



JÚNIOR ASSIS BARRETO BERNARDES

**UMA MODIFICAÇÃO NA HEURÍSTICA DE
SNAY PARA REDUÇÃO DO CUSTO
COMPUTACIONAL DO MÉTODO DOS
GRADIENTES CONJUGADOS**

LAVRAS - MG

2016

JÚNIOR ASSIS BARRETO BERNARDES

**UMA MODIFICAÇÃO NA HEURÍSTICA DE SNAY PARA
REDUÇÃO DO CUSTO COMPUTACIONAL DO MÉTODO
DOS GRADIENTES CONJUGADOS**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Inteligência Computacional e Processamento Digital de Imagens, para a obtenção do título de Mestre.

Orientador

Prof. Dr. Sanderson L. Gonzaga de Oliveira

LAVRAS - MG

2016

**Ficha catalográfica elaborada pelo Sistema de Geração de Ficha
Catalográfica da Biblioteca Universitária da UFLA, com dados
informados pelo próprio autor.**

Bernardes, Júnior Assis Barreto.

Uma modificação na heurística de Snay para redução do custo computacional do método dos gradientes conjugados/ Júnior Assis Barreto Bernardes. – Lavras : UFLA, 2016.

155 p. : il.

Dissertação (mestrado acadêmico) – Universidade Federal de Lavras, 2016.

Orientador: Sanderson L. Gonzaga de Oliveira.

Bibliografia.

1. Redução de profile. 2. Heurísticas. 3. Método dos Gradientes Conjugados condicionado. 4. Matrizes esparsas. I. Universidade Federal de Lavras. II. Título.

JÚNIOR ASSIS BARRETO BERNARDES

**UMA MODIFICAÇÃO NA HEURÍSTICA DE SNAY PARA
REDUÇÃO DO CUSTO COMPUTACIONAL DO MÉTODO
DOS GRADIENTES CONJUGADOS**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Inteligência Computacional e Processamento Digital de Imagens, para a obtenção do título de Mestre.

APROVADA em 17 de junho de 2016.

Prof. Dr. Alessandro Vivas Andrade UFVJM

Prof. Dra. Luciana Pereira de Assis UFVJM

Prof. Dr. Sanderson L. Gonzaga de Oliveira
Orientador

LAVRAS - MG

2016

À minha esposa, grande e eterno amor da minha vida, Carolina, dedico.

AGRADECIMENTOS

Agradeço a Deus por seu o meu caminho, a minha verdade e a minha vida.

Agradeço a minha esposa Carolina pelo amor, carinho, paciência, cumplicidade e apoio, mesmo com todas as minhas limitações e ausências durante o desenvolvimento deste trabalho; por me encher de felicidade e esperança com seu sorriso mesmo nos momentos mais difíceis que passamos até aqui; e por ter aberto mão de tantas coisas para que eu pudesse realizar esse sonho.

Agradeço aos meus pais Janete e Jorge por serem meus mestres e meus professores da vida, verdadeiros exemplos de humildade, generosidade e responsabilidade, e por nunca terem medido esforços para garantir minha educação.

Agradeço ao meu professor, orientador e amigo Sanderson por todos os ensinamentos, correções e sugestões, e pela compreensão nos momentos de dificuldade.

Agradeço aos colegas de mestrado e amigos Guilherme Chagas, pelo compartilhamento do projeto computacional *LinearSystemSolver*, pela liberação da publicação dos pseudocódigos das figuras 16, 17, 18 e 19; e Alexandre Abreu, pelas contribuições no projeto computacional *LinearSystemSolver*.

Agradeço aos meus amigos da Pró-Reitoria de Graduação e da Diretoria de Gestão e Tecnologia da Informação da UFLA pela sincera amizade e companheirismo que demonstraram em todos os momentos.

Ainda, agradeço ao professor Dr. Yifan Hu pela liberação da publicação das imagens das instâncias da *University of Florida sparse-matrix collection*. Finalmente, agradeço os apoios financeiros da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG).

RESUMO

Neste trabalho, é proposta uma heurística de baixo custo computacional para redução de *profile*, baseada na heurística de Snay, e denominada Snay-GL. A resolução de sistemas de equações lineares esparsos e de grande porte é crucial em diversas simulações computacionais nas áreas de ciência e engenharia. Os métodos iterativos são mais adequados para a resolução desses tipos de sistemas, e o Método dos Gradientes Conjugados (MGC) pré-condicionado é um dos métodos iterativos mais proeminentes. Uma localidade de memória adequada favorece a eficiência do MGC pré-condicionado, e essa característica pode ser obtida por ordenações locais geradas por heurísticas para reduções de largura de banda e de *profile*. Dezenas de heurísticas para redução de *profile* têm sido propostas desde a década de 1960. A heurística proposta neste trabalho foi comparada com as potenciais melhores heurísticas para redução de *profile* selecionadas em revisão sistemática. As comparações foram realizadas em relação à redução do custo computacional do MGC pré-condicionado pelo método de Jacobi e pela fatoração incompleta de Cholesky, em quatro conjuntos de instâncias. Os experimentos computacionais realizados neste trabalho confirmam uma bom desempenho da heurística Snay-GL proposta neste trabalho.

Palavras-chave: Redução de *profile*. Heurísticas. Método dos gradientes conjugados condicionado. Matrizes esparsas.

ABSTRACT

In this work, a low-cost algorithm for profile reduction is proposed. This algorithm, termed Snay-GL, is based on Snay's heuristic. The resolution of large sparse linear systems is fundamental in several science and engineering applications. Iterative methods are more suitable to solve these linear systems than the direct methods, and the Conjugated Gradient Method (CGM) is a prominent iterative method for solving large sparse linear systems. One can reduce computational costs using this method by applying a local ordering of the vertices of the corresponding graph to improve cache hit rates. This local ordering can be achieved using a heuristic for bandwidth and profile reductions. Several heuristics for profile reduction have been proposed since the 1960s. The results of the new algorithm are compared to the results of heuristics that were selected as potentially being the best low-cost methods for profile reduction. These results were obtained when solving linear systems using the preconditioned conjugate gradient method. Computational experiments confirm a good performance of the proposed reordering algorithm for the considered sets of linear systems.

Keywords: Profile reduction. Heuristics. Preconditioned Conjugate Gradient Method. Sparse matrices.

LISTA DE FIGURAS

Figura 1	Representação da matriz por meio de um grafo, com o valor do <i>profile</i> representado pelo número de elementos nulos e não nulos contidos na região delimitada pelo traço vermelho, e o valor da largura de banda denotado pelo número de elementos nulos e não nulos contidos na região delimitada pelo traço azul.....	29
Figura 2	Grafo da figura 1 com a numeração dos vértices alterada, com o valor do <i>profile</i> representado pelo número de elementos nulos e não nulos contidos na região delimitada pelo traço vermelho, e o valor da largura de banda denotado pelo número de elementos nulos e não nulos contidos na região delimitada pelo traço azul.	29
Figura 3	Pseudocódigo do algoritmo de Sloan (1989) para encontrar um par de vértices pseudoperiféricos.	39
Figura 4	Pseudocódigo da etapa de renumeração dos vértices da heurística de Sloan (1989).....	42
Figura 5	Exemplo da atribuição de <i>status</i> durante a segunda fase da heurística de Sloan (1989).....	43
Figura 6	Pseudocódigo da etapa de renumeração dos vértices da heurística NSloan.....	45
Figura 7	Pseudocódigo do algoritmo MGPS de Reid e Scott (1999) para encontrar um par de vértices pseudoperiféricos.....	48
Figura 8	Pseudocódigo da segunda fase da heurística MPG.	53
Figura 9	Pseudocódigo do algoritmo de contração da heurística Hu-Scott (HU; SCOTT, 2001).....	55
Figura 10	Etapas de contração, renumeração e prolongamento da heurística Hu-Scott. Na etapa de contração, são escolhidos os vértices para o grafo mais contraído (em vermelho) e montada a matriz de prolongamento <i>P</i> . Na etapa de renumeração, os vértices do grafo mais contraído são renumerados pela heurística de Sloan (1989). Na etapa de prolongamento, valores globais de prioridades para o grafo mais refinado (marcados com um retângulo vermelho) são obtidos por meio do produto da matriz de prolongamento e o vetor permutação que representa a renumeração do grafo mais contraído.	58

Figura 11	Pseudocódigo da heurística multinível de Hu e Scott (2001) para redução de <i>profile</i>	60
Figura 12	Método de Hager (2002) que utiliza trocas adjacentes descendentes para redução do <i>profile</i>	64
Figura 13	Implementação de Reid e Scott (2002) para os métodos de trocas adjacentes descendentes de Hager (2002).	66
Figura 14	Método de que Hager (2002) utiliza trocas adjacentes ascendentes para redução do <i>profile</i>	70
Figura 15	Implementação de Reid e Scott (2002) para os métodos de trocas adjacentes ascendentes de Hager (2002).	72
Figura 16	Pseudocódigo do método dos gradientes conjugados.	77
Figura 17	Pseudocódigo do método dos gradientes conjugados condicionado à esquerda.	79
Figura 18	Pseudocódigo do método dos gradientes conjugados condicionado por <i>split</i>	80
Figura 19	Pseudocódigo da fatoração incompleta de Cholesky com zero <i>fill-in</i>	82
Figura 20	Pseudocódigo do algoritmo para encontrar vértices iniciais para a heurística de Snay	86
Figura 21	Pseudocódigo da etapa de renumeração da heurística de Snay	88
Figura 22	Imagens das matrizes de coeficientes das instâncias da base <i>The University of Florida sparse-matrix Collection</i>	95
Figura 23	Imagens dos grafos relacionados às matrizes de coeficientes das instâncias da base <i>The University of Florida sparse-matrix Collection</i>	96
Figura 24	Speedup do MGCPJ obtido com aplicação das heurísticas, nas instâncias originadas de discretizações da equação da condução do calor por volumes finitos, mostrados nas tabelas 4, 5 e 6. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência.	106
Figura 25	Speedup do MGCPJ obtido com aplicação das 10 heurísticas com os melhores resultados, nas instâncias originadas de discretizações da equação da condução do calor por volumes finitos, mostrados nas tabelas 4, 5 e 6. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência.	106

Figura 26	Speedup do MGCPJ obtido com aplicação das 5 heurísticas com os melhores resultados, nas instâncias originadas de discretizações da equação da condução do calor por volumes finitos, mostrados nas tabelas 4, 5 e 6....	109
Figura 27	Quantidades de memória extra ocupadas pelas heurísticas WBRA-GL, Snay, Snay-GL, Sloan, NSloan, Sloan-MGPS, MPG e Hu-Scott, nas instâncias originadas de discretizações da equação da condução do calor por volumes finitos, mostradas nas tabelas 4, 5 e 6. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência.	112
Figura 28	Quantidades de memória extra ocupadas pelas heurísticas Snay, Snay-GL, Sloan, NSloan, Sloan-MGPS e MPG, nas instâncias originadas de discretizações da equação da condução do calor por volumes finitos, mostradas nas tabelas 4, 5 e 6.....	113
Figura 29	Tempos de execução, em segundos, das heurísticas RCM-GL, RBFS-GL, hGPHH-GL, WBRA-GL, Snay, Snay-GL, Sloan, NSloan, Sloan-MGPS, MPG e Hu-Scott nas instâncias oriundas da discretização da equação da condução do calor por volumes finitos. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência.	113
Figura 30	Tempos de execução, em segundos, das heurísticas RCM-GL, RBFS-GL, hGPHH-GL, NSloan e MPG nas instâncias oriundas da discretização da equação da condução do calor por volumes finitos.....	114
Figura 31	Quantidades de memória extra ocupadas pelas heurísticas WBRA-GL, Snay, Snay-GL, Sloan, NSloan, Sloan-MGPS e MPG, nas instâncias originadas de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada de forma aleatória, mostradas nas tabelas 7 e 8. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência..	119

Figura 32	Quantidades de memória extra ocupadas pelas heurísticas Snay, Snay-GL, Sloan, NSloan, Sloan-MGPS e MPG, nas instâncias originadas de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada de forma aleatória, mostradas nas tabelas 7 e 8.	119
Figura 33	Quantidades de memória extra ocupadas pelas heurísticas WBRA-GL, Snay, Snay-GL, Sloan, NSloan, Sloan-MGPS e MPG, nas instâncias originadas de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada pela curva de Sierpiński modificada, mostradas nas tabelas 9 e 10. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência.	120
Figura 34	Quantidades de memória extra ocupadas pelas heurísticas Snay, Snay-GL, Sloan, NSloan, Sloan-MGPS e MPG, nas instâncias originadas de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada pela curva de Sierpiński modificada, mostradas nas tabelas 9 e 10.	120
Figura 35	Speedup do MGC-IC0 obtido com aplicação das heurísticas, nas instâncias originadas de discretizações da equação da condução do calor por volumes finitos, mostrados nas tabelas 14 e 15. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência. .	129
Figura 36	Speedup do MGC-IC0 obtido com aplicação das 5 heurísticas com os melhores resultados, nas instâncias originadas de discretizações da equação da condução do calor por volumes finitos, mostrados nas tabelas 14 e 15.	129
Figura 37	Speedup do MGC-IC0 obtido com aplicação das heurísticas, nas instâncias originadas de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada de forma aleatória, mostrado nas tabelas 16 e 17. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência.	131

Figura 38	Speedup do MGC-IC0 obtido com aplicação das 5 heurísticas com melhores resultados, nas instâncias originadas de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada de forma aleatória, mostrado nas tabelas 16 e 17.	134
Figura 39	Speedup do MGC-IC0 obtido com aplicação das heurísticas, nas instâncias originadas de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada pela curva de Sierpiński modificada, mostrado nas tabelas 18 e 19. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência. .	134
Figura 40	Speedup do MGC-IC0 obtido com aplicação das 4 heurísticas com melhores resultados, nas instâncias originadas de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada pela curva de Sierpiński modificada, mostrado nas tabelas 18 e 19.	137

LISTA DE TABELAS

Tabela 1	Heurísticas para redução de <i>profile</i> de baixo custo computacional selecionadas na revisão sistemática de literatura.....	33
Tabela 2	11 SELs do conjunto de dados <i>The University of Florida sparse-matrix Collection</i> cujas matrizes de coeficientes são simétricas e positivo-definidas.	94
Tabela 3	Características das máquinas utilizadas para realizar as simulações computacionais.	103
Tabela 4	Resolução de sistemas de equações lineares (de até 23.637 incógnitas) pelo MGCPJ, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de <i>profile</i> . Essas simulações foram realizadas na máquina M4 e esses sistemas de equações lineares foram originados de discretizações da equação da condução do calor por volumes finitos.....	107
Tabela 5	Resolução de sistemas de equações lineares (de 50.592 a 232.052 incógnitas) pelo MGCPJ, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de <i>profile</i> . Essas simulações foram realizadas na máquina M4 e esses sistemas de equações lineares foram originados de discretizações da equação da condução do calor por volumes finitos.....	108
Tabela 6	Resolução de sistemas de equações lineares (de 492.853 e 965.545 incógnitas) pelo MGCPJ, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de <i>profile</i> . Essas simulações foram realizadas na máquina M4 e esses sistemas de equações lineares foram originados de discretizações da equação da condução do calor por volumes finitos.....	109
Tabela 7	Resolução de sistemas de equações lineares (de até 34.238 incógnitas) pelo MGCPJ, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de <i>profile</i> . Essas simulações foram realizadas na máquina M1 e esses sistemas de equações lineares foram originados de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada de forma aleatória.....	115

Tabela 8	Resolução de sistemas de equações lineares (de 75.542 a 277.118 incógnitas) pelo MGCPJ, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de <i>profile</i> . Essas simulações foram realizadas na máquina M1 e esses sistemas de equações lineares foram originados de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada de forma aleatória.....	116
Tabela 9	Resolução de sistemas de equações lineares (de até 105.764 incógnitas) pelo MGCPJ, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de <i>profile</i> . Essas simulações foram realizadas na máquina M2 e esses sistemas de equações lineares foram originados de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada pela curva de Sierpiński modificada.	117
Tabela 10	Resolução de sistemas de equações lineares (de 237.086 a 1.015.004 incógnitas) pelo MGCPJ, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de <i>profile</i> . Essas simulações foram realizadas na máquina M2 e esses sistemas de equações lineares foram originados de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada pela curva de Sierpiński modificada.	118
Tabela 11	Resultados de resoluções de sistemas de equações lineares, com o MGCPJ, e com renumeração dos vértices dada pelas heurísticas para reduções de largura de <i>profile</i> , oriundos das instâncias <i>nasa1824</i> , <i>nasa2910</i> , <i>nasa4704</i> e <i>nasasrb</i> , da <i>University of Florida sparse-matrix collection</i> . Estas simulações foram realizadas nas máquinas M1 (instâncias <i>nasa1824</i> , <i>nasa2910</i> e <i>nasasrb</i>) e M5 (instância <i>nasa4704</i>).	122

Tabela 12	Resultados de resoluções de sistemas de equações lineares, com o MGCPJ, e com renumeração dos vértices dada pelas heurísticas para reduções de largura de <i>profile</i> , oriundos das instâncias <i>sts4098</i> , <i>Pres_Poisson</i> , <i>olafu</i> e <i>raefsky4</i> do conjunto de dados da <i>University of Florida sparse-matrix collection</i> . Estas simulações foram realizadas nas máquinas M3 (instância <i>olafu</i>) e M5 (instâncias <i>sts4098</i> , <i>Pres_Poisson</i> e <i>raefsky4</i>).....	123
Tabela 13	Resultados de resoluções de sistemas de equações lineares, com o MGCPJ, e com renumeração dos vértices dada pelas heurísticas para reduções de largura de <i>profile</i> , oriundos das instâncias <i>textitthermall</i> , <i>2cubes_sphere</i> e <i>offshore</i> do conjunto de dados da <i>University of Florida sparse-matrix collection</i> . Estas simulações foram realizadas nas máquinas M6.....	124
Tabela 14	Resolução de sistemas de equações lineares (de até 23.637 incógnitas) pelo MGC-IC0, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de <i>profile</i> . Essas simulações foram realizadas na máquina M4 e esses sistemas de equações lineares foram originados de discretizações da equação da condução do calor por volumes finitos.....	127
Tabela 15	Resolução de sistemas de equações lineares (de 50.592 a 495.853 incógnitas) pelo MGC-IC0, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de <i>profile</i> . Essas simulações foram realizadas na máquina M4 e esses sistemas de equações lineares foram originados de discretizações da equação da condução do calor por volumes finitos.....	128
Tabela 16	Resolução de sistemas de equações lineares (de até 34,238 incógnitas) pelo MGC-IC0, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de <i>profile</i> . Essas simulações foram realizadas na máquina M1 e esses sistemas de equações lineares foram originados de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada de forma aleatória.....	132

Tabela 17	Resolução de sistemas de equações lineares (de 75,542 a 277,118 incógnitas) pelo MGC-IC0, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de <i>profile</i> . Essas simulações foram realizadas na máquina M1 e esses sistemas de equações lineares foram originados de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada de forma aleatória.....	133
Tabela 18	Resolução de sistemas de equações lineares (de até 68,414 incógnitas) pelo MGC-IC0, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de <i>profile</i> . Essas simulações foram realizadas na máquina M2 e esses sistemas de equações lineares foram originados de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada pela curva de Sierpiński modificada.	135
Tabela 19	Resolução de sistemas de equações lineares (de 105,764 a 467,504 incógnitas) pelo MGC-IC0, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de <i>profile</i> . Essas simulações foram realizadas na máquina M2 e esses sistemas de equações lineares foram originados de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada pela curva de Sierpiński modificada.	136
Tabela 20	Resultados de resoluções de sistemas de equações lineares, com o MGC-IC0, e com renumeração dos vértices dada pelas heurísticas para reduções de largura de <i>profile</i> , oriundos das instâncias <i>nasa1824</i> , <i>nasa2910</i> , <i>sts4098</i> , <i>nasa4704</i> , <i>Pres_Poisson</i> e <i>olafu</i> do conjunto de dados da <i>University of Florida sparse-matrix collection</i> . Estas simulações foram realizadas nas máquinas M1 (<i>olafu</i>) e M3 (<i>nasa1824</i> , <i>nasa2910</i> , <i>sts4098</i> , <i>nasa4704</i> e <i>Pres_Poisson</i>).....	140

Tabela 21	Resultados de resoluções de sistemas de equações lineares, com o MGC-IC0, e com renumeração dos vértices dada pelas heurísticas para reduções de largura de <i>profile</i> , oriundos das instâncias <i>raefsky4</i> , <i>nasasrb</i> , <i>thermal1</i> , <i>2cubes_sphere</i> , <i>offshore</i> , da <i>University of Florida sparse-matrix collection</i> . Estas simulações foram realizadas nas máquinas M3 (instâncias <i>nasasrb</i> , <i>raefsky4</i> e <i>thermal1</i>) e M6 (instâncias <i>2cubes_sphere</i> e <i>offshore</i>).	141
Tabela 22	Heurísticas que proporcionaram o maior valor de <i>speedup</i> para o MGCPJ e para o MGC-IC0 em cada conjunto de instâncias.	142

LISTA DE SIGLAS

RBFS-GL	Busca em largura iniciando por um vértice pseudoperiférico pelo algoritmo de George e Liu (1979) com renumeração invertida ao final
CM	<i>Cuthill-McKee</i>
RCM	<i>Cuthill-McKee reverso</i>
RCM-GL	<i>Cuthill-McKee reverso</i> iniciando por um vértice pseudoperiférico pelo algoritmo de George e Liu (1979)
RCM-GL-FL	aplicação da heurística <i>RCM-GL</i> em malhas de elementos finitos, realizada por Fenves e Law (1983)
EDP	Equação diferencial parcial
hGPHH-GL	Heurística gerada pela <i>Genetic Programming Hyper-Heuristic</i> iniciando por um vértice pseudoperiférico pelo algoritmo de George e Liu (1979)
MGC	Método dos Gradientes Conjugados
MGCPJ	Método dos Gradientes Conjugados Precondicionado por Jacobi
MGC-IC0	Método dos Gradientes Conjugados Precondicionado pela fatoração incompleta de Cholesky com zero <i>fill-in</i>
MPG	<i>Medeiros-Pimenta-Goldenberg</i>
NSloan	<i>Normalized Sloan</i> (heurística de Sloan com normalização dos fatores no cálculo da prioridade)
SEL	Sistema de equações lineares
Sloan-MGPS	Heurística de Sloan (1989) com vértices pseudoperiféricos determinados pelo método MGPS (<i>modified GPS</i>)
Snay-GL	Heurística de Snay (1976) modificada iniciando por um vértice pseudoperiférico pelo algoritmo de George e Liu (1979)
UFLA	Universidade Federal de Lavras
WBRA-GL	<i>Wonder Bandwidth Reduction Algorithm</i> modificado iniciando por um vértice pseudoperiférico pelo algoritmo de George e Liu (1979)

SUMÁRIO

1	INTRODUÇÃO	21
2	REFERENCIAL TEÓRICO	27
2.1	Conceitos básicos	28
2.2	Revisão sistemática da literatura	31
2.3	Algoritmo de George-Liu para encontrar um vértice pseudoperiférico	33
2.4	Heurística Cuthill-McKee reverso iniciada por um vértice pseudoperiférico	34
2.5	Heurísticas baseadas na heurística de Sloan	36
2.5.1	Heurística NSloan	41
2.5.2	Heurística Sloan-MGPS	46
2.6	Heurística de Snay (1976)	49
2.7	Heurística MPG	49
2.8	Heurística Hu-Scott	52
2.9	Métodos de trocas adjacentes	59
2.9.1	Trocas adjacentes descendentes	61
2.9.2	Trocas adjacentes ascendentes	68
2.10	Método dos gradientes conjugados	73
2.10.1	Método dos gradientes conjugados pré-condicionado ..	76
2.10.2	Métodos pré-condicionadores	81
2.11	Estruturas CSR-SSS e CSC-SSS	82
3	HEURÍSTICA SNAY-GL	85
4	METODOLOGIA	91
4.1	Instâncias utilizadas	91
4.1.1	Instâncias oriundas da discretização da equação da condução do calor	91
4.1.2	Instâncias oriundas da discretização da equação de Laplace	92
4.1.3	Instâncias da <i>University of Florida sparse-matrix Collection</i>	93
4.2	Pré-condicionadores utilizados com o MGC	93
4.3	Detalhes de implementação	98
4.4	Ferramentas utilizadas	101
5	SIMULAÇÕES COM RESOLUÇÃO DE SISTEMAS DE EQUAÇÕES LINEARES	104
5.1	Resultados de resoluções de sistemas de equações lineares com o MGCPJ	104

5.1.1	Instâncias oriundas da discretização da equação da condução do calor	106
5.1.2	Instâncias oriundas da discretização da equação de Laplace	112
5.1.3	Instâncias da <i>University of Florida sparse-matrix collection</i>	121
5.2	Resultados de resoluções de sistemas de equações lineares com o MGC-IC0	125
5.2.1	Instâncias oriundas da discretização da equação da condução do calor	126
5.2.2	Instâncias oriundas da discretização da equação de Laplace	130
5.2.3	Instâncias da <i>University of Florida sparse-matrix collection</i>	138
5.3	Resumo dos resultados das resoluções de sistemas de equações lineares com o MGCPJ e com o MGC-IC0	139
6	CONCLUSÃO E TRABALHOS FUTUROS.....	143
	REFERÊNCIAS.....	149

1 INTRODUÇÃO

A resolução de sistemas de equações lineares (SEL) da forma $Ax = b$ é uma das etapas da discretização de equações diferenciais parciais presente em simulações científicas realizadas em diversas áreas (BENZI, 2002). Como a matriz de coeficientes A desses sistemas é de grande porte e esparsa, uma boa parte do custo computacional dessas simulações é devido ao custo computacional da resolução do sistema de equações lineares.

Para sistemas de equações lineares de porte, relativamente, pequeno, métodos diretos encontram soluções exatas com um custo computacional razoável. Porém, para sistemas de equações lineares de grande porte e esparsos, métodos iterativos são utilizados para encontrar uma solução aproximada de maneira escalável. O Método dos Gradientes Conjugados (MGC) (HESTENES; STIEFEL, 1952; LANCZOS, 1952) é um dos métodos iterativos mais utilizados. Para reduzir o custo computacional do MGC, frequentemente, é utilizado o condicionamento, que diminui o número de iterações do MGC.

Além do condicionamento, outra forma de se obter uma redução no custo computacional do MGC é por meio de ordenações locais das equações do sistema de equações lineares (DUFF; MEURANT, 1989). Essas ordenações locais permitem uma melhor exploração da hierarquia de memória das arquiteturas de computadores atuais e das políticas de paginação existentes. Nesse sentido, dados que são utilizados em sequência nas execuções devem ser armazenados em endereços de memória próximos para favorecer o carregamento e manutenção desses dados nos níveis da hierarquia de memória mais próximos do processador, em que a leitura é mais rápida. Com isso, a redução do custo computacional do MGC é devida

à organização das equações do sistema de equações lineares na memória principal, de maneira a melhorar a coerência de *cache* (TEMAM; JALBY, 1992; BURGESS; GILES, 1997).

Uma localidade de memória adequada para o armazenamento das equações de um sistema de equações lineares que favorece a coerência de *cache* pode ser proporcionada por reduções de largura de banda e de *profile* da matriz de coeficientes do sistema de equações lineares, o que resulta em uma ordenação local das matrizes de coeficientes. Seja $A = [a_{ij}]$ uma matriz simétrica $n \times n$. A largura de banda de uma linha $1 \leq i \leq n$ dessa matriz é definida como o número de colunas existentes do primeiro elemento não nulo da linha i até a coluna mais próxima da diagonal principal. A largura de banda de A é igual largura de banda máxima entre todas as linhas de A . Por outro lado, o *profile* de A é definido como a soma da largura de banda de todas as linhas de A . A redução do *profile* de A é realizada por meio da reordenação das equações. Se a matriz A for considerada como a matriz de adjacências de um grafo G , a redução de *profile* de G é realizada por meio da renumeração dos vértices de G . O problema da minimização do *profile* de matrizes é NP-completo (LIN E YUAN, 1994). Com isso, dezenas de heurísticas têm sido propostas desde a década de 1960.

No contexto de resolução de SELs, é desejável que essas heurísticas possuam baixo custo computacional. Nesse caso, considera-se que uma heurística possui baixo custo computacional quando o tempo de execução dessa heurística é suficientemente pequeno de modo que a soma do tempo de execução da heurística com o tempo de execução do MGC pré-condicionado após a renumeração dos vértices pela heurística é menor que o tempo de execução do MGC pré-condicionado sem a renumeração dos vértices. Com

isso, o custo computacional total da resolução será baixo, e a redução do custo computacional do MGC compensará o custo computacional da heurística utilizada. Nesse sentido, há uma evidência, mostrada por Chagas (2015), de que heurísticas baseadas em meta-heurísticas (que expandem o espaço de busca por soluções melhores, empregam busca local etc.), para reduções de largura de banda e de *profile*, não são melhores que uma heurística de baixo custo computacional (que não é baseada em meta-heurística), quando aplicadas para reduzir o custo total de resolução de SELs pelo MGC.

Neste trabalho, é proposta uma heurística de baixo custo computacional para redução de *profile*, denominada Snay-GL, a partir de uma modificação na primeira fase da heurística de Snay (1976). Os resultados dessa heurística são comparados com os resultados de outras 10 heurísticas para reduções de largura de banda e de *profile* selecionadas em revisões sistemáticas de literatura: as heurísticas RCM-GL (GEORGE; LIU, 1981) e hGPHH-GL (KOOHESTANI; POLI, 2011), identificadas por Chagas e Gonzaga de Oliveira (2015) como as melhores heurísticas para redução de largura de banda com baixo custo computacional; as heurísticas RBFS-GL e WBRA-GL, propostas no trabalho de Chagas (2015); e as heurísticas MPG (MEDEIROS; PIMENTA; GOLDENBERG, 1993), NSloan (KUMFERT e POTHEN, 1997), Sloan-MGPS (REID e SCOTT, 1999), Hu-Scott (HU; SCOTT, 2001), de Snay (1976) e de Sloan (1989), selecionadas no trabalho de Bernardes e Gonzaga de Oliveira (2015) como as potenciais melhores heurísticas para redução de *profile* com baixo custo computacional (a heurística RCM-GL (GEORGE; LIU, 1981) também foi selecionada nessa revisão sistemática de literatura). A heurística Snay-GL

apresentou resultados comparáveis com os resultados das demais heurísticas testadas, e obteve os melhores resultados em algumas instâncias de teste.

Nas revisões sistemáticas que foram realizadas, tanto neste trabalho como no trabalho de Chagas e Gonzaga de Oliveira (2015), foram selecionadas apenas heurísticas para reduções de largura de banda e de *profile* com baixo custo computacional. Ordenações locais de vértices do grafo associado com a matriz de adjacências de sistemas de equações lineares podem contribuir para reduzir o custo computacional de um método iterativo, tal como o Método dos Gradientes Conjugados (MGC). Isso é realizado por meio do aumento da quantidade de *cache hits* (DAS et al., 1994; BURGESS; GILES, 1997). Por outro lado, as reduções de *profile* e de largura de banda não são diretamente proporcionais à redução do custo computacional da resolução do sistema de equações lineares por um método iterativo. Claramente, o custo computacional dessa parte da simulação é dada pela soma do custo computacional da heurística para a reordenação de vértices e do custo computacional do resolutor do sistema de equações lineares, e é fundamental que a heurística possua baixo custo computacional.

Devido às características dos problemas da minimização da largura de banda e da minimização de *profile*, há centenas de heurísticas propostas desde a década de 1960 que podem ser utilizadas para se obter uma ordenação local. Além disso, não foram encontrados na literatura trabalhos que avaliassem e comparassem a aplicação de heurísticas para redução de *profile* na redução do custo computacional do MGC preconditionado. Ainda, foi possível perceber que a heurística de Snay (1976), selecionada como uma das potenciais melhores heurísticas para redução de *profile* com baixo custo computacional, possui uma etapa em que pode ser otimizada

e, com isso, se tornar adequada para reduzir o custo computacional do MGC preconditionado. Portanto, é possível justificar o presente trabalho tanto na necessidade de se otimizar a heurística de Snay (1976) com o objetivo de reduzir o custo computacional do MGC preconditionado, como na necessidade de se avaliar e comparar a aplicação das heurísticas para redução de *profile*, com baixo custo computacional, na redução do custo computacional do MGC preconditionado.

O objetivo geral deste trabalho é propor uma heurística modificada, de baixo custo computacional, para redução de *profile* a partir de uma modificação da heurística de Snay (1976), visando à redução do custo computacional do Método dos Gradientes Conjugados preconditionado pelos métodos de Jacobi e de Cholesky, em quatro conjuntos de instâncias de sistemas de equações lineares oriundos de discretizações de equações diferenciais parciais. Os objetivos específicos deste trabalho são:

1. implementar e avaliar as heurísticas para a redução de *profile* com baixo custo computacional, selecionadas em revisões sistemáticas de literatura, como pré-processamento para redução do custo computacional do MGC preconditionado pelo método de Jacobi e pelo método baseado na fatoração incompleta de Cholesky, em cada um dos conjuntos de instâncias utilizados;
2. comparar os resultados, em relação à redução do custo computacional do MGC preconditionado, das heurísticas para a redução de *profile*, selecionadas neste trabalho, com os resultados das melhores heurísticas para a redução de largura de banda, indicadas por Chagas (2015);

3. identificar, entre as heurísticas para reduções de largura de banda e de *profile* com baixo custo computacional selecionadas em revisões sistemáticas de literatura, as heurísticas mais adequadas para redução do custo computacional do MGC preconditionado pelo método de Jacobi em cada um dos conjuntos de instâncias utilizados;
4. identificar, entre as heurísticas para reduções de largura de banda e de *profile* com baixo custo computacional selecionadas em revisões sistemáticas de literatura, as heurísticas mais adequadas para redução do custo computacional do MGC preconditionado pela fatoração incompleta de Cholesky com zero *fill-in* em cada um dos conjuntos de instâncias utilizados.

Este trabalho está organizado como a seguir. No capítulo 2, são apresentados os conceitos básicos para o entendimento deste trabalho e a revisão sistemática realizada para selecionar as possíveis melhores heurísticas com baixo custo computacional para redução de *profile*, bem como são descritas essas heurísticas selecionadas. No capítulo 3, é apresentada a heurística modificada para redução de *profile*, com baixo custo computacional, proposta neste trabalho. No capítulo 4, é apresentada a metodologia adotada para o desenvolvimento deste trabalho, bem como os detalhes de implementação e as ferramentas utilizadas. No capítulo 5, são mostrados os resultados obtidos nas simulações com resoluções de SELs com o MGC preconditionado, com as reduções de largura de banda e de *profile* dadas pela aplicação da heurística proposta neste trabalho e pelas demais heurísticas testadas. Por fim, no capítulo 6, são apresentadas as conclusões obtidas com este trabalho, bem como os trabalhos futuros a serem conduzidos para a continuação deste trabalho.

2 REFERENCIAL TEÓRICO

Neste capítulo, é apresentado o conteúdo necessário para o entendimento deste trabalho. Primeiramente, na seção 2.1, são apresentadas definições sobre grafos e matrizes de adjacências utilizadas neste texto. Na seção 2.2, é mostrada a revisão sistemática de literatura realizada para selecionar as potenciais melhores heurísticas para redução de *profile*. O algoritmo para se encontrar um vértice pseudoperiférico de George e Liu (1979) é descrito na seção 2.3. Apresentam-se, na seção 2.4, as heurísticas Cuthill-McKee (CM) (CUTHILL; MCKEE, 1969), Cuthill-McKee reverso (RCM) (GEORGE, 1971) e Cuthill-McKee reverso com vértice pseudoperiférico pelo algoritmo de George e Liu (1979) (RCM-GL). Na seção 2.5, são apresentadas as heurísticas: de Sloan (1989), NSloan (KUMFERT e POTHEN, 1997) (subseção 2.5.1) e Sloan-MGPS (REID e SCOTT, 1999) (subseção 2.5.2). Além disso, comentam-se as heurísticas SNAY (SNAY, 1976), MPG (MEDEIROS; PIMENTA; GOLDENBERG, 1993) e Hu-Scott (HU; SCOTT, 2001) nas Seções 2.6, 2.7 e 2.8, respectivamente. Os métodos de trocas de Reid e Scott (2002) são mostrados na seção 2.9.

Por fim, mostra-se, na seção 2.10, o método dos gradientes conjugados e o método dos gradientes conjugados preconditionado. O preconditionador de Jacobi e a fatoração incompleta de Cholesky com zero *fill-in* são detalhados na seção 2.10.2. As estruturas de armazenamento de matrizes simétricas esparsas CSC-SSS e CSR-SSS são apresentadas na seção 2.11.

2.1 Conceitos básicos

Um sistema de equações lineares, pode ser escrito na forma matricial como $Ax = b$. Neste trabalho, serão considerados apenas sistemas de equações lineares cuja matriz de coeficientes $A = [a_{ij}]$ é uma matriz $n \times n$ esparsa, simétrica e positivo-definida. Essa matriz A , por sua vez, pode ser considerada uma matriz de adjacências de um grafo não direcionado $G = (V, E)$, em que V é um conjunto finito de vértices e E é um conjunto finito de arestas $e = \{u, v\}$, tal que $u, v \in V$ (GONZAGA DE OLIVEIRA, 2011).

Como exemplo, tem-se a representação de um grafo por meio de uma matriz de adjacências na figura 1. Cada linha da matriz A corresponde a um vértice no grafo da figura 1, ou seja, para $1 \leq i \leq 6$, a linha i da matriz corresponde ao vértice i do grafo. Para o grafo da figura 2 utilizou-se uma ordem diferente da anterior para numerar os vértices do grafo e, conseqüentemente, obteve-se uma matriz de adjacências diferente. Nota-se que a disposição dos coeficientes não nulos na representação matricial do grafo depende da ordem em que se numeram os vértices do grafo para a montagem da matriz. A seguir, têm-se as definições de alguns conceitos básicos necessários para o entendimento deste trabalho.

Definição 1 (largura de banda). *Seja A uma matriz simétrica $n \times n$, com coeficientes a_{ij} e $1 \leq i, j \leq n$. A largura de banda da i -ésima linha é $\beta_i(A) = i - \min((1 \leq j < i) \mid a_{ij} \neq 0)$. A largura de banda $\beta(A)$ é a maior distância de coeficiente não nulo da matriz triangular inferior até a diagonal principal, considerando-se todas as n linhas da matriz, ou seja, $\beta(A) = \max((1 \leq i \leq n) \mid \beta_i(A))$ (GONZAGA DE OLIVEIRA; CHAGAS, 2014). Para uma matriz A assimétrica, a largura de banda da i -ésima linha*

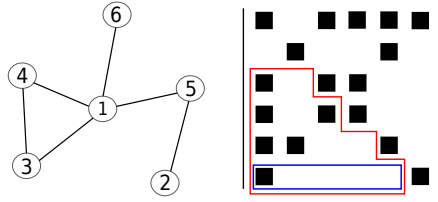


Figura 1 Representação da matriz por meio de um grafo, com o valor do *profile* representado pelo número de elementos nulos e não nulos contidos na região delimitada pelo traço vermelho, e o valor da largura de banda denotado pelo número de elementos nulos e não nulos contidos na região delimitada pelo traço azul.

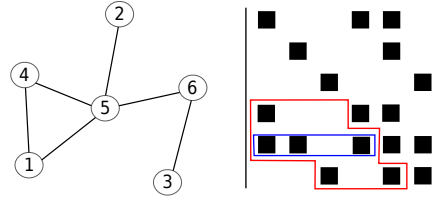


Figura 2 Grafo da figura 1 com a numeração dos vértices alterada, com o valor do *profile* representado pelo número de elementos nulos e não nulos contidos na região delimitada pelo traço vermelho, e o valor da largura de banda denotado pelo número de elementos nulos e não nulos contidos na região delimitada pelo traço azul.

é $\beta_i(A) = \max(|i - j| \mid a_{ij} \neq 0)$, para $1 \leq i, j \leq n$. A largura de banda de A é $\beta(A) = \max(\beta_i(A))$, $1 \leq i \leq n$ (GONZAGA DE OLIVEIRA; CHAGAS, 2014).

Definição 2 (profile). O *profile* de A , uma matriz simétrica, é definido como $profile = \sum_{i=1}^n \beta_i(A)$ (GONZAGA DE OLIVEIRA; CHAGAS, 2014). Seja A uma matriz assimétrica e L e U sejam as partições de A em matrizes triangulares inferior e superior, respectivamente. O *profile* de A é definido como $profile = \sum_{i=1}^n (\beta_i(L) + \beta_i(U))$ (GONZAGA DE OLIVEIRA; CHAGAS, 2014).

Definição 3 (distância). A distância $d(v, u)$ entre dois vértices v e $u \in V$ é o tamanho do menor caminho entre eles (GONZAGA DE OLIVEIRA; CHAGAS, 2014).

Definição 4 (excentricidade). A excentricidade $\ell : V \rightarrow \mathbb{N}$ de um vértice $v \in V$ é dada por $\ell(v) = \max((\forall u \in V) d(v, u))$ (GONZAGA DE OLIVEIRA; CHAGAS, 2014).

