



ALEXANDRE AUGUSTO ALBERTO MOREIRA DE
ABREU

REDUÇÕES DE LARGURA DE BANDA E DE
PROFILE DE MATRIZES POR MAPA
AUTO-ORGANIZÁVEL

LAVRAS - MG

2014

ALEXANDRE AUGUSTO ALBERTO MOREIRA DE ABREU

REDUÇÕES DE LARGURA DE BANDA E DE *PROFILE* DE
MATRIZES POR MAPA AUTO-ORGANIZÁVEL

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

Orientador

Dr. Sanderson Lincoln Gonzaga de Oliveira

LAVRAS - MG

2014

**Ficha Catalográfica Elaborada pela Coordenadoria de Produtos e
Serviços da Biblioteca Universitária da UFLA**

Abreu, Alexandre Augusto Alberto Moreira de.

Reduções de largura de banda e de profile de matrizes por mapa auto-organizável / Alexandre Augusto Alberto Moreira de Abreu. – Lavras : UFLA, 2014.

125 p. : il.

Dissertação (mestrado) – Universidade Federal de Lavras, 2014.

Orientador: Sanderson Lincoln Gonzaga de Oliveira.

Bibliografia.

1. Redes neurais artificiais. 2. Redes de Kohonen. 3. Sistemas de equações lineares. I. Universidade Federal de Lavras. II. Título.

CDD – 003.74

ALEXANDRE AUGUSTO ALBERTO MOREIRA DE ABREU

REDUÇÕES DE LARGURA DE BANDA E DE *PROFILE* DE
MATRIZES POR MAPA AUTO-ORGANIZÁVEL

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

APROVADA em 18 de Novembro de 2014.

Dr. Luiz Nélio Henderson Guedes de Oliveira IPRJ-UERJ

Dr. Wilian Soares Lacerda UFLA

Dr. Sanderson Lincoln Gonzaga de Oliveira

Orientador

LAVRAS - MG

2014

Dedicado à memória de Virgílio Augusto Moreira de Abreu que, apesar de sua breve estadia entre nós, presenteou-nos imensamente com seu amor e alegria.

AGRADECIMENTOS

Agradeço à Ana Rosalva Moreira, Claudemir de Abreu, Aníbal Augusto Moreira de Abreu e Mariana Pereira Bernardo pelo apoio, carinho e compreensão de minha ausência.

Agradeço, em especial, ao professor Dr. Sanderson Lincoln Gonzaga de Oliveira pelas orientações, pelos esforços e pela paciência; ao meu amigo Guilherme Oliveira Chagas pela ajuda e ao professor Dr. Dragan Urosevic pelos envios dos executáveis do VNS-*band*.

Finalmente, agradeço a todos os meu familiares, amigos, professores e colegas de trabalho da Universidade Federal de Lavras.

RESUMO

Neste trabalho, descreve-se a proposta de uma heurística para reduções de largura de banda e de *profile* de matrizes simétricas e assimétricas por mapa auto-organizável unidimensional. Foram realizados experimentos e comparações dos resultados obtidos com resultados de heurísticas importantes para reduções de largura de banda e de *profile*: *Variable neighbourhood search for bandwidth reduction* e *Cuthill-McKee* reverso. Particularmente, foram realizadas simulações com a heurística *Cuthill-McKee* reverso com e sem a utilização de um método para se encontrar um vértice pseudo-periférico. São mostradas simulações com essas heurísticas para reduções de largura de banda e de *profile* de matrizes em dois conjuntos de instâncias da base *Harwell-Boeing* e simulações em três conjuntos de instâncias com sistemas de equações lineares oriundos de discretizações das equações de condução de calor e de Laplace por volumes finitos. Mais especificamente, os sistemas de equações lineares são resolvidos pelo método dos gradientes conjugados condicionado por Jacobi. Verificou-se que a melhor heurística nas simulações realizadas com as matrizes da base *Harwell-Boeing* foi a heurística *Variable neighbourhood search for bandwidth reduction*. A melhor heurística nas simulações com sistemas de equações lineares foi a heurística *Cuthill-McKee* reverso iniciada por um vértice pseudo-periférico. A heurística proposta não obteve resultados competitivos com os resultados obtidos pelas demais heurísticas avaliadas.

Palavras-chave: Largura de banda. *Profile*. Mapas auto-organizáveis. Redes neurais artificiais. Redes de Kohonen. Resolução de sistemas de equações lineares. Método dos gradientes conjugados condicionado por Jacobi.

