Técnico CSE-93-06, Department of Computer Science and Engineering, Auburn University, Junho 1993.
- [41] MENDES, H. A. Um modelo de escalonamento de tarefas para execução eficiente da aplicações MPI em grades computacionais. Tese de Mestrado, Instituto de Computação - Universidade Federal Fluminense, Rio de Janeiro, Brasil, 2004.
- [42] NASCIMENTO, A. P. Aglomeração de tarefas em arquiteturas paralelas com memória distribuída. Tese de Mestrado, Instituto de Computação - Universidade Federal Fluminense, Rio de Janeiro, Brasil, 1999.
- [43] NORMAN, M. G., E THANISCH, P. Models of machines and computation for mapping in multicomputers. *ACM Computing Surveys* 25, 3 (1993), 263–302.

- [44] OH, H., E HA, S. A static scheduling heuristic for heterogeneous processors. In *Euro-Par, Vol. II* (1996), pp. 573–577.
- [45] PALIS, M. A., E LIOU, J.-C. A comparison of general approaches to multi-processor scheduling. *Proceedings of the 11th International Parallel Processing Symposium* (Abril 1997), 152–156.
- [46] PALIS, M. A., LIOU, J.-C., RAJASEKARAN, S., SHENDE, S., E WEI, D. S. L. Online scheduling of dynamic trees. *Parallel Processing Letters* 5, 4 (1995), 635–646.
- [47] PALIS, M. A., LIOU, J.-C., E WEI, D. S. L. Task clustering and scheduling for distributed memory parallel architectures. *IEEE Transactions on Parallel and Distributed Systems* 7, 1 (Maio 1996), 46–55.
- [48] PAPADIMITRIOU, C. H., E YANNAKAKIS, M. Towards an architecture-independent analysis of parallel algorithms. *SIAM Journal on Computing* 19, 2 (1990), 322–328.
- [49] PORTO, S. C. S., KITAJIMA, J. P., E RIBEIRO, C. C. Performance evaluation of a parallel tabu search task scheduling algorithm. *Parallel Computing* 26, 1 (Janeiro 2000), 73–90.
- [50] QUINN, M. J. *Parallel Computing: Theory and Practice*. McGraw-Hill Inc., 1987.
- [51] RANAWEERA, S., E AGRAWAL, D. P. A task duplication based scheduling algorithm for heterogeneous systems. *Proceedings of the 14th International Conference on Parallel and Distributed Processing Symposium (IPDPS-00)* (Maio 2000), 445–450.
- [52] SARKAR, V. *Partitioning and Scheduling Parallel Programs for Multiprocessors*. MIT Press, Cambridge, MA, 1989.
- [53] SHROFF, P., WATSON, D. W., FLANN, N. S., E FREUND, R. F. Genetic simulated annealing for scheduling data-dependent tasks in heterogeneous en-

- vironment. In *5th Heterogeneous Computing Workshop (HCW'96)* (Abril 1996), 98–117.
- [54] SIH, G. C., E LEE, E. A. A compile-time scheduling heuristic for interconnection-constrained heterogeneous processor architectures. *IEEE Transactions on Parallel and Distributed Systems* (Fevereiro 1993), 175–187.
- [55] SINNEN, O., E SOUZA, L. A classification of graph theoretic models for parallel computing. Relatório Técnico RT/005/99, Instituto Superior Técnico, Universidade Técnica de Lisboa, Maio 1999.
- [56] SINNEN, O., E SOUZA, L. Comparison of contention aware list scheduling heuristics for cluster computing. *Workshop on Scheduling and Resource Management for Cluster Computing* (Setembro 2001), 382–387.
- [57] SUTER, F., CASANOVA, H., DESPREZ, F., E BOUDET, V. From heterogeneous task scheduling to heterogeneous mixed data and task parallel scheduling. Relatório Técnico RR-2003-4995, Laboratoire de l'Informatique du Parallélisme, École Normale Supérieure de Lyon, Novembro 2003.
- [58] TOPCUOGLU, H., HARIRI, S., E WU, M. Performance-effective and low-complexity task scheduling for heterogeneous computing. In *IEEE Transactions on Parallel and Distributed Systems* 13(3) (2002), pp. 260–274.
- [59] YANG, T., E GERASOULIS, A. DSC: Scheduling parallel tasks on an unbounded number of processors. Relatório Técnico TRCS94-12, Department of Computer Science, Rutgers University, Fevereiro 1993.
- [60] YANG, T., E GERASOULIS, A. List scheduling with and without communication delays. *Parallel Computing Journal* 19, 12 (1993), 1321–1344.