Definição 5 (diâmetro do grafo). O diâmetro $\Phi(G) = \max((\forall v \in V) \ell(v)) = \max((\forall v, u \in V) d(v, u))$ do grafo $G = (V, E)$ é a maior excentricidade encontrada em $G = (V, E)$ (GONZAGA DE OLIVEIRA; CHAGAS, 2014).

Definição 6 (vértice periférico). Um vértice v é considerado periférico se a sua excentricidade é igual ao diâmetro do grafo, ou seja, $\ell(v) = \Phi(G)$ (GONZAGA DE OLIVEIRA; CHAGAS, 2014).

Definição 7 (vértice pseudoperiférico). Um vértice é considerado pseudoperiférico se sua excentricidade for próxima ao diâmetro do grafo (GIBBS; POOLE; STOCKMEYER, 1976).

Definição 8 (estrutura de nível enraizada). Seja $G = (V, E)$ conexo e simples. Dado um vértice $v \in V$, a estrutura de nível com raiz v e profundidade $\ell(v)$ é o particionamento $\mathcal{L}(v) = \{L_0(v), L_1(v), \dots, L_{\ell(v)}(v)\}$, em que $L_0(v) = \{v\}$ e $L_i(v) = \text{Adj}(L_{i-1}(v)) - \bigcup_{j=0}^{i-1} L_j(v)$, para $i = 1, 2, 3, \dots, \ell(v)$ e $\text{Adj}(\cdot)$ retorna os vértices adjacentes aos vértices do argumento (GONZAGA DE OLIVEIRA; CHAGAS, 2014).

Definição 9 (estrutura de nível). Seja $G = (V, E)$ conexo e simples. A estrutura de nível é o particionamento $\mathcal{K}(v, \dots) = \{K_0(v, \dots), K_1(v, \dots), \dots, K_{\ell(v)}(v, \dots)\}$, em que $v \in K_0(v, \dots)$ e $K_i(v, \dots) = \text{Adj}(K_{i-1}(v, \dots)) - \bigcup_{j=0}^{i-1} K_j(v, \dots)$, para $i = 1, 2, 3, \dots, \ell(v)$ (GONZAGA DE OLIVEIRA; CHAGAS, 2014).

Definição 10 (largura de nível). *A largura de nível $b(\mathcal{L}(u))$ é o número de vértices do nível com mais vértices, ou seja, $b(\mathcal{L}(u)) = \max((0 \leq i \leq \ell(u)) |L_i(u)|)$ (GONZAGA DE OLIVEIRA; CHAGAS, 2014).*

2.2 Revisão sistemática da literatura

Nesta seção, é mostrada a revisão sistemática de literatura que foi realizada para selecionar as potenciais melhores heurísticas para redução de *profile* com baixo custo computacional. Para mais detalhes sobre essa revisão sistemática, veja o trabalho de Bernardes e Gonzaga de Oliveira (2015).

Uma vez que o benefício da redução de *profile* não é o único objetivo, em determinadas situações, no contexto das simulações computacionais, não é trivial identificar qual a melhor heurística para redução de *profile* entre todas heurísticas propostas desde a década de 1960. Na resolução de SELs, por exemplo, a redução de *profile* é desejável para se obter uma ordenação local que torne o método iterativo mais eficiente, mas o custo computacional da heurística também deve ser levado em consideração. Ainda, é possível que a ocupação de memória da heurística também seja um fator limitante na simulação. Com isso, foi realizada uma revisão sistemática de literatura para encontrar heurísticas para redução de *profile* que proporcione boa redução de *profile*, porém com custos computacionais relativamente baixos.

A revisão sistemática foi realizada na base de artigos Scopus (<http://www.scopus.com/>). Na busca, foi utilizada a seguinte chave de busca: **(TITLE-ABS-KEY(profile) OR TITLE-ABS-KEY(envelope)) AND (TITLE-ABS-KEY(reduction) OR TITLE-ABS-KEY(reducing) OR TITLE-ABS-KEY(minimization) OR TITLE-ABS-**

KEY(minimisation) OR **TITLE-ABS-KEY**(minimizing) OR **TITLE-ABS-KEY**(optimization) OR **TITLE-ABS-KEY**(optimizing)) AND (**TITLE-ABS-KEY**(sparse matrix) OR **TITLE-ABS-KEY**(sparse matrices) OR **TITLE-ABS-KEY**(stiffness matrix)), em que o termo definido por **TITLE-ABS-KEY** é buscado no título, *abstract* e palavras-chave dos documentos. Com essa busca, foram obtidos 213 artigos. Foram identificados, entre todos os 213 artigos, aqueles que apresentavam propostas de heurísticas para redução de *profile* ou que apresentavam comparações entre heurísticas para redução de *profile*. Além disso, as referências de todos os artigos sobre redução de *profile* foram analisadas na tentativa de se encontrar outros artigos com proposição ou comparação de heurísticas para redução de *profile*. Com isso, ao final dessa etapa, 74 heurísticas para redução de largura de banda foram identificadas.

Foram analisados, nos artigos encontrados, os resultados de todas essas 74 heurísticas em relação à redução de *profile* e ao custo computacional. Com isso, foi possível identificar oito heurísticas como sendo as heurísticas com os melhores benefícios (i. e. redução de *profile*) por custos computacionais. As heurísticas identificadas são mostradas na tabela 1. Com a exceção da heurística RCM-GL-FL de Fenves e Law (1983), que trata de uma aplicação da heurística RCM-GL específica para malhas de elementos finitos, todas as outras sete heurísticas foram avaliadas neste trabalho. Essas heurísticas são descritas, em detalhes, na continuação deste capítulo.

Tabela 1 Heurísticas para redução de *profile* de baixo custo computacional selecionadas na revisão sistemática de literatura.

Snay (1976)
RCM-GL (GEORGE; LIU, 1981)
RCM-GL-FL (FENVES; LAW, 1983)
Sloan (1989)
MPG (MEDEIROS; PIMENTA; GOLDENBERG, 1993)
NSloan (KUMFERT e POTHEN, 1997)
Sloan-MGPS (REID e SCOTT, 1999)
Hu-Scott (HU; SCOTT, 2001)

2.3 Algoritmo de George-Liu para encontrar um vértice pseudoperiférico

As heurísticas para a redução de *profile* não baseadas em meta-heurísticas, geralmente, são estruturadas em duas etapas. Na primeira etapa, é determinado um vértice inicial, a partir do qual será iniciada a renumeração dos vértices do grafo. A seleção de um vértice pseudoperiférico inicial é justificada pelo fato de que vértices pseudoperiféricos possuem excentricidades grandes e, conseqüentemente, possui a largura de nível, da estrutura de nível enraizada, pequena. Segundo Cuthill e McKee (1969), se a renumeração dos vértices de $G = (V, E)$ é realizada nível a nível de $\mathcal{H}(v, \dots)$, então, $\beta(A) \leq 2b(\mathcal{H}(v, \dots)) - 1$. Ainda, se a estrutura de nível é enraizada, a largura de banda de A não pode ser menor que a largura de nível da estrutura de nível enraizada. Em seguida, na segunda etapa, os vértices do grafo são renumerados, de acordo com o funcionamento da heurística.

George e Liu (1979) propuseram um algoritmo para se encontrar um vértice pseudoperiférico para o início da renumeração por parte da heurística RCM (GEORGE, 1971). Nesse algoritmo, primeiramente, é escolhido um

vértice $v \in V$ qualquer. Em seguida, gera-se a estrutura de nível enraizada $\mathcal{L}(v)$. Um vértice $u \in L_{\ell(v)}(v)$ de grau mínimo é selecionado. Após a construção da estrutura de nível enraizada $\mathcal{L}(u)$, verifica-se se $\ell(u) > \ell(v)$. Se isso ocorrer, o vértice u passa a ser o corrente e todo o processo é repetido. Caso contrário, o algoritmo termina e o vértice v é selecionado como um vértice pseudoperiférico. Para mais detalhes sobre o algoritmo de George e Liu (1979), veja Gonzaga de Oliveira e Chagas (2014, p. 70-71).

2.4 Heurística Cuthill-McKee reverso iniciada por um vértice pseudoperiférico

A heurística Cuthill-McKee reverso (RCM) foi proposta por George (1971) como uma melhoria da heurística Cuthill-McKee (CM) proposta por Cuthill e McKee (1969). Implementações dessas heurísticas podem ser encontradas em *softwares* e bibliotecas, como, por exemplo, MATLAB e *Boost C++ Libraries*.

A única diferença entre as heurísticas CM e RCM é que, na heurística RCM, inverte-se a renumeração final obtida. George (1971) constatou que inverter a renumeração obtida pela heurística CM resulta em um *profile* pelo menos tão bom quanto o *profile* sem inversão da renumeração, e a largura de banda não é alterada (LIU; SHERMAN, 1976; LIU, 1976).

A heurística CM possui três etapas: escolha dos vértices iniciais, construção das estruturas de níveis enraizadas em cada um dos vértices iniciais e renumeração nível a nível. Além dessas três etapas, a heurística RCM possui uma quarta etapa, que consiste na inversão da renumeração obtida na terceira etapa.

Como foi constatado que o vértice inicial interfere no *profile* e na

largura de banda da solução encontrada (CUTHILL; MCKEE, 1969), na primeira etapa das heurísticas CM e RCM, são escolhidos os vértices de grau mínimo. Na segunda etapa, são geradas as estruturas de nível enraizadas de todos os vértices selecionados na primeira etapa. Para cada estrutura de nível enraizada gerada, que possua a menor largura de nível, realiza-se a renumeração dos vértices de cada estrutura de nível enraizada selecionada na segunda etapa. A renumeração é realizada nível a nível da estrutura de nível enraizada. Dessa maneira, a renumeração realizada pelas heurísticas CM e RCM assemelha-se a visitação dos vértices na busca em largura. Após todos os vértices de todas as estruturas de nível enraizadas armazenadas forem renumerados, a renumeração que resultar na menor largura de banda será a renumeração final dos vértices do grafo. Ainda, na heurística RCM, inverte-se a renumeração final. Para mais detalhes sobre as heurísticas CM e RCM, veja Gonzaga de Oliveira e Chagas (2014, p. 15-19).

Além de verificarem que as heurísticas CM e RCM realizavam um esforço computacional desnecessário gerando várias estruturas de nível, Gibbs, Poole e Stockmeyer (1976) popularizaram o uso de um vértice pseudoperiférico como vértice inicial de heurísticas baseadas em estrutura de nível. De fato, a escolha de vértices de grau mínimo, a construção de estruturas de níveis enraizadas para cada um desses vértices e a posterior renumeração de todos os vértices do grafo contribuem significativamente para o aumento do custo computacional e de armazenamento das heurísticas CM e RCM. Com isso, George e Liu (1981) propuseram a substituição da primeira etapa das heurísticas CM e RCM pelo algoritmo de George e Liu (1979). Assim, a renumeração dos vértices de um grafo passaram a ser realizadas iniciando-se apenas de um vértice pseudoperiférico. Neste

trabalho, é testada essa versão da heurística RCM, denominada neste trabalho como RCM-GL. Essa versão da heurística RCM é a heurística que se encontra implementada no MATLAB.

2.5 Heurísticas baseadas na heurística de Sloan

A heurística proposta por Sloan (1986) é composta de duas etapas. Na primeira etapa, de maneira similar ao que ocorre na heurística RCM-GL, são determinados dois vértices pseudoperiféricos. Um deles é utilizado como vértice inicial para a renumeração, como ocorre na heurística RCM-GL, e o outro é utilizado na função de priorização que determina a ordem dos vértices renumerados na segunda etapa. Nessa primeira etapa da heurística de Sloan (1986), cuja estrutura de repetição é similar à estrutura do algoritmo de George e Liu (1979), primeiramente, é escolhido um vértice qualquer v do grafo a ser renumerado. Em seguida, gera-se a estrutura de nível enraizada $\mathcal{L}(v)$. Um conjunto de vértices do último nível de $\mathcal{L}(v)$, denotado por Q , é selecionado. Após a construção de estruturas de níveis $\mathcal{L}(w)$, enraizadas em cada vértice $w \in Q$, verifica-se se $\ell(w) > \ell(v)$. Se isso ocorrer para algum vértice w , o vértice w passa a ser o corrente e todo o processo é repetido. Caso contrário, o algoritmo termina, e o vértice v e o vértice com menor largura de nível, entre os vértices pertencentes a Q , são retornados.

Na segunda etapa da heurística de Sloan (1986), os vértices do grafo são renumerados na ordem decrescente definida por uma função de priorização. Ao calcular a prioridade de um determinado vértice v , são levados em consideração o grau de v e a distância do vértice v em relação a um dos vértices pseudoperiféricos determinados na primeira etapa. Além

disso, são utilizados dois pesos w_1 e w_2 para ponderar a influência dessas duas características no cálculo da prioridade de cada vértice.

No trabalho de Sloan (1989), é proposta uma implementação para a heurística de Sloan (1986). Nessa implementação, duas modificações foram realizadas em relação à heurística original. A primeira modificação proposta foi em relação ao método para encontrar vértices pseudoperiféricos correspondente à primeira etapa da heurística de Sloan (1986). Nesse método os vértices do último nível da estrutura de nível corrente são ordenados em ordem crescente de grau e seleciona-se apenas a primeira metade para construir o conjunto Q . Em vez disso, Sloan (1989) propôs que fosse considerado apenas um vértice de cada grau do último nível da estrutura de nível enraizada corrente para construir o conjunto Q . Segundo Sloan (1989), isso diminui a quantidade de estruturas de níveis a serem construídas no passo seguinte, sem prejudicar a qualidade dos vértices pseudoperiféricos encontrados, pois as excentricidades de vértices com o mesmo grau tendem a ser similares.

O algoritmo proposto por Sloan (1989) para encontrar vértices pseudoperiféricos é mostrado na figura 3. Na linha 2, o primeiro vértice s é escolhido aleatoriamente entre os vértices do grafo que possuem grau mínimo. O laço de repetição da linha 3 até a linha 39 é executado até que não seja encontrado um vértice cuja excentricidade seja maior que a excentricidade do vértice corrente. Na linha 4, é montada a estrutura de nível enraizada no vértice inicial. Na linha 5, os vértices do último nível da estrutura de nível enraizada, construída no passo anterior, são ordenados em ordem crescente de grau. Da linha 6 até a linha 13, é construído o conjunto Q de vértices, contendo um vértice de cada grau entre os vértices do último

nível da estrutura enraizada corrente. Da linha 14 até a linha 25, s e e são inicializados com o vértice com maior excentricidade e menor excentricidade, respectivamente, entre os vértices que ocupam as duas primeiras posições do conjunto Q , e armazena-se o valor da maior excentricidade e da menor largura de nível encontrados nas estruturas de níveis enraizadas nesses vértices. No laço de repetição da linha 27 até a linha 38, verifica-se se existe um vértice i no conjunto Q tal que a excentricidade de i seja maior que a excentricidade máxima encontrada até o momento, e que a largura de nível da estrutura de nível enraizada em i seja menor que a largura de nível mínima encontrada até o momento. Caso exista tal vértice, o mesmo é atribuído ao vértice corrente s (linhas 30 e 31). Caso contrário, verifica-se se a largura de nível da estrutura enraizada no vértice i é menor que a largura de nível mínima encontrada até o momento (linha 34) e, em caso afirmativo, atribui-se o vértice i ao vértice e e atribui-se a largura de banda de i à largura de banda mínima encontrada até o momento (linhas 35 e 36). Ao final, retornam-se os vértices s e e na linha 40.

A segunda modificação foi realizada na função de priorização utilizada para a renumeração dos vértices na segunda etapa. Na heurística de Sloan (1986), na função de priorização, não são permitidos valores negativos como retorno. Segundo Sloan (1989), valores negativos retornados pela função de priorização não afetam as renumerações produzidas e, portanto, não há a necessidade de serem evitados. A função de priorização, proposta por Sloan (1989), é

$$p(v) = w_1 \cdot d(v, e) - w_2 \cdot (\text{Grau}(G, v) + 1), \quad (1)$$

em que $d(v, e)$ é a distância entre os vértices v e e e $\text{Grau}(G, v)$ é o número

Algoritmo: Pseudocódigo do algoritmo de Sloan (1989) para encontrar um par de vértices pseudoperiféricos.

Entrada: grafo conexo $G = (V, E)$.
Saída: vértices pseudoperiféricos s e e .

```

1 início
2    $s \leftarrow \text{verticeComGrauMinimo}(G)$ ;
3   repita
4      $\mathcal{L}(s) \leftarrow \text{BuscaEmLargura}(G, s)$ ;
5      $q \leftarrow \text{ordemCrescenteGrau}(\mathcal{L}_{\ell(s)}(s))$ ;
6      $Q[1] \leftarrow q[1]$ ;
7      $k \leftarrow 1$ ;
8     para (  $j \leftarrow 2; j \leq |\mathcal{L}_{\ell(s)}(s)|; j \leftarrow j + 1$  ) faça
9       se (  $\text{Grau}(G, q[j]) \neq \text{Grau}(G, q[k])$  ) então
10         $Q[k+1] \leftarrow q[j]$ ;
11         $k \leftarrow k + 1$ ;
12      fim-se;
13    fim-para;
14     $\mathcal{L}(Q[1]) \leftarrow \text{BuscaEmLargura}(G, Q[1])$ ;
15     $\mathcal{L}(Q[2]) \leftarrow \text{BuscaEmLargura}(G, Q[2])$ ;
16    se (  $\ell(Q[1]) > \ell(Q[2])$  ) então
17       $s \leftarrow Q[1]$ ;
18       $e \leftarrow Q[2]$ ;
19    fim-se;
20    senão
21       $s \leftarrow Q[2]$ ;
22       $e \leftarrow Q[1]$ ;
23    fim-se;
24     $\ell_{max} \leftarrow \max(\ell(Q[1]), \ell(Q[2]))$ ;
25     $b_{min} \leftarrow \min(b(Q[1]), b(Q[2]))$ ;
26    atualizou  $\leftarrow false$ ;
27    para (  $i = 3; i \leq |Q|$  e não atualizou;  $i \leftarrow i + 1$  ) faça
28       $\mathcal{L}(i) \leftarrow \text{BuscaEmLargura}(Q[i])$ ;
29      se (  $\ell(i) > \ell_{max}$  e  $b(i) < b_{min}$  ) então
30         $s \leftarrow Q[i]$ ;
31         $\ell_{max} \leftarrow \ell(i)$ ;
32        atualizou  $\leftarrow true$ ;
33      fim-se;
34    senão se (  $b_i < b_{min}$  ) então
35       $e \leftarrow Q[i]$ ;
36       $b_{min} \leftarrow b_i$ ;
37    fim-se;
38  fim-para;
39  até que atualizou = false ;
40  retorna  $s, e$ .
41 fim.
```

Figura 3 Pseudocódigo do algoritmo de Sloan (1989) para encontrar um par de vértices pseudoperiféricos.

de vértices adjacentes ao vértice v no grafo G .

Neste trabalho, foi testada a heurística de Sloan (1989), e essa heurística é denominada apenas de Sloan nos próximos capítulos, quando não há ambiguidade. Foram utilizados os valores de $w_1 = 1$ e de $w_2 = 2$, conforme sugerido por Sloan (1986), Sloan (1989). O pseudocódigo da segunda etapa da heurística de Sloan (1989) é mostrado na figura 4. No laço de repetição da linha 2 até a linha 5, atribui-se o *status* inativo para cada um dos vértices do grafo e inicializa-se a prioridade desses vértices por meio da função definida na Equação 1. Nas linhas 6 e 7, insere-se o vértice s na fila de prioridades q e atribui-se o *status* de pré-ativo para esse vértice. No laço de repetição da linha 9 até a linha 37, os vértices pertencentes a q são renumerados em ordem decrescente de prioridade, conforme a seguir. Na linha 10, é escolhido o vértice w com maior prioridade entre os vértices pertencentes à fila de prioridades q , por meio da subrotina *VerticeComMaiorPrioridade*. Esse vértice w é removido da fila de prioridades q (linha 11) e, se o mesmo possuir o *status* pré-ativo (linha 12), para cada vértice adjacente ao vértice w : atualiza-se a prioridade do vértice, subtraindo-se uma unidade no grau de cada vértice na função definida pela Equação 1 (linha 14); e, se o vértice possuir o *status* igual a inativo, atribui-se o *status* pré-ativo ao vértice e insere-se o mesmo na fila de prioridades q (linhas 16 e 17). Posteriormente, o vértice w é renumerado (linha 21) e recebe o *status* de pós-ativo (linha 22). Para cada vértice j , adjacente ao vértice w , que possui o *status* igual a pré-ativo, atualiza-se prioridade do vértice j por meio da subtração de uma unidade no grau do vértice na função definida pela Equação 1, e altera-se o *status* do vértice j para ativo (linhas 26 e 27). Por fim, para cada vértice k , adjacente ao vértice j , atualiza-se a

prioridade do vértice k por meio da subtração de uma unidade no grau do mesmo na função definida pela Equação 1, caso o *status* do vértice j não seja pós-ativo (linha 29); e altera-se o *status* do vértice k para pré-ativo e insere-se o vértice k na fila de prioridades q , caso o *status* do vértice k seja inativo (linhas 31 e 32).

Durante a segunda etapa da heurística de Sloan (1989), os vértices recebem um *status*, conforme mostrado na figura 5, adaptada de Sloan (1989). Os vértices ainda não inseridos na fila de prioridades possuem o *status* inativo. Ao se inserir um determinado vértice na fila de prioridades de vértices candidatos a serem renumerados, atribui-se a esse vértice o *status* de pré-ativo. Esse *status* se mantém até que ocorra uma das seguintes situações: o vértice é escolhido para ser renumerado ou é adjacente ao vértice escolhido para ser renumerado. No primeiro caso, o *status* do vértice passa a ser pós-ativo e, no segundo caso, o *status* do vértice passa a ser ativo. Em ambos casos, os vértices adjacentes a esse vértice são verificados e, caso possuam *status* inativo, esses vértices são inseridos na fila de prioridades e recebem o *status* pré-ativo. No início da execução da heurística, o vértice s é inserido na fila de prioridades (recebe o *status* pré-ativo) e, em seguida, é escolhido para ser renumerado (recebe o *status* pós-ativo). Com isso, seus vértices adjacentes são verificados para que a renumeração prossiga.

2.5.1 Heurística NSloan

Kumfert e Pothen (1997) propuseram uma heurística a partir de três modificações na heurística de Sloan (1986). Uma dessas modificações foi proposta para melhorar a qualidade das renumerações obtidas, e as outras duas foram propostas para diminuir o custo computacional da heurística de

Algoritmo: Pseudocódigo da etapa de renumeração dos vértices da heurística de Sloan (1989).

Entrada: grafo conexo $G = (V, E)$ e vértices pseudoperiféricos s e e .
Saída: renumeração S com $|V|$ entradas $s(1), s(2), \dots, s(|V|)$.

```

1 início
2   para cada (  $v \in V(G)$  ) faça
3      $status_v \leftarrow inativo$ ;
4      $p_v \leftarrow w_1 \cdot d(v, e) - w_2 \cdot (Grau(G, v) + 1)$ ;
5   fim-para-cada;
6    $q \leftarrow \{s\}$ ;
7    $status_q \leftarrow preativo$ ;
8    $i \leftarrow 1$ ;
9   enquanto (  $q \neq \emptyset$  ) faça
10     $w \leftarrow VerticeComMaiorPrioridade(q)$ ;
11     $q \leftarrow q - \{w\}$ ;
12    se (  $status_w = preativo$  ) então
13      para cada (  $j \in Adj(w)$  ) faça
14         $p_j \leftarrow p_j + w_2$ ;
15        se (  $status_j = inativo$  ) então
16           $status_j \leftarrow preativo$ ;
17           $q \leftarrow q \cup \{j\}$ ;
18        fim-se;
19      fim-para-cada;
20    fim-se;
21     $s(i) \leftarrow w$ ;
22     $status_w \leftarrow positivo$ ;
23     $i \leftarrow i + 1$ ;
24    para cada (  $j \in Adj(w)$  ) faça
25      se (  $status_j = preativo$  ) então
26         $status_j \leftarrow ativo$ ;
27         $p_j \leftarrow p_j + w_2$ ;
28      para cada (  $k \in Adj(j)$  ) faça
29        se (  $status_k \neq positivo$  ) então  $p_k \leftarrow p_k + w_2$ ;
30        se (  $status_k = inativo$  ) então
31           $status_k \leftarrow preativo$ ;
32           $q \leftarrow q \cup \{k\}$ ;
33        fim-se;
34      fim-para-cada;
35    fim-se;
36  fim-para-cada;
37 fim-enquanto;
38 retorna  $S$ .
39 fim.
```

Figura 4 Pseudocódigo da etapa de renumeração dos vértices da heurística de Sloan (1989).

Sloan (1986). Essas três modificações são descritas a seguir.

A primeira modificação foi em relação ao algoritmo para encontrar

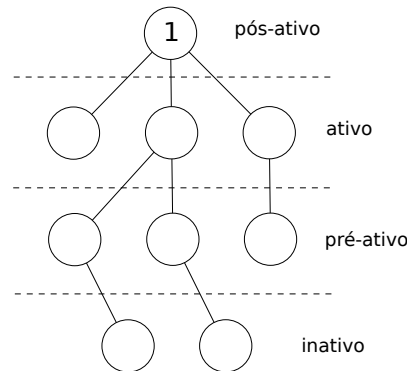


Figura 5 Exemplo da atribuição de *status* durante a segunda fase da heurística de Sloan (1989).

Fonte: Sloan (1989) (adaptação)

vértices pseudoperiféricos, correspondente à primeira etapa da heurística de Sloan (1986). Em virtude do número de vértices do último nível da estrutura de nível enraizada corrente, considerado pelo algoritmo de Sloan (1986), conforme comentado na seção 2.5, Kumfert e Pothen (1997) propuseram a utilização do algoritmo de Duff Reid e Scott (1989). Na verdade, o algoritmo de Duff Reid e Scott (1989) é idêntico ao algoritmo proposto por Sloan (1989) para encontrar um par de vértices pseudoperiféricos apresentado na seção 2.5. Ambos foram propostos no mesmo ano e, como não há citação entre os artigos, é provável que os autores não estavam cientes dos trabalhos equivalentes.

A segunda modificação proposta por Kumfert e Pothen (1997) na heurística de Sloan (1986) é referente à função de priorização proposta por Sloan (1986). Na Equação 1, por exemplo, correspondente à função de priorização proposta por Sloan (1989), é possível notar que no cálculo da prioridade de cada vértice há dois termos principais que são ponderados: o grau do vértice e a distância desse vértice ao vértice e . A escala desses dois

termos é diferente. Logo, não é possível a utilização de pesos normalizados. A normalização desses dois termos permite que a normalização seja feita de forma equilibrada. Dessa maneira, Kumfert e Pothen (1997) propuseram a seguinte normalização para a função de priorização utilizada na heurística de Sloan (1986):

$$p_v = w_1 \cdot d(v, e) - w_2 \cdot \lfloor \text{dist}(s, e) / \Delta \rfloor \cdot (\text{Grau}(G, v) + 1), \quad (2)$$

em que Δ é o grau do vértice com maior grau entre os vértices do grafo. Com isso, tem-se que o valor de $\lfloor \text{dist}(s, e) / \Delta \rfloor \cdot (\text{Grau}(G, v) + 1)$ varia de 0 a $\text{dist}(s, e)$, que é o mesmo domínio no intervalo de valores $\text{dist}(v, e)$, uma vez que os vértices s e e são determinados de maneira que um esteja no último nível da estrutura de nível enraizada do outro. A heurística proposta por Kumfert e Pothen (1997) é denominada NSloan (*normalized Sloan*), e o pseudocódigo dessa heurística é mostrado na figura 6.

Neste trabalho, foram utilizados os valores $w_1 = 2$ e $w_2 = 1$. Kumfert e Pothen (1997) sugeriram a utilização de um dos seguintes pares de valores: $w_1 = 2$ e $w_2 = 1$, ou $w_1 = 1$ e $w_2 = 8$. A partir de uma investigação exploratória, nas instâncias utilizadas neste trabalho, verificou-se que os valores $w_1 = 2$ e $w_2 = 1$ obtiveram melhores resultados.

A última modificação presente na heurística NSloan, em relação à heurística de Sloan (1986), consiste na estrutura de dados utilizada para implementar a fila de prioridades q . Em Sloan (1986) e em Sloan (1989), respectivamente, comenta-se que há a possibilidade de se utilizar uma estrutura de árvore binária ou de *heap*, cujas operações de inserção, remoção e alteração de prioridades são de complexidade $O(n)$ e cuja operação de busca pelo elemento com maior prioridade possui complexidade $O(1)$.

Algoritmo: Pseudocódigo da etapa de renumeração dos vértices da heurística NSloan.

Entrada: grafo conexo $G = (V, E)$ e vértices pseudoperiféricos s e e .
Saída: renumeração S com $|V|$ entradas $s(1), s(2), \dots, s(|V|)$.

```

1 início
2   para cada (  $v \in V(G)$  ) faça
3      $status_v \leftarrow inativo$ ;
4      $p_v \leftarrow w_1 \cdot d(v, e) - w_2 \cdot \lfloor dist(s, e) / \Delta \rfloor \cdot (Grau(G, v) + 1)$ ;
5   fim-para-cada;
6    $q \leftarrow \{s\}$ ;
7    $status_s \leftarrow preativo$ ;
8    $i \leftarrow 1$ ;
9   enquanto (  $q \neq \emptyset$  ) faça
10     $w \leftarrow VerticeComMaiorPrioridade(q)$ ;
11     $q \leftarrow q - \{w\}$ ;
12    se (  $status_w = preativo$  ) então
13      para cada (  $j \in Adj(w)$  ) faça
14         $p_j \leftarrow p_j + w_2 \cdot \lfloor dist(s, e) / \Delta \rfloor \cdot (Grau(G, j) + 1)$ ;
15        se (  $status_j = inativo$  ) então
16           $status_j \leftarrow preativo$ ;
17           $q \leftarrow q \cup \{j\}$ ;
18        fim-se;
19      fim-para-cada;
20    fim-se;
21     $s(i) \leftarrow w$ ;
22     $status_w \leftarrow positivo$ ;
23     $i \leftarrow i + 1$ ;
24    para cada (  $j \in Adj(w)$  ) faça
25      se (  $status_j = preativo$  ) então
26         $status_j \leftarrow ativo$ ;
27         $p_j \leftarrow p_j + w_2 \cdot \lfloor dist(s, e) / \Delta \rfloor \cdot (Grau(G, j) + 1)$ ;
28        para cada (  $k \in Adj(j)$  ) faça
29          se (  $status_k \neq positivo$  ) então
30             $p_k \leftarrow p_k + w_2 \cdot \lfloor dist(s, e) / \Delta \rfloor \cdot (Grau(G, k) + 1)$ ;
31            se (  $status_k = inativo$  ) então
32               $status_k \leftarrow preativo$ ;
33               $q \leftarrow q \cup \{k\}$ ;
34            fim-se;
35          fim-para-cada;
36        fim-se;
37      fim-para-cada;
38    fim-enquanto;
39 retorna  $S$ .
40 fim.
```

Figura 6 Pseudocódigo da etapa de renumeração dos vértices da heurística NSloan.

Porém, nesses trabalhos, o tamanho das instâncias eram de até algumas centenas de vértices. Dessa maneira, o uso de uma estrutura de lista não

ordenada, cujas operações de inserção, remoção e alteração de prioridades são triviais e a operação de busca possui complexidade $O(n)$, foi a estrutura mais eficiente na implementação dessas heurísticas. Com a realização de simulações com instâncias de tamanhos maiores, na ordem de até uma centena de milhares de vértices, Kumfert e Pothen (1997) constataram que a utilização de uma estrutura de *heap* binária é mais eficiente, e propuseram sua utilização na implementação da fila de prioridades na heurística NSloan.

2.5.2 Heurística Sloan-MGPS

Reid e Scott (1999) propuseram uma heurística para redução de *profile* a partir de modificações na heurística de Sloan (1986). A primeira delas é em relação ao método para encontrar um par de vértices pseudoperiféricos, que corresponde à primeira etapa da heurística de Sloan (1986). Especificamente, Reid e Scott (1999) propuseram a utilização de duas estratégias diferentes na etapa de escolha de vértices do último nível da estrutura de nível enraizada corrente. Para limitar a quantidade de estruturas de níveis enraizadas a serem construídas no passo seguinte, Reid e Scott (1999) propuseram escolher cinco vértices do último nível, ou todos os vértices caso a quantidade de vértices no último nível seja menor que cinco, em ordem crescente de grau, e descartar aqueles vértices que são adjacentes a vértices já escolhidos. Além disso, após a escolha dos vértices s e e iniciais, Reid e Scott (1999) propuseram que o vértice de maior excentricidade, entre os dois vértices, seja designado como o vértice s inicial, e o vértice de menor excentricidade seja designado como o vértice e . Esse algoritmo foi denominado MGPS (*modified* Gibbs-Poole-Stockmeyer) pelos autores, devido à semelhança com a estrutura de repetição do algoritmo referente

à primeira etapa da heurística GPS (GIBBS; POOLE; STOCKMEYER, 1976).

O pseudocódigo do algoritmo MGPS (REID e SCOTT, 1999) é mostrado na figura 7. A execução do algoritmo é similar à execução do algoritmo de Sloan (1989) mostrado na figura 3. A primeira diferença está no laço de repetição da linha 8 até a linha 11, em que a quantidade de vértices escolhidos é limitada (linha 8), e é verificado se o vértice corrente não é adjacente a um vértice já escolhido (linha 9). Vale informar que a função $Adj(Q)$ retorna o conjunto de vértices adjacentes a vértices pertencentes a Q . A outra diferença está no final do algoritmo, da linha 38 até a linha 42, em que há uma troca do vértice s com o vértice e , caso a excentricidade do vértice e seja maior que a excentricidade do vértice s .

Outra modificação proposta por Reid e Scott (1999), para a heurística de Sloan (1986), é em relação à escolha da estrutura de dados para implementar a fila de prioridades que armazena os vértices elegíveis a cada iteração da segunda etapa da heurística de Sloan (1986). Segundo Sloan (1986) e Sloan (1989), a estrutura de dados de lista não ordenada é eficiente para instâncias pequenas (até algumas centenas de vértices). Porém, para instâncias maiores, é desejável a utilização de estruturas robustas, como árvore binária ou *heap* binária, conforme observado por Sloan (1986) e Sloan (1989), e proposto por Kurfert e Pothen (1997). Com isso, Reid e Scott (1999) propuseram a utilização de um limiar para definir a escolha da estrutura de dados a ser utilizada para implementar a fila de prioridades. Caso o número de vértices na fila de prioridades q seja inferior ao limiar estabelecido, utiliza-se uma estrutura de lista não ordenada. Caso contrário, utiliza-se uma estrutura de *heap* binária. O valor do limiar estabelecido

Algoritmo: Pseudocódigo do algoritmo MGPS de Reid e Scott (1999) para encontrar um par de vértices pseudoperiféricos.

Entrada: grafo conexo $G = (V, E)$.
Saída: Vértices pseudoperiféricos s e e .

```

1 início
2    $s \leftarrow \text{verticeComGrauMinimo}(G)$ ;
3   repita
4      $\mathcal{L}(s) \leftarrow \text{BuscaEmLargura}(G, s)$ ;
5      $q \leftarrow \text{ordemCrescenteGrau}(\mathcal{L}_{\ell(s)}(s))$ ;
6      $Q \leftarrow \emptyset$ ;  $j \leftarrow 1$ ;
7     enquanto (  $j \leq |\mathcal{L}_{\ell(s)}(s)|$  e  $|Q| < 5$  ) faça
8       se (  $q[j] \notin \text{Adj}(Q)$  ) então  $Q \leftarrow Q \cup \{q[j]\}$ ;
9        $j \leftarrow j + 1$ ;
10    fim-enquanto;
11     $\mathcal{L}(Q[1]) \leftarrow \text{BuscaEmLargura}(G, Q[1])$ ;
12     $\mathcal{L}(Q[2]) \leftarrow \text{BuscaEmLargura}(G, Q[2])$ ;
13    se (  $\ell(Q[1]) > \ell(Q[2])$  ) então
14       $s \leftarrow Q[1]$ ;  $e \leftarrow Q[2]$ ;
15    fim-se;
16    senão
17       $s \leftarrow Q[2]$ ;  $e \leftarrow Q[1]$ ;
18    fim-se;
19     $\ell_{\max} \leftarrow \max(\ell(Q[1]), \ell(Q[2]))$ ;
20     $b_{\min} \leftarrow \min(b(Q[1]), b(Q[2]))$ ;
21    atualizou  $\leftarrow \text{false}$ ;
22    para (  $i = 3; i \leq |Q|$  e não atualizou;  $i \leftarrow i + 1$  ) faça
23       $\mathcal{L}(i) \leftarrow \text{BuscaEmLargura}(Q[i])$ ;
24      se (  $\ell(i) > \ell_{\max}$  e  $b(i) < b_{\min}$  ) então
25         $s \leftarrow Q[i]$ ;
26         $\ell_{\max} \leftarrow \ell(i)$ ;
27        atualizou  $\leftarrow \text{true}$ ;
28      fim-se;
29      senão se (  $b_i < b_{\min}$  ) então
30         $e \leftarrow Q[i]$ ;  $b_{\min} \leftarrow b_i$ ;
31      fim-se;
32    fim-para;
33    até que (atualizou = false);
34    se (  $\ell(e) > \ell_s$  ) então
35       $aux \leftarrow s$ ;
36       $s \leftarrow e$ ;
37       $e \leftarrow aux$ ;
38    fim-se;
39    retorna  $s, e$ ;
40 fim.
```

Figura 7 Pseudocódigo do algoritmo MGPS de Reid e Scott (1999) para encontrar um par de vértices pseudoperiféricos.

por Reid e Scott (1999) foi 100. Neste trabalho, como as instâncias eram majoritariamente maiores que o limiar estabelecido por Reid e Scott (1999),

utilizou-se apenas a estrutura *heap* binária. Além disso, foram utilizados os valores $w_1 = 1$ e $w_2 = 2$ na função de priorização, conforme sugerido por Reid e Scott (1999).

2.6 Heurística de Snay (1976)

A heurística de Snay (1976) para redução de *profile* possui duas etapas: a determinação de vértices pseudoperiféricos iniciais e a renumeração dos vértices do grafo. Na primeira etapa, são selecionados 10 vértices, ou todos os vértices caso a quantidade de vértices do grafo seja menor que 10, como candidatos a vértices iniciais para a realização da renumeração dos vértices na segunda etapa. Na segunda etapa, realiza-se uma renumeração dos vértices do grafo a partir de cada vértice selecionado na primeira etapa. Por fim, escolhe-se a renumeração com o menor valor de *profile*. A heurística de Snay (1976) é descrita em detalhes no capítulo 3.

2.7 Heurística MPG

A heurística MPG, proposta por Medeiros, Pimenta e Goldenberg (1993) para redução de *profile*, também é dividida em duas etapas. Na primeira etapa, dois vértices pseudoperiféricos são determinados por meio do algoritmo proposto por Sloan (1989) e mostrado na figura 3.

Na segunda etapa, ocorre a renumeração dos vértices do grafo. Para isso, são mantidos duas filas de prioridades, denominadas q e t . Na fila de prioridades t , são mantidos os vértices que são elegíveis para a serem renumerados e, na fila de prioridades q , são mantidos os vértices que já estão em t ou que são elegíveis para serem inseridos em t . A atualização das filas de prioridades q e t é feita com base no grau corrente de cada vértice.

O grau corrente de um vértice v é o número de vértices w adjacentes a v , tal que w não seja renumerado e $w \notin q$. Em cada iteração, todos os vértices pertencentes a t , cujo grau corrente seja 0, são renumerados em ordem crescente de grau. Além disso, os vértices pertencentes a t , cujo grau corrente seja maior que 1, são removidos de t . Por fim, caso a fila de prioridades t esteja vazia, os vértices pertencentes a q , cuja prioridade seja igual à prioridade máxima entre os vértices de q , ou menor em uma unidade, são inseridos em t . A prioridade de um vértice é dada, inicialmente, pela mesma fórmula proposta por Sloan (1989), representada pela Equação 1, com os valores $w_1 = 1$ e $w_2 = 2$. Durante a execução da heurística, o termo $(Grau(G, v) + 1)$ da Equação 1 é substituído pelo grau corrente do vértice.

Na primeira iteração, é inserido o vértice inicial s na fila de prioridades q . Nas demais iterações, é inserido um novo vértice na fila de prioridades q que seja adjacente a um vértice pertencente a t e que ainda não tenha sido renumerado. Isso faz com que o vértice escolhido seja adjacente a um vértice que já esteja para ser renumerado e garante que as diferenças entre as renumerações dos vértices adjacentes sejam pequenas. A escolha do vértice a ser inserido em q é feita de maneira a maximizar a função π , definida como se segue:

$$\pi(v) = 2 \cdot p(v) + 2 \cdot p_{max}(Adj(v) \cap t) + 3 \cdot a_v, \quad (3)$$

em que $p(v)$ é a prioridade do vértice v dada pela Equação 1, $p_{max}(A)$ é a maior prioridade entre os vértices do conjunto A e a_v é o número de vértices w adjacentes a v , tal que $w \in q$.