ABSTRACT

In this work, we described the proposal of a heuristic for reductions of bandwidth and profile of symmetric and asymmetric matrices using one-dimensional self-organizing map. Experiments and comparisons of results obtained here were performed in relation to results of important heuristics, namely, Variable neighbourhood search for bandwidth reduction and reverse Cuthill-McKee. Simulations were performed with the reverse Cuthill-McKee, using and not using a method of finding a pseudo-peripheral vertex. Simulations performed with these heuristics were shown for two sets of Harwell-Boeing Collection instances, and for three sets of instances with systems of linear equations obtained from discretization of the equations of heat conduction and of Laplace by finite volumes. The systems of linear equations were solved using the Jacobi preconditioned conjugate gradient method. According to results, the best heuristic in the simulations performed with Harwell-Boeing Collection was the Variable neighbourhood search for bandwidth reduction. The best heuristic in the simulations performed with systems of linear equations was the reverse Cuthill-McKee, begun by a pseudo-peripheral vertex. Therefore, the heuristic proposed here did not show competitive results when compared to results obtained from other assessed heuristics.

Keywords: Bandwidth. Profile. Self organizing maps. Artificial neural networks. Kohonen network. Solving a linear equation system. Jacobi preconditioned conjugate gradient method.

LISTA DE FIGURAS

Figura 1	Grafo não direcionado e sua representação por matriz de adjacência.	20
Figura 2	Pseudocódigo da heurística <i>Cuthill-McKee</i>	25
Figura 3	Algoritmo do método dos gradientes conjugados.	34
Figura 4	Método dos gradientes conjugados preconditionado.....	37
Figura 5	Esquema do <i>i</i> -ésimo neurônio de uma RNA, adaptado de Vellasco (2007).....	42
Figura 6	Exemplo de topologia de mapa auto-organizável, adaptado de Zuben e Attux (2007).	47
Figura 7	Pseudocódigo de um algoritmo de treinamento de um MAO.	48
Figura 8	Pseudocódigo da heurística <i>MAO-band</i>	55
Figura 9	Exemplo de renumeração dos vértices de um grafo e de reordenamento da matriz de adjacência correspondente feito pela heurística <i>MAO-band</i>	58
Figura 10	Gráfico das ocupações de memória da heurística <i>VNS-band</i> com as instâncias resultantes da discretização da equação de condução do calor por volumes finitos.....	93
Figura 11	Gráfico das ocupações de memória da heurística <i>MAO-band</i> com as instâncias resultantes da discretização da equação de condução do calor por volumes finitos.....	93
Figura 12	Gráfico das ocupações de memória da heurística <i>VNS-band</i> com as instâncias resultantes da discretização da equação de Laplace por volumes finitos numeradas aleatoriamente.	101
Figura 13	Gráfico das ocupações de memória da heurística <i>MAO-band</i> com as instâncias resultantes da discretização da equação de Laplace por volumes finitos numeradas aleatoriamente.	101
Figura 14	Gráfico das ocupações de memória da heurística <i>VNS-band</i> com as instâncias resultantes da discretização da equação de Laplace por volumes finitos numeradas por uma curva de preenchimento de espaço.	108
Figura 15	Gráfico das ocupações de memória da heurística <i>MAO-band</i> com as instâncias resultantes da discretização da equação de Laplace por volumes finitos numeradas por uma curva de preenchimento de espaço.	109

LISTA DE TABELAS

Tabela 1	Simulações com a heurística CMr para as instâncias bastante pequenas.	65
Tabela 2	Simulações com a heurística CMr-pseudo para as instâncias bastante pequenas.	66
Tabela 3	Simulações com a heurística VNS- <i>band</i> para as instâncias bastante pequenas.	67
Tabela 4	Simulações com a heurística MAO- <i>band</i> para as instâncias bastante pequenas.	68
Tabela 5	Simulações com a heurística CMr para as instâncias pequenas.	72
Tabela 6	Simulações com a heurística CMr-pseudo para as instâncias pequenas.	74
Tabela 7	Simulações com a heurística VNS- <i>band</i> para as instâncias pequenas.	76
Tabela 8	Simulações com a heurística MAO- <i>band</i> para as instâncias pequenas.	78
Tabela 9	Resolução dos SELs sem reduções de largura de banda e de <i>profile</i>	84
Tabela 10	Resolução dos SELs com reduções de largura de banda e de <i>profile</i> pela heurística CMr.	84
Tabela 11	Resolução dos SELs com reduções de largura de banda e de <i>profile</i> pela heurística VNS- <i>band</i>	85
Tabela 12	Resolução dos SELs com reduções de largura de banda e de <i>profile</i> pela heurística MAO- <i>band</i>	85
Tabela 13	Resolução dos SELs condicionados por Jacobi sem reduções de largura de banda e de <i>profile</i>	87
Tabela 14	Resolução dos SELs condicionados por Jacobi com reduções de largura de banda e de <i>profile</i> pela heurística CMr.	87
Tabela 15	Resolução dos SELs condicionados por Jacobi com reduções de largura de banda e de <i>profile</i> pela heurística CMr-pseudo.	87
Tabela 16	Resolução dos SELs condicionados por Jacobi com reduções de largura de banda e de <i>profile</i> pela heurística VNS- <i>band</i>	88

Tabela 17	Resolução dos SELs preconditionados por Jacobi com reduções de largura de banda e de <i>profile</i> pela heurística MAO- <i>band</i>	88
Tabela 18	Ocupações de memória das heurísticas VNS- <i>band</i> e MAO- <i>band</i> com as instâncias resultantes da discretização da equação de condução do calor por volumes finitos.....	92
Tabela 19	Resolução dos SELs numerados aleatoriamente pelo MGCPJ sem reduções de largura de banda e de <i>profile</i>	97
Tabela 20	Resolução dos SELs numerados aleatoriamente pelo MGCPJ com reduções de largura de banda e de <i>profile</i> pela heurística CMr.	97
Tabela 21	Resolução dos SELs numerados aleatoriamente pelo MGCPJ com reduções de largura de banda e de <i>profile</i> pela heurística CMr-pseudo.	97
Tabela 22	Resolução dos SELs numerados aleatoriamente pelo MGCPJ com reduções de largura de banda e de <i>profile</i> pela heurística VNS- <i>band</i>	98
Tabela 23	Resolução dos SELs numerados aleatoriamente pelo MGCPJ com reduções de largura de banda e de <i>profile</i> pela heurística MAO- <i>band</i>	98
Tabela 24	Ocupações de memória das heurísticas VNS- <i>band</i> e MAO- <i>band</i> com as instâncias resultantes da discretização da equação de Laplace por volumes finitos numeradas aleatoriamente.	100
Tabela 25	Resolução dos SELs numerados por uma curva de preenchimento de espaço pelo MGCPJ sem reduções de largura de banda e de <i>profile</i>	104
Tabela 26	Resolução dos SELs numerados por uma curva de preenchimento de espaço pelo MGCPJ com reduções de largura de banda e de <i>profile</i> pela heurística CMr.	104
Tabela 27	Resolução dos SELs numerados por uma curva de preenchimento de espaço pelo MGCPJ com reduções de largura de banda e de <i>profile</i> pela heurística CMr-pseudo. .	105
Tabela 28	Resolução dos SELs numerados por uma curva de preenchimento de espaço pelo MGCPJ com reduções de largura de banda e de <i>profile</i> pela heurística MAO- <i>band</i>	105
Tabela 29	Resolução dos SELs numerados por uma curva de preenchimento de espaço pelo MGCPJ com reduções de largura de banda e de <i>profile</i> pela heurística VNS- <i>band</i>	106

Tabela 30	Ocupações de memória das heurísticas <i>VNS-band</i> e <i>MAO-band</i> com as instâncias resultantes da discretização da equação de Laplace por volumes finitos numeradas por uma curva de preenchimento de espaço.	108
-----------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----

LISTA DE SIGLAS

CM	<i>Cuthill-McKee</i>
CMr	<i>Cuthill-McKee reverso</i>
CMr-pseudo	<i>Cuthill-McKee reverso iniciado por um vértice pseudo-periférico</i>
MGC	Método dos Gradientes Conjugados
MGCPJ	Método dos Gradientes Conjugados Precondicionado por Jacobi
MAO	Mapa Auto-Organizável
MAO-band	Mapa Auto-Organizável para redução de largura de Banda
RNA	Redes Neurais Artificiais
SEL	Sistemas de Equações Lineares
UFLA	Universidade Federal de Lavras
VNS-band	<i>Variable Neighbourhood Search for Bandwidth Reduction</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Justificativa	16
1.2	Objetivos	17
1.2.1	Objetivos específicos	18
2	REFERENCIAL TEÓRICO	19
2.1	Grafo, matriz de adjacência e conceitos básicos	19
2.2	Largura de banda	22
2.3	<i>Profile</i>	23
2.4	Heurísticas Cuthill-McKee e Cuthill-McKee reverso .	24
2.5	Vértice pseudo-periférico	26
2.6	Heurísticas recentes para o problema de minimização de largura de banda e de <i>profile</i>	27
2.7	Método dos gradientes conjugados	31
2.8	Precondicionamento de sistemas de equações lineares	34
2.8.1	Método dos gradientes conjugados precondicionado ..	36
2.8.2	Técnicas de precondicionamento	38
2.9	Redes neurais	39
2.9.1	Redes neurais artificiais	41
2.9.2	Especificações	42
2.9.3	Treinamento de redes neurais artificiais	43
2.9.4	Redes neurais artificiais e redução de <i>profile</i>	45
2.9.5	Mapas auto-organizáveis	46
3	MAPA AUTO-ORGANIZÁVEL APLICADO AOS PROBLEMAS DE REDUÇÃO DE LARGURA DE BANDA E DE <i>PROFILE</i>	49
3.1	Metodologia de pesquisa	49
3.2	Metodologia de desenvolvimento	50
3.3	Detalhes da implementação da heurística MAO- <i>band</i>	53
3.4	Ferramentas utilizadas	60
3.5	Metodologia utilizada nas simulações	60
4	RESULTADOS E ANÁLISES	62
4.1	Simulações com as heurísticas CMr, CMr-pseudo, VNS- <i>band</i> e MAO- <i>band</i> com matrizes da base <i>Harwell-Boeing</i>	63
4.1.1	Resultados e análises das 33 instâncias bastante pequenas	64
4.1.1.1	Redução de largura de banda	69