O pseudocódigo da segunda etapa da heurística MPG é mostrado na figura 8. Inicialmente, na linha 2, são inicializadas as filas de prioridades

q e t como conjuntos vazios. Da linha 3 até a linha 7, são inicializados os atributos de todos os vértices do grafo: ao grau corrente, é atribuído o valor do grau do vértice no grafo (linha 4), à prioridade, é atribuído o valor calculado pela Equação 1 (linha 5) e, ao atributo a_v , correspondente ao número de adjacências a vértices pertencentes a q , é atribuído o valor 0 (linha 6). Posteriormente, em cada iteração do laço de repetição da linha 9 até a linha 43, um conjunto não vazio de vértices é renumerado. Essa é a principal diferença da heurística MPG em relação às heurísticas de Snay (1976), de Sloan (1989), NSloan (KUMFERT e POTHEN, 1997) e Sloan-MGPS (REID e SCOTT, 1999), apresentadas nas Subseções de 2.5 a 2.6, nas quais apenas um vértice é renumerado em cada iteração. No início da iteração, verifica-se se a fila de prioridades q encontra-se vazia (linha 10). Em caso afirmativo, insere-se o vértice inicial s em q . Caso contrário, escolhe-se entre todos os vértices não renumerados e não pertencentes a q , e que sejam adjacentes a algum vértice pertencente a t , o vértice cujos atributos maximizam a função π , definida pela Equação 3 (linha 14) e insere-se esse vértice em q (linha 16). Em seguida, os atributos de todos os vértices são atualizados: subtrai-se do grau corrente de cada vértice a quantidade de vértices adjacentes ao mesmo que já se encontram renumerados ou que pertençam a q (linha 19), recalcula-se a prioridade de cada vértice de acordo com o novo valor do grau corrente (linha 20) e recalcula-se o número de adjacências de cada vértice a vértices pertencentes a q (linha 21). Por fim, ordena-se os vértices da lista de prioridades t em ordem crescente de grau na linha 26 e, no laço de repetição da linha 27 até a linha 37, percorre-se a lista de prioridades t , renumera-se (linha 29) e remove-se das listas de prioridades q e t os vértices visitados cujo grau corrente seja igual a 0, e remove-se de t os

vértices visitados cujo grau corrente seja maior que 1 (linha 34). Caso a fila de prioridades t torne-se vazia, percorre-se a fila de prioridades q em ordem crescente de grau (linhas 38 e 39), e insere-se em t os vértices visitados cuja prioridade seja igual à prioridade máxima entre todos os vértices de q , ou menor em uma unidade (linha 40).

2.8 Heurística Hu-Scott

Hu e Scott (2001) propuseram uma heurística multinível para redução de *profile*, denominada aqui de heurística Hu-Scott. Em uma abordagem multinível, grafos contraídos (*coarsed graphs*) são gerados de maneira iterativa até que uma condição estabelecida, geralmente relacionada ao tamanho máximo do grafo mais contraído (*coarsest graph*), seja satisfeita. Um grafo contraído é um grafo que encapsula, em menos vértices e arestas, a informação contida no grafo refinado (*fine graph*), que contém um número maior de vértices e arestas.

A heurística Hu-Scott possui as seguintes etapas, conforme previsto em uma estratégia multinível: contração, resolução do grafo mais contraído, prolongamento e refinamento. No contexto da heurística Hu-Scott, a resolução do grafo mais contraído consiste na renumeração dos vértices desse grafo com o objetivo de reduzir o *profile*. Em outros contextos, essa solução pode ser o desenho (HU, 2005) ou o particionamento (HENDRICKSON; LELAND, 1995) do grafo, ou ainda a solução do problema do caixeiro-viajante (WALSHAW, 2002), por exemplo.

Segundo Hu e Scott (2001), o algoritmo de contração utilizado na primeira etapa da heurística Hu-Scott é baseado no algoritmo de Ruge e Stüben (1986), que particiona o conjunto de vértices $V(G)$ do grafo G em dois

Algoritmo: Segunda fase da heurística MPG.

Entrada: grafo conexo $G = (V, E)$ e vértices pseudoperiféricos s e e .

Saída: renumeração S com $|V|$ entradas $s(1), s(2), \dots, s(|V|)$.

```

1 início
2    $q \leftarrow \emptyset; t \leftarrow \emptyset;$ 
3   para cada (  $v \in V(G)$  ) faça
4      $d_v \leftarrow \text{Grau}(G, v);$ 
5      $p_v \leftarrow d(v, e) - 2 \cdot d_v;$ 
6      $a_v \leftarrow 0;$ 
7   fim-para-cada;
8    $i \leftarrow 1;$ 
9   enquanto (  $i < |V|$  ) faça
10    se (  $q = \emptyset$  ) então
11       $q \leftarrow q \cup \{s\};$ 
12       $n \leftarrow s;$ 
13    fim-se;
14    senão
15       $n \leftarrow \text{EscolheVerticeComMaior}\pi(\text{Adj}(t) - (q \cup \{s(1) \dots s(i-1)\}));$ 
16       $q \leftarrow q \cup \{n\};$ 
17    fim-se;
18    para cada (  $w \in V(G)$  ) faça
19       $d_w \leftarrow |\text{Adj}(w) - (q \cup \{s(1) \dots s(i-1)\})|;$ 
20       $p_w \leftarrow d(w, e) - 2 \cdot d_w;$ 
21       $a_w \leftarrow |\text{Adj}(w) \cap q|;$ 
22    fim-para-cada;
23    para cada (  $u \in \text{Adj}(n) \cap q$  ) faça
24      se (  $d_u = 1$  ) então  $t \leftarrow t \cup \{u\};$ 
25    fim-para-cada;
26     $t \leftarrow \text{ordemCrescenteGrau}(t);$ 
27    para cada (  $x \in t$  ) faça
28      se (  $d_x = 0$  ) então
29         $s(i) \leftarrow x;$ 
30         $q \leftarrow q - \{x\}; t \leftarrow t - \{x\};$ 
31         $i \leftarrow i + 1;$ 
32      fim-se;
33      senão
34        se (  $d_x > 1$  ) então  $t \leftarrow t - \{x\};$ 
35      fim-se;
36    fim-para-cada;
37    se (  $t = \emptyset$  ) então
38       $q \leftarrow \text{ordemCrescenteGrau}(q);$ 
39      para cada (  $y \in q$  ) faça
40        se (  $p_y \geq p_{\max}(q) - 1$  ) então  $t \leftarrow t \cup \{y\};$ 
41      fim-para-cada;
42    fim-se;
43  fim-enquanto;
44  retorna  $S$ .
45 fim.
```

Figura 8 Pseudocódigo da segunda fase da heurística MPG.

conjuntos V_C e V_F , tal que $V_F = V(G) - V_C$. De maneira geral, no algoritmo de contração de Hu e Scott (2001), escolhe-se um conjunto de vértices V_C do grafo refinado $V(G)$ para compor o grafo contraído, de modo a garantir que um vértice pertencente ao conjunto V_F seja adjacente a, pelo menos, um vértice do conjunto V_C . Para isso, escolhe-se entre os vértices do conjunto V_U , em que $V_U = V(G) - (V_C \cup V_F)$, aquele com o maior número de adjacências a vértices do conjunto V_F . Conforme afirmado por Hu e Scott (2001), essa abordagem permite uma distribuição melhor dos vértices de V_C do que a abordagem gulosa de Ruge e Stüben (1986).

O pseudocódigo da etapa de contração da heurística Hu-Scott é mostrado na figura 9. Nas linhas 2 e 3 são inicializados os conjuntos de vértices V_C e V_U . Na linha 4, para cada vértice do grafo, o valor do grau do vértice é estabelecido como o ganho desse vértice. No laço de repetição da linha 5 até a linha 19, são escolhidos vértices de V_U para serem inseridos em V_C . Para isso, da linha 6 até a linha 12, é escolhido o vértice com maior ganho. Esse vértice é, então, removido no conjunto V_U (linha 13) e inserido no conjunto V_C (linha 14). Posteriormente, na linha 16, cada vértice adjacente ao vértice de maior ganho é removido do conjunto V_U para evitar um aumento desnecessário do tamanho do conjunto V_C com a inserção posterior de tais vértices. Na linha 17, incrementa-se em uma unidade o ganho dos vértices adjacentes a esses vértices removidos de V_U . Esse incremento favorece a escolha de um desses vértices nas próximas iterações, para que seja garantida existência de adjacências entre vértices de V_C e V_F .

Após a escolha do conjunto de vértices do grafo contraído V_C , por meio do algoritmo de contração, é construído um mapeamento entre os vértices desse conjunto e os vértices do grafo refinado $V(G)$. Esse

Algoritmo: Etapa de contração da heurística HuScott.

Entrada: grafo conexo $G = (V, E)$.
Saída: conjunto V_C de vértices de $V(G)$.

```

1 início
2    $V_C \leftarrow \emptyset$ ;
3    $V_U \leftarrow V(G)$ ;
4   para cada (  $i \in V(G)$  ) faça  $ganho_i \leftarrow Grau(i, G)$ ;
5   enquanto (  $V_U \neq \emptyset$  ) faça
6      $ganho_{max} \leftarrow 0$ ;
7     para cada (  $j \in V_U$  ) faça
8       se (  $ganho_j > ganho_{max}$  ) então
9          $ganho_{max} \leftarrow ganho_j$ ;
10         $v_{max} \leftarrow j$ ;
11      fim-se;
12    fim-para-cada;
13     $V_U \leftarrow V_U - \{v_{max}\}$ ;
14     $V_C \leftarrow V_C \cup \{v_{max}\}$ ;
15    para cada (  $k \in Adj(v_{max}) \cap V_U$  ) faça
16       $V_U \leftarrow V_U - \{k\}$ ;
17      para cada (  $l \in Adj(k) \cap V_U$  ) faça  $ganho_l \leftarrow ganho_l + 1$ ;
18    fim-para-cada;
19  fim-enquanto;
20  retorna  $V_C$ .
21 fim.
```

Figura 9 Pseudocódigo do algoritmo de contração da heurística Hu-Scott (HU; SCOTT, 2001).

mapeamento é representado por uma matriz, denominada matriz de prolongamento. Essa matriz é utilizada para definir as adjacências entre os vértices de V_C no grafo contraído e para realizar a etapa de prolongamento após a renumeração do grafo mais contraído. A matriz de prolongamento

$P_{|V(G)| \times |V_C|} = [p_{ij}]$ é definida como

$$p_{ij} = \begin{cases} 1 & \text{se } V(G)[i] = V_C[j] \\ \frac{1}{mdeg(i)} & \text{se } V(G)[i] \in Adj(V_C[j]) \text{ e } V(G)[i] \neq V_C[j] \\ 0 & \text{caso contrário,} \end{cases} \quad (4)$$

em que $mdeg(i) = |\{v \in V_C | v \in Adj(V(G)[i])\}|$ e $Adj(v)$ retorna o conjunto de vértices de $V(G)$ que são adjacentes a v no grafo refinado.

A partir dos vértices escolhidos para o conjunto V_C e matriz de prolongamento, são determinadas as adjacências entre os vértices de V_C para construção do grafo contraído. Para isso, constrói-se a matriz de adjacências A_C do grafo contraído por meio do produto de *Garlekin* da matriz de adjacências A_G do grafo refinado G e a matriz de prolongamento P : $A_C = P^T A_G P$. Com isso, é possível construir o grafo G_C , que contém os vértices de V_C e as arestas representadas na matriz de adjacências A_C .

Essa etapa de contração, que compreende a construção do conjunto V_C , da matriz de prolongamento P e do grafo contraído G_C , é repetida recursivamente até que seja atingido o tamanho máximo definido para o grafo mais contraído, ou até que a contração não diminua o número de vértices do grafo refinado. Neste trabalho, foi estabelecido o valor 100 com o tamanho máximo para o grafo mais contraído, conforme foi proposto por Hu e Scott (2001).

Após o término da etapa de contração, o último grafo construído, que será o mais contraído, é reenumerado com o objetivo de se reduzir o *profile*. Em Hu e Scott (2001), foram utilizadas a heurística de Sloan (1989) e a heurística Hybrid (KUMFERT e POTHEN, 1997) para esse fim. Como os

melhores resultados foram obtidos pela heurística de Sloan (1989), utilizou-se, neste trabalho, essa heurística na etapa de renumeração do grafo mais contraído. Em relação aos pesos utilizados na heurística de Sloan (1989) para a renumeração do grafo mais contraído, foram utilizados os valores $w_1 = 1$ e $w_2 = 2$. (HU; SCOTT, 2001) propuseram a utilização de um dos seguintes pares de valores: $w_1 = 1$ e $w_2 = 2$, ou $w_1 = 16$ e $w_2 = 1$. Neste trabalho, foi realizada uma investigação exploratória, e os melhores resultados foram obtidos com os valores $w_1 = 1$ e $w_2 = 2$.

A partir da renumeração do grafo mais contraído, são realizadas as etapas de prolongamento e refinamento dessa renumeração obtida. O prolongamento consiste na extensão da renumeração, do grafo contraído para o grafo imediatamente mais refinado na ordem das chamadas recursivas. O objetivo dessa extensão da renumeração é construir um vetor de valores globais de prioridades p_F^0 para os vértices do grafo imediatamente mais refinado. Essa extensão é realizada por meio do produto da matriz de prolongamento P com o vetor permutação que representa a renumeração: $p_F^0 = Pp_C$, em que p_C é o vetor permutação que representa a renumeração do grafo mais contraído. Na figura 10, são mostradas as etapas de contração, renumeração e prolongamento. Com isso, de acordo com a estrutura da matriz de prolongamento P , mostrada na Equação 4, o valor de prioridade global de um vértice do grafo mais refinado que esteja no grafo mais contraído será igual à numeração do mesmo no grafo mais contraído. Por outro lado, o valor de prioridade global de um vértice do grafo mais refinado, que não esteja no grafo mais contraído, será igual à média dos valores de prioridades globais dos vértices adjacentes a esse vértice que pertençam ao grafo mais contraído.

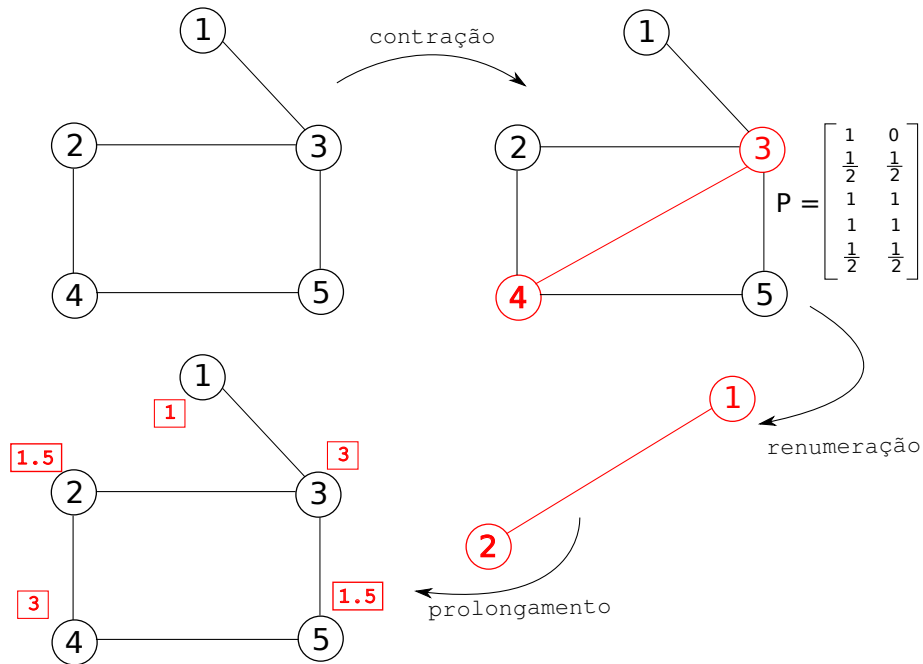


Figura 10 Etapas de contração, renomeação e prolongamento da heurística Hu-Scott. Na etapa de contração, são escolhidos os vértices para o grafo mais contraído (em vermelho) e montada a matriz de prolongamento P . Na etapa de renomeação, os vértices do grafo mais contraído são renomeados pela heurística de Sloan (1989). Na etapa de prolongamento, valores globais de prioridades para o grafo mais refinado (marcados com um retângulo vermelho) são obtidos por meio do produto da matriz de prolongamento e o vetor permutação que representa a renomeação do grafo mais contraído.

O refinamento consiste em adaptar e melhorar a renomeação estendida com o objetivo de obter uma redução do *profile* do grafo imediatamente mais refinado. Para isso, é utilizada uma variação da heurística de Sloan (1989) que utiliza um vetor numérico para definir os valores globais no cálculo das prioridades dos vértices. Com isso, a fórmula definida na Equação 1 passa a ser $p(v) = w_1 \cdot p_F^0[v] - w_2 \cdot (\text{Grau}(G, v) + 1)$, em que $p_F^0[v]$ é o valor definido em p_F^0 na posição que o vértice v ocupa no

conjunto $V(G)$, e apenas a linha 4 do pseudocódigo apresentado na figura 4 é modificada para contemplar o novo cálculo de prioridade. Essa renumeração adaptada e melhorada é utilizada, sucessivamente, em relação às chamadas recursivas para realizar o prolongamento e o refinamento posteriores.

O pseudocódigo da heurística Hu-Scott é mostrado na figura 11. Da linha 2 até a linha 6, ocorre o passo base da função recursiva: caso o tamanho máximo do grafo mais contraído seja atingido (linha 2), o grafo contraído é renumerado pela heurística de Sloan (1989) apresentada na seção 2.5 (linha 3), cujo pseudocódigo é mostrado na figura 4. Retorna-se, na linha 5, o vetor permutação relacionado à renumeração obtida. Na linha 7, ocorre o início da etapa de contração, por meio do algoritmo de contração mostrado na figura 9. Da linha 8 até a linha 11 ocorre a finalização da etapa de contração: construção da matriz de prolongamento por meio da definição mostrada na Equação 4 (linha 8), determinação das adjacências do grafo mais contraído por meio do produto de *Garlekin* (linha 10) e construção do grafo mais contraído (linha 11). A função *MatrizAdjacencias(G)*, utilizada na linha 9, retorna a matriz de adjacências do grafo G . Na linha 12, ocorre a chamada recursiva da heurística Hu-Scott, a partir do grafo contraído construído na etapa de contração. Após o retorno da renumeração da chamada recursiva, ocorre as etapas de prolongamento e refinamento dessa renumeração nas linhas 13 e 14, respectivamente. Por fim, a renumeração resultante é retornada na linha 15.

2.9 Métodos de trocas adjacentes

Entende-se por métodos de trocas qualquer estratégia que busca diminuir o *profile* a partir de trocas simétricas de linhas e de colunas

Algoritmo: HeurísticaHuScott.

Entrada: grafo conexo $G = (V, E)$.
Saída: vetor permutação referente à renumeração de G .

```

1 início
2   se (  $|V| < menorTamanhoDesrefinado$  ) então
3     // Etapa de renumeração do grafo mais contraído
4     Sloan( $G$ );
5      $p_C \leftarrow permutacaoRenumeracao(G)$ ;
6     retorna  $p_C$ .
7   fim-se;
8   // Etapa de contração
9    $V_C \leftarrow ContracaoHuScott(G)$ ;
10   $P \leftarrow MatrizProlongamento(G, V_C)$ ;
11   $A_F \leftarrow MatrizAdjacencias(G)$ ;
12   $A_C \leftarrow P^T A_F P$ ;
13   $G_C \leftarrow ConstroiGrafo(V_C, A_C)$ ;
14  // Chamada recursiva
15   $p_C \leftarrow HeuristicaHuScott(G_C)$ ;
16  // Etapa de prolongamento
17   $p_F^0 \leftarrow P_C$ ;
18  // Etapa de refinamento
19   $p_F \leftarrow RefinamentoSloan(G, p_F)$ ;
20  retorna  $p_F$ .
21 fim.
  
```

Figura 11 Pseudocódigo da heurística multinível de Hu e Scott (2001) para redução de *profile*.

de uma matriz simétrica (HAGER, 2002). No trabalho de Hager (2002), foram apresentados métodos que buscam reduzir o *profile* a partir de trocas especializadas em vez de fazer trocas exaustivas entre todos os pares possíveis. Nesses métodos, são aplicadas trocas sequenciais em um conjunto de linhas adjacentes, de maneira ascendente ou descendente, de modo que o *profile* seja garantidamente menor. Esses métodos podem ser aplicados tanto para redução de *profile* da matriz original quanto para reduzir o *profile* da matriz resultante após as permutações realizadas por heurísticas para reduções de largura de banda e de *profile*.

Os métodos propostos por Hager (2002) identificam, a partir de uma matriz A , um conjunto de linhas adjacentes cujas trocas sequenciais resultam em uma permutação da matriz original que possua um valor menor de *profile*. Após efetuar as trocas sequenciais das linhas do conjunto encontrado o método faz as trocas das respectivas colunas para manter a simetria da matriz. Essas trocas sequenciais, feitas pelo método sobre as colunas de A para manter a simetria, não afetam o *profile* da matriz após as trocas das linhas. Isso acontece porque as trocas entre colunas não alteram a quantidade de elementos nulos acima das fronteiras das colunas, dado que a fronteira de uma coluna é representada pelo primeiro elemento não nulo da mesma.

As trocas sequenciais de linhas adjacentes podem ser realizadas de maneira descendente ou ascendente. Nas subseções 2.9.1 e 2.9.2, são apresentadas, em detalhes, as trocas adjacentes descendentes e ascendentes, respectivamente.

2.9.1 Trocas adjacentes descendentes

Dada uma matriz simétrica A de ordem n , trocas adjacentes descendentes são aquelas trocas sequenciais feitas da linha k com a linha $k+1$, da linha $k+1$ com a linha $k+2$, \dots , da linha $l-1$ com a linha l , dado que $k < l \leq n$. Conforme mostrado por Hager (2002), é possível determinar a mudança obtida no *profile* de A , denotado por $D_{k:l}(A)$, por meio das trocas adjacentes descendentes efetuadas entre as linhas de k a l da matriz A , com

$k < l \leq n$, da seguinte maneira:

$$D_{k:l}(A) = |\{j \geq k : k < f_j(A) \leq l\}| - \left(\sum_{j \in \mathcal{F}_k} (\min\{l, g_j(A) - 1\} - k) \right) \quad (5)$$

em que $f_j(A) = \min\{i : 1 \leq i \leq j, a_{ij} \neq 0\}$ corresponde à linha onde se encontra a fronteira da coluna j , $\mathcal{F}_k = \{j : f_j(A) = k\}$ corresponde ao conjunto de colunas cujas fronteiras estão na linha k e $g_j(A) = \min\{i : f_j(A) < i, a_{ij} \neq 0\}$ corresponde ao segundo elemento não nulo da coluna j , caso exista, ou $g_j(A) = n + 1$ caso não exista o segundo elemento não nulo.

O primeiro termo da definição de $D_{k:l}$ na equação (5) corresponde à quantidade de elementos nulos que entrarão no *envelope* de A após as trocas adjacentes descendentes das linhas de k a l . Isto é, corresponde à quantidade de colunas j , a partir de k , cujas fronteiras estão abaixo da linha k e acima da linha l . Após a subida das linhas de $k + 1$ a l , com as trocas adjacentes descendentes, o *profile* é aumentado pela subida das fronteiras das colunas j .

Por outro lado, o somatório presente na definição de $D_{k:l}$ na (5) corresponde à diminuição do *profile* causado pela descida da linha k até a posição l . Nesse processo, acontece a descida da posição das fronteiras das colunas $j \in \mathcal{F}_k$, cujas fronteiras estão na linha k . As fronteiras que estão na linha k descerão até a posição em que se encontra o segundo elemento não nulo. Caso não existirem elementos não nulos na coluna j entre as linhas k e l , a fronteira da coluna j descerá até a linha l .

A partir da definição (5), Hager (2002) propôs o método apresentado pelo algoritmo mostrado na figura 12, que identifica um conjunto de linhas,

cujas trocas adjacentes descendentes resulta no menor aumento do *profile*, e efetua as respectivas trocas caso esse aumento seja negativo. Para identificar o conjunto de linhas a serem trocadas, o método inicia da parte de baixo da matriz e vai aumentando iterativamente o tamanho máximo do conjunto. Para cada valor de k , variando de maneira decrescente entre $n - 1$ e 1 , o algoritmo escolhe o valor de L , com $k < L \leq n$, que resulte no menor valor de $D_{k:l}$ e garanta que $D_{k:L} < 0$. Após a escolha do L , caso exista, o algoritmo realiza as trocas adjacentes descendentes entre as linhas de k a L .

O algoritmo mostrado na figura 12 retorna um vetor p , que representa a matriz permutação P , buscando reduzir o *profile* da matriz PAP^T . Nas *linhas de 1 a 3*, o vetor p recebe a representação da matriz identidade de ordem n . Nas *linhas de 4 a 26*, o algoritmo busca, para cada valor de k entre $n - 1$ e 1 (*linha 4*), um valor de l entre $k + 1$ e n (*linha 7*), cujo valor de $D_{k:l}(A)$ seja mínimo (*linhas 14 a 17*) e negativo (*linha 19*), efetuando as trocas ascendentes de k até L quando encontrar um valor de L que satisfaça tais condições de $D_{k:l}(A)$ (*linhas de 20 a 24*). Para encontrar o valor de $D_{k:l}(A)$ em cada caso, o algoritmo primeiro calcula o aumento de *profile* ocasionado pelas trocas descendentes da linha k até a posição l (*linha 8*) e depois calcula a redução proporcionada pelas mesmas trocas descendentes (*linhas de 9 a 12*), realizando e armazenando a diferença entre o aumento e a redução encontrados (*linha 13*). Após o cálculo de $D_{k:l}(A)$, o algoritmo verifica se o mesmo é o menor encontrado até o momento (*linha 14*) e armazena-o, caso seja (*linhas de 15 a 16*).

Para reduzir o espaço de busca por valores de l , que resultem em *profiles* menores por meio de valores de $D_{k:l}$ negativos, Hager (2002) identificou duas propriedades na definição (5) que permitem definir dois

Algoritmo: Método de Hager (2002) que utiliza trocas adjacentes descendentes para redução do *profile*.

```

Entrada: Matriz simétrica  $A$  de ordem  $n$ .
Saída: Vetor  $p$  contendo uma permutação de linhas e colunas de  $A$ .
// o vetor  $p$  inicia-se representando a matriz identidade  $I_n$ .
1 para (  $i \leftarrow 1; i \leq n; i \leftarrow i + 1$  ) faça
2    $p[i] = i$ ;
3 fim-para;
// para cada valor de  $k$ , o algoritmo busca um valor de  $l$  que resulte
// numa redução do profile de  $A$ .
4 para (  $k \leftarrow n - 1; k \geq 1; k \leftarrow k - 1$  ) faça
5    $menorAumento \leftarrow +\infty$ ;
6    $L \leftarrow NIL$ ;
// para cada valor de  $l$  entre  $k + 1$  e  $n$ 
7   para (  $l \leftarrow k + 1; l \leq n; l \leftarrow l + 1$  ) faça
// calcula-se o aumento do profile obtido pelas trocas
// descendentes de  $k$  até  $l$ 
8      $aumentoParcial \leftarrow |\{j \geq k : k < f_j(A) \leq l\}|$ ;
// calcula-se a redução do profile obtida pelas trocas
// descendentes de  $k$  até  $l$ 
9      $reducaoParcial \leftarrow 0$ ;
10    para cada (  $j \in \{j : f_j(A) = k\}$  ) faça
11       $reducaoParcial \leftarrow reducaoParcial + (\min\{l, g_j(A) - 1\} - k)$ ;
12    fim-para-cada;
// calcula-se a contribuição final das trocas descendentes de
//  $k$  até  $l$ 
13     $aumentoTotal \leftarrow aumentoParcial - reducaoParcial$ ;
// armazena-se o valor de  $L$  que proporcione o menor aumento de
// profile
14    se (  $(aumentoTotal < menorAumento)$  ) então
15       $menorAumento \leftarrow aumentoTotal$ ;
16       $L \leftarrow l$ ;
17    fim-se;
18  fim-para;
// realiza-se as trocas descendentes caso o menor aumento
// encontrado seja negativo
19  se (  $(menorAumento < 0)$  ) então
20     $q \leftarrow p[k]$ ;
21    para (  $i \leftarrow k; i \leq L - 1; i \leftarrow i + 1$  ) faça
22       $p[i] \leftarrow p[i + 1]$ ;
23    fim-para;
24     $p[L] = q$ ;
25  fim-se;
26 fim-para;
27 retorna  $p$ .
```

Figura 12 Método de Hager (2002) que utiliza trocas adjacentes descendentes para redução do *profile*.

limites superiores para os valores de l em cada iteração. A função *min* presente no somatório do segundo termo da equação (5) permanece constante a partir do momento em que l atinge o valor de $g_j(A) - 1$. Desta maneira, pode-se estabelecer que, na busca por valores de l em cada iteração, deve-se procurar até o valor limite de $\max\{g_j(A) : j \in \mathcal{F}\}$, visto que deve ser considerada a condição de l não ser selecionada na função *min* em todos os termos do respectivo somatório.

Ainda, como objetivo do método é, em cada iteração k , buscar uma redução do *profile*, não há necessidade de procurar valores de l que resultem em cenários em que o aumento no valor de *profile*, representado pelo primeiro termo da definição (5), seja maior que a diminuição do *profile*, representada pelo segundo termo da definição (5). A partir disso, tem-se que os cenários desejados são aqueles em que $|\{j \geq k : k < f_j(A) \leq l\}| < (\sum_{j \in \mathcal{F}_k} (\min\{l, g_j(A) - 1\} - k))$. Nota-se que o aumento no primeiro termo é proporcional ao aumento no valor l , já que, quanto maior o valor de l , maior é a possibilidade de haver colunas cujas fronteiras estão nas linhas entre k e l . O somatório presente no segundo termo, é limitado superiormente por $\sum_{j \in \mathcal{F}_k} (g_j(A) - 1 - k)$, visto que $\min\{l, g_j(A) - 1\} \leq g_j(A) - 1$. Portanto, deve-se limitar a busca de até o maior valor de l que resulte em $|\{j \geq k : k < f_j(A) \leq l\}| < \sum_{j \in \mathcal{F}_k} (g_j(A) - 1 - k)$, para que haja redução líquida do *profile*.

Reid e Scott (2002) propuseram uma implementação mais eficiente para os métodos de trocas descendentes de Hager (2002). A parte central dessa implementação, em que é encontrado a linha l para uma dada linha de entrada k cujas trocas adjacentes descendentes de k até l resulta na maior redução de *profile*, é apresentada no algoritmo mostrado na figura 13.

Algoritmo: Implementação de Reid e Scott (2002) para os métodos de trocas adjacentes descendentes de Hager (2002).

Entrada: Matriz simétrica A de ordem n e índice k de uma linha de A .
Saída: Índice l de uma linha de A .

```

1  $ganho\_liquido \leftarrow 0$ ;
2  $ganho\_maximo \leftarrow 0$ ;
  // número de colunas cuja primeira entrada encontra-se na linha  $k$ .
3  $ganho \leftarrow |colunas\_com\_primeira\_entrada(k)|$ ;
4 para (  $(m \leftarrow k + 1; k \leq n; k \leftarrow k + 1)$  ) faça
  // número de colunas cuja primeira entrada encontra-se na linha  $k$ 
  // e cuja segunda entrada encontra-se na linha  $m$ .
5    $colunas\_inalteradas \leftarrow$ 
      $|\{x | x \in colunas\_com\_primeira\_entrada(k) \cap colunas\_com\_segunda\_entrada(m)\}|$ ;
6    $ganho \leftarrow ganho - colunas\_inalteradas$ ;
7   se (  $ganho = 0$  ) então break.
8    $perda \leftarrow |colunas\_com\_primeira\_entrada(m)|$ ;
9    $ganho\_liquido \leftarrow ganho\_liquido + ganho - perda$ ;
10  se (  $ganho\_liquido > ganho\_maximo$  ) então
11     $ganho\_maximo \leftarrow ganho\_liquido$ ;
12     $l \leftarrow m$ ;
13  fim-se;
14 fim-para;
15 retorna  $l$ .
```

Figura 13 Implementação de Reid e Scott (2002) para os métodos de trocas adjacentes descendentes de Hager (2002).

No algoritmo mostrado na figura 13, o ganho corresponde ao valor que será diminuído do *profile* após as trocas adjacentes descendentes devido à descida dos coeficientes que se encontram na linha k . Porém, há colunas que, mesmo com a descida da primeira entrada não diminuirá o valor *profile*, uma vez que a segunda entrada está posicionada na linha m . Por outro lado, a perda corresponde ao valor que será somado ao *profile* após as trocas devido à subida de coeficientes que se encontram na linha m . O ganho líquido corresponde à alteração do valor do *profile* devido à soma do valor da perda e à subtração do valor do ganho. Após a inicialização das variáveis *net_gain* e *max_gain* nas linhas 1 e 2, o ganho é calculado, na linha 3, a partir do número de colunas cuja primeira entrada encontra-se na linha k . Posteriormente, no laço de repetição da linha 4 até a linha 14, para cada linha m pertencente

ao intervalo de $k + 1$ até n , é subtraído do ganho o número de colunas cuja primeira entrada encontra-se na linha k e cuja segunda entrada encontra-se na linha m (linhas 5 e 6). Se o ganho for igual a 0, o algoritmo para (linha 7). Caso contrário, a perda é calculada a partir do número de colunas cuja primeira entrada encontra-se na linha m (linha 8) e o ganho líquido é atualizado com a diferença entre o ganho e perda calculada para a linha m (linha 9). Se esse ganho líquido for maior que o ganho máximo até o momento (linha 10), o ganho líquido da iteração torna-se o ganho máximo (linha 11) e a coluna m é salva (linha 12). Na linha 15, retorna-se a coluna que resultou no maior ganho considerando todas as iterações.

Para tornar eficiente o cálculo dos valores do ganho e da perda, são utilizados, na implementação de Reid e Scott (2002) dos métodos de trocas adjacentes descendentes de Hager (2002), três vetores auxiliares. No vetor $num_first(i)$, é armazenado o número de colunas cuja primeira entrada está posicionada na linha i . No vetor $num_second(i)$, é armazenado o número de colunas cuja primeira entrada encontra-se na linha k e cuja segunda entrada encontra-se na linha i . No vetor $second(j)$, é armazenado o índice da segunda entrada da coluna j , caso a primeira entrada da linha j esteja na linha k , ou 0, caso contrário. Além disso, são utilizados dois outros vetores para facilitar a revisão dos vetores acima em cada iteração: $start(i)$, que contém o índice de uma coluna cuja primeira entrada está posicionada na linha i , e $next(j)$, que contém o índice de outra coluna cuja primeira entrada está posicionada na mesma linha em que está posicionada a primeira entrada da coluna j . A atualização desses vetores em cada iteração possui baixo custo computacional, e permite que a implementação do algoritmo mostrado na figura 13 seja eficiente.

2.9.2 Trocas adjacentes ascendentes

Assim como as trocas adjacentes descendentes, é possível estabelecer trocas adjacentes ascendentes. Dada uma matriz simétrica A de ordem n , trocas adjacentes ascendentes são aquelas trocas sequenciais feitas da linha k com a linha $k-1$, da linha $k-1$ com a linha $k-2$, \dots , da linha $l+1$ com a linha l , dado que $1 \leq l < k$.

De maneira semelhante, conforme mostrado por Hager (2002), é possível determinar a mudança obtida no *profile* de A , denotado por $U_{k:l}(A)$, por meio das trocas adjacentes ascendentes efetuadas entre as linhas de l a k da matriz A , novamente com $1 \leq l < k$, da seguinte maneira:

$$U_{k:l}(A) = \left(\sum_{\{j:a_{kj} \neq 0\}} \max\{f_j(A) - l, 0\} \right) - |\{j : a_{kj} = 0, k > f_j(A) \geq l\}| \quad (6)$$

em que novamente $f_j(A) = \min\{i : 1 \leq i \leq j, a_{ij} \neq 0\}$ corresponde à linha onde se encontra a fronteira da coluna j .

O primeiro termo da definição de $U_{k:l}$ na equação (6) corresponde ao aumento do *profile* causado pelas trocas adjacentes ascendentes das linhas de l a k . Corresponde à soma das distâncias, em relação à linha l , das fronteiras das colunas j , cujas fronteiras estão abaixo da linha l e as colunas possuem elementos não nulos associados à linha k . Após as trocas adjacentes, a linha k subirá até a posição l , fazendo com que as colunas j passem a ter suas fronteiras na posição k .

Já o segundo termo da definição de $U_{k:l}$ na (6) refere-se a diminuição do *profile* causado pelas respectivas trocas adjacentes. Esse termo corresponde à quantidade de colunas j , que possuem elementos nulos

associados à linha k , e cujas fronteiras estão entre as linhas l e $k - 1$. As trocas adjacentes ascendentes faz com que todos os elementos, de todas as colunas, compreendidos entre as linhas l e $k - 1$, desçam uma posição. Com isso, as colunas j terão suas fronteiras uma posição abaixo e o *profile* será reduzido em uma unidade para cada coluna j .

A partir da definição (6), Hager (2002) propôs o método apresentado pelo algoritmo mostrado na figura 14, que identifica, de maneira similar ao algoritmo 12, um conjunto de linhas, cujas trocas adjacentes ascendentes resulta no menor aumento do *profile*, e efetua as respectivas trocas caso esse aumento seja negativo. Para identificar o conjunto de linhas a serem trocadas, o método inicia da parte de cima da matriz e vai aumentando iterativamente o tamanho máximo do conjunto. Para cada valor de k , variando de maneira ascendente entre 2 e n , o algoritmo escolhe o valor de L , com $1 \leq L < k$, que resulte no menor valor de $U_{k:l}$ e garanta que $U_{k:L} < 0$. Após a escolha do L , caso exista, o algoritmo faz as trocas adjacentes ascendentes entre as linhas de L a k .

O algoritmo mostrado na figura 14 também retorna um vetor p , que representa a matriz permutação P , buscando reduzir o *profile* da matriz PAP^T . Nas *linhas de 1 a 3*, o vetor p recebe a representação da matriz identidade de ordem n . Nas *linhas de 4 a 26*, o algoritmo busca, para cada valor de k entre 2 e n (*linha 4*), um valor de l entre 1 e $k - 1$ (*linha 7*), cujo valor de $U_{k:l}(A)$ seja mínimo (*linhas de 14 a 17*) e negativo (*linha 19*), efetuando as trocas ascendentes de k até L (*linhas de 20 a 24*) quando encontrar um valor de L que satisfaça tais condições de $U_{k:l}(A)$. Para encontrar o valor de $U_{k:l}(A)$ em cada caso, o algoritmo primeiro calcula a redução de *profile* ocasionada pelas trocas ascendentes da linha k até a

Algoritmo: Método de que Hager (2002) utiliza trocas adjacentes ascendentes para redução do *profile*.

Entrada: Matriz simétrica A de ordem n .
Saída: Vetor p contendo uma permutação de linhas e colunas de A .
// o vetor p inicia-se representando a matriz identidade I_n .

```

1 para (  $i \leftarrow 1; i \leq n; i \leftarrow i + 1$  ) faça
2    $p[i] = i$ 
3 fim-para;
// para cada valor de  $k$ , o algoritmo busca um valor de  $l$  que reduza o
// profile de  $A$ .
4 para (  $k \leftarrow 2; k \leq n; k \leftarrow k + 1$  ) faça
5    $menorAumento \leftarrow +\infty$ ;
6    $L \leftarrow NIL$ ;
// para cada valor de  $l$  entre 1 e  $k$ 
7   para (  $l \leftarrow 1; l < k; l \leftarrow l + 1$  ) faça
// calcula-se a redução do profile obtida pelas trocas
// ascendentes de  $k$  até chegar em  $l$ 
8      $reducaoParcial \leftarrow |\{j : a_{kj} = 0, k > f_j(A) \geq l\}|$ ;
// calcula-se o aumento do profile obtido pelas trocas
// ascendentes de  $k$  até chegar em  $l$ 
9      $aumentoParcial \leftarrow 0$ ;
10    para cada (  $j \in \{j : a_{kj} \neq 0\}$  ) faça
11       $aumentoParcial \leftarrow aumentoParcial + (\max\{l, f_j(A) - l, 0\})$ ;
12    fim-para-cada;
// calcula-se a contribuição final das trocas ascendentes de  $k$ 
// até  $l$ 
13     $aumentoTotal \leftarrow aumentoParcial - reducaoParcial$ ;
// armazena-se o valor de  $l$  que proporcionou o menor aumento
// de profile
14    se (  $aumentoTotal < menorAumento$  ) então
15       $menorAumento \leftarrow aumentoTotal$ ;
16       $L \leftarrow l$ 
17    fim-se;
18  fim-para;
// efetua-se as trocas ascendentes caso o menor aumento encontrado
// seja negativo
19  se (  $menorAumento < 0$  ) então
20     $q \leftarrow p[k]$ ;
21    para (  $i \leftarrow k; i \geq L + 1; i \leftarrow i - 1$  ) faça
22       $p[i] \leftarrow p[i - 1]$ ;
23    fim-para;
24     $p[L] = q$ ;
25  fim-se;
26 fim-para;
27 retorna  $p$ .
```

Figura 14 Método de que Hager (2002) utiliza trocas adjacentes ascendentes para redução do *profile*.

posição l (*linha 8*) e depois calcula o aumento proporcionado pelas mesmas trocas ascendentes (*linhas de 9 a 12*), realizando e armazenando a diferença entre o aumento e a redução encontrados (*linha 13*).