4.1.1.2	Redução de <i>profile</i>	70
4.1.2	Resultados e análises das 80 instâncias pequenas	71
4.1.2.1	Redução de largura de banda.....	80
4.1.2.2	Redução de <i>profile</i>	81
4.2	Simulações com SELs oriundos da discretização da condução do calor	82
4.2.1	SELs resolvidos pelo MGC sem condicionamento .	83
4.2.2	SELs resolvidos pelo MGC condicionado por Jacobi	86
4.2.3	Ocupações de memória	91
4.3	Simulações com SELs oriundos da discretização da equação de Laplace.....	94
4.3.1	Simulações com SELs numerados aleatoriamente	95
4.3.1.1	Reduções de largura de banda e de <i>profile</i> e resolu- ções dos SELs pelo MGCPJ	96
4.3.1.2	Ocupações de memória	99
4.3.2	Simulações com SELs numerados por uma curva de preenchimento de espaço	102
4.3.2.1	Reduções de largura de banda e de <i>profile</i> e resolu- ções dos SELs pelo MGCPJ	103
4.3.2.2	Ocupações de memória	107
5	CONCLUSÕES E TRABALHOS FUTUROS	110
5.1	Considerações finais.....	111
5.1.1	MAO- <i>band</i> : proposta de uma heurística para re- duções de largura de banda e de <i>profile</i> por mapa auto-organizável	111
5.1.2	Experimentos com instâncias da base <i>Harwell-Boeing</i>	112
5.1.3	Ocupações de memória	113
5.1.4	Simulações de reduções de largura de banda e de <i>profile</i> com resolução do SEL pelo MGCPJ	114
5.2	Trabalhos futuros	116
	REFERÊNCIAS.....	120

1 INTRODUÇÃO

Diversos problemas científicos, na engenharia ou industriais, como em dinâmica de fluidos ou modelagem de objetos e jogos 3D, são descritos por equações diferenciais parciais. O método dos elementos finitos é, atualmente, um dos mais populares para a resolução de equações diferenciais parciais e consiste em transformar uma equação diferencial contínua em um sistema de equações algébricas. Sistemas de equações lineares desse tipo, também, podem ser obtidos pelo método dos volumes finitos, por exemplo.

Tal sistema de equações lineares (SEL) possui a forma $Ax = b$, em que A é a matriz de coeficientes, x é o vetor de incógnitas e b é o vetor de termos independentes. SELs desse tipo, geralmente, possuem uma representação matricial de grande porte e esparsa. Dessa forma, o custo computacional exigido para se armazenar e processar tais matrizes é elevado. A resolução de tais sistemas lineares esparsos é a etapa que mais demanda tempo de processamento computacional.

Sistemas lineares esparsos, também, são oriundos de outras aplicações que não têm origens em equações diferenciais parciais. Como exemplos, desenho e análise de circuitos digitais, sistemas de redes de energia elétrica, processos de engenharia química, modelos econômicos e sistemas de produção industrial (BENZI, 2002).

A resolução de um SEL pode ser feita por métodos diretos ou iterativos. Como exemplos de métodos diretos, temos as decomposições LU e de Cholesky. Métodos diretos são usados em códigos industriais em razão de sua confiabilidade. Além de robustos, os métodos diretos tendem a demandar uma quantidade previsível de recursos de memória e processamento computacional. No geral, métodos diretos têm sido utilizados em problemas

como em dinâmica de fluidos e outros exemplos já citados não oriundos de equações diferenciais parciais (BENZI, 2002).

Como os métodos diretos têm baixa escalabilidade em relação ao processamento e à utilização de memória, principalmente, em problemas tridimensionais, advindos de equações diferenciais parciais (BENZI, 2002), os SELs, representados por matrizes esparsas de grande porte são, geralmente, resolvidas por métodos iterativos.

Métodos iterativos requerem menos recursos de memória e de processamento, principalmente, em situações em que a precisão desejada é relativamente pequena. Benzi (2002), entretanto, ressalva que tais métodos não possuem a confiabilidade dos métodos diretos. Em alguns casos os métodos iterativos falham e o condicionamento é necessário, apesar de não ser suficiente, para garantir a convergência em um tempo razoável.