Assim como no algoritmo 12, para reduzir o espaço de busca por valores de l , garantam uma redução do *profile* por meio de valores negativos de $U_{k:l}$, Hager (2002) mostrou que apenas deverão ser considerados valores de l maiores que aquele valor de l_1 que faz com que

$$\sum_{\{j:a_{kj}\neq 0\}} \max\{l, f_j(A) - l, 0\} > |\{j : k > f_j(A) \geq 1\}|$$

já que $|\{j : k > f_j(A) \geq 1\}|$ limita superiormente o valor de $|\{j : a_{kj} = 0, k > f_j(A) \geq l\}|$. Desta maneira, como o primeiro termo decresce com o aumento de l , há uma garantia que, para qualquer valor de $l > l_1$, o primeiro termo será menor que o segundo, resultando em uma redução do *profile* representada por $U_{k:l}$.

Reid e Scott (2002) propuseram uma implementação mais eficiente para os métodos de trocas ascendentes de Hager (2002). A parte central dessa implementação, em que é encontrado a linha l para uma dada linha de entrada k cujas trocas adjacentes ascendentes de k até l resulta na maior redução de *profile*, é apresentada no algoritmo mostrado na figura 15.

No algoritmo mostrado na figura 15, a perda corresponde ao valor que será somado ao *profile* após as trocas adjacentes ascendentes devido à descida dos coeficientes que se encontram na linha k . Ainda, há um acréscimo no valor da perda correspondente ao número de colunas que possuem a primeira entrada na linha m e que possuem alguma entrada na linha k . Por outro lado, o ganho corresponde ao valor que será somado

Algoritmo: Implementação de Reid e Scott (2002) para os métodos de trocas adjacentes ascendentes de Hager (2002).

Entrada: Matriz simétrica A de ordem n e índice k de uma linha de A .
Saída: Índice l de uma linha de A .

```

1  $ganho\_liquido \leftarrow 0$ ;
2  $ganho\_maximo \leftarrow 0$ ;
3  $perda \leftarrow |colunas\_com\_primeira\_entrada(k)|$ ;
4 para (  $m \leftarrow k - 1; k \geq 1; k \leftarrow k + 1$  ) faça
5      $acrescimo\_perda \leftarrow |\{x|x \in colunas\_com\_primeira\_entrada(m) \cap A[k]\}|$ ;
6      $perda \leftarrow perda + acrescimo\_perdas$ ;
7      $ganho \leftarrow |colunas\_com\_primeira\_entrada(m)|$ ;
8      $ganho\_liquido \leftarrow ganho\_liquido + ganho - perda$ ;
9     se (  $ganho\_liquido > ganho\_maximo$  ) então
10          $ganho\_maximo \leftarrow ganho\_liquido$ ;
11          $l \leftarrow m$ ;
12     fim-se;
13 fim-para;
14 retorna  $l$ .
```

Figura 15 Implementação de Reid e Scott (2002) para os métodos de trocas adjacentes ascendentes de Hager (2002).

ao *profile* após as trocas devido à descida de coeficientes que se encontram na linha m . O ganho líquido corresponde à alteração do valor do *profile* devido à soma do valor da perda e da subtração do valor do ganho. Após a inicialização das variáveis *net_gain* e *max_gain* nas linhas 1 e 2, perda é calculada, na linha 3, a partir do número de colunas cuja primeira entrada encontra-se na linha k . Posteriormente, no laço de repetição da linha 4 até a linha 13, para cada linha m pertencente ao intervalo de $k - 1$ até 1, é somado ao valor da perda o número de colunas, que possuem alguma entrada na linha m (linhas 7), e o ganho líquido é atualizado com a diferença entre o ganho e perda calculada para a linha m (linha 8). Se esse ganho líquido for maior que o ganho máximo até o momento (linha 9), o ganho líquido da iteração torna-se o ganho máximo (linha 10) e a coluna m é salva (linha 11). Na linha 14, retorna-se a coluna que resultou no maior ganho considerando todas as iterações.

Para tornar eficiente o cálculo dos valores do ganho e da perda, são utilizados, na implementação de Reid e Scott (2002) dos métodos de trocas adjacentes descendentes de Hager (2002), três vetores auxiliares. No vetor $first(j)$, é armazenado o índice da primeira entrada da coluna j . No vetor $num_first(i)$, é armazenado o número de colunas cuja primeira entrada está posicionada na linha i . No vetor $gains(i)$, é armazenado o ganho acumulado das linhas que são adjacentes correspondente ao somatório de $num_first(j)$ para as linhas j de i a n .

2.10 Método dos gradientes conjugados

Apresenta-se, nesta seção, uma breve descrição sobre o método dos gradientes conjugados (MGC). Essa descrição é baseada na construção do método apresentada por e Shewchuk (1994) e Burden e Faires (2010).

O método dos gradientes conjugados foi proposto por Hestenes e Stiefel (1952) e Lanczos (1952), inicialmente, como um método direto para resolução de sistemas lineares cuja matriz de coeficientes é simétrica e positivo-definida. Porém, como método direto, o MGC é pior que o método da eliminação Gaussiana com pivotamento pois, apesar de possuir a mesma quantidade de iterações, o MGC possui iterações com um custo computacional maior (BURDEN; FAIRES, 2010). Dessa forma, o MGC é útil quando aplicado como um método de aproximação iterativa para a resolução de sistemas lineares esparsos e de grande porte.

Seja um sistema de equações lineares na forma $Ax = b$, em que $A \in \mathbb{R}^{n \times n}$, x e $b \in \mathbb{R}^n$. A solução desse sistema de equações lineares pode ser

encontrada por meio da minimização da função $f(x)$, definida como

$$f(x) = \frac{x^T A x}{2} - b^T x, \quad (7)$$

com $x \in \mathbb{R}^n$. Dada a função f , o vetor x^* é uma solução do sistema de equações lineares $Ax = b$ se, e somente se, x^* minimiza f (BURDEN; FAIRES, 2010).

A função $f(x)$ pode ser reescrita da seguinte maneira:

$$f(x) = \frac{1}{2} \sum_{i=1}^n \left(\sum_{j=1}^n a_{ij} x_j \right) x_i - \sum_{i=1}^n b_i x_i + c. \quad (8)$$

Os pontos de mínimo de uma função correspondem a um subconjunto dos pontos críticos dessa função. Os pontos críticos de uma função são encontrados nas ocorrências em que o valor da derivada dessa função é igual a zero ($\vec{\nabla} f = 0$). Além disso, o vetor gradiente da função $\vec{\nabla} f(x)$ representa a direção e o sentido de maior crescimento da função f . Ao calcular o gradiente $\vec{\nabla} f$, por meio do cálculo da derivada de $f(x)$ em relação ao vetor $x = [x_1, x_2, \dots, x_n]^T$, tem-se:

$$\vec{\nabla} f = \begin{bmatrix} \frac{\partial}{\partial x_1} f(x) \\ \frac{\partial}{\partial x_2} f(x) \\ \vdots \\ \frac{\partial}{\partial x_n} f(x) \end{bmatrix}.$$

A derivada de $f(x)$ em relação à x_i é, então:

$$\frac{\partial}{\partial x_i} f = \frac{1}{2} \sum_{j=1}^n a_{ij} x_j + \frac{1}{2} \sum_{j=1}^n a_{ji} x_j - b_i. \quad (9)$$

Porém, como a matriz A é simétrica, tem-se que $\vec{\nabla}f = \frac{1}{2}Ax + \frac{1}{2}A^T x - b = Ax - b$. O resultado é um vetor também pertencente a \mathbb{R}^n , que satisfaz $\vec{\nabla}f(x^*) = Ax^* - b = 0$. Esse vetor possui a mesma magnitude do vetor resíduo r , denotado por $r = b - Ax$, e sentido contrário.

É possível construir um método de otimização para a solução do sistema de equações lineares $Ax = b$ é por meio de um esquema iterativo, em que, em cada passo, a procura pela solução é feita na direção inversa do vetor resíduo. Dada uma solução inicial x_0 para o sistema de equações lineares, com $r_0 = b - Ax_0$, o passo de busca desse método seria:

$$x_k = x_{k+1} + \alpha r_k, \quad (10)$$

em que α é a magnitude do passo em direção a r . Porém, segundo Burden e Faires (2010), esse método possui convergência lenta. O método dos gradientes conjugados possui a mesma estrutura iterativa, mas possui outra abordagem de busca que garante uma convergência mais rápida que o método anterior. Inicia-se o método dos gradientes conjugados escolhendo-se uma solução inicial x_0 , uma solução aproximada de $Ax^* = b$. Considere que, inicialmente, $r_0 = b - Ax_0$ e $d_1 = r_0$, em que r_k é o vetor resíduo na iteração k e d_k é o vetor de direção de busca na iteração k , com $k = 0, 1, \dots, n - 1$. O método dos gradientes conjugados constrói os vetores r e d , a cada iteração, por

$$r_k = r_{k-1} - \alpha_k A d_k \quad (11)$$

e

$$d_{k+1} = r_k + \beta_k d_k, \quad (12)$$

em que

$$\alpha_k = \frac{r_{k-1}^T r_{k-1}}{d_k^T A d_k} \quad (13)$$

e

$$\beta_k = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}. \quad (14)$$

A solução em cada iteração x_k é, então, calculada pela fórmula

$$x_k = x_{k-1} + \alpha_k d_k. \quad (15)$$

Com as equações 11, 12, 13, 14 e 15, forma-se o algoritmo iterativo do método dos gradientes conjugados. Esse algoritmo é mostrado na figura 16. Inicialmente, define-se a precisão numérica ε desejada. Nas linhas 3 a 5, inicializam-se os vetores x, r e d . As variáveis k e *erro* são inicializadas na linha 6, em que k é o contador para a estrutura de repetição *enquanto* e o *erro* é a variável para o critério de parada. A cada iteração atualiza-se $\text{erro} \leftarrow \frac{\|x_k - x_{k-1}\|_\infty}{\|x_k\|_\infty}$, em que $\|x\|_\infty = \max\{|x_1|, |x_2|, \dots, |x_n|\}$. Se $\text{erro} \geq \varepsilon$, o algoritmo para.

O método dos gradientes conjugados converge para a solução em no máximo n iterações. Para mais detalhes sobre o método dos gradientes conjugados, veja Shewchuk (1994) e Burden e Faires (2010, p. 479), por exemplo.

2.10.1 Método dos gradientes conjugados preconditionado

Embora o método dos gradientes conjugados seja um resolutor rápido em relação aos métodos diretos, há casos em que a convergência desse método é lenta. Segundo Saad (2003, p. 261), “isso ocorre com frequência em sistemas de equações lineares (SEL) oriundos de problemas

Algoritmo: método dos gradientes conjugados.

```

1 início
  // define a precisão numérica  $\varepsilon$ , ex.  $10^{-16}$ 
2  $\varepsilon \leftarrow 10^{-16}$ ;
3  $x_0 \leftarrow 0$ ; // estimativa inicial
4  $r_0 \leftarrow b - Ax_0$ ;
5  $d_0 \leftarrow r_0$ ;
6  $k \leftarrow 1$ ;  $erro \leftarrow \varepsilon + 1$ ;
  //  $n$  é a dimensão de  $x$ 
7 enquanto (  $(k < n) \wedge (erro > \varepsilon)$  ) faça
8    $\alpha_k \leftarrow \frac{r_{k-1}^T r_{k-1}}{d_k^T A d_k}$ ; // equação 13
9    $x_k \leftarrow x_{k-1} + \alpha_k d_k$ ; // equação 15
10   $r_k \leftarrow r_{k-1} - \alpha_k A d_k$ ; // equação 11
11   $erro \leftarrow \frac{\|x_k - x_{k-1}\|_\infty}{\|x_k\|_\infty}$ ;
12   $k \leftarrow k + 1$ ;
13   $\beta_k \leftarrow \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$ ; // equação 14
14   $d_{k+1} \leftarrow r_k + \beta_k d_k$ ; // equação 12
15 fim-enquanto;
16 fim.
```

Figura 16 Pseudocódigo do método dos gradientes conjugados.
 Fonte: Chagas (2015)

em dinâmica dos fluídos e dispositivos eletrônicos”. Com isso, métodos de condicionamento de SEL são utilizadas para acelerar a convergência do método dos gradientes conjugados.

O condicionamento da matriz de coeficientes de um SEL é realizado de forma que seja gerado um SEL equivalente ao SEL original mas que possa ser resolvido mais rapidamente e que apresente a mesma solução. Embora não seja garantido que a resolução de um SEL condicionado convirja com menos iterações, isso geralmente ocorre (SAAD, 2003, p. 261).

Considere um SEL $Ax = b$. Em termos simples, os métodos de

precondicionamento encontram uma matriz M não singular, ou seja, uma matriz que possua inversa, em que o resultado da operação $P^{-1}A$ tenha um número de condição menor que A . Além disso, M deverá ser de tal forma que, a resolução de $Mx = b$ apresente o menor custo computacional possível. Isso porque, em cada iteração, esse SEL deverá ser resolvido. Ainda, no caso do método dos gradientes conjugados preconditionado, a matriz M também deve ser simétrica e positivo-definida.

Como Saad (2003, p. 262) explica, “há três formas de preconditionamento: à esquerda, à direita e o preconditionamento conhecido como *split*”. Especificamente, neste trabalho, utilizou-se os preconditionamentos à esquerda e *split*.

A matriz M pode ser aplicada à esquerda do SEL $Ax = b$. Nesse caso, gera-se o SEL preconditionado

$$M^{-1}Ax = M^{-1}b. \quad (16)$$

No preconditionamento *split* utiliza-se a matriz M em sua forma fatorada $M = M_L M_R$, em que as matrizes M_L e M_R são matrizes triangulares. Com isso, o preconditionador pode ser separado em

$$M_L^{-1} A M_R^{-1} u = M_L^{-1} b, \quad x \equiv M_R^{-1} u. \quad (17)$$

Neste trabalho, utilizou-se o preconditionamento à esquerda e o preconditionamento *split*. O preconditionamento à esquerda, apresentado na equação 16, foi utilizado com o método preconditionador de Jacobi. O preconditionamento *split*, apresentado na equação 17, foi utilizado com o método incompleto de Cholesky. Esses métodos preconditionadores são

detalhados na subseção 2.10.2.

A seguir, são apresentados os pseudocódigos do método dos gradientes conjugados preconditionado à esquerda e pelo *split*. Esses métodos são apresentados nas figuras 17 e 18, respectivamente. Note que, no algoritmo da figura 17, utiliza-se a matriz preconditionadora M e a variável z_k para alterar o algoritmo do método dos gradientes conjugados mostrado na figura 16.

Algoritmo: método dos gradientes conjugados preconditionado à esquerda.

```

1 início
  // define a precisão numérica  $\varepsilon$ , ex.  $10^{-16}$ 
2  $\varepsilon \leftarrow 10^{-16}$ ;
3  $x_0 \leftarrow 0$ ; // estimativa inicial
4  $r_0 \leftarrow b - Ax_0$ ;
5  $z_0 \leftarrow M^{-1}r_0$ ;
6  $d_0 \leftarrow r_0$ ;
7  $k \leftarrow 1$ ;  $erro \leftarrow \varepsilon + 1$ ;
  //  $n$  é a dimensão de  $x$ 
8 enquanto (  $(k < n) \wedge (erro > \varepsilon)$  ) faça
9    $\alpha_k \leftarrow \frac{r_{k-1}^T z_{k-1}}{d_k^T A d_k}$ ;
10   $x_k \leftarrow x_{k-1} + \alpha_k d_k$ ;
11   $r_k \leftarrow r_{k-1} - \alpha_k A d_k$ ;
12   $z_k \leftarrow M^{-1}r_k$ ;
13   $erro \leftarrow \frac{\|x_k - x_{k-1}\|_\infty}{\|x_k\|_\infty}$ ;
14   $k \leftarrow k + 1$ ;
15   $\beta_k \leftarrow \frac{r_k^T r_k}{z_{k-1}^T z_{k-1}}$ ;
16   $d_{k+1} \leftarrow z_k + \beta_k d_k$ ;
17 fim-enquanto;
18 fim.
```

Figura 17 Pseudocódigo do método dos gradientes conjugados preconditionado à esquerda.

Fonte: Saad (2003) (adaptação)

No algoritmo da figura 18, utiliza-se a matriz M na forma de uma fatoração incompleta de Cholesky, ou seja, $M = LL^T$. Isso é possível pois pode-se utilizar o preconditionador do tipo *split*, mostrado na equação 17, para que a simetria seja mantida. Com isso, pode-se reescrever a equação 17 na forma $L^{-1}AL^{-T}u = L^{-1}b$, em que $x = L^{-T}u$. Além disso, a variável z_k é também utilizada. Para mais detalhes sobre o método dos gradientes conjugados preconditionado, veja Saad (2003, p. 262).

Algoritmo: método dos gradientes conjugados preconditionado por *split*.

```

1 início
  // define a precisão numérica  $\varepsilon$ , ex.  $10^{-16}$ 
2    $\varepsilon \leftarrow 10^{-16}$ ;
3    $x_0 \leftarrow 0$ ; // estimativa inicial
4    $r_0 \leftarrow b - Ax_0$ ;
5    $z_0 \leftarrow L^{-1}r_0$ ;
6    $d_0 \leftarrow L^{-T}z_0$ ;
7    $k \leftarrow 1$ ;  $erro \leftarrow \varepsilon + 1$ ;
  //  $n$  é a dimensão de  $x$ 
8   enquanto (  $(k < n) \wedge (erro > \varepsilon)$  ) faça
9      $\alpha_k \leftarrow \frac{z_{k-1}^T z_{k-1}}{d_k^T A d_k}$ ;
10     $x_k \leftarrow x_{k-1} + \alpha_k d_k$ ;
11     $z_k \leftarrow z_{k-1} - \alpha_k L^{-1} A d_k$ ;
12     $erro \leftarrow \frac{\|x_k - x_{k-1}\|_\infty}{\|x_k\|_\infty}$ ;
13     $k \leftarrow k + 1$ ;
14     $\beta_k \leftarrow \frac{z_k^T z_k}{z_{k-1}^T z_{k-1}}$ ;
15     $d_{k+1} \leftarrow L^{-T} z_k + \beta_k d_k$ ;
16  fim-enquanto;
17 fim.
```

Figura 18 Pseudocódigo do método dos gradientes conjugados preconditionado por *split*.

Fonte: Saad (2003) (adaptação)

2.10.2 Métodos preconditionadores

Alguns métodos diretos e iterativos para a resolução de SEL são utilizados como preconditionadores. Tem-se o método de Cholesky, como exemplo de método direto utilizado como preconditionador, e o método de Jacobi (1845) como método iterativo. Contudo, esses métodos são aplicados de forma parcial pois, nesse caso, objetivo desses métodos não é a resolução do SEL.

O método de Jacobi (1845) é um resolutor clássico de SEL e é um dos métodos de preconditionamento mais simples. Seja A uma matriz $n \times n$, a matriz D é composta pela diagonal principal de A e a matriz M preconditionadora de Jacobi é definida por $M = D$.

Na fatoração de Cholesy, a matriz A é fatorada em L e L^T em que $A = LL^T$, em que L é a matriz estritamente triangular inferior somada com a matriz D . Para o preconditionamento, utiliza-se a fatoração incompleta de Cholesky. Com isso, calcula-se $M = LL^T$. Segundo Benzi (2002), os autores Meijerink e van der Vorst (1977) foram os primeiros autores a utilizarem a fatoração incompleta de Cholesky como preconditionador do MGC. Especificamente, neste trabalho, utilizou-se a fatoração incompleta de Cholesky com zero *fill-in* (IC0) para realizar o preconditionamento das matrizes de coeficientes dos SEL. O IC0 mantém, na matriz M , a estrutura de matriz A sem realizar *fill-in*. Em outras palavras, se $a_{ij} \neq 0$, então, $m_{ij} \neq 0$ e se $a_{ij} = 0$, então, $m_{ij} = 0$, para $1 \leq i, j \leq n$, $A = [a_{ij}]$ e $M = [m_{ij}]$. Mostra-se, na figura 19, o algoritmo da fatoração incompleta de Cholesky com zero *fill-in* (GOLUB; VAN LOAN, 1996, p. 535). O algoritmo recebe uma matriz A simétrica e retorna a matriz M preconditionadora.

Para mais detalhes sobre o método preconditionador de Jacobi, veja

Algoritmo: IC0.

Entrada: matriz $A = [a_{ij}]$ simétrica $n \times n$;
Saída: matriz $M = [m_{ij}]$ condicionadora;
1 início
2 $M \leftarrow A$; // copia-se os elementos de A para M
3 para ($k \leftarrow 1$; $k \leq n$; $k \leftarrow k+1$) **faça**
4 $m_{kk} \leftarrow \sqrt{m_{kk}}$;
5 para ($i \leftarrow k+1$; $i \leq n$; $i \leftarrow i+1$) **faça**
6 se ($m_{ik} \neq 0$) **então** $m_{ik} \leftarrow \frac{m_{ik}}{m_{kk}}$;
7 fim-para;
8 para ($j \leftarrow k+1$; $j \leq n$; $j \leftarrow j+1$) **faça**
9 para ($i \leftarrow j$; $i \leq n$; $i \leftarrow i+1$) **faça**
10 se ($m_{ij} \neq 0$) **então** $m_{ij} \leftarrow m_{ij} - m_{ik} * m_{jk}$;
11 fim-para;
12 fim-para;
13 fim-para;
14 retorna M ;
15 fim.

Figura 19 Pseudocódigo da fatoração incompleta de Cholesky com zero *fill-in*.
 Fonte: Golub e Van Loan (1996)

Saad (2003, p. 284). Para mais detalhes sobre a fatoração incompleta de Cholesky com zero *fill-in*, veja Golub e Van Loan (1996, p. 535).

2.11 Estruturas CSR-SSS e CSC-SSS

Pode-se economizar espaço em memória apenas armazenando-se as informações essenciais das matrizes esparsas, por exemplo, somente os elementos não nulos e os índices. Para isso, há vários esquemas de armazenamento de matrizes esparsas e Saad (2003, p. 89-92) apresenta alguns.

Os esquemas de armazenamento de matrizes esparsas *Compressed Sparse Row/Column Sparse Symmetric Skyline* (CSR-SSS/CSC-SSS),

utilizados neste trabalho, são a combinação de outros dois esquemas conhecidos: *Compressed Sparse Row/Column* (CSR/CSC) e *Skyline*. Nos esquemas de armazenamento CSR-SSS/CSC-SSS são utilizados quatro vetores para armazenar uma matriz esparsa A : um vetor para armazenar os coeficientes da diagonal principal, assim como no esquema *Skyline*; e outros três vetores, assim como nos esquemas CSR/CSC, em que se armazenam os coeficientes não nulos da matriz estritamente triangular inferior (ou superior), os índices das colunas (CSR-SSS) ou os índices das linhas (CSC-SSS) desses coeficientes não nulos, e as posições em que cada linha (CSR-SSS) ou coluna (CSC-SSS) são iniciadas nos vetores dos coeficientes e índices, respectivamente. Para detalhes sobre os esquemas de armazenamento de matrizes esparsas CSR/CSC e *Skyline*, veja Saad (2003, p. 90) e Jennings (1966) respectivamente. Para mais detalhes sobre o esquema de armazenamento de matrizes esparsas CSR-SSS/CSC-SSS, veja Gonzaga de Oliveira (2015, p. 66-68).

A seguir, um exemplo das estruturas CSR-SSS e CSC-SSS é mostrado. Para isso, considere a seguinte matriz de adjacências do primeiro grafo da figura 10:

$$A = \begin{bmatrix} \mathbf{1} & 0 & 1 & 0 & 0 \\ 0 & \mathbf{1} & 1 & 1 & 0 \\ 1 & 1 & \mathbf{1} & 0 & 1 \\ 0 & 1 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 1 & 0 & \mathbf{1} \end{bmatrix}$$

A matriz A pode ser armazenada no formato CSR-SSS da seguinte forma:

$$\begin{aligned}
 dCoef &= [1 \ 1 \ 1 \ 1 \ 1], \\
 linhaPtr &= [0 \ 0 \ 1 \ 3 \ 4], \\
 colunaInd &= [1 \ 2 \ 2 \ 3], \\
 coef &= [1 \ 1 \ 1 \ 1],
 \end{aligned}$$

em que $dCoef$ é o vetor dos coeficientes da diagonal principal, $linhaPtr$ é o vetor de onde se inicia cada linha nos vetores $colInd$ e $coef$, $colunaInd$ é o vetor dos índices das colunas e $coef$ são os coeficientes da matriz estritamente triangular inferior.

De forma similar ao formato CSR-SSS, a matriz A pode ser armazenada no formato CSC-SSS da seguinte forma:

$$\begin{aligned}
 dCoef &= [1 \ 1 \ 1 \ 1 \ 1], \\
 colunaPtr &= [0 \ 3 \ 4 \ 5 \ 6 \ 6], \\
 linhaInd &= [0 \ 0 \ 1 \ 3 \ 4], \\
 coef &= [1 \ 1 \ 1 \ 1],
 \end{aligned}$$

em que o $colunaPtr$ é o vetor de onde se inicia cada coluna nos vetores $colInd$ e $coef$, $linhaInd$ é o vetor dos índices das linhas.

3 HEURÍSTICA SNAY-GL

Nesse capítulo, mostra-se, em detalhes, a heurística de Snay (1976) para redução de *profile*. Além disso, apresenta-se a heurística Snay-GL, proposta neste trabalho como forma de reduzir o custo computacional da heurística de Snay (1976).

Considere o grafo $G = (V, E)$, em que V e E são conjuntos de vértices e de arestas, respectivamente. A heurística de Snay (1976) recebe o vértice v inicial e referencia-se esse vértice na primeira entrada da renumeração $S = \{s(1), s(2), \dots, s(|V|)\}$. Para essa etapa, Snay (1976) propôs um algoritmo que retorna 10 vértices pseudoperiféricos iniciais para a heurística. Considere que a distância $d(v, u)$ entre dois vértices $u, v \in V$ é o tamanho do menor caminho entre u e v . O método para a determinação dos candidatos a vértice inicial, proposto por Snay (1976) e mostrado no algoritmo da figura 20 extraída de Gonzaga de Oliveira e Chagas (2014), consiste em considerar, a partir de um vértice aleatório $v \in V$, um conjunto D com *cinco* vértices mais distantes de v . Considera-se, também, o conjunto Q com cinco vértices mais distantes do vértice $x \in D$, em que x é um dos vértices mais distantes de v . Os conjuntos de vértices D e Q contêm os candidatos a vértice inicial da heurística de Snay (1976). Ao escolher os vértices mais distantes de um vértice inicial v aleatório, a heurística de Snay (1976) considera, de maneira implícita, os vértices do(s) último(s) nível(is) da estrutura de nível enraizada de v .

No algoritmo mostrado na figura 20, utiliza-se a estrutura de nível para se determinar os vértices mais distantes a um determinado vértice. O algoritmo recebe um grafo $G = (V, E)$ e tem como saída um conjunto de vértices pseudoperiféricos $D \cup Q$. Na condição da linha 2, exige-se que o

Algoritmo: *VerticesPseudoPerifericosSnay*

```

Entrada: grafo  $G = (V, E)$ ;
Saída: conjunto de vértices pseudoperiféricos  $D \cup Q$ ;
1 início
2   se (  $|V| < 10$  ) então retorna;
3    $v \leftarrow \text{VerticeAleatorio}(V)$ ;
   // a estrutura de nível enraizada do vértice  $v$  é construída
4    $\mathcal{L}(v) \leftarrow \text{BuscaEmLargura}(v)$ ;
5    $D \leftarrow \emptyset$ ; // conterà  $v$  vértices mais distantes de  $v$ 
6    $k \leftarrow 0$ ;
7    $u \leftarrow \emptyset$ ; // um dos vértices mais distantes de  $v$ 
8    $i \leftarrow 0$ ;
9   enquanto (  $i < 5$  ) faça
10    para cada (  $w \in L_{\ell(v)-k}(v)$  ) faça
11       $D \leftarrow D \cup \{w\}$ ;
12       $i \leftarrow i + 1$ ;
   // um vértice  $u$  mais distante de  $v$  é determinado
13    se (  $u = \emptyset$  ) então
14       $u \leftarrow w$ ;
   // a estrutura de nível enraizada do vértice  $u$ 
15       $\mathcal{L}(u) \leftarrow \text{BuscaEmLargura}(u)$ ; // é construída
16    fim-se;
   // sai da estrutura de repetição para cada
17    se (  $i \geq 5$  ) então break;
18    fim-para-cada;
19     $k \leftarrow k + 1$ ;
20  fim-enquanto;
21   $Q \leftarrow \emptyset$ ; // conterà  $v$  vértices mais distantes de  $u$ 
22   $k \leftarrow 0$ ;
23   $i \leftarrow 0$ ;
24  enquanto (  $i < 5$  ) faça
25    para cada (  $w \in L_{\ell(u)-k}(u) - D$  ) faça
26       $Q \leftarrow Q \cup \{w\}$ ;
27       $i \leftarrow i + 1$ ;
   // sai da estrutura de repetição para cada
28    se (  $i \geq 5$  ) então break;
29    fim-para-cada;
30     $k \leftarrow k + 1$ ;
31  fim-enquanto;
32  retorna  $D \cup Q$ .
33 fim.

```

Figura 20 Pseudocódigo do algoritmo para encontrar vértices iniciais para a heurística de Snay

Fonte: Gonzaga de Oliveira e Chagas (2014)

grafo tenha, pelo menos, 10 vértices. Um vértice aleatório é atribuído a v na linha 3. Na linha 4, constrói-se a estrutura de nível enraizada de v . A construção do conjunto de vértices mais distantes a v inicia-se na linha 5.

A variável k , inicializada na linha 6, e utilizada para que $|D| = 5$, pois o(s) último(s) nível(is) da estrutura de nível enraizada de v pode(m) não ter cinco vértices. A variável u , inicializada na linha 7, é utilizada para armazenar um vértice mais distante a v . A variável i , inicializada na linha 8, é utilizada na estrutura de repetição das linhas 9 a 20, em que se inserem os cinco vértices em D na linha 11. Constrói-se a estrutura de nível enraizada de u na linha 15. Da mesma forma, cinco vértices mais distantes a u são inseridos em Q , nas linhas 21 a 31. Por simplicidade da apresentação, mostram-se os dois trechos de código similares em seguida, das linhas 6 a 20 e das linhas 22 a 31. Finalmente, retorna-se $D \cup Q$, na linha 32.

Na etapa de renumeração da heurística de Snay (1976), mostrada no algoritmo da figura 21 extraído de Gonzaga de Oliveira e Chagas (2014), renumera-se, a cada iteração i , o vértice com o menor grau, ao considerar, como candidatos, os vértices adjacentes a vértices já renumerados e também os vértices adjacentes a esses vértices. Nesse caso, para o cálculo do grau de um vértice, são desconsideradas as adjacências a vértices renumerados ou a vértices adjacentes a vértices renumerados.

Conforme descrito em Gonzaga de Oliveira e Chagas (2014), a heurística de Snay (1976) recebe o vértice v inicial e referencia-se esse vértice na primeira entrada da renumeração S (linha 2). Nas linhas 3 e 4, estabelecem-se como vazios os conjuntos de vértices esperançosos (H), que corresponde aos vértices adjacentes a vértices já renumerados, e candidatos. O vértice com menor grau é escolhido no laço de repetição das linhas 5 a 25, sem considerar as adjacências a vértices não processados e não esperançosos. No início da iteração i , na linha 6, os novos vértices esperançosos são os vértices não numerados adjacentes ao último vértice renumerado. Nas linhas

Algoritmo: Etapa de renumeração da heurística de Snay (1976).

Entrada: grafo conexo $G = (V, E)$ e um vértice inicial $v \in V$.
Saída: vetor S para V , com $|V|$ entradas $s(1), s(2), \dots, s(|V|)$.

```

1 início
2    $S(1) \leftarrow v$ ; // inicia-se a renumeração pelo vértice inicial
3    $H \leftarrow \emptyset$ ; // inicializa-se os conjuntos de vértices
4    $candidatos \leftarrow \emptyset$ ; // esperançosos e candidatos como vazios
5   para (  $i \leftarrow 2; i \leq |V|; i \leftarrow i + 1$  ) faça
        // vértices não renumerados adjacentes a  $s(i-1)$ 
        // tornam-se vértices esperançosos e candidatos
6    $NovosEsperanosos \leftarrow Adj(G, s[i-1]) - \{s(1), s(2), \dots, s(i-1)\}$ ;
7    $H \leftarrow H \cup NovosEsperanosos$ ;
8    $candidatos \leftarrow candidatos \cup NovosEsperanosos$ ;
9   para cada (  $u \in NovosEsperanosos$  ) faça
        // torna candidatos os vértices não processados do
        // segundo nível de adjacência de  $s(i-1)$ 
10     $candidatos \leftarrow candidatos \cup (Adj(G, u) - \{s(1), s(2), \dots, s(i-1)\})$ ;
11  fim-para-cada;
12   $min \leftarrow +\infty$ ;
13   $v_{min} \leftarrow \emptyset$ ;
        // escolhe-se o candidato  $w$  com menor grau, sem contar as
        // adjacências a não esperançosos e a não processados
14  para cada (  $w \in candidatos$  ) faça
        // número de vértices não processados e
        // não esperançosos adjacentes a  $w$ 
15   $grauRel_w \leftarrow |Adj(G, u) - (\{s(1), s(2), \dots, s(i-1)\} \cup H)|$ ;
        // os vértices esperançosos são favorecidos
        // em relação aos demais vértices candidatos
16  se (  $w \in H$  ) então  $grauRel_w \leftarrow grauRel_w - 1$ ;
17  se (  $grauRel_w < min$  ) então
18     $min \leftarrow grauRel_w$ ;
19     $v_{min} \leftarrow w$ ;
20  fim-se;
21  fim-para-cada;
        // inclui o candidato escolhido na sequência
22   $s(i) \leftarrow v_{min}$ ;
        // se o candidato escolhido era vértice esperançoso, retira-o
        // de  $H$ 
23   $H \leftarrow H - \{v_{min}\}$ ;
        // o vértice escolhido é retirado também do conjunto
24   $candidatos \leftarrow candidatos - \{v_{min}\}$ ; // de candidatos
25  fim-para;
26  retorna  $S$ .
27 fim.
```

Figura 21 Pseudocódigo da etapa de renumeração da heurística de Snay
 Fonte: Gonzaga de Oliveira e Chagas (2014)

7 e 8, respectivamente, o conjunto de vértices esperançosos (H) e o conjunto

de vértices candidatos são acrescidos desses novos vértices esperançosos. Nas linhas 9 a 11, o conjunto de vértices candidatos é acrescido dos vértices adjacentes dos novos vértices esperançosos. A limitação dos vértices candidatos, conforme observado por Gonzaga de Oliveira e Chagas (2014), é para tornar o algoritmo mais rápido do que considerar todos os vértices não renumerados como candidatos.

Na estrutura de repetição nas linhas 14 a 21, o vértice candidato escolhido para ser renumerado em uma iteração é o vértice que possui o menor grau, calculado na linha 15 sem considerar as adjacências a vértices não esperançosos e não processados. Além disso, os vértices esperançosos são favorecidos em relação aos demais vértices candidatos, na linha 16. O vértice v_{min} a ser renumerado na iteração i é referenciado na entrada $s(i)$, na linha 22. Após ser renumerado, o vértice v_{min} deixa de ser tanto um vértice (na linha 23), como um vértice candidato (linha 24). O algoritmo retorna a renumeração S na linha 26.

Como pode ser observado, na primeira etapa da heurística de Snay (1976), são escolhidos 10 vértices pseudoperiféricos para iniciar a renumeração na segunda etapa heurística. A necessidade de se executar a segunda etapa para cada um dos 10 vértices, para que se possa escolher renumeração que com o menor valor de *profile*, prejudica significativamente o custo computacional da heurística de Snay (1976). Diante disso, como as heurísticas devem possuir baixo custo computacional para possibilitar a redução do custo computacional da resolução de sistemas de equações lineares, foi proposta, neste trabalho, a utilização do algoritmo de George e Liu (1979) na primeira etapa da heurística de Snay (1976). O algoritmo de George e Liu (1979) é um algoritmo clássico para a determinação de

um vértice pseudoperiférico, e encontra-se implementado junto ao RCM (GEORGE, 1971) no MATLAB. Com isso, tem-se a necessidade de executar a segunda etapa da heurística de Snay (1976) apenas uma vez. Essa nova heurística é denominada de Snay-GL.

Além disso, outra modificação foi proposta para a heurística de Snay (1976). Ao escolher um vértice v para ser renumerado, inserem-se os vértices adjacentes ao vértice v na lista de candidatos, bem como os vértices adjacentes aos vértices adjacentes ao vértice v . É possível notar que são tornados candidatos os vértices do primeiro e segundo nível da estrutura de nível enraizada em v . Foi testada, neste trabalho, a possibilidade de se inserir os vértices até um nível genérico n . Porém, verificou-se que quanto maior o valor de n piores eram as renumerações produzidas em relação à redução de *profile*, pois vértices distantes recebiam numerações próximas, e isso causa um aumento no valor de *profile*. Assim, essa modificação foi descartada e não está contemplada na heurística Snay-GL.

4 METODOLOGIA

Neste capítulo, apresentam-se os procedimentos metodológicos adotados neste trabalho. As instâncias utilizadas nas simulações computacionais são detalhadas na subseção 4.1. Na subseção, são apresentados os detalhes de implementação dos condicionadores utilizados com o MGC. Por fim, apresentam-se, na subseção 4.3, os detalhes das implementações das heurísticas utilizadas no desenvolvimento deste trabalho, as especificações das máquinas e as ferramentas utilizadas nas simulações computacionais.

4.1 Instâncias utilizadas

Nesta seção, são apresentadas as instâncias de matrizes e de SELs utilizadas neste trabalho. Para os testes com a resolução de SELs, foram utilizados quatro conjuntos de instâncias. O primeiro conjunto é composto de SELs oriundos da discretização da equação da condução do calor e é detalhado na subseção 4.1.1. No segundo e terceiro conjuntos, foram utilizados SELs oriundos da discretização da equação de Laplace, e esses conjuntos são apresentados na subseção 4.1.2. O quarto conjunto contém instâncias da coleção de matrizes esparsas da Universidade da Flórida, oriundas de diversas áreas de aplicação, conforme mostrado na seção 4.1.3.

4.1.1 Instâncias oriundas da discretização da equação da condução do calor

Foram utilizados oito SELs (de 4.846 a 965.545 incógnitas), oriundos da discretização da equação da condução do calor, gerados pelo projeto

computacional de Gonzaga de Oliveira, Oliveira e Chagas (2015). Nesse projeto computacional, foram utilizadas as condições de contorno de Dirichlet e o domínio discretizado foi um quadrado unitário, com os valores das condições de contorno nos lados superior, esquerdo e direito iguais a dez e, no lado inferior, igual a zero. Ainda, Gonzaga de Oliveira, Oliveira e Chagas (2015) utilizaram o método dos volumes finitos para a discretização da equação da condução do calor, em que as malhas são compostas pelo refinamento de *Delaunay* e diagramas de *Voronoi*. A equação da condução do calor é uma equação parcial parabólica e é definida por $\frac{\partial}{\partial t}(\phi\rho) = \nabla \cdot \left(\frac{k\nabla}{c_p}\phi\right)$, em que t é o tempo, ρ é a massa volumétrica de um corpo sólido, ϕ é a temperatura, k é a condutibilidade térmica e c_p é o calor específico.

4.1.2 Instâncias oriundas da discretização da equação de Laplace

Os SELs oriundos da discretização da equação de Laplace foram gerados pelo projeto computacional de Gonzaga de Oliveira, Kischinhevsky e Tavares (2013). Assim como os SELs oriundos da discretização da equação da condução do calor, a discretização da equação de Laplace também foi realizada pelo método dos volumes finitos. A equação de Laplace é uma equação diferencial parcial elíptica e é definida por $\nabla^2\phi = 0$ em $\Omega \in \mathbb{R}^2$, $\phi = f$ em $\partial\Omega$, em que ϕ é a variável dependente da equação diferencial parcial, Ω é um domínio limitado em \mathbb{R}^2 e f é uma função suave definida no contorno $\delta\Omega$. Gonzaga de Oliveira, Kischinhevsky e Tavares (2013) utilizaram o quadrado unitário como domínio. Foram utilizadas as condições de contorno definidas nos lados superior, inferior e esquerdo com um valor definido único f e no lado direito com um valor diferente.

As instâncias de SELs oriundas da discretização da equação de Laplace, foram divididas em dois conjuntos. No primeiro conjunto, têm-se os SELs gerados de forma aleatória. Esse conjunto possui sete SELs de 7.322 a 277.118 incógnitas. No segundo conjunto, têm-se os SELs gerados na ordem da curva de Sierpiński modificada (VELHO; FIGUEIREDO; GOMES, 1999; GONZAGA DE OLIVEIRA; KISCHINHEVSKY, 2008). Esse conjunto possui oito SELs de 16.922 a 1.015.004 incógnitas.

4.1.3 Instâncias da *University of Florida sparse-matrix Collection*

Onze sistemas de equações lineares, com matrizes de coeficientes simétricas e positivo-definidas, pertencentes ao conjunto de dados da *University of Florida sparse-matrix collection* (DAVIS; HU, 2011) foram testados neste trabalho. Na tabela 2, são apresentadas as características dessas instâncias. Nas figuras 22 e 23, são mostradas as matrizes de coeficientes e as visualizações dos grafos relacionados com essas instâncias. Para alguns sistemas de equações lineares, há mais de um vetor de termos independentes nesse conjunto de instâncias. Como o objetivo deste trabalho é a avaliação de heurísticas para redução de *profile*, e essas heurísticas são aplicadas apenas na matriz de coeficientes, foi considerado apenas o primeiro vetor de termos independentes dos sistemas de equações lineares que possuem mais de um vetor de termos independentes.

4.2 Precondicionadores utilizados com o MGC

Segundo Saad (2003), condicionamento é um fator chave para o sucesso da aplicação dos métodos iterativos baseados no subespaço de

Tabela 2 11 SELs do conjunto de dados *The University of Florida sparse-matrix Collection* cujas matrizes de coeficientes são simétricas e positivo-definidas.

Nome	Tamanho	β	<i>profile</i>	Densidade	Tipo de problema
<i>nasa1824</i>	1824	239	205547	1,18%	estrutural
<i>nasa2910</i>	2910	859	525745	2,06%	estrutural
<i>sts4098</i>	4098	3323	5217389	0,43%	estrutural ¹
<i>nasa4707</i>	4704	423	917562	0,47%	estrutural
<i>Pres_Poisson</i>	14822	12583	9789525	0,33%	dinâmica de fluido computacional
<i>olafu</i>	16146	593	4951980	0,39%	estrutural
<i>raefsky4</i>	19779	11786	19611188	0,34%	estrutural
<i>nasasrb</i>	54870	893	20311330	0,09%	estrutural
<i>thermal1</i>	82654	80916	175625317	0,01%	térmico
<i>2cubes_sphere</i>	101492	100407	483241271	0,02%	eletromagnetismo ¹
<i>offshore</i>	259789	237738	3588201815	0,01%	eletromagnetismo ¹

¹ Método dos elementos finitos.

Krylov, como é o caso do MGC, devido ao aumento da eficiência e da robustez que o condicionamento proporciona. Neste trabalho, foram utilizados dois condicionadores: o de Jacobi e a fatoração incompleta de Cholesky.

O condicionador de Jacobi é trivial, é consiste na matriz inversa da matriz diagonal extraída da matriz de coeficientes do sistema de equações lineares a ser condicionado. Dessa maneira, o condicionador de Jacobi foi implementado juntamente com o MGC. Esse MGC condicionado pelo método de Jacobi é denominado de MGCPJ neste trabalho.

Em relação à fatoração incompleta de Cholesky, neste trabalho foi implementada a fatoração com zero *fill-in* proposta por Golub e Van Loan (1996). Trata-se da fatoração de Cholesky em $A = LL^T$ com a manutenção dos elementos nulos de A em L e, conseqüentemente, em L^T . A geração do condicionador a partir da fatoração incompleta de Cholesky com zero

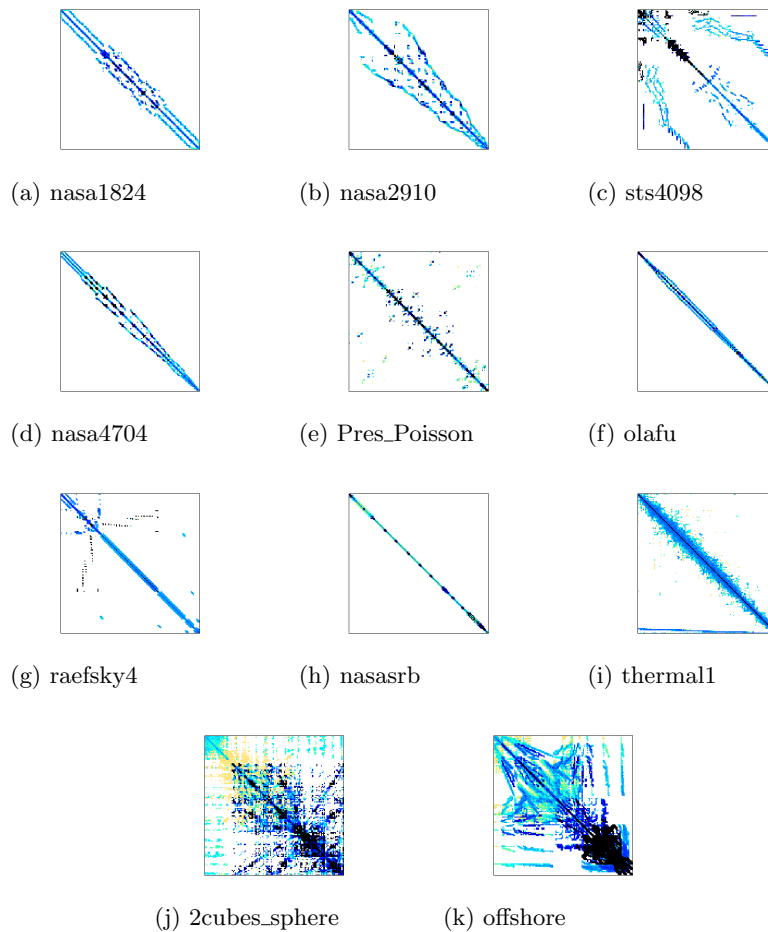


Figura 22 Imagens das matrizes de coeficientes das instâncias da base *The University of Florida sparse-matrix Collection*.

Fonte: Hu (2016)

fill-in foi implementada separadamente do MGC. O MGC, preconditionado pela fatoração incompleta de Cholesky com zero *fill-in*, é denominado de MGC-IC0 neste trabalho.

Golub e Van Loan (1996) afirmaram que essa fatoração pode não funcionar para algumas instâncias. Segundo Mantueffel (1980), uma

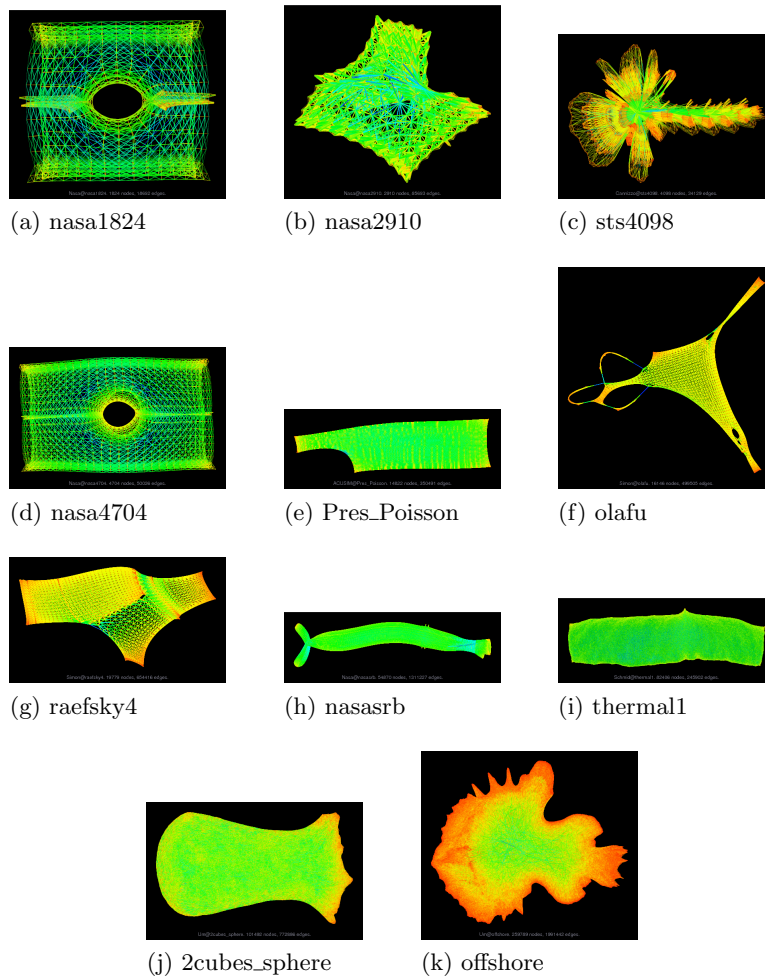


Figura 23 Imagens dos grafos relacionados às matrizes de coeficientes das instâncias da base *The University of Florida sparse-matrix Collection*.
 Fonte: Hu (2016)

condição suficiente para a estabilidade da fatoração incompleta de Cholesky é que a matriz de coeficientes deve ser positivo-definida e possuir dominância da diagonal principal. Diz-se que uma matriz quadrada $A_{n \times n}$ possui dominância da diagonal principal quando ocorre que $a_{ii} \geq \sum_{j=1, j \neq i}^n a_{ij}$, $\forall i \in [1, n]$. Quando essa propriedade não é atendida, há a possibilidade da

existência de pivôs nulos ou negativos durante a fatoração, causando erro de execução pelo cálculo de raiz quadrada de número negativo ou divisão por zero.

Ao examinar as instâncias utilizadas nas simulações, verificou-se que as instâncias da *University of Florida sparse-matrix Collection* não possuíam dominância da diagonal principal. Nas execuções do MGC-IC0 nessas instâncias, ocorreram pivôs negativos em 8 das 11 instâncias selecionadas. Dessa maneira, foi necessário modificar a fatoração incompleta de Cholesky de Golub e Van Loan (1996) de maneira a contornar esse problema.

No trabalho de Benzi (2002), são identificadas algumas estratégias para corrigir esse problema dos pivôs não positivos. A estratégia mais simples é substituir o valor do pivô não positivo para um valor positivo qualquer. Essa estratégia foi proposta por Kershaw (1978), e sua justificativa é que há poucos pivôs não positivos e, portanto, a substituição arbitrária dos mesmos é pouco prejudicial à qualidade do preconditionador a ser gerado. Segundo Benzi (2002), a escolha do número positivo para fazer a substituição não é simples. Outra estratégia é o deslocamento da diagonal principal, proposto por Mantueffel (1980). Esse deslocamento consiste em, antes da fatoração, construir uma matriz $\tilde{A} = A + \alpha \cdot \text{diag}(A)$, de modo que a condição da dominância da diagonal principal seja atendida. A desvantagem dessa estratégia é o custo computacional necessário para se determinar, empiricamente, o valor de α . Mantueffel (1980) propôs que o valor inicial de α seja 0 e, sempre que a fatoração incompleta de Cholesky falhar, deve-se incrementar o valor de α e reiniciar a fatoração. É garantido que existe um valor α^* para α para o qual a dominância da

diagonal seja alcançada. Ainda, Benzi (2002) identificou outras estratégias ((AJIZ; JENNINGS, 1984), (BENZI; TUMA, 2003) e (TISMENETSKY, 1991)), porém essas estratégias apresentam alto custo computacional.

Neste trabalho, foi utilizada a estratégia de Kershaw (1978), por ser a estratégia com o menor custo computacional entre as estratégias identificadas por Benzi (2002). A escolha do valor positivo, para substituir o valor do pivô negativo, foi o oposto do valor do pivô negativo, uma vez que não ocorreram pivôs nulos nas simulações.

4.3 Detalhes de implementação

Nesta seção, são apresentados os detalhes das implementações realizadas. Para o desenvolvimento deste trabalho, utilizou-se como base o projeto *LinearSystemSolver* desenvolvido e mostrado em detalhes por Chagas (2015). Nesse projeto computacional, haviam sido implementadas as heurísticas para redução de largura de banda selecionadas nas revisões sistemáticas realizadas por Chagas e Gonzaga de Oliveira (2015) e Gonzaga de Oliveira e Chagas (2015a).

A heurística RCM-GL (GEORGE; LIU, 1981), selecionada como uma das possíveis melhores heurísticas para redução de *profile* com baixo custo computacional, já fazia parte do projeto computacional. Para mais detalhes sobre o projeto computacional *LinearSystemSolver*, veja Chagas (2015, p.74-77). Os autores das heurísticas MPG (MEDEIROS; PIMENTA; GOLDENBERG, 1993), NSloan (KUMFERT e POTHEN, 1997), Sloan-MGPS (REID e SCOTT, 1999), Hu-Scott (HU; SCOTT, 2001), de Snay (1976) e de Sloan (1989), e dos métodos de trocas de Reid e Scott (2002), foram consultados sobre a possibilidade de disponibilização dos códigos-

fonte ou dos programas executáveis das heurísticas. Alguns autores não responderam e outros responderam que não tinham mais acesso ao código fonte ou ao programa executável da heurística. Com isso, essas heurísticas foram implementadas neste trabalho na linguagem de programação C++ e acrescentadas ao projeto computacional de Chagas (2015). É importante ressaltar que o objetivo deste trabalho não foi superar os resultados das implementações originais das heurísticas, mas obter implementações razoavelmente eficientes para possibilitar a comparação dos resultados dessas heurísticas com os resultados da heurística proposta neste trabalho. Por outro lado, as implementações dessas heurísticas foram testadas e calibradas a partir da garantia de que as implementações das heurísticas eram compatíveis com os algoritmos propostos originalmente.

Especificamente, os resultados obtidos pelas implementações em C++ das heurísticas foram comparados com os resultados publicados originalmente pelos autores. As heurísticas NSloan (KUMFERT e POTHEN, 1997), Sloan-MGPS (REID e SCOTT, 1999), Hu-Scott (HU; SCOTT, 2001) e de Sloan (1989) possuem parâmetros importantes que podem alterar os resultados. Investigações exploratórias foram realizadas para estabelecer os valores desses parâmetros mas, de maneira geral, os valores que resultaram nos melhores resultados foram aqueles sugeridos pelos autores: $w_1 = 1$ e $w_2 = 2$ (associados com os critérios global e local, relacionado com a distância de cada vértice até o vértice final e e relacionado com o grau de cada vértice, respectivamente) para as heurísticas Sloan-MGPS (REID e SCOTT, 1999), Hu-Scott (HU; SCOTT, 2001) e de (SLOAN, 1989), ; e $w_1 = 2$ e $w_2 = 1$ para a heurística NSloan (KUMFERT e POTHEN, 1997). Foram seguidos rigorosamente as recomendações

presentes no trabalho de Sloan (1989) e também as recomendações descritas em <http://www.hsl.rl.ac.uk/archive/specs/mc40.pdf> (HSL, 2016), em um código-fonte na linguagem de programação Fortran, para implementar a heurística Sloan na linguagem de programação C++. Da mesma maneira, foi estudado o código-fonte da heurística Sloan-MGPS na linguagem de programação Fortran, disponível em <http://www.hsl.rl.ac.uk/catalogue/mc60.html> (HSL, 2016), para implementar essa heurística na linguagem de programação C++. Os resultados das versões das heurísticas Sloan-MGPS e de Sloan (1989) na linguagem de programação C++, em relação à redução de *profile*, foram melhores que os resultados mostrados por Reid e Scott (1999). Os resultados da versão da heurística NSloan na linguagem de programação C++, em relação à redução de *profile*, foram similares aos resultados obtidos pela versão original da heurística (KUMFERT e POTHEN, 1997). Os resultados da versão da heurística MPG na linguagem de programação C++, em relação à redução de *profile*, foram melhores do que os resultados obtidos pela versão original da heurística (MEDEIROS; PIMENTA; GOLDENBERG, 1993). Em todos esses casos de comparações com os resultados originais, foram utilizadas as mesmas instâncias testadas nos trabalhos originais.

A heurística de Hu e Scott (2001) é mais complicada de ser implementada do que as demais heurísticas. Essa heurística foi implementada na linguagem de programação C++, mas os resultados dessa implementação foram inferiores em relação aos resultados mostrados por Hu e Scott (2001). A implementação dessa heurística na linguagem de programação C++ realizada neste trabalho não é uma implementação

razoavelmente eficiente dessa heurística. Porém, em investigações exploratórias, percebeu-se que essa heurística é uma heurística com alto custo computacional, principalmente por realizar multiplicações de matrizes. Então, não foram realizados esforços de programação para se obter uma implementação eficiente dessa heurística pois, certamente, essa heurística não pode ser competitiva com as demais heurísticas testadas quando aplicadas para reduzir o custo computacional de um método iterativo de resolução de sistemas de equações lineares. Como descrito, um algoritmo de reordenamento deve reduzir significativamente a largura de banda ou o *profile* a um custo computacional baixo; esse algoritmo não pode possuir alto custo computacional quando comparado a outras heurísticas para essa tarefa.

4.4 Ferramentas utilizadas

Utilizou-se a IDE *Netbeans* 8.0.2 para auxiliar na escrita dos códigos em linguagem *C++*. O compilador utilizado foi o *G++*, versão 4.8.2. Além disso, utilizou-se a biblioteca *GNU Multiple Precision Floating-point computations with correct-Rounding* (MPFR) para se obter maior precisão nos cálculos na resolução dos SELs. Utilizou-se precisão de 512 *bits* na biblioteca *GNU MPFR*, com exceção das simulações com instâncias relacionadas à equação de Laplace em que a precisão utilizada foi de 256 *bits*.

Na tabela 3, são mostradas as características de cada máquina utilizada para realizar as simulações computacionais. O MGC preconditionado e as instâncias que fizeram parte das simulações, executadas em cada uma das máquinas, foram:

- Máquina 1: sistemas de equações lineares originados das discretizações da equação de Laplace por volumes finitos (GONZAGA DE OLIVEIRA; KISCHINHEVSKY; TAVARES, 2013), ordenados de forma aleatória e resolvidos pelo MGCPJ e pelo MGC-IC0; as instâncias *nasa1824*, *nasa2910* e *nasasrb*, do conjunto de dados da *University of Florida sparse-matrix Collection*, resolvidas pelo MGCPJ; e a instância *olafu*, do conjunto de dados da *University of Florida sparse-matrix Collection*, resolvida pelo MGC-IC0;
- Máquina 2: sistemas de equações lineares originados das discretizações da equação de Laplace por volumes finitos (GONZAGA DE OLIVEIRA; KISCHINHEVSKY; TAVARES, 2013) e resolvidos pelo MGCPJ e pelo MGC-IC0. Esses sistemas de equações lineares foram originalmente ordenados pela sequência dada pela curva de Sierpiński modificada;
- Máquina 3: as instâncias *olafu* e *thermal1*, do conjunto de dados da *University of Florida sparse-matrix Collection*, resolvidas pelo MGCPJ, e as instâncias *nasa1824*, *nasa2910*, *nasasrb*, *sts4098*, *nasa4704*, *Pres_Poisson*, *raefsky4*, e *thermal1*, do conjunto de dados da *University of Florida sparse-matrix Collection*, resolvidas pelo MGC-IC0;
- Máquina 4: sistemas de equações lineares originados das discretizações da equação da condução do calor por volumes finitos (GONZAGA DE OLIVEIRA; OLIVEIRA; CHAGAS, 2015), resolvidos pelo MGCPJ e pelo MGC-IC0;
- Máquina 5: as instâncias *sts4098*, *nasa4704*, *Pres_Poisson*, e

raefsky4, do conjunto de dados da *University of Florida sparse-matrix Collection*, resolvidas pelo MGCPJ;

- Máquina 6: as instâncias *2cubes_sphere* e *offshore*, do conjunto de dados da *University of Florida sparse-matrix Collection*, resolvidas pelo MGCPJ e pelo MGC-IC0.

Tabela 3 Características das máquinas utilizadas para realizar as simulações computacionais.

Máquina	Processador	Quantidade de mem. <i>cache</i>	Quantidade de mem. principal	kernel Linux
M1	Intel® Core™ i3-2120 CPU 3.30GHz × 4	3 MB	8 GB DDR3 1.333GHz	3.13.0-39-generic
M2	Intel Xeon™ E5620 CPU 2.40GHz × 8	12 MB	24 GB DDR3 1.333GHz	3.13.0-44-generic
M3	Intel® Core™ i5-3570 CPU 3.40GHz × 4	6 MB	8 GB DDR3 1.333GHz	3.13.0-37-generic
M4	Intel® Core™ I3-550 CPU 3.20GHz × 4	4 MB	16 GB DDR3 1.333GHz	3.13.0-39-generic
M5	Intel® Core™ i7-4510U CPU 2.00GHz × 4	4 MB	8 GB DDR3 1.6GHz	3.16.0-23-generic
M6	Intel® Core™ i7-4790K CPU 4.00GHz × 8	8 MB	12 GB DDR3 1.6GHz	3.19.0-31-generic

5 SIMULAÇÕES COM RESOLUÇÃO DE SISTEMAS DE EQUAÇÕES LINEARES

Neste capítulo, mostram-se os resultados dos testes com a resolução de SELs pelo MGC preconditionado pelo método de Jacobi (MGCPJ) e pelo MGC preconditionado pela fatoração incompleta de Cholesky com zero *fill-in* (MGC-IC0). Para isso, foram utilizadas quatro conjuntos de SELs, conforme descrito na seção 4.1. Para cada conjunto de instâncias, resolveu-se os SELs pelo MGCPJ e pelo MGC-IC0 com e sem as reduções de largura de banda e de *profile*. Essas reduções são obtidas pelas melhores heurísticas para redução de largura de banda com baixo custo computacional, selecionadas por Chagas e Gonzaga de Oliveira (2015), Gonzaga de Oliveira e Chagas (2015a), e pelas heurísticas para redução de *profile* com baixo custo computacional selecionadas por Bernardes e Gonzaga de Oliveira (2015). Cada heurística e cada MGC pré-condicionado foram executados três vezes em cada instância. Ainda, é válido ressaltar que todas as execuções foram sequenciais.

Nas seções 5.1 e 5.2, são mostrados os resultados obtidos pelo MGCPJ e pelo MGC-IC0, respectivamente. Na seção 5.3, é mostrado um resumo dos resultados do MGCPJ e do MGC-IC0 em cada conjunto de instâncias.

5.1 Resultados de resoluções de sistemas de equações lineares com o MGCPJ

Nesta seção, são apresentados os resultados das resoluções dos sistemas de equações lineares dos conjuntos de instâncias apresentados na seção 4.1, por meio do MGCPJ, com e sem as reduções de largura de banda

e de *profile* dadas pela aplicação das heurísticas testadas. Nas Subseções 5.1.1 e 5.1.2, são mostrados os resultados das resoluções dos sistemas de equações lineares oriundos da discretização das equações da condução do calor e de Laplace, respectivamente. Na subseção 5.1.3, são mostrados os resultados das resoluções dos sistemas de equações lineares da *University of Florida sparse-matrix collection*.

As tabelas mostradas nesta seção, com os resultados do MGCPJ, são estruturadas como a seguir. Na primeira coluna, é exibida a dimensão n da matriz de coeficientes do respectivo sistema de equações lineares (número de vértices do grafo) ou o nome da instância da *University of Florida sparse-matrix collection*. Na segunda coluna, há o nome da heurística utilizada para reduções de *profile* e largura de banda. Os resultados em relação ao *profile* e à largura de banda são mostrados na terceira e quarta colunas, respectivamente. Na quinta e sexta colunas, são apresentados, respectivamente, os resultados das heurísticas em relação ao custo computacional, em segundos, e a ocupação de memória, em *mebibytes*. O número de iterações e o custo computacional, em segundos, do MGCPJ são mostrados na sétima e oitava colunas, respectivamente. Os desvios padrões e os coeficientes de variação, referentes ao custo computacional do MGCPJ, são mostrados na nona e décima colunas, respectivamente. Note que é mostrado “Sem renumeração” na primeira linha do conjunto de simulações para cada instância. Isso significa que nenhuma heurística de reordenação foi utilizada. Com esse resultado, é possível verificar o *speedup* obtido por meio da utilização da heurística de reordenação, que consiste na divisão do tempo de execução do MGCPJ somado ao tempo de execução da heurística pelo tempo de execução do MGCPJ sem a utilização de heurística

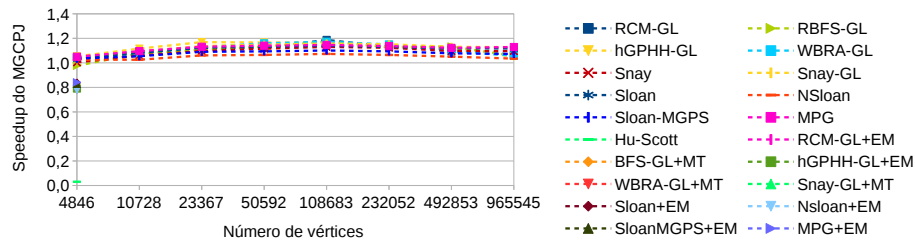


Figura 24 Speedup do MGCPJ obtido com aplicação das heurísticas, nas instâncias originadas de discretizações da equação da condução do calor por volumes finitos, mostrados nas tabelas 4, 5 e 6. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência.

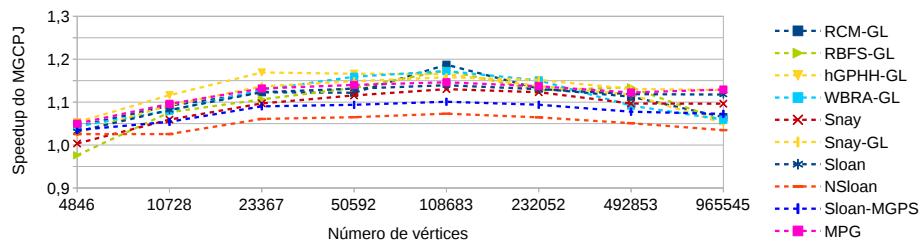


Figura 25 Speedup do MGCPJ obtido com aplicação das 10 heurísticas com os melhores resultados, nas instâncias originadas de discretizações da equação da condução do calor por volumes finitos, mostrados nas tabelas 4, 5 e 6. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência.

para reordenação, e é mostrado na última coluna.

5.1.1 Instâncias oriundas da discretização da equação da condução do calor

Os resultados médios obtidos pelo MGCPJ, nas instâncias oriundas da discretização da equação da condução calor, são apresentados nas tabelas 4, 5 e 6. Esses resultados são ilustrados nas figuras 24, 25 e 26 (sem os resultados das heurísticas que obtiveram os piores resultados nesse conjunto de instâncias).

Tabela 4 Resolução de sistemas de equações lineares (de até 23.637 incógnitas) pelo MGCPJ, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de *profile*. Essas simulações foram realizadas na máquina M4 e esses sistemas de equações lineares foram originados de discretizações da equação da condução do calor por volumes finitos.

n	Heurística	β	<i>profile</i>	Heurística		MGCPJ		σ	C_v	Speedup
				t(s)	m.(MiB)	iter.	t(s)			
4846	Without reordering	4769	9116750	-	-	319	9,97	0,04	0,46%	-
	RCM-GL	161	468110	0,005	0	319	9,55	0,19	2,01%	1,043
	RBFS-GL	163	467970	0,002	0	319	10,21	0,07	0,66%	0,976
	hGPHH-GL	161	518637	0,005	0	319	9,46	0,05	0,52%	1,053
	WBRA-GL	171	441298	0,063	1,03	319	9,49	0,04	0,46%	1,044
	Snay	1047	352116	0,284	0	319	9,65	0,01	0,14%	1,004
	Snay-GL	1123	364222	0,030	0	319	9,67	0,04	0,41%	1,028
	Sloan	774	297591	0,006	0	319	9,66	0,01	0,07%	1,031
	NSloan	608	472650	0,003	0	319	9,72	0,09	0,89%	1,025
	Sloan-MGPS	658	288047	0,023	0	319	9,61	0,02	0,18%	1,035
	MPG	1124	317356	0,006	0	319	9,49	0,01	0,11%	1,050
	Hu-Scott	3501	640372	317,169	100,46	319	9,73	0,02	0,22%	0,030
	RCM-GL+EM	476	358017	2,709	0	319	9,72	0,03	0,28%	0,802
	RBFS-GL+EM	504	378118	2,770	0	319	9,72	0,01	0,13%	0,798
	hGPHH-GL+EM	377	391204	2,965	0	319	9,64	0,02	0,16%	0,791
	WBRA-GL+MT	419	397060	3,019	0,43	319	9,75	0,01	0,13%	0,781
	Snay-GL+EM	1088	384289	2,464	0,09	319	9,56	0,01	0,09%	0,829
	Sloan+EM	919	268663	2,350	0	319	9,54	0,04	0,39%	0,839
	NSloan+EM	601	348643	3,056	0,09	319	9,73	0,01	0,05%	0,780
	SloanMGPS+EM	635	256907	2,390	0	319	9,64	0,05	0,55%	0,829
MPG+EM	1077	307725	2,374	0	319	9,47	0,01	0,09%	0,842	
10728	Without reordering	10626	45314579	-	-	462	37,35	0,36	0,96%	-
	RCM-GL	270	1579179	0,020	0	462	34,19	0,06	0,17%	1,092
	RBFS-GL	274	1590382	0,012	0	462	34,69	0,07	0,20%	1,076
	hGPHH-GL	275	1768638	0,020	0	462	33,44	0,04	0,12%	1,116
	WBRA-GL	306	1680767	0,220	2,58	462	34,36	0,11	0,32%	1,080
	Snay	1289	1064597	0,870	0,26	462	34,39	0,46	1,33%	1,059
	Snay-GL	1400	1137463	0,100	0,17	462	34,07	0,07	0,21%	1,093
	Sloan	1391	1012101	0,020	0	462	34,51	0,07	0,20%	1,082
	NSloan	890	1511606	0,014	0	462	36,41	0,05	0,13%	1,026
	Sloan-MGPS	1249	1023132	0,090	0	462	35,34	0,12	0,33%	1,054
MPG	2282	1097773	0,020	0	462	34,06	0,13	0,37%	1,096	
23367	Without reordering	23167	216212086	-	-	671	124,39	0,53	0,42%	-
	RCM-GL	313	4664523	0,070	0	671	110,69	2,01	0,82%	1,123
	RBFS-GL	314	4662697	0,047	0	671	112,37	2,34	2,04%	1,106
	hGPHH-GL	314	5129407	0,080	0	671	106,32	0,26	0,25%	1,169
	WBRA-GL	681	5618347	0,750	7,73	671	109,46	1,84	1,68%	1,129
	Snay	1824	3427644	2,960	0,69	671	110,43	1,94	1,76%	1,097
	Snay-GL	1810	3428553	0,290	0,26	671	109,14	0,11	0,09%	1,137
	Sloan	2565	3419063	0,070	0	671	110,67	0,06	0,05%	1,123
	NSloan	2057	5252921	0,051	0	671	117,18	0,05	0,05%	1,061
	Sloan-MGPS	2650	3614398	0,330	0	671	113,77	0,12	0,13%	1,090
MPG	3988	3775704	0,060	0	671	109,85	0,26	0,24%	1,132	

Tabela 5 Resolução de sistemas de equações lineares (de 50.592 a 232.052 incógnitas) pelo MGCPJ, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de *profile*. Essas simulações foram realizadas na máquina M4 e esses sistemas de equações lineares foram originados de discretizações da equação da condução do calor por volumes finitos.

n	Heurística	β	<i>profile</i>	Heurística		MGCPJ		σ	C_v	Speedup
				t(s)	m.(MiB)	iter.	t(s)			
50592	Without reordering	50461	1020411959	-	-	970	391,19	0,13	0,03%	-
	RCM-GL	647	17502999	0,26	0	970	348,44	10,71	3,07%	1,122
	RBFS-GL	650	17518537	0,19	0	970	345,01	6,23	1,81%	1,133
	hGPHH-GL	640	19173913	0,26	0	970	335,17	0,48	0,14%	1,166
	WBRA-GL	706	18363549	2,61	15,98	970	334,85	2,12	0,63%	1,159
	Snay	2941	10776787	10,80	1,29	970	339,83	0,37	0,11%	1,116
	Snay-GL	3039	10486508	1,06	0,52	970	339,47	0,22	0,06%	1,149
	Sloan	6025	11975091	0,23	0	970	345,52	0,31	0,09%	1,131
	NSloan	4314	16059399	0,09	0	970	367,24	0,25	0,07%	1,065
	Sloan-MGPS	3914	11784471	1,12	0	970	356,52	0,41	0,11%	1,094
MPG	9026	12000823	0,15	0,09	970	342,95	0,39	0,11%	1,140	
108683	Without reordering	108216	4725435534	-	-	1398	1238,76	29,02	2,34%	-
	RCM-GL	868	54290089	0,25	0	1398	1043,15	2,57	0,25%	1,187
	RBFS-GL	868	54309266	0,12	0	1398	1055,17	1,41	0,13%	1,174
	hGPHH-GL	868	59697870	0,26	0	1399	1062,85	5,69	0,53%	1,165
	WBRA-GL	754	54548762	7,98	35,70	1400	1095,00	1,59	0,14%	1,123
	Snay	4281	33664485	29,12	2,98	1398	1066,89	1,28	0,12%	1,130
	Snay-GL	4432	34768930	3,06	2,32	1398	1066,73	0,11	0,01%	1,158
	Sloan	10272	36886113	0,75	2,32	1398	1086,46	1,19	0,11%	1,139
	NSloan	7826	52122701	0,19	2,32	1398	1154,12	0,73	0,06%	1,073
	Sloan-MGPS	12656	38154730	4,18	2,32	1398	1121,04	0,35	0,03%	1,101
MPG	15172	38512718	0,34	2,32	1399	1080,37	0,57	0,05%	1,146	
232052	Without reordering	231672	21652820640	-	-	2034	3814,37	0,33	0,01%	-
	RCM-GL	1471	183551861	0,58	0	2039	3370,62	5,59	0,17%	1,131
	RBFS-GL	1471	183479875	0,31	0	2038	3382,99	5,79	0,17%	1,127
	hGPHH-GL	1471	200679371	0,61	0	2030	3352,42	20,39	0,61%	1,138
	WBRA-GL	1972	177829270	30,03	65,74	2029	3284,84	15,75	0,48%	1,151
	Snay	6212	102404593	94,28	6,19	2030	3301,87	4,88	0,15%	1,123
	Snay-GL	6531	105533138	9,89	4,21	2032	3301,56	4,04	0,12%	1,152
	Sloan	14640	119489765	2,46	4,13	2032	3372,41	5,53	0,16%	1,130
	NSloan	23577	174602136	0,44	4,13	2032	3582,52	6,59	0,18%	1,065
	Sloan-MGPS	26056	124108449	14,87	4,21	2032	3471,88	3,54	0,14%	1,094
MPG	22919	118186600	0,88	4,13	2031	3354,02	8,77	0,26%	1,137	

Tabela 6 Resolução de sistemas de equações lineares (de 492.853 e 965.545 incógnitas) pelo MGCPJ, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de *profile*. Essas simulações foram realizadas na máquina M4 e esses sistemas de equações lineares foram originados de discretizações da equação da condução do calor por volumes finitos.

n	Heurística	β	profile	Heurística		MGCPJ		σ	C_v	Speedup
				t(s)	m.(MiB)	iter.	t(s)			
492853	Without reordering	492100	97893937993	-	-	2927	11619,20	11,49	0,10%	-
	RCM-GL	1805	510308269	1,48	0	2925	10440,46	12,96	0,22%	1,113
	RBFS-GL	1811	510450511	0,93	0	2932	10241,11	8,86	0,09%	1,134
	hGPHH-GL	1814	557734280	1,57	0	2932	10395,07	16,07	0,15%	1,118
	WBRA-GL	2671	590763978	101,77	222,88	2935	10539,18	37,65	0,36%	1,092
	Snay	10273	329516129	323,58	12,97	2935	10270,53	26,60	0,26%	1,097
	Snay-GL	8858	313512518	29,92	10,07	2932	10246,57	6,46	0,06%	1,131
	Sloan	44208	373429562	8,42	9,99	2934	10378,44	16,51	0,16%	1,119
	NSloan	28222	510609978	1,19	9,93	2932	11054,62	41,74	0,38%	1,051
	Sloan-MGPS	43022	391323009	54,07	9,99	2928	10727,11	13,94	0,13%	1,078
	MPG	55654	387601624	2,64	9,96	2932	10348,92	2,15	0,02%	1,122
	965545	Without reordering	964827	377848438952	-	-	4358	34061,44	11,67	0,03%
RCM-GL		2201	1332260636	2,21	0	4346	31978,36	37,83	0,12%	1,065
RBFS-GL		2453	1447960230	1,14	0	4360	32247,29	39,08	0,12%	1,056
hGPHH-GL		2315	1495703784	2,37	0	4370	32392,77	15,64	0,05%	1,051
WBRA-GL		3603	1469674194	259,95	256,07	4362	31901,95	19,41	0,06%	1,059
Snay		13292	950057407	1049,95	26,09	4341	30016,38	205,23	0,68%	1,096
Snay-GL		14204	972342545	103,61	17,39	4346	30087,75	268,24	0,89%	1,128
Sloan		42459	1007346790	22,79	16,98	4365	30453,62	41,81	0,14%	1,118
NSloan		56571	1379371308	1,90	17,07	4362	32913,96	59,64	0,18%	1,035
Sloan-MGPS		60821	1007161537	166,78	17,07	4361	31589,23	77,27	0,24%	1,073
MPG		77152	982945106	6,02	17,24	4360	30154,23	60,63	0,22%	1,129

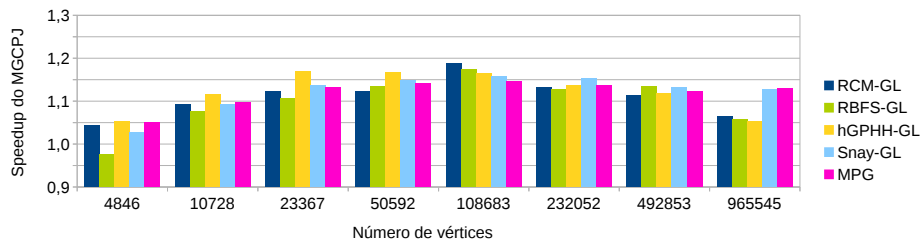


Figura 26 Speedup do MGCPJ obtido com aplicação das 5 heurísticas com os melhores resultados, nas instâncias originadas de discretizações da equação da condução do calor por volumes finitos, mostrados nas tabelas 4, 5 e 6.

Observa-se, com base nos resultados mostrados na tabela 4, que a heurística de Hu e Scott (2001), e a aplicação dos métodos de trocas de Reid e Scott (2002) em conjunto com as demais heurísticas, foram superadas por várias outras heurísticas quando aplicados à instância de 4.846 vértices, em relação à redução do custo computacional do MGCPJ. Ainda, é possível notar que os métodos de trocas foram efetivos na redução de *profile*, porém essa redução não foi proporcional à redução do custo computacional do MGCPJ. Dessa maneira, tanto a heurística de Hu e Scott (2001) como a aplicação dos métodos de trocas não foram testados nas demais simulações realizadas com o MGCPJ neste trabalho.

Na revisão sistemática realizada (BERNARDES; GONZAGA DE OLIVEIRA, 2015), a heurística de Hu e Scott (2001) foi selecionada como uma das possíveis melhores heurísticas para redução de *profile* pois não foi possível afirmar que os resultados dessa heurística foram superados pelos resultados de outra heurística. Além disso, é necessário ressaltar que os custos computacionais da heurística de Hu e Scott (2001) não foram mostrados no trabalho original. Com isso, após a verificação do custo alto de execução relacionado às operações de multiplicação de matrizes, e de uma ocupação de memória significativamente maior em relação à ocupação de memória das demais heurísticas, devido à geração de vários grafos intermediários na implementação realizada neste trabalho, a heurística de Hu e Scott (2001) não foi considerada como uma das possíveis melhores heurísticas de reordenação com *baixo custo computacional*.

Em relação aos métodos de trocas, os mesmos são aplicados de forma iterativa e alternada entre o método ascendente e o método descendente. Os resultados apresentados na tabela 4 foram obtidos com a utilização de

apenas uma iteração alternada, e não 5 iterações como foi proposto por Reid e Scott (2002). Com isso, não é possível diminuir ainda mais o número de iterações para se obter um custo computacional menor.

Nas tabelas 4, 5 e 6, é possível observar que a heurística RBFS-GL apresentou a menor valor de largura de banda pelo maior número de vezes, e que as heurísticas de Snay (1976) e de Sloan (1989) apresentaram o menor valor de *profile* pelo maior número de vezes. A heurística hGPHH-GL obteve o melhor resultado em relação à redução do custo computacional do MGCPJ nas instâncias de até 50.592 vértices. A heurística RCM-GL (GEORGE; LIU, 1981) obteve o melhor resultado em relação à redução do custo computacional do MGCPJ na instância de 108.683 vértices. As heurísticas MPG (MEDEIROS; PIMENTA; GOLDENBERG, 1993) e Snay-GL foram consideradas as melhores heurísticas para redução do custo de processamento do MGCPJ pois essas heurísticas estiveram entre as 4 heurísticas que obtiveram os melhores valores de *speedup* em quase todas as instâncias. Em particular, as heurísticas Snay-GL e MPG obtiveram os melhores resultados em relação à redução do custo computacional do MGCPJ nas instâncias de 232.052 e 965.545 vértices, respectivamente. Além disso, é possível verificar que a heurística que apresentou o menor custo computacional foi a heurística RBFS-GL, e a mesma obteve o melhor resultado em relação à redução do custo computacional do MGCPJ na instância de 492.853 vértices.