Os limites entre métodos diretos e iterativos têm se tornado cada vez menos claros com a utilização de diversas ideias e técnicas dos métodos diretos no campo dos métodos iterativos. Como consequência, os métodos iterativos têm se tornado cada vez mais confiáveis (BENZI, 2002). O método dos gradientes conjugados (MGC) é um exemplo de método iterativo eficiente muito utilizado na prática (SAAD, 2003).

Este trabalho está dividido em cinco capítulos. No capítulo 1, são introduzidos os assuntos envolvidos nesse trabalho, a justificativa e os objetivos do mesmo. O referencial teórico necessário a um melhor entendimento desse trabalho é apresentado no capítulo 2.

No capítulo 3, as metodologias de pesquisa e desenvolvimento deste trabalho são descritas. São apresentados os detalhes de implementação de uma heurística para reduções de largura de banda e de *profile* baseada em

mapa auto-organizável. Os resultados e as análises dos experimentos realizados são apresentados no capítulo 4. Finalmente, no capítulo 5, são apresentadas algumas conclusões sobre este trabalho e proposições de trabalhos futuros.

A seguir, na seção 1.1, são apresentadas as justificativas que motivaram o desenvolvimento deste trabalho. Em seguida, na seção 1.2, são apresentados os objetivos gerais e específicos deste trabalho.

1.1 Justificativa

A resolução de sistemas lineares esparsos é a etapa que mais demanda tempo de processamento computacional em problemas diversos. Para grandes estruturas mecânicas, encontradas em situações práticas, entre 30% e 50% do custo computacional, está em se resolver os sistemas de equações lineares; esse custo pode chegar a 80% em problemas dinâmicos ou de otimização estrutural (KAVEH, 2004).

Métodos para resolução de SELs diretos são, geralmente, baseados na eliminação gaussiana. A eliminação de Gauss, que é um método direto para resolução de sistema de equações lineares, pode ser realizada em tempo $O(n \cdot \beta^2)$ para uma matriz com largura de banda β , quando $\beta \ll n$, que é significativamente melhor do que o típico algoritmo $O(n^3)$ para uma matriz de coeficientes $n \times n$. Veja Lim et al. (2006) e Lim, Rodrigues e Xiao (2006) para detalhes sobre esse assunto.

Papadimitriou (1976) demonstrou que o problema da minimização da largura de banda é um problema NP-Difícil, para mais detalhes veja Oliveira e Chagas (2014, p. 6). Encontrar a menor largura de banda por força bruta, por exemplo, torna-se uma tarefa impraticável. Seriam $n!$ so-

luções possíveis para uma matriz de ordem n . Por isso, “desde a década de 1960, foram propostas, provavelmente, centenas de heurísticas aplicadas nos problemas de redução de largura de banda ou de *profile*” (OLIVEIRA; CHAGAS, 2014, p. 7).

As heurísticas realizam permutações nas linhas e colunas das matrizes na intenção de que a estrutura da matriz resultante se torne mais compacta, com os coeficientes não nulos próximos à diagonal principal (OLIVEIRA; CHAGAS, 2014). Ao se resolver um sistema de equações lineares permutado, é possível que seja necessário um número menor de operações aritméticas e/ou de armazenamento do que o sistema original (LIU, 1976; OLIVEIRA; CHAGAS, 2014). As permutações nas linhas e colunas das matrizes simétricas são equivalentes a se renumerar os vértices do grafo correspondente sem alterar as adjacências.

Com as reduções de largura de banda e de *profile*, pretende-se reduzir o custo de armazenamento e processamento das matrizes que representam os SELs, assim como reduzir o custo computacional necessário para sua resolução. Não se tem conhecimento, até o momento, de utilização de redes neurais artificiais nos problemas de redução de largura de banda ou de *profile*. Neste trabalho, apresenta-se uma proposta de aplicação de redes neurais artificiais no problema em questão. Com a utilização de mapas auto-organizáveis, pretende-se realizar as reduções de largura de banda e de *profile* com custo computacional baixo.

1.2 Objetivos

O objetivo geral deste trabalho é propor e verificar o desempenho computacional de uma heurística, baseada em mapa auto-organizável, para

realizar reduções de largura de banda e de *profile* de matrizes esparsas simétricas e assimétricas que representam sistemas de equações lineares. Para se alcançar esse objetivo, necessita-se que sejam alcançados os seguintes objetivos específicos.

1.2.1 Objetivos específicos

Os objetivos específicos estão divididos nos tópicos seguintes.