Com as reduções de largura de banda e de *profile*, ocorreu a redução do custo computacional do MGCPJ. Porém, a redução do custo computacional do MGCPJ não ocorreu de forma proporcional às reduções de largura de banda e de *profile*. Ainda, observa-se que não há variação

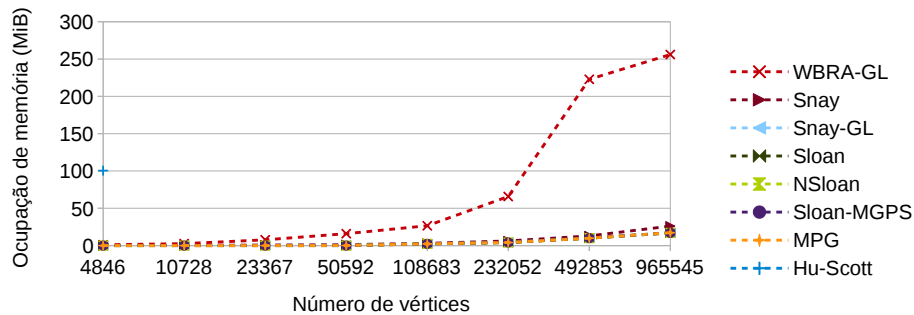


Figura 27 Quantidades de memória extra ocupadas pelas heurísticas WBRA-GL, Snay, Snay-GL, Sloan, NSloan, Sloan-MGPS, MPG e Hu-Scott, nas instâncias originadas de discretizações da equação da condução do calor por volumes finitos, mostradas nas tabelas 4, 5 e 6. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência.

significativa no número de iterações do MGCPJ com as reduções de largura de banda e de *profile*.

As heurísticas RCM-GL, RBFS-GL e hGPHH-GL são algoritmos *in-place*, isto é, não possuem ocupação extra de memória. Nas figuras 27 e 28 (sem as ocupações das heurísticas WBRA-GL e Hu-Scott, que foram as heurísticas com as maiores ocupações de memória), são mostrados os custos de armazenamento das outras 8 heurísticas. Ainda, nas figuras 29 e 30, são mostrados os tempos de execução, em segundos, das heurísticas para reduções de largura de banda e de *profile*.

5.1.2 Instâncias oriundas da discretização da equação de Laplace

Nas tabelas 7 e 8, são apresentados os resultados das resoluções dos SELs, oriundos da discretização da equação de Laplace e numerados de forma aleatória, com o MGCPJ. Nas tabelas 9 e 10, são apresentados os resultados das resoluções dos SELs, oriundos da discretização da equação de Laplace e numerados de pela curva de Sierpiński modificada, com o MGCPJ.

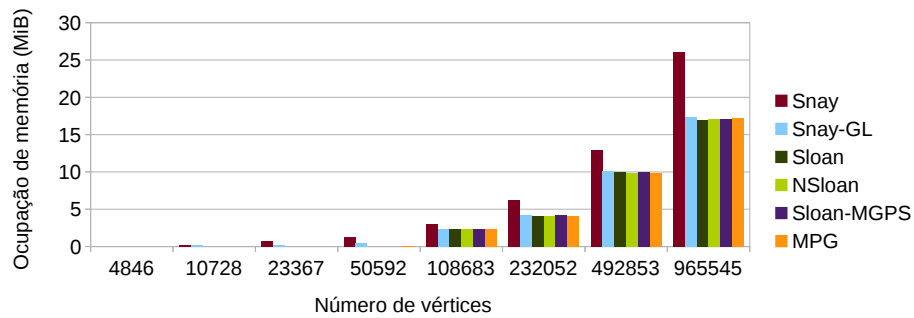


Figura 28 Quantidades de memória extra ocupadas pelas heurísticas Snay, Snay-GL, Sloan, NSloan, Sloan-MGPS e MPG, nas instâncias originadas de discretizações da equação da condução do calor por volumes finitos, mostradas nas tabelas 4, 5 e 6.

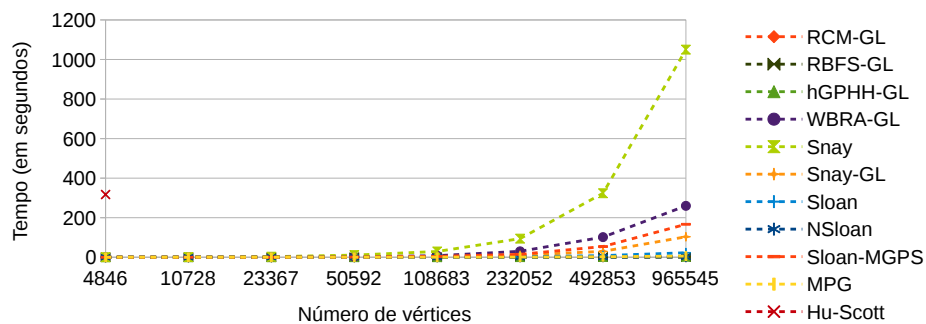


Figura 29 Tempos de execução, em segundos, das heurísticas RCM-GL, RBFS-GL, hGPHH-GL, WBRA-GL, Snay, Snay-GL, Sloan, NSloan, Sloan-MGPS, MPG e Hu-Scott nas instâncias oriundas da discretização da equação da condução do calor por volumes finitos. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência.

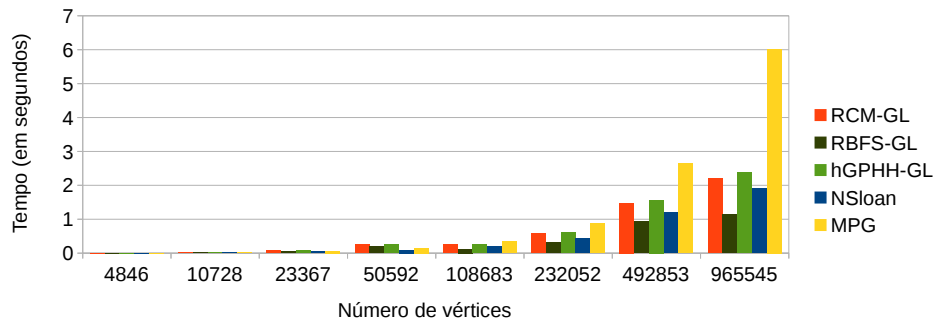


Figura 30 Tempos de execução, em segundos, das heurísticas RCM-GL, RBFS-GL, hGPHH-GL, NSloan e MPG nas instâncias oriundas da discretização da equação da condução do calor por volumes finitos.

Observa-se que houve um valor pequeno de *speedup* positivo ao utilizar a heurística hGPHH-GL na instância de 16.922 vértices (veja tabela 9). No entanto, valores negativos de *speedup* do MGCPJ foram obtidos ao aplicar as demais heurísticas para reduções de largura de banda e de *profile* nas demais instâncias.

Em particular, a heurística Snay foi testada apenas nas três menores instâncias de cada conjunto, pois os seus resultados, em relação à redução do custo computacional do MGC-IC0, foram superados pelos resultados da heurística Snay-GL, conforme pode ser verificado nas tabelas 7 e 9.

Como descrito, as heurísticas RCM-GL, RBFS-GL, e hGPHH-GL heurísticas são algoritmos *in-place*, e não possuem ocupação extra de memória. Nas figuras 31 e 32, são mostrados os custos de armazenamento das outras 7 heurísticas testadas nas instâncias oriundas da discretização da equação de Laplace com numeração das linhas dada de forma aleatória. Nas figuras 33 e 34, são mostrados os custos de armazenamento das outras 7 heurísticas testadas nas instâncias oriundas da discretização da equação de Laplace com numeração das linhas dada pela curva de Sierpiński modificada.

Tabela 7 Resolução de sistemas de equações lineares (de até 34.238 incógnitas) pelo MGCPJ, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de *profile*. Essas simulações foram realizadas na máquina M1 e esses sistemas de equações lineares foram originados de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada de forma aleatória.

n	Heurística	β	<i>profile</i>	Heurística		MGCPJ		σ	C_v	Speedup
				t(s)	m.(MiB)	iter.	t(s)			
7322	Sem renum.	7248	16083808	-	-	498	10,39	0,02	0,16%	-
	RCM-GL	80	396652	0,010	0	498	11,12	0,04	0,37%	0,934
	RBFS-GL	80	396709	0,003	0	498	12,54	0,07	0,55%	0,828
	hGPHH-GL	80	406461	0,010	0	498	11,18	0,03	0,29%	0,929
	WBRA-GL	152	411821	0,050	0,73	498	12,61	0,08	0,61%	0,821
	Snay	1859	447914	0,240	0,17	498	14,12	0,10	0,03%	0,724
	Snay-GL	1465	453745	0,030	0,26	498	14,27	0,12	0,83%	0,727
	Sloan	391	375254	0,010	0,17	498	14,21	0,05	0,38%	0,731
	NSloan	197	424476	0,040	0,17	498	14,50	0,01	0,07%	0,715
	Sloan-MGPS	312	374571	0,030	0,17	498	14,20	0,02	0,14%	0,730
	MPG	800	608671	0,010	0,17	498	14,52	0,03	0,17%	0,715
15944	Sem renum.	15902	76482022	-	-	745	33,74	0,21	0,63%	-
	RCM-GL	121	1149442	0,020	0	745	38,65	0,04	0,10%	0,873
	RBFS-GL	122	1149179	0,010	0	745	42,96	0,01	0,01%	0,785
	hGPHH-GL	124	1231692	0,020	0	745	38,56	0,29	0,76%	0,875
	WBRA-GL	175	1271533	0,110	1,29	745	36,08	0,15	0,42%	0,932
	Snay	5862	1586436	1,020	0,52	745	47,01	0,06	0,13%	0,702
	Snay-GL	6333	1634345	0,110	0,34	745	46,98	0,08	0,17%	0,717
	Sloan	484	982693	0,020	0,34	745	46,89	0,02	0,04%	0,719
	NSloan	218	1222337	0,020	0,26	745	48,95	0,05	0,11%	0,689
	Sloan-MGPS	481	1002661	0,100	0,26	745	47,30	0,14	0,30%	0,712
	MPG	1277	1612396	0,030	0,34	745	47,93	0,20	0,41%	0,704
34238	Sem renum.	34059	357518296	-	-	1069	105,36	0,41	0,39%	-
	RCM-GL	194	3411077	0,040	0	1069	113,75	0,90	0,79%	0,926
	RBFS-GL	192	3413781	0,020	0	1069	122,25	0,32	0,26%	0,862
	hGPHH-GL	192	3759478	0,040	0	1069	113,30	0,11	0,10%	0,930
	WBRA-GL	272	3750387	0,290	2,90	1069	113,50	0,54	0,48%	0,926
	Snay	21625	5150148	4,830	1,03	1069	145,24	0,53	0,36%	0,702
	Snay-GL	21625	5154351	0,480	0,86	1069	144,71	0,10	0,07%	0,726
	Sloan	917	2578022	0,060	0,77	1069	144,12	0,03	0,02%	0,731
	NSloan	357	3608666	0,030	0,86	1069	152,94	0,14	0,09%	0,689
	Sloan-MGPS	795	2671240	0,300	0,86	1069	145,66	0,07	0,05%	0,722
	MPG	2322	3986576	0,060	0,86	1069	146,09	0,04	0,03%	0,721

Simulações realizadas na máquina M1.

Tabela 8 Resolução de sistemas de equações lineares (de 75.542 a 277.118 incógnitas) pelo MGCPJ, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de *profile*. Essas simulações foram realizadas na máquina M1 e esses sistemas de equações lineares foram originados de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada de forma aleatória.

n	Heurística	β	<i>profile</i>	Heurística		MGCPJ		σ	C_v	Speedup
				t(s)	m.(MiB)	iter.	t(s)			
75542	Sem renum.	75490	1744941733	-	-	1540	327,95	0,38	0,11%	-
	RCM-GL	274	12086129	0,09	0	1540	361,89	0,62	0,17%	0,906
	RBFS-GL	273	12107123	0,04	0	1540	372,46	0,81	0,22%	0,880
	hGPHH-GL	277	13793938	0,10	0	1540	360,27	0,22	0,06%	0,910
	WBRA-GL	436	13766127	0,88	6,72	1540	372,62	1,02	0,27%	0,878
	Snay	47789	28972039	41,67	2,06	1540	468,18	0,61	0,13%	0,643
	Snay-GL	47833	28978947	3,80	1,03	1540	473,61	0,02	0,01%	0,687
	Sloan	1521	8981209	0,17	1,03	1540	471,57	0,21	0,04%	0,695
	NSloan	534	12805249	0,07	1,03	1540	507,60	0,04	0,01%	0,646
	Sloan-MGPS	1236	9245713	1,01	1,03	1540	480,93	0,19	0,04%	0,680
MPG	4020	14107424	0,15	1,03	1540	481,41	0,06	0,01%	0,681	
101780	Sem renum.	101583	3169282786	-	-	2173	630,79	0,36	0,06%	-
	RCM-GL	405	21399542	0,13	0	2173	682,61	1,81	0,26%	0,924
	RBFS-GL	404	21404400	0,06	0	2173	699,69	0,09	0,01%	0,901
	hGPHH-GL	407	24041332	0,14	0	2173	684,12	0,30	0,04%	0,922
	WBRA-GL	697	24026871	1,53	7,50	2173	698,40	0,32	0,05%	0,901
	Snay	64553	45830895	61,06	2,82	2173	886,46	0,12	0,01%	0,666
	Snay-GL	64282	45857748	5,55	2,15	2173	889,57	2,30	0,26%	0,705
	Sloan	8845	14909417	0,32	2,15	2173	882,19	2,60	0,29%	0,715
	NSloan	7602	21266761	0,09	2,15	2173	950,94	2,53	0,27%	0,663
	Sloan-MGPS	8420	15400014	1,91	2,15	2173	902,26	2,78	0,31%	0,698
MPG	10502	24115880	0,24	2,15	2173	905,61	2,50	0,28%	0,696	
192056	Sem renum.	191738	11329772559	-	-	2383	1304,62	0,72	0,06%	-
	RCM-GL	360	42578191	0,24	0	2382	1436,77	3,39	0,24%	0,908
	RBFS-GL	363	42634929	0,12	0	2383	1471,47	0,22	0,02%	0,887
	hGPHH-GL	364	48308977	0,27	0	2383	1429,24	0,12	0,01%	0,913
	WBRA-GL	617	48295345	3,25	13,65	2383	1466,81	2,59	0,18%	0,887
	Snay	112715	158031137	262,18	5,91	2384	1835,22	6,77	0,37%	0,622
	Snay-GL	120017	160185995	25,03	3,44	2384	1827,85	1,43	0,08%	0,704
	Sloan	1963	30916653	0,70	3,52	2384	1815,26	0,49	0,03%	0,718
	NSloan	750	44537494	0,18	3,52	2384	1967,73	0,25	0,01%	0,663
	Sloan-MGPS	1759	31863871	4,23	3,52	2384	1858,45	1,06	0,06%	0,700
MPG	5366	47979879	0,47	3,52	2384	1852,76	0,37	0,02%	0,704	
277118	Sem renum.	277019	23512579029	-	-	2771	2235,58	2,78	0,12%	-
	RCM-GL	421	74726891	0,35	0	2771	2382,51	5,18	0,22%	0,938
	RBFS-GL	424	74852621	0,17	0	2771	2428,08	0,76	0,03%	0,921
	hGPHH-GL	427	84714895	0,38	0	2771	2328,01	3,04	0,13%	0,960
	WBRA-GL	713	84585649	6,20	18,47	2771	2420,37	0,21	0,01%	0,921
	Snay	107539	310401674	516,59	10,49	2771	3032,36	0,75	0,02%	0,630
	Snay-GL	107563	310418685	48,13	3,82	2771	3030,94	2,07	0,07%	0,726
	Sloan	2243	55586226	1,17	3,82	2771	3036,44	0,68	0,02%	0,736
	NSloan	909	77343800	0,26	3,82	2771	3294,12	0,01	0,01%	0,679
	Sloan-MGPS	2084	57032215	7,91	4,34	2771	3197,94	2,39	0,07%	0,697
MPG	7281	89227523	0,81	4,34	2771	3199,94	1,26	0,04%	0,698	

Simulações realizadas na máquina M1.

Tabela 9 Resolução de sistemas de equações lineares (de até 105.764 incógnitas) pelo MGCPJ, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de *profile*. Essas simulações foram realizadas na máquina M2 e esses sistemas de equações lineares foram originados de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada pela curva de Sierpiński modificada.

n	Heurística	β	<i>profile</i>	Heurística		MGCPJ		σ	C_v	Speedup
				t(s)	m.(MiB)	iter.	t(s)			
16922	Sem renum.	16921	1710910	-	-	767	51,20	0,02	0,04%	-
	RCM-GL	115	1252527	0,010	0	767	52,56	0,73	1,40%	0,974
	RBFS-GL	117	1253216	0,005	0	767	52,79	0,40	0,76%	0,970
	hGPHH-GL	119	1321688	0,020	0	767	51,08	0,07	0,14%	1,002
	WBRA-GL	178	1327087	0,180	1,55	767	53,03	0,07	0,14%	0,962
	Snay	6212	1508415	1,560	0,51	767	73,54	0,11	0,15%	0,682
	Snay-GL	5576	1535250	0,160	0,43	767	73,36	0,05	0,07%	0,696
	Sloan	571	1074251	0,020	0,52	767	73,95	0,02	0,03%	0,692
	NSloan	229	1336588	0,010	0,26	767	74,89	0,01	0,02%	0,684
	Sloan-MGPS	462	1093326	0,120	0,26	767	74,59	0,05	0,06%	0,685
MPG	1231	1750944	0,030	0,52	767	74,92	0,08	0,10%	0,683	
39716	Sem renum.	39715	6309342	-	-	1144	187,88	0,04	0,02%	-
	RCM-GL	195	4376986	0,050	0	1144	210,13	0,52	0,25%	0,894
	RBFS-GL	198	4380812	0,020	0	1144	211,13	0,20	0,10%	0,890
	hGPHH-GL	192	4770829	0,050	0	1144	209,49	0,31	0,15%	0,897
	WBRA-GL	280	4764403	0,570	3,35	1385	259,05	0,36	0,14%	0,724
	Snay	22597	6548607	8,670	1,20	1144	268,97	0,30	0,11%	0,677
	Snay-GL	22507	6596678	0,870	1,03	1144	269,06	0,17	0,06%	0,696
	Sloan	830	3342149	0,080	1,03	1144	267,87	0,23	0,09%	0,701
	NSloan	372	4634523	0,030	1,03	1144	286,20	0,11	0,04%	0,656
	Sloan-MGPS	831	3461255	0,430	1,03	1144	275,08	0,24	0,09%	0,682
MPG	2358	5222214	0,080	1,03	1144	274,12	0,17	0,06%	0,685	
68414	Sem renum.	68413	14882117	-	-	1514	430,09	0,03	0,01%	-
	RCM-GL	238	9598308	0,080	0	1514	480,92	1,09	0,23%	0,894
	RBFS-GL	240	9605665	0,030	0	1514	485,29	1,02	0,21%	0,886
	hGPHH-GL	236	10705920	0,080	0	1514	480,96	0,39	0,08%	0,894
	WBRA-GL	316	10702498	1,220	5,15	1514	473,28	1,20	0,25%	0,906
	Snay	43837	21823074	36,430	2,06	1514	609,39	0,11	0,02%	0,666
	Snay-GL	43854	21828427	3,590	1,62	1514	609,68	0,78	0,13%	0,701
	Sloan	1284	7093207	0,170	0,95	1514	608,71	0,16	0,03%	0,706
	NSloan	442	10128234	0,060	0,95	1514	655,91	0,10	0,01%	0,656
	Sloan-MGPS	1082	7326579	0,960	0,95	1514	625,25	0,06	0,01%	0,687
MPG	2811	11460778	0,160	0,95	1514	623,09	0,44	0,07%	0,690	
105764	Sem renum.	105763	29560801	-	-	1846	815,79	0,04	0,04%	-
	RCM-GL	311	18180951	0,130	0	1846	899,11	2,09	0,23%	0,907
	RBFS-GL	307	18183664	0,050	0	1846	904,16	2,29	0,25%	0,902
	hGPHH-GL	309	20753083	0,140	0	1846	900,93	1,75	0,20%	0,905
	WBRA-GL	523	20706588	2,330	8,25	2210	1071,93	0,31	0,03%	0,759
	Snay-GL	49982	48279650	9,140	2,23	1846	1144,56	0,07	0,01%	0,707
	Sloan	1756	13247695	0,320	2,23	1846	1159,37	0,35	0,03%	0,703
	NSloan	602	19321158	0,100	1,20	1846	1254,65	0,76	0,06%	0,650
	Sloan-MGPS	1512	13685106	1,880	1,89	1846	1191,15	0,15	0,01%	0,684
	MPG	4447	20523176	0,270	1,20	1846	1166,83	2,19	0,19%	0,699

Simulações realizadas na máquina M2.

Tabela 10 Resolução de sistemas de equações lineares (de 237.086 a 1.015.004 incógnitas) pelo MGCPJ, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de *profile*. Essas simulações foram realizadas na máquina M2 e esses sistemas de equações lineares foram originados de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada pela curva de Sierpiński modificada.

n	Heurística	β	<i>profile</i>	Heurística		MGCPJ		σ	C_v	Speedup
				t(s)	m.(MiB)	iter.	t(s)			
237086	Sem renum.	237085	115804392	-	-	2611	2629,25	8,69	0,33%	-
	RCM-GL	391	56621430	0,29	0	2612	2865,81	7,98	0,28%	0,917
	RBFS-GL	392	56725752	0,12	0	2611	2869,30	1,04	0,04%	0,916
	hGPHH-GL	393	64411087	0,30	0	2612	2852,31	3,88	0,14%	0,922
	WBRA-GL	671	64251019	6,87	15,47	2612	2815,44	0,52	0,02%	0,932
	Snay-GL	116472	215858651	48,28	6,51	2612	3630,73	4,75	0,13%	0,715
	Sloan	2044	41300807	1,01	6,53	2612	3660,82	3,25	0,09%	0,718
	NSloan	812	59082821	0,24	6,45	2612	3981,41	2,10	0,05%	0,660
	Sloan-MGPS	1898	42561396	6,20	6,45	2612	3777,10	1,33	0,04%	0,695
	MPG	5707	64716730	0,74	6,45	2612	3717,71	1,27	0,03%	0,707
467504	Sem renum.	467503	382386929	-	-	3446	6972,41	2,65	0,04%	-
	RCM-GL	448	130166482	0,57	0	3446	7433,93	3,43	0,05%	0,938
	RBFS-GL	445	130547416	0,24	0	3446	7444,73	4,57	0,06%	0,937
	hGPHH-GL	445	149299971	0,60	0	3446	7422,67	6,17	0,08%	0,939
	WBRA-GL	721	148990820	16,10	25,97	3446	7402,24	3,32	0,04%	0,940
	Snay-GL	133459	911618989	239,34	11,04	3449	9650,03	20,42	0,21%	0,705
	Sloan	2498	95551358	2,35	9,92	3449	9676,20	13,96	0,14%	0,720
	NSloan	911	135054695	0,47	9,75	3449	10563,17	35,21	0,33%	0,660
	Sloan-MGPS	2251	98187318	14,22	9,84	3449	10053,35	49,93	0,50%	0,693
	MPG	7842	152632760	1,65	10,95	3449	9974,37	66,00	0,66%	0,699
750446	Sem renum.	750445	911516500	-	-	4246	13659,77	5,64	0,07%	-
	RCM-GL	461	224589050	0,93	0	4245	14562,95	4,39	0,03%	0,938
	RBFS-GL	462	225259320	0,38	0	4245	14673,63	2,30	0,02%	0,931
	hGPHH-GL	471	257304543	0,96	0	4246	14629,19	7,03	0,05%	0,934
	WBRA-GL	727	257141087	27,66	41,23	4246	15194,03	0,54	0,01%	0,897
	Snay-GL	219527	2556636349	733,21	20,74	4232	19103,57	539,83	2,83%	0,689
	Sloan	2441	164952184	3,95	20,75	4232	19222,56	36,40	0,19%	0,710
	NSloan	946	232760320	0,76	20,88	4232	20942,06	9,65	0,05%	0,652
	Sloan-MGPS	2264	169448464	24,42	20,88	4232	19856,75	38,68	0,19%	0,687
	MPG	7986	265072969	2,79	20,36	4232	19351,55	22,96	0,12%	0,706
1015004	Sem renum.	1015003	1580908606	-	-	4557	20025,09	14,55	0,07%	-
	RCM-GL	462	316593383	0,98	0	4557	20726,11	84,12	0,41%	0,966
	RBFS-GL	462	317205568	0,51	0	4557	20656,64	11,44	0,06%	0,969
	hGPHH-GL	465	363030399	1,10	0	4557	20708,71	33,82	0,16%	0,967
	WBRA-GL	727	362460259	19,38	20,90	4557	20654,06	7,76	0,04%	0,969
	Snay-GL	201048	4197030591	707,63	26,85	4557	20521,96	17,59	0,09%	0,943
	Sloan	2484	233117499	4,08	26,78	4557	20586,92	37,33	0,18%	0,973
	MPG	8107	375103029	2,38	26,61	4557	20682,31	38,37	0,19%	0,968

Simulações realizadas na máquina M2 com exceção da instância de 1.015.004 vértices que foi testada na máquina M3.

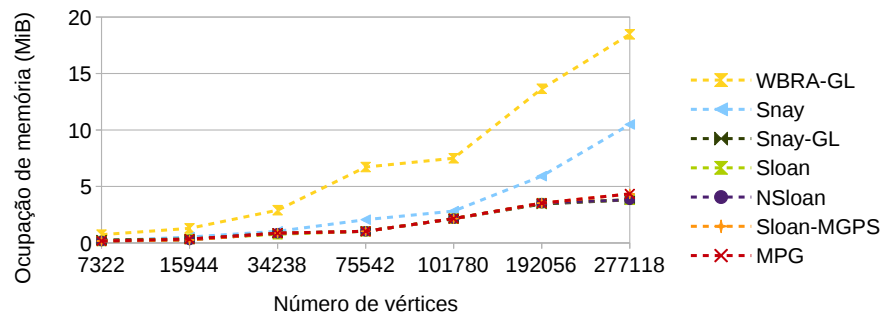


Figura 31 Quantidades de memória extra ocupadas pelas heurísticas WBRA-GL, Snay, Snay-GL, Sloan, NSloan, Sloan-MGPS e MPG, nas instâncias originadas de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada de forma aleatória, mostradas nas tabelas 7 e 8. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência.

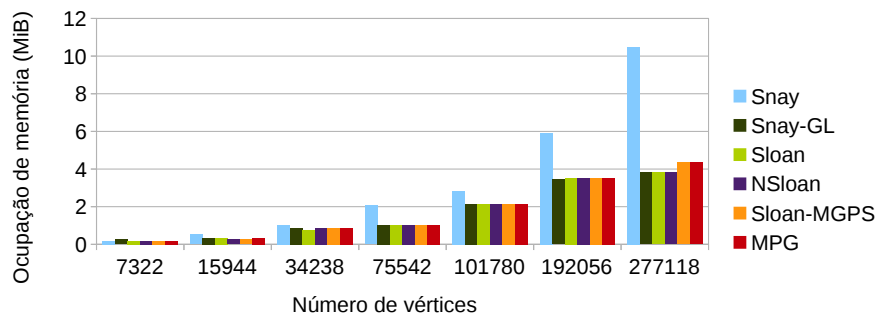


Figura 32 Quantidades de memória extra ocupadas pelas heurísticas Snay, Snay-GL, Sloan, NSloan, Sloan-MGPS e MPG, nas instâncias originadas de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada de forma aleatória, mostradas nas tabelas 7 e 8.

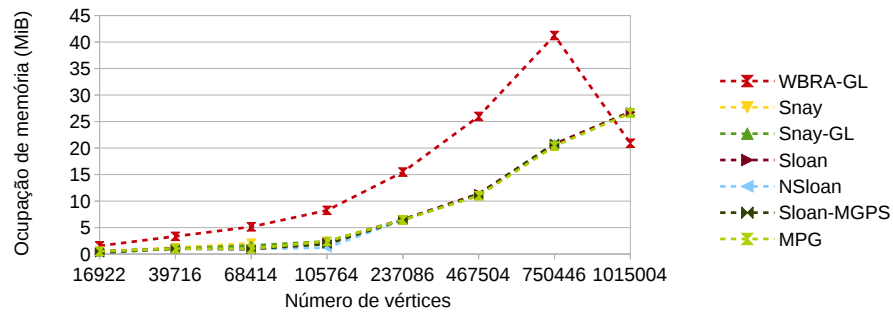


Figura 33 Quantidades de memória extra ocupadas pelas heurísticas WBRA-GL, Snay, Snay-GL, Sloan, NSloan, Sloan-MGPS e MPG, nas instâncias originadas de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada pela curva de Sierpiński modificada, mostradas nas tabelas 9 e 10. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência.

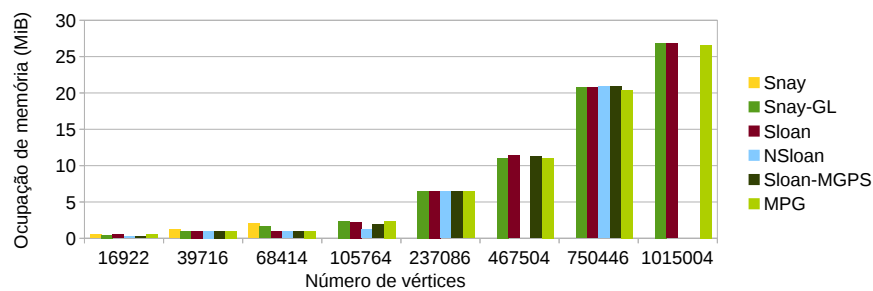


Figura 34 Quantidades de memória extra ocupadas pelas heurísticas Snay, Snay-GL, Sloan, NSloan, Sloan-MGPS e MPG, nas instâncias originadas de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada pela curva de Sierpiński modificada, mostradas nas tabelas 9 e 10.

5.1.3 Instâncias da *University of Florida sparse-matrix collection*

Os resultados médios obtidos pelo MGCPJ, nas instâncias do conjunto de dados da *University of Florida sparse-matrix collection*, são apresentados nas tabelas 11, 12 e 13. Nesse conjunto de instâncias, foram testadas 7 heurísticas para reduções de largura de banda e de *profile*: RCM-GL (GEORGE; LIU, 1981), RBFS-GL (CHAGAS, 2015), hGPHH-GL (KOOHESTANI; POLI, 2011), Snay-GL, de Sloan (1989), Sloan-MGPS (REID e SCOTT, 1999) e MPG (MEDEIROS; PIMENTA; GOLDENBERG, 1993), pois foram as heurísticas que apresentaram os melhores resultados no conjunto de instâncias oriundas da discretização da equação do calor e nos conjuntos de instâncias oriundas da discretização da equação de Laplace. É possível verificar, nos resultados mostrados nas tabelas 4, 5 e 6, que a heurística Snay foi dominada pela heurística Snay-GL quando aplicada para redução do custo computacional do MGCPJ. Em particular, as heurísticas Sloan, MPG, Sloan-MGPS e Snay-GL apresentaram resultados competitivos em relação à redução do custo computacional do MGC-IC0 nas instâncias oriundas da discretização da equação da condução do calor, como pode ser observado nos resultados mostrados nas tabelas 4, 5 e 6.

Nota-se que a heurística RBFS-GL obteve o maior valor de *speedup* quando aplicada na instância *Pres_Poisson*. No entanto, valores negativos de *speedup* do MGCPJ foram obtidos na aplicação das heurísticas para reduções de largura de banda e de *profile* nas outras 10 instâncias. Ainda, observa-se que as heurísticas que apresentaram os melhores resultados em relação à redução de *profile* foram as heurísticas Sloan-MGPS (REID e

Tabela 11 Resultados de resoluções de sistemas de equações lineares, com o MGCPJ, e com renumeração dos vértices dada pelas heurísticas para reduções de largura de *profile*, oriundos das instâncias *nasa1824*, *nasa2910*, *nasa4704* e *nasasrb*, da *University of Florida sparse-matrix collection*. Estas simulações foram realizadas nas máquinas M1 (instâncias *nasa1824*, *nasa2910* e *nasasrb*) e M5 (instância *nasa4704*).

Inst.	Heurística	β	<i>profile</i>	Heurística		JPCGM		σ	C _v	Speedup
				t(s)	m.(MiB)	iter.	t(s)			
<i>nasa1824</i>	Semrenum.	239	205547			1350	24,09	0,14	0,58%	-
	RCM-GL	282	229770	0,004	0	1350	25,01	0,10	0,38%	0,963
	RBFS-GL	283	232069	0,001	0	1349	24,81	0,01	0,04%	0,971
	hGPHH-GL	293	291203	0,003	0	1350	24,60	0,14	0,56%	0,979
	SnayGL	1455	300294	0,305	0	1349	25,10	0,03	0,13%	0,948
	Sloan	1303	186725	0,005	0	1350	25,06	0,17	0,67%	0,961
	NSloan	415	284963	0,002	0	1351	25,65	0,14	0,54%	0,939
	Sloan-MGPS	1102	190128	0,012	0	1347	24,92	0,18	0,71%	0,966
MPG	1519	516936	0,010	0	1350	25,68	0,16	0,61%	0,938	
<i>nasa2910</i>	Semrenum.	859	525745			1846	133,17	0,32	0,24%	-
	RCM-GL	875	522223	0,018	0	1846	142,52	1,02	0,72%	0,934
	RBFS-GL	866	518826	0,004	0	1846	142,60	0,27	0,19%	0,934
	hGPHH-GL	869	1288759	0,016	0	1846	140,07	1,08	0,77%	0,951
	SnayGL	2676	2863227	1,859	0	1843	144,42	0,40	0,27%	0,910
	Sloan	2015	456322	0,018	0	1839	138,05	0,15	0,11%	0,965
	NSloan	1327	955899	0,010	0	1844	145,48	0,49	0,34%	0,915
	Sloan-MGPS	2010	460149	0,016	0	1842	139,36	0,16	0,12%	0,955
MPG	2708	2288760	0,038	0	1842	146,69	0,02	0,01%	0,908	
<i>nasa4704</i>	Semrenum	423	917562			4248	189,98	0,08	0,04%	-
	RCM-GL	419	918658	0,009	0	4245	201,15	0,81	0,40%	0,944
	RBFS-GL	435	928275	0,003	0	4245	200,86	0,80	0,40%	0,946
	hGPHH-GL	450	1079926	0,009	0	4244	201,79	4,37	2,16%	0,941
	SnayGL	4032	1193396	1,620	0,09	4247	203,06	0,65	0,32%	0,928
	Sloan	3084	834354	0,024	0	4244	203,81	0,62	0,30%	0,932
	NSloan	678	1076453	0,005	0	4247	210,09	2,46	1,17%	0,904
	Sloan-MGPS	2753	808577	0,056	0	4246	203,29	1,20	0,59%	0,934
MPG	3680	2716364	0,074	0	4244	211,56	0,04	0,02%	0,898	
<i>nasasrb</i>	Semrenum.	893	20311330			25326	27902,24	2,28	0,01%	-
	RCM-GL	586	19448635	0,26	0	25327	30241,97	14,94	0,05%	0,923
	RBFS-GL	779	20164371	0,07	0	25319	30067,30	90,53	0,30%	0,928
	hGPHH-GL	806	20545002	0,23	0	25326	28461,38	54,27	0,19%	0,980
	SnayGL	41895	363658668	580,23	1,32	25319	30213,73	20,52	0,07%	0,906
	Sloan	5063	19055047	0,34	1,03	25320	29892,50	147,67	0,49%	0,933
	NSloan	4865	23564619	0,08	1,03	25326	31307,06	24,46	0,08%	0,891
	Sloan-MGPS	4932	18682599	0,62	1,03	25320	29712,80	60,37	0,20%	0,939
MPG	45896	346820836	13,67	1,03	25326	32795,47	61,17	0,19%	0,850	

Tabela 12 Resultados de resoluções de sistemas de equações lineares, com o MGCPJ, e com renumeração dos vértices dada pelas heurísticas para reduções de largura de *profile*, oriundos das instâncias *sts4098*, *Pres_Poisson*, *olafu* e *raefsky4* do conjunto de dados da *University of Florida sparse-matrix collection*. Estas simulações foram realizadas nas máquinas M3 (instância *olafu*) e M5 (instâncias *sts4098*, *Pres_Poisson* e *raefsky4*).

<i>Inst.</i>	Heurística	β	<i>profile</i>	Heurística		JPCGM		σ	C_v	Speedup
				t(s)	m.(MiB)	iter.	t(s)			
<i>sts4098</i>	Semrenum.	3323	5217389			590	19,59	0,01	0,03%	-
	RCM-GL	1165	2084237	0,007	0	590	22,48	0,07	0,30%	0,871
	RBFS-GL	1343	2110715	0,002	0	589	20,74	0,01	0,04%	0,944
	hGPHH-GL	1171	2981815	0,007	0	588	21,81	0,39	1,80%	0,898
	SnayGL	3955	1851729	1,800	0,17	589	21,22	0,04	0,19%	0,851
	Sloan	3195	518163	0,023	0,17	589	20,53	0,01	0,01%	0,953
	NSloan	3020	2505064	0,007	0,17	589	21,76	0,01	0,05%	0,900
	Sloan-MGPS	3351	461998	0,073	0,17	590	20,60	0,01	0,03%	0,948
	MPG	3729	961548	0,017	0,17	588	21,11	0,08	0,38%	0,927
<i>Pres_Poisson</i>	Semrenum.	12583	9789525			1009	309,45	1,27	0,41%	-
	RCM-GL	326	3009635	0,060	0	1012	297,49	0,99	0,33%	1,040
	RBFS-GL	368	3089264	0,025	0	1009	276,47	0,49	0,18%	1,119
	hGPHH-GL	364	3130744	0,059	0	1009	292,71	0,46	0,16%	1,057
	SnayGL	2079	7306759	6,757	0,17	1009	316,35	0,59	0,19%	0,958
	Sloan	642	2827171	0,066	0,26	1012	295,06	0,39	0,13%	1,049
	NSloan	594	3951006	0,044	0,26	1012	327,74	0,38	0,12%	0,944
	Sloan-MGPS	582	2834035	0,156	0,26	1009	293,98	0,94	0,32%	1,052
	MPG	14168	26556694	3,845	0,34	1009	297,32	0,02	0,01%	1,028
<i>olafu</i>	Semrenum.	593	4951980			16146	6833,33	5,10	0,07%	-
	RCM-GL	553	5029301	0,146	0	16146	7253,32	32,13	0,44%	0,942
	RBFS-GL	582	5025044	0,070	0	16146	7168,24	19,04	0,27%	0,953
	hGPHH-GL	573	5165776	0,132	0	16146	7189,16	10,30	0,14%	0,950
	SnayGL	5708	7148721	6,396	0,26	16146	7223,69	7,27	0,10%	0,945
	Sloan	6173	4768547	0,146	0,26	16146	7154,39	2,24	0,03%	0,955
	NSloan	4760	7334811	0,105	0,26	16146	7290,16	0,58	0,01%	0,937
	Sloan-MGPS	7775	4489770	0,171	0,26	16146	7218,71	3,00	0,04%	0,947
	MPG	14467	29376748	4,268	0,34	16146	7491,18	9,28	0,12%	0,912
<i>raefsky4</i>	Semrenum.	11786	19611188			11245	5862,29	1,05	0,02%	-
	RCM-GL	991	12553981	0,130	0	11157	6293,30	3,03	0,05%	0,931
	RBFS-GL	1171	12251822	0,030	0	11158	6467,80	12,69	0,20%	0,906
	hGPHH-GL	1141	13120923	0,110	0	11182	6313,03	397,88	6,30%	0,929
	SnayGL	17602	34715770	39,160	0,43	11225	6587,44	4,73	0,07%	0,885
	Sloan	6550	8587731	0,180	0,26	11180	6199,92	29,26	0,47%	0,946
	NSloan	2242	15308534	0,070	0,26	11246	6785,80	21,77	0,32%	0,864
	Sloan-MGPS	8378	7841072	0,340	0,26	11245	6411,41	4,27	0,07%	0,914
	MPG	18201	74604715	6,790	0,26	11248	6803,42	1,01	0,01%	0,861

Tabela 13 Resultados de resoluções de sistemas de equações lineares, com o MGCPJ, e com renumeração dos vértices dada pelas heurísticas para reduções de largura de *profile*, oriundos das instâncias *textitthermal1*, *2cubes_sphere* e *offshore* do conjunto de dados da *University of Florida sparse-matrix collection*. Estas simulações foram realizadas nas máquinas M6.