- Estudo dos problemas de redução de largura de banda e de *profile* para melhor entendimento do problema. Esse estudo envolve a pesquisa e a compreensão dos principais algoritmos e meta-heurísticas utilizados na resolução desses problemas.
- Estudo de técnicas de aprendizado não supervisionado, principalmente redes neurais artificiais.
- Desenvolvimento e investigação da viabilidade de utilização de mapa auto-organizável para realizar reduções de largura de banda e de *profile* de matrizes esparsas simétricas e assimétricas que representam sistemas de equações lineares.
- Realização de experimentos e comparações dos resultados obtidos com uma heurística clássica e o possível estado da arte para o problema de redução de largura de banda.
- Avaliar, experimentalmente, a redução do custo computacional envolvido na resolução de sistemas de equações lineares proporcionado por cada método de redução de largura de banda e de *profile*.

2 REFERENCIAL TEÓRICO

Neste capítulo, são introduzidos alguns conceitos para melhor entendimento deste trabalho. Primeiramente, é apresentada a relação entre grafo e matriz de adjacência e alguns conceitos básicos, na seção 2.1. Na sequência, são apresentados os conceitos de largura de banda na seção 2.2 e de *profile* na seção 2.3.

Duas meta-heurísticas clássicas para o problema de minimização de largura de banda e de *profile* são apresentadas na seção 2.4. Na seção 2.5, são apresentados alguns conceitos sobre vértice pseudo-periférico e uma forma de encontrá-lo. São apresentadas algumas meta-heurísticas recentes para o problema de minimização de largura de banda, na seção 2.6.

O método dos gradientes conjugados é apresentado na seção 2.7. O condicionamento de sistemas de equações lineares e assuntos relacionados são abordados na seção 2.8.

Na seção 2.9, são introduzidos os conceitos de redes neurais e suas relações com a computação. São apresentadas as definições de redes neurais artificiais e suas especificações. A seguir, na subseção 2.9.3, são introduzidos os conceitos de treinamento de redes neurais artificiais e os processos de aprendizado supervisionado e não supervisionado. Ao final deste capítulo, na subseção 2.9.5, são abordados os mapas auto-organizáveis.

2.1 Grafo, matriz de adjacência e conceitos básicos

Pode-se representar um grafo, por meio de uma matriz de adjacência, como na Figura 1. Nessa representação, cada linha $1 \leq i \leq n$ de uma matriz A de dimensão $n \times n$, corresponde a um vértice v_i do grafo $G(V, E)$, não direcionado e não ponderado, em que $V = \{v_1, v_2, \dots, v_n\}$ é um conjunto finito

de vértices e $E = \{e_1, e_2, \dots, e_m\}$ é um conjunto finito de arestas representadas por pares não ordenados de vértices. Em um grafo não direcionado, uma aresta pode ser representada por $e_1 = \{v_1, v_2\}$, por exemplo.

As colunas $1 \leq j \leq n$ da matriz A representam as adjacências, ou seja, a existência de uma aresta entre os vértices v_i e v_j , caso $a_{i,j} \neq 0$. Por exemplo, na Figura 1, o vértice 1 é adjacente aos vértices 2 e 3 somente. Então, a matriz de adjacência que o representa possui elementos não nulos nas colunas 2 e 3 da linha 1. A diagonal principal sempre conterá elementos não nulos. Os elementos não nulos são representados por quadrados pretos e os elementos nulos por quadrados brancos na Figura 1.

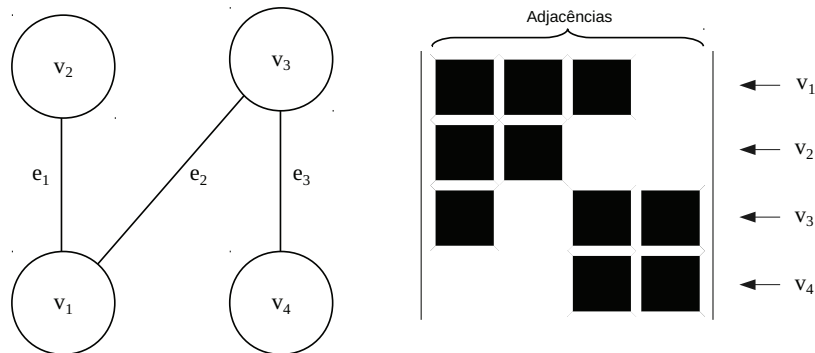


Figura 1 Grafo não direcionado e sua representação por matriz de adjacência.

Considere o grafo $G(V, E)$ descrito acima. Seguem as seguintes definições (OLIVEIRA; CHAGAS, 2014):

Definição 1 (adjacência) *Dois vértices são adjacentes se existe uma aresta entre eles. O conjunto de vértices adjacentes a v é $Adj(G, v) = \{u \in V : \{v, u\} \in E\}$.*

Definição 2 (grau de um vértice) *O grau de um vértice v é a quantidade de vértices em seu conjunto de vértices adjacentes, ou seja,*

$\text{Grau}(G, v) = |\text{Adj}(G, v)|$.

Definição 3 (caminho) *Caminho é uma sequência de vértices, de forma que existe uma aresta entre esses vértices desde o primeiro até o último vértice dessa sequência.*

Definição 4 (tamanho de um caminho) *O tamanho de um caminho é a sua quantidade de arestas.*

Definição 5 (distância) *A distância $d(u, v)$ entre u e $v \in V$ é o tamanho do menor caminho entre esses vértices.*

Definição 6 (excentricidade) *A excentricidade $\ell : V \rightarrow \mathbb{N}$ de um vértice $v \in V$ é definida como $\ell(v) = \max_{u \in V} (d(v, u))$.*

Definição 7 (diâmetro do grafo) *O diâmetro $\Phi(G) = \max_{v \in V} (\ell(v))$ é a maior excentricidade encontrada em um grafo $G(V, E)$.*

Definição 8 (vértice periférico) *Um vértice v é periférico se $\Phi(G) = \ell(v)$, ou seja, sua excentricidade é igual ao diâmetro do grafo.*

Definição 9 (estrutura de nível enraizada) *A estrutura de nível enraizada em v é o particionamento $\mathcal{L}(v) = \{L_0(v), L_1(v), \dots, L_{\ell(v)}(v)\}$, de profundidade $\ell(v)$, 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 $i = 1, 2, \dots, \ell(v)$ e $\text{Adj}(\cdot)$ é o conjunto de vértices adjacente ao vértice do argumento.*

Definição 10 (largura de nível) *A largura de nível é o número de vértices do nível com mais vértices e pode ser definida como $b(\mathcal{L}(v)) = \max_{0 \leq i \leq \ell(v)} |L_i(v)|$.*

A estrutura de nível enraizada é o particionamento de V em níveis de vértices com a mesma distância para o vértice raiz. Como exemplo, considere o grafo da Figura 1. Escolhendo-se o vértice 1 arbitrariamente, tem-se que $\ell(1) = 2$. O particionamento dos vértices em níveis, partindo-se do vértice 1, é $L_0(1) = \{1\}$, $L_1(1) = \{2, 3\}$ e $L_2(1) = \{4\}$. Percebe-se que o número de partições, ou níveis, de $\mathcal{L}(v)$ é $\ell(v) + 1$. Nesse exemplo, a largura de nível é $b(\mathcal{L}(v)) = |L_1(1)| = 2$.

A seguir, são apresentadas as definições de largura de banda e de *profile* nas seções 2.2 e 2.3, respectivamente.

2.2 Largura de banda

Pode-se definir a largura de banda de uma matriz como a maior distância de um atributo não nulo até a diagonal principal. A largura de banda β de uma matriz A , de dimensões $n \times n$, pode ser definida como

$$\beta = \beta(A) = \max\{\beta_i(A) \mid 1 \leq i \leq n\}, \quad (1)$$

$$\beta_i(A) = i - \min\{j \mid a_{ij} \neq 0\}. \quad (2)$$

$\beta_i(A)$ é denominada largura de banda da i -ésima linha da matriz A . Na Figura 1, a largura de banda $\beta_3(A) = 3 - 1 = 2$ é a maior distância de um elemento não nulo até a diagonal principal, que ocorre na linha 3.

A largura de banda pode ser encontrada diretamente em um grafo $G(V, E)$, considerando-se a função $f : V \rightarrow \{1, 2, \dots, n\}$ que rotula os vértices,

em que $|V| = n$, da seguinte forma:

$$\beta(f) = \max\{|f(u) - f(v)| \mid \forall (u, v) \in E\}. \quad (3)$$

No exemplo da Figura 1, temos $\beta(f) = 3 - 1 = 2$ se considerarmos a rotulação $f : f(v_1) = 1, f(v_2) = 2, f(v_3) = 3, f(v_4) = 4$.

O problema de minimização da largura de banda pode ser definido como a busca por uma rotulação f^* que minimize $\beta(f)$, ou seja:

$$\beta(f^*) = \min_f \beta(f) = \min_f \max_{(u,v) \in E} |f(u) - f(v)|. \quad (4)$$

O problema de minimização de largura de banda é um problema NP-Difícil (PAPADIMITRIOU, 1976). Para um grafo $G(V, E)$ em que $|V| = n$, existem $n!$ rotulações possíveis.