<i>Inst.</i>	Heurística	β	<i>profile</i>	Heurística		JPCGM		σ	C_v	Speedup
				t(s)	m.(MiB)	iter.	t(s)			
<i>thermal1</i>	Semrenum.	80916	175625317			1885	455,72	0,75	0,17%	-
	RCM-GL	220	12017373	0,09	0	1885	562,44	1,21	0,22%	0,810
	RBFS-GL	228	12244583	0,05	0,98	1885	556,77	0,12	0,02%	0,818
	hGPHH-GL	240	12997244	0,10	0	1885	556,30	0,26	0,05%	0,819
	SnayGL	2670	20232166	1,02	3,11	1885	522,63	0,44	0,09%	0,870
	Sloan	889	10487409	0,19	3,09	1885	537,41	0,19	0,04%	0,848
	NSloan	429	13393908	0,12	3,09	1885	594,25	0,12	0,02%	0,767
	SloanMGPS	661	10677120	0,69	3,01	1885	559,05	0,09	0,02%	0,814
MPG	16857	10958622	0,17	3,09	1885	529,42	0,65	0,12%	0,861	
<i>2cubes_sphere</i>	Semrenum.	100407	483241271			33	13,82	0,05	0,38%	-
	RCM-GL	4709	268149672	0,25	0	33	17,55	0,05	0,29%	0,776
	RBFS-GL	4847	280851796	0,14	0	33	17,62	0,02	0,11%	0,778
	hGPHH-GL	4693	345191689	0,25	0	33	17,55	0,07	0,37%	0,776
	SnayGL	25715	231759298	21,51	2,15	33	17,14	0,03	0,15%	0,358
	Sloan	11186	186478091	14,66	2,15	33	17,21	0,04	0,23%	0,434
	NSloan	9203	346819754	0,24	2,15	33	18,11	0,12	0,68%	0,753
	SloanMGPS	13446	200449820	5,80	2,15	33	17,29	0,04	0,21%	0,599
MPG	95371	460302437	21,54	2,15	33	17,61	0,02	0,13%	0,353	
<i>offshore</i>	Semrenum.	237738	3588201815			1226	1952,44	1,73	0,09%	-
	RCM-GL	21035	2634951939	0,66	0	1226	2494,48	0,30	0,01%	0,782
	RBFS-GL	20032	2660219387	0,38	0	1226	2520,47	2,65	0,11%	0,775
	hGPHH-GL	23859	3897866179	0,70	0	1228	2519,27	9,04	0,36%	0,775
	SnayGL	99875	1184107609	166,66	6,89	1228	2446,45	2,07	0,08%	0,747
	Sloan	121957	1837918281	72,14	4,73	1237	2417,41	10,46	0,43%	0,784
	NSloan	102633	3264868562	0,72	4,64	1226	2637,57	11,08	0,42%	0,740
	SloanMGPS	124658	1510670726	141,63	4,73	1226	2489,40	9,35	0,38%	0,742
MPG	253828	4262147507	260,51	4,81	1230	2438,28	1,25	0,05%	0,723	

SCOTT, 1999) e de Sloan (1989), considerando o número de vezes que a heurística obteve o melhor resultado. Em relação à redução de largura de banda, considerando a mesma métrica, a heurística RCM-GL apresentou os melhores resultados. Além disso, é possível verificar que a heurística que apresentou o menor custo computacional foi a heurística RBFS-GL.

5.2 Resultados de resoluções de sistemas de equações lineares com o MGC-IC0

Nesta seção, são apresentados os resultados das resoluções dos sistemas de equações lineares dos conjuntos de instâncias apresentados na seção 4.1, por meio do MGC-IC0, com e sem as reduções de largura de banda e de *profile* dadas pela aplicação das heurísticas testadas. Nas Subseções 5.2.1 e 5.2.2, são mostrados os resultados das resoluções dos sistemas de equações lineares oriundos da discretização das equações da condução do calor e de Laplace, respectivamente. Na subseção 5.2.3, são mostrados os resultados das resoluções dos sistemas de equações lineares of the *University of Florida sparse-matrix collection*.

As tabelas mostradas nesta seção com os resultados do MGC-IC0 são estruturadas como a seguir. Na primeira coluna, é exibida a dimensão n da matriz de coeficientes do respectivo sistema de equações lineares (número de vértices do grafo) ou o nome da instância da *University of Florida sparse-matrix collection*. Na segunda coluna, há o nome da heurística utilizada para reduções de *profile* e largura de banda. Os resultados em relação ao *profile* e à largura de banda são mostrados na terceira e quarta colunas, respectivamente. Na quinta e sexta colunas, são apresentados, respectivamente, os resultados das heurísticas em relação ao

custo computacional, em segundos, e a ocupação de memória, em *mebibytes*. Os custos computacionais do IC0 e do MGC preconditionado são mostrados separadamente, na sétima e oitava colunas. Decidiu-se discriminar esses dados, pois as renumerações produzidas pelas heurísticas também afetam o custo computacional do IC0. O número de iterações e o custo computacional total, em segundos, do MGC-IC0, são mostrados na nona e décima colunas. Os desvios padrões e os coeficientes de variação, referentes ao custo computacional total do MGC-IC0, são mostrados na décima primeira e décima segunda colunas. Note que é mostrado “Sem renumeração” na primeira linha do conjunto de simulações para cada instância. Isso significa que nenhuma heurística de reordenação foi utilizada. Com esse resultado, é possível verificar o *speedup* obtido por meio da utilização da heurística de reordenação, que consiste na divisão do tempo de execução do MGC-IC0 somado ao tempo de execução da heurística pelo tempo de execução do MGC-IC0 sem a utilização de heurística para reordenação, e é mostrado na última coluna.

5.2.1 Instâncias oriundas da discretização da equação da condução do calor

Os resultados médios obtidos pelo MGC-IC0 nas instâncias oriundas da discretização da equação da condução calor são mostrados nas tabelas 14 e 15. Os *speedups* do MGC-IC0 obtido com aplicação das heurísticas são ilustrados nas figuras 35 e 36.

Observa-se que as heurísticas que apresentaram os melhores resultados em relação às reduções de largura de banda e de *profile*, considerando o número de vezes que a heurística obteve o melhor resultado,

Tabela 14 Resolução de sistemas de equações lineares (de até 23.637 incógnitas) pelo MGC-IC0, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de *profile*. Essas simulações foram realizadas na máquina M4 e esses sistemas de equações lineares foram originados de discretizações da equação da condução do calor por volumes finitos.

n	Heurística	β	<i>profile</i>	Heurística		IC0	MGC	MGC-IC0		σ	C _v	Speedup
				t(s)	m.(MiB)			t(s)	iter.			
4846	Without reordering	4769	9116750	-	-	2,29	5,81	144	8,10	0,02	0,26%	-
	RCM-GL	152	425223	0,005	0	3,24	3,50	92	6,74	0,02	0,34%	1,201
	RBFS-GL	165	448906	0,002	0	3,31	3,65	95	6,96	0,06	0,80%	1,163
	hGPHH-GL	152	471590	0,006	0	3,29	4,04	106	7,33	0,01	0,19%	1,104
	WBRA-GL	198	495965	0,069	1,03	3,31	3,89	101	7,20	0,02	0,33%	1,114
	Snay	1108	360746	0,285	0,09	3,23	3,16	84	6,39	0,02	0,35%	1,213
	Snay-GL	1123	364222	0,030	0,17	3,23	3,19	85	6,42	0,02	0,33%	1,256
	Sloan	882	297009	0,006	0,17	3,24	3,21	85	6,45	0,02	0,29%	1,255
	NSloan	554	448633	0,004	0	3,30	4,04	106	7,33	0,03	0,35%	1,104
	Sloan-MGPS	809	299210	0,025	0	3,26	3,26	86	6,52	0,02	0,37%	1,238
	MPG	1084	305791	0,006	0,17	3,24	3,44	91	6,67	0,02	0,36%	1,213
	Hu-Scott	2350	398032	331,284	100,47	3,22	3,28	89	6,50	0,02	0,24%	0,024
10728	Without reordering	10626	45314579	-	-	11,77	20,69	206	32,46	0,27	0,82%	-
	RCM-GL	242	1504910	0,014	0	16,71	11,36	133	28,07	0,28	1,01%	1,156
	RBFS-GL	241	1455453	0,007	0	15,86	11,77	137	27,63	0,11	0,42%	1,175
	hGPHH-GL	249	1645616	0,015	0	15,32	12,96	152	28,28	0,10	0,34%	1,147
	WBRA-GL	300	1641665	0,203	2,58	15,74	12,44	145	28,18	0,22	0,77%	1,144
	Snay	1418	1087614	0,880	0,26	16,07	10,25	121	26,33	0,12	0,44%	1,193
	Snay-GL	1391	1127110	0,100	0,26	16,17	10,26	121	26,44	0,27	1,01%	1,223
	Sloan	1415	1041059	0,020	0,26	16,15	10,53	124	26,67	0,28	1,03%	1,216
	NSloan	1191	1525236	0,010	0,17	16,51	13,05	150	29,56	0,11	0,36%	1,098
	Sloan-MGPS	1521	1056012	0,100	0,26	16,21	10,61	124	26,82	0,16	0,60%	1,206
	MPG	2087	1099888	0,020	0,17	16,12	11,30	133	27,42	0,12	0,43%	1,183
	23367	Without reordering	23167	216212086	-	-	71,05	70,76	302	141,81	2,21	1,56%
RCM-GL		367	4913698	0,050	0	93,52	36,80	192	130,32	0,45	0,35%	1,088
RBFS-GL		355	4786316	0,020	0	93,43	38,12	198	131,55	0,73	0,56%	1,078
hGPHH-GL		376	5521227	0,050	0	90,82	41,63	219	132,45	0,29	0,22%	1,070
WBRA-GL		671	5597225	0,710	7,73	92,58	39,99	208	132,57	0,78	0,59%	1,064
Snay		1823	3413219	2,940	0,69	94,68	32,67	176	127,35	0,76	0,60%	1,088
Snay-GL		1841	3580526	0,310	0,51	94,92	32,62	176	127,55	0,02	0,02%	1,109
Sloan		2973	3578074	0,080	0,60	95,48	33,07	177	128,55	1,60	1,25%	1,102
NSloan		1590	5012507	0,030	0,26	97,29	41,49	217	138,78	0,37	0,27%	1,022
Sloan-MGPS		2713	3565599	0,330	0,26	95,80	33,46	178	129,26	0,90	0,70%	1,094
MPG		3060	3493791	0,060	0,34	95,71	35,80	193	131,51	0,56	0,43%	1,078

Tabela 15 Resolução de sistemas de equações lineares (de 50.592 a 495.853 incógnitas) pelo MGC-IC0, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de *profile*. Essas simulações foram realizadas na máquina M4 e esses sistemas de equações lineares foram originados de discretizações da equação da condução do calor por volumes finitos.

n	Heurística	β	<i>profile</i>	Heurística		IC0	MGC	MGC-IC0		σ	C_v	Speedup
				t(s)	m.(MiB)			t(s)	iter.			
50592	Sem renum,	50461	1020411959	-	-	383,76	220,27	431	604,03	0,83	0,14%	-
	RCM-GL	553	16182346	0,11	0	479,33	115,46	278	594,79	4,01	0,67%	1,0150
	RBFS-GL	541	16420191	0,06	0	478,28	118,42	283	596,70	3,01	0,50%	1,0120
	hGPHH-GL	547	18159371	0,12	0	471,42	128,22	314	599,64	0,15	0,03%	1,0070
	WBRA-GL	706	18363549	2,49	15,98	478,02	125,58	301	603,60	2,57	0,43%	0,9970
	Snay	2928	10727635	9,33	1,38	481,77	102,84	257	584,61	1,02	0,17%	1,0170
	Snay-GL	2953	10825865	0,97	1,03	481,69	102,65	256	584,35	1,81	0,31%	1,0320
	Sloan	4590	11711775	0,24	1,12	489,28	104,06	258	593,34	1,05	0,18%	1,0180
	NSloan	2568	16461468	0,08	1,12	498,61	131,84	316	630,45	3,18	0,50%	0,9580
	Sloan-MGPS	5150	11866029	1,22	1,12	489,88	104,47	255	594,35	2,83	0,48%	1,0140
MPG	6604	11565237	0,14	1,12	492,15	110,73	275	602,89	2,37	0,39%	1,0020	
108683	Sem renum,	108216	4725435534	-	-	1963,07	702,21	627	2665,28	3,56	0,13%	-
	RCM-GL	911	55350648	0,25	0	2184,5	351,83	400	2536,34	6,95	0,27%	1,0510
	RBFS-GL	907	55649180	0,13	0	2166,44	357,33	408	2523,77	2,04	0,08%	1,0560
	hGPHH-GL	885	59355159	0,27	0	2177,27	399,50	452	2576,77	3,39	0,13%	1,0340
	WBRA-GL	1764	61146321	9,31	26,55	2164,43	375,23	428	2539,66	3,87	0,15%	1,0460
	Snay	3887	33205366	28,59	2,92	2248,01	317,88	369	2565,89	10,85	0,42%	1,0270
	Snay-GL	4363	34535667	2,98	2,58	2254,04	315,62	366	2569,67	8,47	0,33%	1,0360
	Sloan	8243	36198196	0,73	2,75	2305,87	321,32	370	2627,19	9,67	0,37%	1,0140
	NSloan	8680	52203136	0,23	2,75	2322,17	408,26	453	2730,42	21,82	0,80%	0,9760
	Sloan-MGPS	7700	36496358	4,00	2,75	2309,2	327,70	371	2636,90	17,16	0,65%	1,0090
MPG	14267	36552907	0,33	2,75	2297,64	344,29	396	2641,93	11,17	0,42%	1,0090	
232052	Sem renum,	231672	21652820640	-	-	9338,85	2167,06	908	11505,91	19,05	0,17%	-
	RCM-GL	1231	168178362	0,55	0	9726,02	1080,58	573	10806,60	21,12	0,20%	1,0650
	RBFS-GL	1275	167101014	0,28	0	9731,16	1125,55	595	10856,70	8,85	0,08%	1,0600
	hGPHH-GL	1228	180118365	0,63	0	9810,81	1239,59	654	11050,40	3,65	0,03%	1,0410
	WBRA-GL	1958	184101780	30,14	65,74	9751,17	1177,3	621	10928,47	13,14	0,12%	1,0500
	Snay	5818	99995456	92,28	6,34	10023,30	986,05	533	11009,34	36,06	0,33%	1,0360
	Snay-GL	5928	101242969	9,55	5,22	10016,25	985,42	532	11001,67	4,26	0,04%	1,0450
	Sloan	26898	121163861	2,59	5,22	10257,78	995,94	534	11253,72	23,31	0,21%	1,0220
	NSloan	12010	166693574	0,45	5,22	10396,03	1274,85	657	11670,89	61,79	0,53%	0,9860
	Sloan-MGPS	26456	122023395	14,85	5,13	10424,15	1014,25	534	11438,41	50,49	0,44%	1,0050
MPG	25623	117501685	0,80	5,22	10279,02	1070,18	574	11349,20	10,50	0,09%	1,0140	
492853 ¹	Sem renum,	492100	97893937993	-	-	31921,28	4952,97	1322	36874,24	15,12	0,04%	-
	RCM-GL	2050	564919856	0,79	0	37319,27	2698,11	834	40017,38	40,90	0,10%	0,9214
	RBFS-GL	1832	547906746	0,39	0	37280,46	2769,42	856	40049,88	25,22	0,06%	0,9207
	Snay-GL	9745	324635599	25,30	11,79	37891,73	2431,46	771	40323,19	45,04	0,36%	0,9139

¹Simulações realizadas na máquina M1.

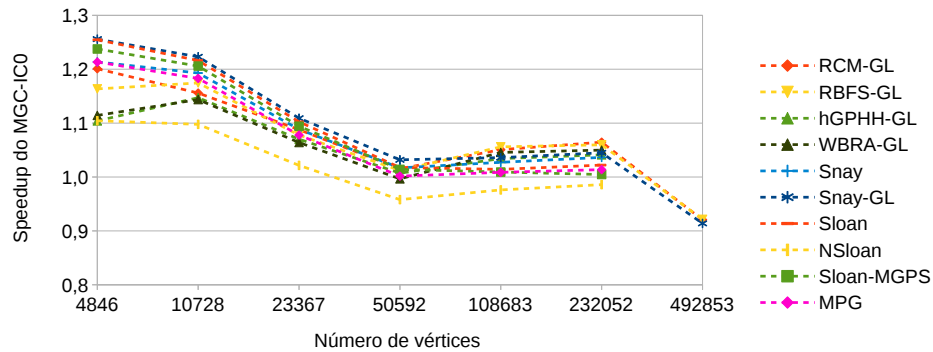


Figura 35 Speedup do MGC-IC0 obtido com aplicação das heurísticas, nas instâncias originadas de discretizações da equação da condução do calor por volumes finitos, mostrados nas tabelas 14 e 15. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência.

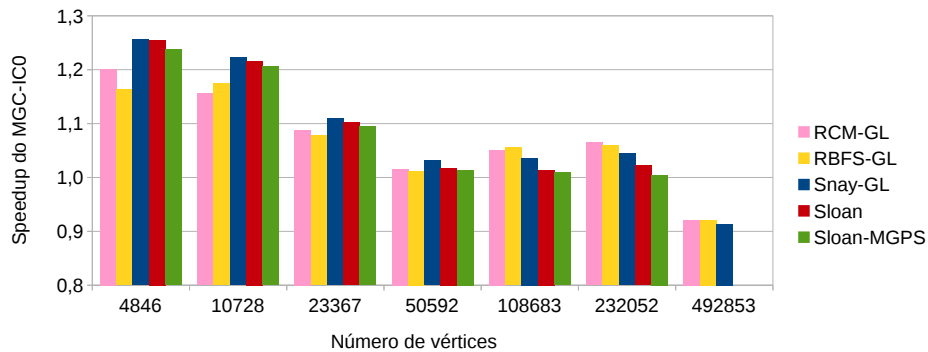


Figura 36 Speedup do MGC-IC0 obtido com aplicação das 5 heurísticas com os melhores resultados, nas instâncias originadas de discretizações da equação da condução do calor por volumes finitos, mostrados nas tabelas 14 e 15.

foram as heurísticas RBFS-GL e de Snay (1976), respectivamente. Em relação ao consumo de memória e ao tempo de execução, os resultados apresentados pelas heurísticas foram similares aos resultados mostrados na seção 5.1, uma vez que foram utilizadas as mesmas instâncias para teste.

Em relação à redução do custo computacional do MGC-IC0, observa-se que a heurística de Hu e Scott (2001) e os métodos de trocas adjacentes (REID; SCOTT, 2002), aplicados em conjunto com as demais heurísticas, foram superados por várias heurísticas quando aplicados à instância de 4.846 vértices. Portanto, a heurística de Hu e Scott (2001) e os métodos de trocas adjacentes não foram aplicados nas instâncias maiores e nos demais conjuntos de dados. Além disso, observa-se que a heurística Snay-GL obteve o melhor resultado nas instâncias de até 50.592 vértices, e as heurísticas RBFS-GL e RCM-GL obtiveram os melhores resultados nas instâncias de 108.683 e 232.052 vértices, respectivamente. Essas heurísticas foram aplicadas à instância de 492.853 vértices, porém essas simulações resultaram em *speedup* negativos.

Com as reduções de largura de banda de *profile*, ocorreu a redução do custo computacional do MGC-IC0. Porém, a redução do custo computacional do MGC-IC0 não ocorreu de forma proporcional às reduções de largura de banda e de *profile*. Ainda, em relação ao MGC-IC0, observa-se que há uma variação significativa no número de iterações com as reduções de largura de banda e de *profile*.

5.2.2 Instâncias oriundas da discretização da equação de Laplace

Os resultados médios obtidos pelo MGC-IC0 nas instâncias oriundas da discretização da equação de Laplace, com numeração das linhas de forma

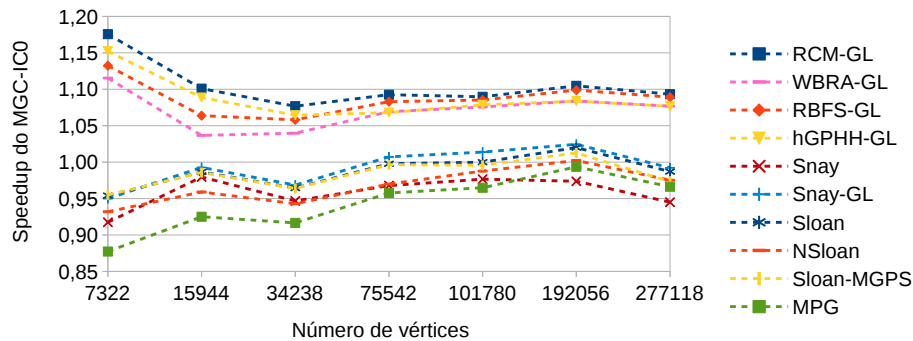


Figura 37 Speedup do MGC-IC0 obtido com aplicação das heurísticas, nas instâncias originadas de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada de forma aleatória, mostrado nas tabelas 16 e 17. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência.

aleatória, são mostrados nas tabelas 16 e 17, e os *speedups* do MGC-IC0 obtido com aplicação das heurísticas são ilustrados nas figuras 37 e 38. Os resultados médios obtidos pelo MGC-IC0 nas instâncias oriundas da discretização da equação de Laplace, com numeração das linhas dada pela curva de Sierpiński modificada, são mostrados nas tabelas 18 e 19, e os *speedups* do MGC-IC0 obtido com aplicação das heurísticas são ilustrados nas figuras 39 e 40.

Observa-se que as heurísticas que apresentaram os melhores resultados em relação às reduções de largura de banda e de *profile*, considerando o número de vezes que a heurística obteve o melhor resultado, foram as heurísticas RCM-GL e de Sloan (1989), respectivamente. A heurística RCM-GL obteve os melhores resultados em relação à redução do custo computacional do MGC-IC0 nesses dois conjuntos de instâncias. Em particular, a heurística Snay foi testada apenas nas três menores instâncias de cada conjunto, pois os seus resultados, em relação à redução do custo

Tabela 16 Resolução de sistemas de equações lineares (de até 34,238 incógnitas) pelo MGC-IC0, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de *profile*. Essas simulações foram realizadas na máquina M1 e esses sistemas de equações lineares foram originados de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada de forma aleatória.

n	Heurística	β	<i>profile</i>	Heurística		IC0	MGC	MGC-IC0		σ	C_v	Speedup
				t(s)	m.(MiB)	t(s)	t(s)	iter.	t(s)			
7322	Sem renum,	7248	16083808	-	-	3,36	7,66	271	11,01	0,01	0,05%	-
	RCM-GL	80	397878	0,005	0	4,17	5,19	201	9,36	0,01	0,12%	1,176
	RBFS-GL	80	398377	0,003	0	4,16	5,56	202	9,72	0,01	0,06%	1,132
	hGPHH-GL	80	406461	0,006	0	4,18	5,36	208	9,55	0,02	0,24%	1,152
	WBRA-GL	154	412219	0,051	0,73	4,18	5,64	205	9,82	0,04	0,43%	1,115
	Snay	1853	448728	0,240	0,26	4,73	7,02	199	11,76	0,28	2,39%	0,918
	Snay-GL	1852	448722	0,030	0,26	4,53	7,03	199	11,56	0,07	0,59%	0,950
	Sloan	407	377055	0,010	0,17	4,52	7,01	199	11,53	0,02	0,19%	0,954
	NSloan	178	425289	0,004	0,17	4,57	7,25	204	11,81	0,01	0,12%	0,932
	Sloan-MGPS	296	374434	0,031	0,17	4,52	6,98	197	11,50	0,08	0,71%	0,955
MPG	846	610304	0,010	0,17	4,39	8,15	228	12,54	0,09	0,75%	0,877	
15944	Sem renum,	15902	76482022	-	-	19,38	25,79	396	45,17	0,08	0,17%	-
	RCM-GL	120	1149124	0,020	0	24,45	16,56	290	41,01	0,04	0,11%	1,101
	RBFS-GL	124	1148009	0,010	0	24,58	17,87	291	42,45	0,04	0,09%	1,064
	hGPHH-GL	121	1219929	0,020	0	24,21	17,26	304	41,47	0,07	0,16%	1,089
	WBRA-GL	176	1233347	0,110	1,29	24,71	18,76	305	43,46	0,04	0,10%	1,037
	Snay	5862	1586436	1,030	0,43	23,33	21,75	281	45,08	0,24	0,52%	0,980
	Snay-GL	6290	1629561	0,110	0,34	23,54	21,85	282	45,39	0,13	0,29%	0,993
	Sloan	477	982724	0,030	0,26	23,66	22,10	286	45,76	0,24	0,53%	0,986
	NSloan	218	1222337	0,010	0,26	23,71	23,37	298	47,09	0,15	0,31%	0,959
	Sloan-MGPS	473	1003462	0,100	0,26	23,58	22,15	284	45,72	0,11	0,23%	0,986
MPG	1222	1587797	0,030	0,26	23,44	25,36	323	48,80	0,37	0,76%	0,925	
34238	Sem renum,	34609	357518296	-	-	108,04	81,50	565	189,53	0,60	0,32%	-
	RCM-GL	192	3413888	0,040	0	125,85	50,13	411	175,98	0,47	0,27%	1,077
	RBFS-GL	191	3416318	0,020	0	126,75	52,36	412	179,11	0,75	0,42%	1,058
	hGPHH-GL	193	3756415	0,040	0	125,15	52,87	434	178,02	0,03	0,02%	1,064
	WBRA-GL	267	3750738	0,290	2,90	127,10	54,89	431	181,99	0,67	0,37%	1,040
	Snay	21625	5150148	5,500	0,95	127,70	67,00	391	194,70	1,36	0,70%	0,947
	Snay-GL	21625	5153902	0,560	0,86	128,33	66,80	391	195,13	0,34	0,17%	0,969
	Sloan	917	2578022	0,060	0,77	128,93	67,54	395	196,46	0,94	0,48%	0,964
	NSloan	366	3608451	0,030	0,86	128,03	72,94	419	200,98	0,70	0,35%	0,943
	Sloan-MGPS	845	2672048	0,300	0,86	129,25	67,08	388	196,33	1,24	0,63%	0,964
MPG	2111	3965061	0,060	0,86	129,23	77,49	448	206,72	0,42	0,20%	0,917	

Tabela 17 Resolução de sistemas de equações lineares (de 75,542 a 277,118 incógnitas) pelo MGC-IC0, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de *profile*. Essas simulações foram realizadas na máquina M1 e esses sistemas de equações lineares foram originados de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada de forma aleatória.

n	Heurística	β	<i>profile</i>	Heurística		IC0	MGC	MGC-IC0		σ	C_v	Speedup
				t(s)	m.(MiB)			t(s)	iter.			
75542	Sem renum.	75490	1744941733	-	-	590,68	268,22	815	858,91	1,15	0,13%	-
	RCM-GL	275	12096902	0,09	0	621,62	164,43	607	786,05	0,65	0,08%	1,093
	RBFS-GL	275	12128940	0,04	0	624,21	168,84	616	793,05	1,40	0,18%	1,083
	hGPHH-GL	277	13793938	0,10	0	624,75	179,41	663	804,16	0,14	0,02%	1,068
	WBRA-GL	459	13739132	0,87	6,72	627,12	175,33	639	802,45	1,52	0,19%	1,069
	Snay	47789	28972039	38,00	2,20	638,59	210,94	567	849,53	0,93	0,11%	0,968
	Snay-GL	47833	28978947	3,36	1,03	639,07	210,36	566	849,43	1,35	0,16%	1,007
	Sloan	1521	8981209	0,18	1,11	646,83	213,75	572	860,58	5,46	0,63%	0,998
	NSloan	533	12805775	0,07	1,11	645,73	239,81	628	885,54	2,39	0,27%	0,970
	Sloan-MGPS	1236	9245713	1,05	1,11	645,84	215,07	568	860,90	1,98	0,23%	0,996
MPG	4020	14107424	0,15	1,11	649,90	246,83	653	896,73	5,38	0,60%	0,958	
101780	Sem renum.	101583	3169282786	-	-	1093,56	405,13	907	1498,68	0,13	0,01%	-
	RCM-GL	407	21336387	0,13	0	1126,20	248,99	670	1375,19	0,64	0,05%	1,090
	RBFS-GL	406	21346316	0,06	0	1127,17	253,62	676	1380,79	0,71	0,05%	1,085
	hGPHH-GL	408	23967214	0,13	0	1123,87	265,42	716	1389,29	1,81	0,13%	1,079
	WBRA-GL	697	24026871	1,53	7,50	1128,75	262,94	701	1391,70	2,27	0,16%	1,076
	Snay	64553	45830895	61,65	2,91	1155,45	317,39	632	1472,85	3,70	0,25%	0,977
	Snay-GL	64282	45857748	5,85	2,49	1154,29	318,10	634	1472,40	0,41	0,03%	1,014
	Sloan	8624	15003839	0,34	2,58	1176,73	321,63	640	1498,36	0,09	0,01%	1,000
	NSloan	7730	21163363	0,09	2,58	1164,64	352,38	685	1517,02	8,25	0,54%	0,988
	Sloan-MGPS	8420	15400014	1,98	2,58	1180,21	323,12	635	1503,33	7,40	0,49%	0,996
MPG	10570	24189624	0,24	2,58	1184,92	368,18	723	1553,09	0,60	0,04%	0,965	
192056	Sem renum.	191738	11329772559	-	-	4115,15	1045,04	1215	5160,18	10,96	0,21%	-
	RCM-GL	360	42577946	0,24	0	4029,35	641,39	909	4670,73	3,90	0,08%	1,105
	RBFS-GL	360	42627911	0,11	0	4036,74	659,31	927	4696,05	4,22	0,09%	1,099
	hGPHH-GL	364	48325681	0,26	0	4056,77	704,00	1000	4760,77	3,95	0,08%	1,084
	WBRA-GL	616	48302407	3,26	13,65	4062,07	695,39	975	4757,46	3,26	0,07%	1,084
	Snay	112715	158031137	290,69	5,75	4185,27	823,49	853	5008,76	14,58	0,29%	0,974
	Snay-GL	120095	160105234	27,40	4,53	4190,03	819,54	849	5009,57	6,11	0,12%	1,024
	Sloan	1963	30916661	0,76	4,62	4218,95	839,61	866	5058,56	18,91	0,37%	1,020
	NSloan	753	44591269	0,17	4,62	4223,12	925,10	932	5148,22	8,17	0,16%	1,002
	Sloan-MGPS	1733	31860210	4,35	4,62	4248,93	842,24	857	5091,17	33,18	0,65%	1,013
MPG	5366	47979879	0,47	4,62	4240,38	952,53	971	5192,91	28,96	0,56%	0,994	
277118	Sem renum.	277019	23512579029	-	-	8562,78	1765,92	1430	10328,70	3,22	0,03%	-
	RCM-GL	420	74697812	0,34	0	8371,11	1075,95	1066	9447,06	7,03	0,07%	1,093
	RBFS-GL	419	74759303	0,17	0	8387,01	1095,76	1078	9482,77	2,16	0,02%	1,089
	hGPHH-GL	425	84682829	0,38	0	8415,59	1177,67	1168	9593,26	5,38	0,06%	1,077
	WBRA-GL	714	84487732	6,16	18,47	8421,79	1163,83	1146	9585,62	3,27	0,03%	1,077
	Snay	107539	310401674	554,91	9,82	9001,44	1372,94	1003	10374,38	8,44	0,08%	0,945
	Snay-GL	107563	310418685	48,78	5,52	8997,83	1362,75	997	10360,58	18,90	0,18%	0,992
	Sloan	2243	55586225	1,22	5,60	9070,26	1392,47	1012	10462,73	14,92	0,14%	0,987
	NSloan	908	77425832	0,26	5,60	9054,98	1535,26	1086	10590,24	66,37	0,63%	0,975
	Sloan-MGPS	2094	57033429	7,39	5,60	9206,05	1398,90	1001	10604,95	68,30	0,64%	0,973
MPG	6920	89231836	0,81	5,60	9113,18	1575,63	1128	10688,81	43,96	0,41%	0,966	

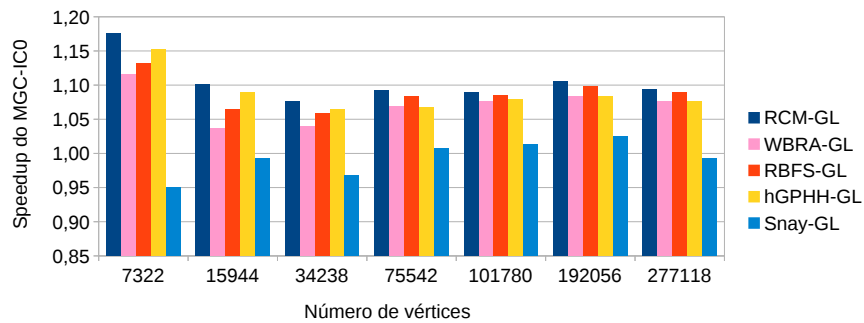


Figura 38 Speedup do MGC-IC0 obtido com aplicação das 5 heurísticas com melhores resultados, nas instâncias originadas de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada de forma aleatória, mostrado nas tabelas 16 e 17.

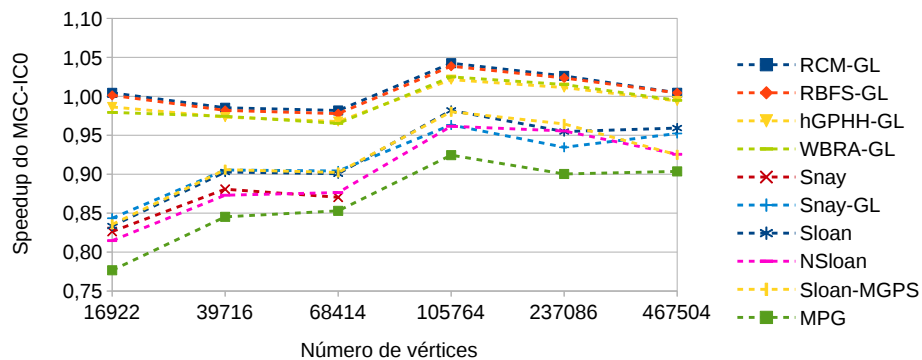


Figura 39 Speedup do MGC-IC0 obtido com aplicação das heurísticas, nas instâncias originadas de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada pela curva de Sierpiński modificada, mostrado nas tabelas 18 e 19. Os testes realizados não foram contínuos, e as linhas tracejadas indicam tendência.

Tabela 18 Resolução de sistemas de equações lineares (de até 68,414 incógnitas) pelo MGC-IC0, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de *profile*. Essas simulações foram realizadas na máquina M2 e esses sistemas de equações lineares foram originados de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada pela curva de Sierpiński modificada.

n	Heurística	β	<i>profile</i>	Heurística		IC0	MGC	MGC-IC0		σ	C_v	Speedup
				t(s)	m.(MiB)	t(s)	t(s)	iter.	t(s)			
16922	Sem renum,	16921	1710910	-	-	33,24	25,79	295	59,06	0,14	0,25%	-
	RCM-GL	115	1253675	0,018	0	32,69	26,09	299	58,78	0,22	0,38%	1,004
	RBFS-GL	115	1255425	0,007	0	32,70	26,28	300	58,98	0,04	0,07%	1,001
	hGPHH-GL	118	1322460	0,019	0	32,72	27,14	311	59,86	0,08	0,13%	0,986
	WBRA-GL	176	1326189	0,176	1,55	32,88	27,25	311	60,13	0,49	0,81%	0,979
	Snay	5939	1505964	1,555	0,52	34,17	35,74	289	69,91	0,08	0,11%	0,826
	Snay-GL	5569	1529357	0,161	0,34	34,28	35,58	288	69,86	0,13	0,18%	0,843
	Sloan	571	1074341	0,025	0,34	34,45	36,39	293	70,84	0,16	0,23%	0,833
	NSloan	229	1336588	0,010	0,17	34,26	38,22	306	72,48	0,18	0,25%	0,815
	Sloan-MGPS	485	1093937	0,120	0,17	34,25	36,32	291	70,57	0,13	0,19%	0,835
MPG	1214	1752814	0,031	0,34	34,53	41,49	332	76,01	0,19	0,25%	0,777	
39716	Sem renum,	38715	6309342	-	-	191,46	92,62	437	283,72	0,11	0,04%	-
	RCM-GL	195	4377424	0,040	0	195,05	92,86	446	287,91	0,13	0,05%	0,985
	RBFS-GL	198	4380812	0,020	0	195,34	93,58	448	288,92	0,48	0,16%	0,982
	hGPHH-GL	197	4773104	0,040	0	194,25	97,22	467	291,46	0,34	0,12%	0,973
	WBRA-GL	280	4762429	0,560	3,35	195,02	95,60	458	290,63	0,27	0,09%	0,974
	Snay	22597	6548607	8,630	1,18	187,98	125,52	424	313,50	0,28	0,09%	0,881
	Snay-GL	22573	6567001	0,870	1,03	187,82	124,68	421	312,50	0,20	0,06%	0,905
	Sloan	830	3342182	0,080	1,03	187,89	126,29	426	314,18	0,50	0,16%	0,903
	NSloan	364	4632710	0,030	1,03	188,36	136,61	455	324,96	0,28	0,08%	0,873
	Sloan-MGPS	831	3461255	0,430	1,03	187,72	125,06	419	312,77	0,46	0,15%	0,906
MPG	2358	5222214	0,080	1,03	189,70	145,88	489	335,58	0,11	0,03%	0,845	
68414	Sem renum,	68413	14882117	-	-	621,85	205,26	567	827,12	0,52	0,06%	-
	RCM-GL	238	9598308	0,080	0	624,12	218,25	589	842,37	0,33	0,04%	0,982
	RBFS-GL	241	9608251	0,040	0	625,43	220,47	593	845,90	1,58	0,19%	0,978
	hGPHH-GL	238	10712844	0,090	0	622,24	232,47	628	854,71	2,07	0,24%	0,968
	WBRA-GL	315	10703573	1,220	5,15	625,37	230,36	620	855,73	1,08	0,13%	0,965
	Snay	43837	21823074	42,100	2,22	625,85	282,38	549	908,23	0,54	0,06%	0,870
	Snay-GL	43854	21828427	4,210	1,46	626,16	284,38	553	910,55	3,32	0,36%	0,904
	Sloan	1284	7093207	0,170	1,03	627,52	290,70	562	918,22	1,04	0,11%	0,901
	NSloan	446	10132670	0,060	1,03	629,22	314,48	599	943,70	1,59	0,17%	0,876
	Sloan-MGPS	1082	7326579	0,970	1,46	627,47	288,27	555	915,74	1,08	0,12%	0,902
MPG	2811	11460778	0,160	1,03	637,07	332,47	642	969,54	1,29	0,13%	0,853	

Tabela 19 Resolução de sistemas de equações lineares (de 105,764 a 467,504 incógnitas) pelo MGC-IC0, com renumeração de vértices dada pelas heurísticas para reduções de largura de banda e de *profile*. Essas simulações foram realizadas na máquina M2 e esses sistemas de equações lineares foram originados de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada pela curva de Sierpiński modificada.

n	Heurística	β	<i>profile</i>	Heurística		IC0	MGC	MGC-IC0		σ	C_r	Speedup
				t(s)	m.(MiB)			t(s)	iter.			
105764	Sem renum,	105763	29560801	-	-	1708,76	396,11	691	2104,87	1,82	0,09%	-
	RCM-GL	306	18161623	0,13	0	1612,34	406,62	705	2018,96	2,23	0,11%	1,042
	RBFS-GL	306	18183385	0,07	0	1609,20	417,28	721	2026,48	1,23	0,06%	1,039
	hGPHH-GL	308	20752577	0,15	0	1604,55	456,04	790	2060,59	3,18	0,15%	1,021
	WBRA-GL	522	20711096	2,28	8,25	1608,44	442,96	766	2051,40	1,31	0,06%	1,025
	Snay-GL	4982	48279650	9,06	2,40	1645,45	530,72	674	2176,17	3,12	0,14%	0,963
	Sloan	1756	13247712	0,32	2,32	1602,37	540,96	677	2143,33	1,62	0,08%	0,982
	NSloan	602	19321158	0,10	2,40	1592,96	595,91	732	2188,87	6,38	0,29%	0,962
	Sloan-MGPS	1516	13690853	1,87	2,40	1605,13	540,45	671	2145,58	3,56	0,17%	0,980
	MPG	4447	20523176	0,27	2,40	1669,39	607,16	759	2276,55	1,93	0,08%	0,924
237086	Sem renum,	237085	115804392	-	-	8532,65	1228,89	956	9761,54	4,66	0,05%	-
	RCM-GL	391	56621430	0,29	0	8223,28	1288,89	991	9512,17	2,56	0,03%	1,026
	RBFS-GL	392	56725752	0,15	0	8217,34	1319,85	1014	9537,19	5,65	0,06%	1,024
	hGPHH-GL	398	64285994	0,34	0	8223,60	1429,04	1102	9652,64	5,12	0,05%	1,011
	WBRA-GL	672	64236202	6,87	15,47	8206,33	1402,49	1078	9608,82	7,11	0,07%	1,015
	Snay-GL	116472	215858651	48,28	6,51	8711,43	1685,10	933	10396,53	7,12	0,07%	0,935
	Sloan	2044	41300756	1,01	6,53	8530,52	1692,31	949	10222,83	15,36	0,15%	0,955
	NSloan	812	58999595	0,24	6,44	8347,22	1868,54	1020	10215,76	56,78	0,56%	0,956
	Sloan-MGPS	1898	42561396	6,12	6,45	8427,26	1687,15	935	10114,41	26,85	0,27%	0,965
	MPG	5707	64716730	0,76	6,45	8894,88	1946,72	1063	10841,60	11,18	0,10%	0,900
467504	Sem renum,	467503	382386929	-	-	32586,49	2948,99	1180	35535,49	8,21	0,02%	-
	RCM-GL	448	130166482	0,58	0	32069,17	3305,69	1321	35374,85	29,53	0,08%	1,005
	RBFS-GL	450	130399787	0,31	0	32114,15	3265,03	1304	35379,18	11,09	0,03%	1,004
	hGPHH-GL	452	149122278	0,67	0	31984,63	3762,82	1473	35747,46	12,43	0,03%	0,994
	WBRA-GL	722	148936289	16,04	25,97	32106,75	3595,98	1438	35702,73	3,07	0,01%	0,995
	Snay-GL	133459	911700269	238,78	11,06	32920,95	4156,13	1187	37077,08	513,99	1,39%	0,952
	Sloan	2498	95551358	2,34	11,38	32804,87	4238,19	1209	37043,06	44,67	0,12%	0,959
	NSloan	914	135038141	0,47	11,23	33583,17	4808,06	1317	38391,23	376,48	0,98%	0,926
	Sloan-MGPS	2291	98184536	14,22	11,27	34070,09	4335,46	1218	38405,55	229,96	0,60%	0,925
	MPG	7842	152632760	1,64	11,02	34432,05	4891,75	1379	39323,80	606,08	1,54%	0,904

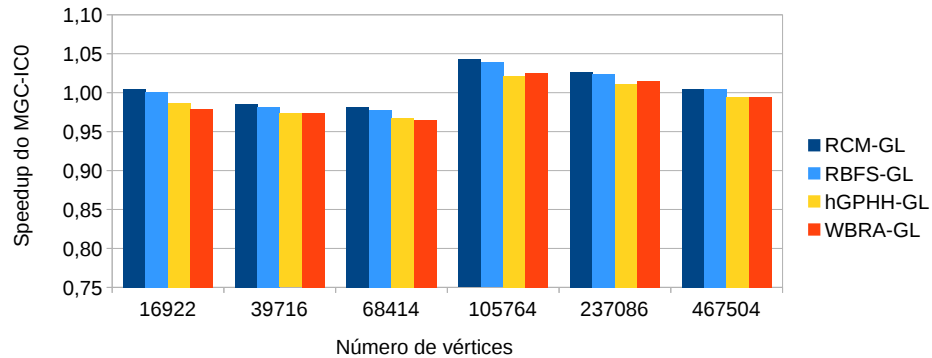


Figura 40 Speedup do MGC-IC0 obtido com aplicação das 4 heurísticas com melhores resultados, nas instâncias originadas de discretizações da equação de Laplace por volumes finitos, com numeração das linhas dada pela curva de Sierpiński modificada, mostrado nas tabelas 18 e 19.

computacional do MGC-IC0, foram superados pelos resultados da heurística Snay-GL, conforme pode ser verificado nas tabelas 16 e 18. As ocupações de memórias das heurísticas foram similares às ocupações de memórias ilustradas nas figuras 31 e 33 mostradas na subseção 5.1.2, uma vez que as instâncias utilizadas foram as mesmas.

É possível verificar que, de maneira geral, as reduções do custo computacional do MGC-IC0 com as reduções de largura de banda e de *profile* nas instâncias oriundas da discretização da equação de Laplace, ordenadas pela curva de Siérpinski modificada, foram significativamente menores em relação aos resultados obtidos com as instâncias ordenadas de forma aleatória. Aparentemente, o preconditionador IC0 gera uma matriz preconditionadora melhor quando o sistema de equações lineares é montado por uma curva de preenchimento de espaço, como a curva de Sierpiński modificada (considerando que a curva de preenchimento de espaço proporciona localidade de memória). Isso porque, com as renumerações

realizadas pelas heurísticas, o número de iterações do MGC-IC0 aumentou em quase todas as instâncias.

5.2.3 Instâncias da *University of Florida sparse-matrix collection*

No conjunto de instâncias da *University of Florida sparse-matrix collection*, foram testadas 7 heurísticas para reduções de largura de banda e de *profile*: RCM-GL (GEORGE; LIU, 1981), RBFS-GL (CHAGAS, 2015), hGPHH-GL (KOOHESTANI; POLI, 2011), Snay-GL, de Sloan (SLOAN, 1989), Sloan-MGPS (REID e SCOTT, 1999) e MPG (MEDEIROS; PIMENTA; GOLDENBERG, 1993), pois foram as heurísticas que apresentaram os melhores resultados no conjunto de instâncias oriundas da discretização da equação do calor e nos conjuntos de instâncias oriundas da discretização da equação de Laplace. É possível verificar, nos resultados mostrados nas tabelas 14 e 15, que a heurística Snay foi dominada pela heurística Snay-GL quando aplicada para redução do custo computacional do MGC-IC0. Em particular, as heurísticas Sloan, MPG, Sloan-MGPS e Snay-GL apresentaram resultados competitivos em relação à redução do custo computacional do MGC-IC0 nas instâncias oriundas da discretização da equação da condução do calor, como pode ser observado nos resultados mostrados nas tabelas 14 e 15.

Os resultados médios obtidos pelo MGC-IC0 nessas 11 instâncias são mostrados nas tabelas 20 e 21. A heurística Snay-GL obteve os melhores resultados em relação à redução do custo computacional do MGC-IC0 quando aplicada às instâncias *nasa4704*, *Pres_Poisson*, e *thermal1*. A heurística hGPHH-GL obteve os melhores resultados quando aplicada

às instâncias *nasa2910*, *2cubes_sphere*, e *offshore*. A heurística Sloan-MGPS obteve os melhores resultados quando aplicada às instâncias *raefsky4* (obtendo um fator de *speedup* de 27,147) e *olafu*. A heurística RBFS-GL foi a única heurística resultou em uma redução do custo computacional do MGC-IC0 quando aplicado na instância *sts4098*. Nas instâncias *nasa1824* e *nasasrb*, não foram obtidos fatores positivos de *speedup* pelas heurísticas. Além disso, a partir dos resultados obtidos nas instâncias *nasa1824* e *nasasrb*, é possível verificar que fatores negativos de *speedup* podem ser obtidos não for usada uma heurística adequada para a reordenação da instância em que está sendo aplicada.

5.3 Resumo dos resultados das resoluções de sistemas de equações lineares com o MGCPJ e com o MGC-IC0

Na tabela 22, é apresentado um resumo dos resultados das resoluções de sistemas de equações lineares, com o MGCPJ e com o MGC-IC0, em cada conjunto de instâncias utilizado. Para cada conjunto de instâncias, são indicadas as heurísticas que proporcionaram o maior valor de *speedup* para o MGCPJ e para o MGC-IC0. Ainda, para o conjunto de instâncias da *University of Florida sparse-matrix collection*, é mostrado o número de instâncias em que as heurísticas indicadas proporcionaram a maior redução do custo computacional do MGCPJ e do MGC-IC0.

Tabela 20 Resultados de resoluções de sistemas de equações lineares, com o MGC-IC0, e com renumeração dos vértices dada pelas heurísticas para reduções de largura de *profile*, oriundos das instâncias *nasa1824*, *nasa2910*, *sts4098*, *nasa4704*, *Pres_Poisson* e *olafu* do conjunto de dados da *University of Florida sparse-matrix collection*. Estas simulações foram realizadas nas máquinas M1 (*olafu*) e M3 (*nasa1824*, *nasa2910*, *sts4098*, *nasa4704* e *Pres_Poisson*).

Inst.	Heurística	β	<i>profile</i>	Heurística		IC0	MGC	MGC-IC0		σ	C _v	Speedup
				t(s)	m.(MiB)			t(s)	iter.			
<i>nasa1824</i>	Sem renum.	239	205547			0,89	5,08	231	5,97	0,15	2,49%	-
	RCM-GL	277	233341	0,006	0	1,00	15,15	680	16,15	3,63	22,46%	0,370
	RBFS-GL	282	231228	0,002	0	0,92	12,80	577	13,72	5,08	37,05%	0,435
	hGPHH-GL	294	295102	0,003	0	0,87	9,91	447	10,78	0,61	5,68%	0,553
	WBRA-GL	239	205547	0,013	0,77	0,91	5,11	231	6,02	0,21	3,49%	0,990
	Snay-GL	1475	300980	0,026	0	0,99	8,97	401	9,96	0,26	2,63%	0,598
	Sloan	1288	185878	0,004	0	0,98	13,98	627	14,96	2,00	13,38%	0,399
	Sloan-MGPS	1164	189173	0,010	0	0,96	14,29	642	15,25	2,54	16,64%	0,391
	MPG	1510	525015	0,011	0	0,95	9,89	447	10,84	0,03	0,25%	0,550
<i>nasa2910</i>	Sem renum.	859	525745			6,13	253,36	2910	259,49	0,50	0,19%	-
	RCM-GL	879	523616	0,020	0	6,09	254,16	2910	260,25	2,07	0,79%	0,997
	RBFS-GL	868	531951	0,010	0	6,10	252,18	2910	258,28	1,67	0,65%	1,005
	hGPHH-GL	869	1289065	0,020	0	5,64	250,65	2910	256,29	0,95	0,37%	1,012
	WBRA-GL	859	525745	0,510	24,83	6,13	252,56	2910	258,69	1,37	0,53%	1,001
	Snay-GL	2700	2903494	0,170	0	6,46	254,94	2910	261,40	0,08	0,03%	0,992
	Sloan	2015	456292	0,020	0	6,17	251,74	2910	257,91	1,17	0,45%	1,006
	Sloan-MGPS	2010	458040	0,020	0	6,08	250,47	2910	256,55	1,09	0,42%	1,011
	MPG	2697	2295308	0,040	0	6,11	256,33	2910	262,44	1,41	0,54%	0,989
<i>sts4098</i>	Sem renum.	3323	5217389			2,92	6,03	129	8,95	0,30	3,30%	-
	RCM-GL	1165	2084237	0,007	0	3,53	9,05	198	12,58	0,12	0,95%	0,710
	RBFS-GL	1343	2110715	0,002	0	3,68	5,15	115	8,83	0,20	2,21%	1,012
	hGPHH-GL	1171	2981815	0,007	0	3,27	32,36	725	35,63	0,18	0,52%	0,251
	WBRA-GL	1741	3612966	0,627	31,45	3,15	8,11	182	11,27	0,40	3,56%	0,751
	Snay-GL	3928	1955574	0,138	0	4,20	41,53	923	45,73	0,20	0,43%	0,195
	Sloan	3471	529840	0,020	0	3,95	37,39	834	41,34	3,15	7,61%	0,216
	Sloan-MGPS	3787	526731	0,080	0	3,93	28,32	633	32,25	3,46	10,74%	0,277
	MPG	3729	961548	0,020	0	4,09	34,99	781	39,08	0,49	1,25%	0,229
<i>nasa4704</i>	Sem renum.	423	917562			6,46	127,79	2041	134,25	0,10	0,08%	-
	RCM-GL	419	918199	0,009	0	5,96	84,37	1345	90,33	0,40	0,45%	1,486
	RBFS-GL	419	918078	0,007	0	6,01	139,94	2243	145,95	0,98	0,67%	0,920
	hGPHH-GL	450	1079787	0,008	0	5,95	122,77	1972	128,72	0,54	0,42%	1,043
	WBRA-GL	423	917562	0,150	6,01	5,97	127,65	2041	133,62	1,00	0,75%	1,004
	Snay-GL	4009	1195778	0,119	0,09	6,49	35,72	575	42,21	6,79	16,09%	3,172
	Sloan	3084	834575	0,022	0,09	6,49	244,42	3931	250,91	1,05	0,42%	0,535
	Sloan-MGPS	2750	808709	0,053	0,09	6,42	285,92	4605	292,34	11,73	4,01%	0,459
	MPG	3696	2739606	0,068	0,09	6,25	136,31	2190	142,56	6,93	4,86%	0,941
<i>Pres_Poisson</i>	Sem renum.	12583	9789525			139,91	111,13	279	251,04	6,13	2,44%	-
	RCM-GL	365	3092536	0,060	0	141,34	50,72	131	192,06	0,34	0,18%	1,307
	RBFS-GL	331	3024736	0,020	0	137,78	104,38	269	242,16	1,01	0,42%	1,037
	hGPHH-GL	292	2999328	0,060	0	140,23	60,13	155	200,36	0,24	0,12%	1,253
	WBRA-GL	531	3088964	0,990	1,23	140,14	69,40	178	209,54	1,14	0,55%	1,192
	Snay-GL	1781	8247803	0,740	0	140,18	40,49	101	180,67	0,02	0,01%	1,384
	Sloan	586	2829459	0,070	0	140,46	53,88	139	194,34	0,39	0,20%	1,291
	Sloan-MGPS	697	2830899	0,160	0	140,63	53,95	139	194,58	0,25	0,13%	1,289
	MPG	13841	25901484	4,080	0	139,92	60,63	156	200,55	0,25	0,13%	1,227
<i>olafu</i>	Sem renum.	593	4951980			201,42	8477,77	16146	8679,19	10,80	0,12%	-
	RCM-GL	553	5029301	0,140	0	204,01	8291,91	16146	8495,92	51,86	0,61%	1,022
	RBFS-GL	582	5025044	0,060	0	202,86	8298,90	16146	8501,76	10,32	0,12%	1,021
	hGPHH-GL	532	5213195	0,130	0	200,81	8211,87	16146	8412,68	26,86	0,32%	1,032
	WBRA-GL	593	4951980	1,910	11,34	202,16	8318,54	16146	8520,71	36,15	0,42%	1,018
	Snay-GL	5709	7141254	0,690	0,17	214,55	8304,82	16146	8519,37	2,38	0,03%	1,019
	Sloan	5533	4494210	0,160	0,17	202,57	8200,25	16146	8402,82	19,90	0,24%	1,033
	Sloan-MGPS	7775	4488774	0,160	0,26	202,42	8156,58	16146	8359,00	50,40	0,60%	1,038
	MPG	14467	29376748	4,020	0,34	207,08	8376,09	16146	8583,17	36,36	0,42%	1,011

Tabela 21 Resultados de resoluções de sistemas de equações lineares, com o MGC-IC0, e com renumeração dos vértices dada pelas heurísticas para reduções de largura de *profile*, oriundos das instâncias *raefsky4*, *nasasrb*, *thermal1*, *2cubes_sphere*, *offshore*, da *University of Florida sparse-matrix collection*. Estas simulações foram realizadas nas máquinas M3 (instâncias *nasasrb*, *raefsky4* e *thermal1*) e M6 (instâncias *2cubes_sphere* e *offshore*).

I. Heurística	β	<i>profile</i>	Heurística		IC0	MGC	MGC-IC0		σ	C, Speedup			
			t(s)	m.(MiB)			t(s)	iter.			t(s)		
<i>raefsky4</i>	Sem renum.	11786	19611188			324,18	13520,69	19779	13844,87	9,46	0,07%	-	
	RCM-GL	991	12553981	0,16	0	319,19	220,71	319	539,90	0,68	0,13%	25,636	
	RBFS-GL	1171	12251822	0,08	0,09	326,11	373,73	541	699,84	150,84	21,55%	19,781	
	hGPHH-GL	973	13195914	0,17	0	325,20	2599,76	3769	2924,96	0,37	0,01%	4,733	
	WBRA-GL	1909	13251290	5,03	46,83	318,91	10013,93	14615	10332,84	6104,22	59,08%	1,339	
	Sloan	17602	34715834	2,89	0,17	337,76	5835,41	8390	6173,17	1,81	0,03%	2,242	
	Sloan	8378	7813447	0,27	0	332,84	3741,64	5427	4074,48	2,03	0,05%	3,398	
	Sloan-MGPS	4626	8630077	0,37	0,17	315,99	193,64	280	509,63	0,45	0,09%	27,147	
	MPG	18290	76709235	7,49	0	320,29	3554,96	5105	3875,25	28,45	0,73%	3,566	
	<i>nasasrb</i>	Sem renum.	893	20311330			1874,03	75735,68	54870	77609,71	19,23	0,02%	-
RCM-GL		916	20536983	0,34	0	2273,96	76848,75	54870	79122,71	87,95	0,11%	0,981	
RBFS-GL		880	20560224	0,17	0,68	1856,53	76763,74	54870	78620,27	31,08	0,04%	0,987	
hGPHH-GL		916	20966663	0,38	0	1879,44	76705,31	54870	78584,75	48,89	0,06%	0,988	
WBRA-GL		893	20311330	6,08	20,02	1857,92	76499,38	54870	78357,3	55,93	0,07%	0,990	
Sloan		43323	365126109	36,85	1,37	1887,02	77935,52	54870	79822,54	209,6	0,26%	0,972	
Sloan		5063	19055047	0,43	1,29	1880,37	77013,09	54870	78893,46	179,96	0,23%	0,984	
Sloan-MGPS		4962	18805784	0,65	1,29	1879,12	76815,19	54870	78694,31	22,03	0,03%	0,986	
MPG		36925	342411948	13,16	1,29	1861,21	80765,06	54870	82626,27	33,84	0,04%	0,939	
<i>thermal1</i>		Sem renum.	228	12121283	0,10	0,00	1000,94	601,85	1468	1602,79	1,38	0,09%	-
	RCM-GL	241	12137584	0,05	0,86	981,67	1480,08	3480	2461,75	116,35	4,73%	0,651	
	RBFS-GL	235	12978967	0,12	0	975,41	2250,10	5291	3225,51	8,50	0,26%	0,497	
	hGPHH-GL	2738	20256126	1,19	2,32	916,72	43881,87	82654	44798,59	118,02	0,26%	0,036	
	Sloan	884	10584658	0,25	1,03	976,89	212,96	507	1189,85	3,89	0,33%	1,346	
	Sloan	693	10687441	0,88	2,58	979,45	214,96	510	1194,41	4,66	0,39%	1,342	
	Sloan-MGPS	2883	10727185	0,18	2,49	979,21	211,94	507	1191,15	3,40	0,29%	1,345	
	MPG	100407	483241271	22,98	1,89	971,71	33062,84	82654	34034,55	375,32	1,10%	0,047	
	<i>2cubes_sphere</i>	Sem renum.	100407	483241271			1967,48	11,63	13	1979,11	11,12	0,56%	-
		RCM-GL	4730	274402970	0,23	0	1957,50	10,36	12	1967,86	13,13	0,67%	1,006
RBFS-GL		4797	266978630	0,14	0	1962,11	10,36	12	1972,47	16,75	0,85%	1,003	
hGPHH-GL		4789	363309520	0,26	0	1935,83	11,48	13	1947,31	2,11	0,11%	1,016	
Sloan		25715	231940284	229,64	1,96	2011,27	9,69	11	2020,96	7,67	0,38%	0,879	
Sloan		10720	189423784	14,42	2,49	1992,05	10,16	12	2002,21	20,99	1,05%	0,981	
Sloan-MGPS		13367	201170041	5,65	2,49	2034,98	10,30	12	2045,28	1,65	0,08%	0,965	
Sloan		25715	231760179	20,36	2,49	2014,83	9,67	11	2024,50	8,80	0,43%	0,968	
MPG		97614	541448073	22,98	1,89	1928,67	10,98	12	1939,65	17,06	0,88%	1,008	
<i>offshore</i>		Sem renum.	237738	3588201815			13447,42	898,27	394	14345,69	5,48	0,04%	-
	RCM-GL	20535	2646186630	0,66	0	13002,06	981,62	432	13983,68	53,53	0,38%	1,026	
	RBFS-GL	21425	2670172280	0,35	0	12955,25	1008,24	447	13963,49	56,50	0,40%	1,027	
	hGPHH-GL	21594	3475148244	0,68	0	12700,19	938,00	411	13638,19	18,55	0,14%	1,052	
	WBRA-GL	27967	3527879744	8038,83	10074,33	12747,87	942,16	415	13690,03	15,51	0,11%	0,660	
	Sloan	99875	1184107609	155,3	5,94	13375,75	671,66	301	14047,41	56,19	0,40%	1,010	
	Sloan	121232	1890408046	69,83	5,76	13195,90	865,44	400	14061,34	32,57	0,23%	1,015	
	Sloan-MGPS	126023	1448051810	127,85	5,76	13425,48	476,17	212	13901,65	426,44	3,07%	1,023	
	MPG	256493	4132444959	231,48	5,67	12950,18	936,15	424	13886,33	59,13	0,43%	1,016	

Tabela 22 Heurísticas que proporcionaram o maior valor de *speedup* para o MGCPJ e para o MGC-IC0 em cada conjunto de instâncias.

Método	Equação da condução do calor	Equação de Laplace		<i>The University of Florida sparse-matrix collection</i> (número de instâncias)	Resultados mostrados na seção
		ordem aleatória	ordem dada pela curva de Sierpiński modificada		
MGCPJ	Snay-GL and MPG	-	-	-	5.1
MGC-IC0	RCM-GL	RCM-GL	RCM-GL	Snay-GL (3) hGPHH-GL (3) Sloan-MGPS (2) RBFS-GL (1)	5.2

6 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho, uma heurística alternativa para redução de *profile* foi proposta. Essa heurística, denominada Snay-GL, é baseada na heurística de Snay (1976) e foi proposta com o objetivo de reduzir o custo computacional da heurística de Snay (1976). Na heurística Snay-GL, é utilizado o método de George e Liu (1979) para redução do número de vértices pseudoperiféricos determinados para se iniciar a fase de renumeração.

Os resultados da heurística Snay-GL foram comparados com os resultados de outras 10 heurísticas para reduções de largura de banda de *profile*, na redução do custo computacional do Método dos Gradientes Conjuntados preconditionado, em quatro conjuntos de sistemas de equações lineares. Essas 10 heurísticas foram selecionadas em revisões sistemáticas de literatura, e podem ser consideradas como as potenciais heurísticas de baixo custo computacional no estado da arte para reduções de largura de banda e de *profile* (BERNARDES; GONZAGA DE OLIVEIRA, 2015; CHAGAS; GONZAGA DE OLIVEIRA, 2015; GONZAGA DE OLIVEIRA; CHAGAS, 2015a). As implementações das heurísticas realizadas neste trabalho obtiveram resultados comparáveis, em relação às implementações originais das heurísticas, em testes realizados com as mesmas instâncias utilizadas nos trabalhos originais. Além disso, os métodos de trocas adjacentes (REID; SCOTT, 2002) também foram avaliados com diversas heurísticas. Em particular, os resultados obtidos neste trabalho mostraram que esses métodos de trocas adjacentes, apesar de reduzir o *profile* das instâncias, não são efetivos para reduzir os custos computacionais do MGCPJ e do MGC-IC0.

A heurística Snay-GL, proposta neste trabalho, resultou no melhor

speedup do MGC-IC0 quando aplicada nas instâncias *nasa4704* (uma instância de engenharia estrutural), *Pres_Poisson* (um sistema de equações lineares originado de um problema de dinâmica de fluido computacional) e *thermal1* (uma instância originada de problema térmico de estado estacionário de elementos finitos não estruturados), contidas no conjunto de dados da *University of Florida sparse-matrix collection*; e nas instâncias de até 50.592 incógnitas oriundas da discretização da equação da condução do calor. Ainda, a heurística Snay-GL resultou no melhor *speedup* do MGCPJ quando aplicado na instância de 232.052 incógnitas oriunda da discretização da equação da condução do calor. Projetada como uma heurística para redução de *profile*, essa heurística reduz largamente o *profile* da matriz com custos baixos de processamento e armazenamento.

Em simulações com três conjuntos de dados, oriundos de discretizações das equações de Laplace e da condução do calor por volumes finitos) e compostos de quase 1.000.000 de incógnitas, a heurística *in-place* RCM-GL (GEORGE; LIU, 1981) obteve os melhores resultados, em média, em relação à redução do custo computacional do MGC-IC0. É necessário ressaltar que essa heurística está contida no *software* MATLAB (<<http://www.mathworks.com/products/matlab/>>).

Já em relação à redução do custo computacional do MGCPJ, as heurísticas MPG (MEDEIROS; PIMENTA; GOLDENBERG, 1993) e Snay-GL, proposta neste trabalho, obtiveram os melhores resultados nas duas maiores instâncias oriundas da discretização da equação da condução do calor. Porém, não é possível afirmar que há uma heurística única e ótima que supera todas as demais heurísticas em todas as instâncias. Por outro lado, nos experimentos com os conjuntos de instâncias oriundas da discretização

da equação de Laplace e da *University of Florida sparse-matrix collection*, houve valores positivos de *speedup* do MGCPJ apenas em duas instâncias, e as heurística hGPHH-GL e RBFS-GL obteve os melhores resultados nesses casos, respectivamente. Nas outras instâncias foram obtidos valores negativos de *speedup* do MGCPJ ao aplicar as heurísticas para reduções de largura de banda e de *profile*. Especificamente nas instâncias oriundas da discretização da equação de Laplace por volumes finitos, com numeração dada na forma aleatória, cada vértice possui exatamente três adjacências (GONZAGA DE OLIVEIRA; KISCHINHEVSKY; TAVARES, 2013). Com isso, provavelmente, a renumeração dos vértices não aumenta o número de *cache hits* na execução do MGCPJ. Além disso, um número grande de *cache misses* do MGCPJ pode ter ocorrido quando as heurísticas foram aplicadas nas instâncias oriundas da discretização da equação de Laplace por volumes finitos, com numeração dada pela curva de Sierpiński (GONZAGA DE OLIVEIRA; KISCHINHEVSKY; TAVARES, 2013). Provavelmente, a razão para isso é que a curva de preenchimento de espaço proporciona uma localidade adequada de dados na memória para o MGCPJ e, então, heurísticas de renumeração não são úteis nesse caso. Em relação às instâncias da *University of Florida sparse-matrix collection*, não foi possível ter acesso aos detalhes da geração dessas instâncias, tais como a forma como foram numeradas e a quantidade média de adjacências. Apesar disso, a partir das evidências obtidas com os experimentos realizados neste trabalho, é possível concluir que o sistema de equações lineares deve ser estudado antes de se aplicar uma heurística para redução de largura de banda ou de *profile* com o objetivo de reduzir o custo computacional do MGCPJ (ou do MGC-IC0 e, provavelmente, de outro resolutor de sistemas de equações lineares).

Conforme mostrado pelo experimento computacional descrito neste trabalho, de forma conclusiva, a escolha de uma heurísticas é altamente dependente da utilização de uma instância específica. Por exemplo, em simulações com 11 sistemas de equações lineares contidos no conjunto de dados da *University of Florida sparse-matrix collection* (DAVIS; HU, 2011), o melhor fator de *speedup* do MGC-IC0 foi obtido pelas heurísticas Snay-GL (em três instâncias), hGPHH-GL (em três instâncias), Sloan-MGPS (em duas instâncias) e RBFS-GL (em uma instância), e nenhuma redução foi obtida quando heurísticas para reduções de largura de banda e de *profile* foram aplicadas em duas instâncias testadas (*nasa1824* e *nasasrb*).

O experimento computacional descrito neste trabalho não garante a superioridade de uma heurística sobre as demais em relação à renumeração dos vértices de um grafo. Existem diversos sistemas de equações lineares diferentes, de forma que compreende diversas aplicações e métodos numéricos. De forma consistente com a literatura, a heurística ideal para renumeração de vértices depende largamente da estrutura da matriz.

O uso de uma heurística que reduz razoavelmente a largura de banda ou o *profile* da matriz com custos baixos de processamento e armazenamento, tal como no caso das heurísticas mencionadas, pode ser melhor que usar uma heurística que reduz largamente a largura de banda ou o *profile* da matriz, mas com custos altos de processamento e armazenamento. Em particular, uma heurística para redução de *profile*, tal como a heurística Snay-GL, pode ser muito útil se uma estrutura de dados dependente do *profile* for utilizada, como a estrutura *skyline* (JENNINGS, 1966; FELIPPA, 1975). Além disso, mesmo se uma heurística baseada em meta-heurística reduz mais a largura de banda e o *profile* da matriz e,

portanto, reduz o custo de armazenamento do sistema de equações lineares (dependendo da estrutura de dados utilizada), um possível consumo alto de memória de tal heurística pode limitar o uso da mesma em estações de trabalho com uma quantidade pequena de memória principal.

Em particular, o maior número de heurísticas que não são baseadas em meta-heurísticas foi proposto na década de 1970, e esse número tem diminuído (CHAGAS, 2015). Nos últimos 20 anos, a tendência é projetar heurísticas baseadas em meta-heurísticas para reduções de largura de banda e de *profile* (CHAGAS; GONZAGA DE OLIVEIRA, 2015; BERNARDES; GONZAGA DE OLIVEIRA, 2015). As simulações apresentadas neste trabalho proporciona a evidência de que essa tendência pode não ser uma estratégia adequada para investigar esse tópico. Por outro lado, na tabela 22 são mostrados as seis possíveis heurísticas no estado da arte em relação à redução do custo computacional do MGCPJ e do MGC-IC0 (e, possivelmente, também em relação ao MGC preconditionado ou até mesmo em relação a outro resolutor iterativo preconditionado). Então, para se obter reduções razoáveis de largura de banda e de *profile* em muitos casos, a heurística mais simples (de ser implementada e mantida) e menos custosa (em relação ao tempo de execução e ao custo de armazenamento) é sugerida para ser implementada em uma nova aplicação que utiliza um resolutor iterativo e preconditionado. Assim, sugere-se a escolha da heurística nesta ordem: as heurísticas RBFS-GL, hGPHH-GL, RCM-GL, Snay-GL, Sloan-MGPS e MPG.

Como continuação deste trabalho, espera-se implementar e avaliar os seguintes preconditionadores: Multigrid Algébrico, ILUT, *Successive Over-Relaxation* (SOR), SOR Simétrico (SSOR) e Gauss-Seidel. Mais

detalhadamente, espera-se examinar a efetividade dos esquemas com computação de preconditionadores baseados em fatoração incompleta ou aproximada, bem como preconditionadores inversos aproximados. Essas técnicas devem ser usadas como preconditionadores do Método dos Gradientes Conjugados e do método *Generalized Minimal Residual* (GMRES) (SAAD; SCHULTZ, 1986) para avaliar o seu desempenho computacional em conjunto com heurísticas de reordenação. Abordagens paralelas dos métodos acima também serão investigadas.

Precisões estendidas (256-bit e 512-bit) foram utilizadas neste trabalho. Claramente, essas precisões reduzem erros de arredondamento. Porém, precisões como essas aumentam o tempo de execução em um fator grande, e esse tempo pode não ser aceitável em resoluções de problemas reais. Espera-se analisar o efeito da aritmética de dupla precisão em estudos futuros.

REFERÊNCIAS

- AJIZ, M. A.; JENNINGS, A. A robust incomplete Choleski-conjugate gradient algorithm. **International Journal for Numerical Methods in Engineering**, v. 20, n.5, p. 949-966, 1984.
- BENZI, M. Preconditioning techniques for large linear systems: a survey. **Journal of Computational Physics**, v. 182, n. 2, p. 418-477, 2002.
- BENZI, M.; TUMA, M. A robust incomplete factorization preconditioner for positive definite matrices. **Numerical Linear Algebra with Applications**, v. 10, n.5, p. 385-400, 2003.
- BERNARDES, J. A. B.; GONZAGA DE OLIVEIRA, S. L. G. A systematic review of heuristics for profile reduction of symmetric matrices. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE, 2015, Reykjavík. **Proceedings...** Reykjavík: ICCS, 2015. p. 221-230.
- BURDEN, R. L.; FAIRES, J. D. **Numerical Analysis**. 10. ed. Boston: Brooks/Cole Cengage Learning, 2010. 877 p.
- BURGESS, D. A.; GILES, M. B. Renumbering unstructured grids to improve the performance of codes on hierarchical memory machines. **Advances in Engineering Software**, v. 28, p. 189-201, 1997.
- CHAGAS, G. C. **Avaliação de heurísticas para a redução de largura de banda para a aceleração da convergência do método dos gradientes conjugados**. 160 p. Dissertação (Mestrado em Ciência da Computação)-Universidade Federal de Lavras, Lavras, 2015.
- CHAGAS, G. O.; GONZAGA DE OLIVEIRA, S. L. Metaheuristic-based heuristics for symmetric-matrix bandwidth reduction: a systematic review. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE, 2015, Reykjavík, Iceland. **Proceedings...** Reykjavík: ICCS, 2015.

CUTHILL, E.; MCKEE, J. Reducing the bandwidth of sparse symmetric matrices. In: NATIONAL CONFERENCE, 24., 1969, New York.

Proceedings... New York: ACM, 1969. p. 157-172.

DAS, R. et al. The design and implementation of a parallel unstructured Euler solver using software primitive. **AIAA Journal**, v. 32, n. 3, p. 489-496, 1994.

DAVIS, T. A.; HU, Y. The University of Florida sparse matrix collection. **ACM Transactions on Mathematical Software**, v. 38, n.1, p. 1-25, 2011.

DUFF, I. S.; MEURANT, G. A. The effect of ordering on preconditioned conjugate gradients. **BIT Numerical Mathematics**, Berlin, v. 29, n. 4, p. 635-657, 1989.

DUFF, I. S.; REID, J. K.; SCOTT, J. A. The use of profile reduction algorithms with a frontal code. **International Journal for Numerical Methods in Engineering**, v. 28, p. 2555-2568, 1989.

FELIPPA, C. A. Solution of linear equations with skyline-stored symmetric matrix. **Computers and Structures**, v. 5, n. 1, p. 13-29, 1975.

FENVES, S. J.; LAW, K. H. A two-step approach to finite element ordering. **International Journal for Numerical Methods in Engineering**, v. 19, n. 6, p. 891-911, 1983.

GEORGE, J. A. **Computer implementation of the finite element method**. 1971. 228 p. Tese (Doutorado)-Stanford University, Stanford, 1971.

GEORGE, J. A.; LIU, J. W. H. An implementation of a pseudoperipheral node finder. **ACM Transactions on Mathematical Software**, New York, v. 5, n. 3, p. 284-295, 1979.

GEORGE, A.; LIU, J. W. H. **Computer Solution of Large Sparse Positive Definite Systems**. New York: Prentice-Hall, 1981. 314 p.

GIBBS, N. E.; POOLE, W. G.; STOCKMEYER, P. K. Algorithm for reducing the bandwidth and profile of a sparse matrix. **SIAM Journal on Numerical Analysis**, Philadelphia, v. 13, n. 2, p. 236-250, 1976.

GOLUB, G. H.; VAN LOAN, C. F. **Matrix Computations**. 3. ed. Baltimore: The Johns Hopkins University, 1996. 687 p.

GONZAGA DE OLIVEIRA, S. L. **Algoritmos e seus Fundamentos**. Lavras: Ed. UFLA, 2011. 420 p.

GONZAGA DE OLIVEIRA, S. L. **Introdução à geração de malhas triangulares**. São Carlos: SBMAC, 2015. 94 p. Disponível em: <http://www.sbmac.org.br/arquivos/notas/livro_79.pdf>. Acesso em: 23 jun. 2016.

GONZAGA DE OLIVEIRA, S. L.; CHAGAS, G. O. A systematic review of heuristics for symmetric-matrix bandwidth reduction methods not based on metaheuristics. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 2015, Porto de Galinhas. **Anais...** Porto de Galinhas: SBPO, 2015a. p. 3304-3315.

GONZAGA DE OLIVEIRA, S. L.; CHAGAS, G. O. **Introdução a heurísticas para redução de largura de banda de matrizes**. São Carlos: SBMAC, 2014. 106 p. Disponível em: <http://www.sbmac.org.br/arquivos/notas/livro_75.pdf>. Acesso em: 23 jun. 2016.

GONZAGA DE OLIVEIRA, S. L.; KISCHINHEVSKY, M. Sierpiński curve for total ordering of a graph-based adaptive simplicial-mesh refinement for finite volume discretizations. In: CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL, 2008, Belém. **Anais...** Belém: CNMAC, 2008. p. 581-585.

GONZAGA DE OLIVEIRA, S. L.; KISCHINHEVSKY, M.; TAVARES, J. M. R. S. Novel graph-based adaptive triangular mesh refinement for finite-volume discretizations. **Computer Modeling in Engineering and Science**, Porto, v.95, n. 2, p. 119-141, 2013.

GONZAGA DE OLIVEIRA, S. L.; OLIVEIRA, F. S.; CHAGAS, G. O. A novel approach to the weighted Laplacian formulation applied to 2D Delaunay triangulations. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE AND ITS APPLICATIONS, 2015, Banff. **Proceedings...** Banff: ICCSA, 2015.

HAGER, W. W. Minimizing the profile for a symmetric matrix. **SIAM Journal on Scientific Computing**, v. 23, n. 5, p. 1799-1816, 2002.

HENDRICKSON, B.; LELAND, R. A Multi-Level Algorithm For Partitioning Graphs. IN: IEEE/ACM Supercomputing Conference, San Diego, CA, USA. **Proceedings...** San Diego: SC95, 1995.

HESTENES, M. R.; STIEFEL, E. Methods of Conjugate Gradients for Solving Linear Systems. **Journal of Research of the National Bureau of Standards**, Essex, v. 49, n. 36, p. 409-436, 1952.

HSL Mathematical Software Library. Oxfordshire: STFC Rutherford Appleton Laboratory, 1963-2016. Disponível em: <<http://www.hsl.rl.ac.uk>>. Acesso em: 23 jun. 2016.

HU, Y. Efficient and high quality force-directed graph drawing. **The Mathematica Journal**, v. 10, p. 37-71, 2005.

HU, Y. **A Gallery of Large Graphs**. Disponível em: <<http://yifanhu.net/GALLERY/GRAPHS/index.html>>. Acesso em: 10 jan. 2016.

HU, Y.; SCOTT, J. A. A multilevel algorithm for wavefront reduction. **SIAM Journal on Scientific Computing**, v. 23, n. 4, p. 1352-1375, 2001.

JACOBI, C. G. J. Über eine neue Auflösungsart der bei der methode der kleinsten quadrate vorkommenden linearen gleichungen, astron. **Nachrichten**, Stuttgart, v. 22, p. 297, 1845.

JENNINGS, A. A Compact Storage Scheme for the Solution of Symmetric Linear Simultaneous Equations. **The Computer Journal**, v. 9, n. 3, p. 281-285, 1966.

KERSHAW, D. S. A The incomplete Cholesky-conjugate gradient method for the iterative solution of systems of linear equations. **Journal of Computational Physics**, v. 26, n. 1, p. 43-65, 1978.

KOOHESTANI, B.; POLI, R. A genetic programming approach to the matrix bandwidth-minimization problem. In: INTERNATIONAL CONFERENCE ON PARALLEL PROBLEM SOLVING FROM NATURE, 11th, 2011, Berlim. **Proceedings...** Berlim: PPSN, 2011. p. 482-491.

KUMFERT, G.; POTHEN, A. Two improved algorithms for envelope and wavefront reduction. **BIT Numerical Mathematics**, v. 37, n. 3, p. 559-590, 1997.

LANCZOS, C. Solutions of systems of linear equations by minimized iterations. **Journal of Research of the National Bureau of Standards**, Washington, v. 49, n. 3, p. 33-53, 1952.

LIN, Y. X.; YUAN, J. J. Profile minimization problem for matrices and graphs. **Acta Mathematicae Applicatae Sinica**, v.10, n. 1, p. 107-122, 1994.

LIU, J. W. **On reducing the profile of sparse symmetric matrices.** 1976. 208 p. Tese (Doutorado em Ciência da Computação)-University of Waterloo, Waterloo, 1976.

LIU, W.; SHERMAN, A. H. Comparative analysis of the Cuthill-McKee and reverse Cuthill-McKee ordering algorithms for sparse matrices. **SIAM**

Journal on Numerical Analysis, Philadelphia, v. 13, n. 2, p. 198-213, 1976.

MANTUEFFEL, T. A. An incomplete factorization technique for positive definite linear systems. **Mathematics of Computation**, v. 34, n. 150, p. 473-497, 1980.

MEDEIROS, S. R. P.; PIMENTA, P. M.; GOLDENBERG, P. Algorithm for profile and wavefront reduction of sparse matrices with a symmetric structure. **Engineering computations**, v. 10, n. 3, p. 257-266, 1993.

MEIJERINK, J. A.; VAN DER VORST, H. A. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. **Mathematics of Computation**, v. 31, n. 137, p. 148-162, 1977.

REID, J. K.; SCOTT, J. A. Ordering symmetric sparse matrices for small profile and wavefront. **International Journal for Numerical Methods in Engineering**, v. 45, n. 12, p. 1737-1755, 1999.

REID, J. K.; SCOTT, J. A. Implementing Hager's exchange methods for matrix profile reduction. **ACM Transactions on Mathematical Software**, v. 28, n. 4, p. 377-391, 2002.

RUGE, J.; STÜBEN, K. Algebraic Multigrid. In: MCCORMICK, S. F. **Multigrid Methods**. Philadelphia: SIAM, 1987. cap. 4, p. 73-130.

SAAD, Y. **Iterative Methods for Sparse Linear Systems**. 2. ed. Philadelphia: SIAM, 2003. 517 p.

SAAD, Y.; SCHULTZ, M. H. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. **SIAM Journal on Scientific and Statistical Computing**, v. 7, p. 856-869, 1986.

SHEWCHUK, J. R. **An introduction to the conjugate gradient method without the agonizing pain**. Pittsburgh, USA: School of Computer Science, 1994. 64 p. Disponível em <<https://www.cs.cmu.edu/quake-papers/painless-conjugate-gradient.pdf>>. Acesso em: 23 jun. 2016.

SLOAN, S. W. A Fortran program for profile and wavefront reduction. **International Journal for Numerical Methods in Engineering**, v. 28, n. 11, p. 2651-2679, 1989.

SLOAN, S. W. An algorithm for profile and wavefront reduction of sparse matrices. **International Journal for Numerical Methods in Engineering**, v. 23, p. 1693-1704, 1986.

SNAY, R. A. Reducing the profile of sparse symmetric matrices. **Bulletin Geodesique**, v. 50, n. 4, p. 341-352, 1976.

TEMAM, O.; JALBY, W. Characterizing the Behavior of Sparse Algorithms on Caches. In: ACM/IEEE Conference on Supercomputing, 1992, Minneapolis. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 1992. p. 578-587.

TISMENETSKY, M. A new preconditioning technique for solving large sparse linear systems. **Linear Algebra and its Applications**, v. 154, p. 331-353, 1991.

VELHO, L.; FIGUEIREDO, L. H.; GOMES, J. Hierarchical generalized triangle strips. **The Visual Computer**, v. 15, n. 1, p. 21-35, 1999.

WALSHAW, C. A Multilevel Approach to the Travelling Salesman Problem. **Operations Research**, v. 50, n. 5, p. 862-877, 2002.