2.3 Profile

O *profile* pode ser definido como a soma da largura de banda de todas as linhas de uma matriz. Formalmente, pode-se definir o *profile* de uma matriz A de dimensão $n \times n$ como:

$$profile(A) = \sum_{i=1}^n \beta_i(A). \quad (5)$$

O *profile* da matriz presente na Figura 1, por exemplo, é $profile(A) = \beta_1(A) + \beta_2(A) + \beta_3(A) + \beta_4(A) = 0 + 1 + 2 + 1 = 4$. Analogamente à largura de banda, pode-se definir o *profile* de um grafo da seguinte forma:

$$profile(G) = \sum_{u \in V} \max((f(u) - f(v)) \mid (u, v) \in E, f(u) > f(v)). \quad (6)$$

A seguir são apresentados dois métodos clássicos para as reduções de largura de banda e de *profile*, os métodos *Cuthill-McKee* e *Cuthill-McKee reverso*. A descrição desses métodos foi baseada em Oliveira e Chagas (2014).

2.4 Heurísticas Cuthill-McKee e Cuthill-McKee reverso

Cuthill e McKee (1969) propuseram um método que reduz a largura de banda de um grafo que pode ser representado por uma matriz de adjacências simétrica. Esse método, conhecido como *Cuthill-McKee* (CM), é semelhante à busca em largura que pode ser realizada em um grafo. A diferença entre a busca em largura convencional está na ordem em que os vértices são visitados na heurística CM. A heurística CM visita os vértices por ordem crescente de grau de cada vértice. O grau de um vértice é a quantidade de vértices adjacentes a este. Ou seja, com base em um vértice inicial, a heurística CM visitará o seus vértices adjacentes começando com o de menor grau. Depois visitará os adjacentes a esses vértices e, assim, sucessivamente até percorrer todo o grafo, sempre respeitando a ordem crescente de grau. A ordem, em que cada vértice é visitado, representará sua rotulação. Assim, o vértice inicial será rotulado como 1, o seu adjacente de menor grau será rotulado como 2 e, assim, sucessivamente.

Foi constatado por Cuthill e McKee (1969) que a escolha do vértice inicial influencia na largura de banda e no *profile* do resultado final. Os autores propuseram que o vértice inicial fosse o vértice de grau mínimo. Com essa abordagem, obtém-se uma configuração matricial na qual os vértices de maior grau ficam posicionados no centro da matriz.

O algoritmo presente na Figura 2, apresentado por Liu (1976), mostra a heurística CM. Seus parâmetros de entrada são um grafo conexo

$G = (V, E)$, em que V é o conjunto de vértices e E é o conjunto de arestas, e um vértice inicial $v \in V$. Sua saída será uma sequência de vértices presente no vetor α .

Algoritmo: *Cuthill-McKee*

Entrada: grafo conexo $G = (V, E)$ e vértice inicial $v \in V$.
Saída: vetor α ordenado de dimensão $|V|$.

```

1  $i \leftarrow 1$ ;
2  $j \leftarrow 1$ ;
3  $\alpha(1) \leftarrow v$ ;
4 enquanto (  $i < |V|$  ) faça
5   para cada (  $\{ \text{vértice } w \in \text{adj}(\alpha(j)) \} - \{ \alpha(1), \dots, \alpha(i) \}$ ,
   em ordem crescente de grau ) faça
6      $i \leftarrow i + 1$ ;
7      $\alpha(i) \leftarrow w$ ;
8   fim-para-cada;
9    $j \leftarrow j + 1$ ;
10 fim-enquanto;
11 retorna  $\alpha$ ; // Retorna a sequência no vetor  $\alpha$ 
```

Figura 2 Pseudocódigo da heurística *Cuthill-McKee*.

Inicialmente, o algoritmo inicializa as variáveis i e j nas linhas 1 e 2, respectivamente. Na linha 3, o vértice inicial v é atribuído à primeira posição de α . O laço de repetição das linhas 4 a 10 repetirá $|V| - 1$ vezes, em que $|V|$ é a quantidade de vértices do grafo. Na linha 7, insere-se em α os vértices adjacentes a $\alpha(j)$ que, ainda, não foram inseridos, em ordem crescente de grau. Isso será feito $|V| - 1$ vezes, pois os adjacentes do último vértice já estarão presentes em α , desde que o grafo seja conexo.

A heurística *Cuthill-McKee* reverso (CMr) é uma modificação no método *Cuthill-McKee* proposta por George (1971). A diferença da heurística CM para a heurística CMr está apenas na ordem final dos vértices.

Na heurística CMr, ordena-se os vértices em ordem inversa. Com a heurística CMr, a largura de banda é a mesma encontrada na heurística CM; porém, muitas vezes, o *profile* é reduzido utilizando-se a heurística CMr. Mais detalhes podem ser encontrados em Oliveira e Chagas (2014).

Na subseção 2.6, são apresentadas algumas meta-heurísticas recentes para o problema de minimização de largura de banda e de *profile*. A seguir, são apresentados alguns conceitos sobre vértices pseudo-periféricos.

2.5 Vértice pseudo-periférico

A solução encontrada pela heurística CM, CMr e outras heurísticas, pode ser influenciada pela escolha do vértice inicial. Inicialmente, Cuthill e McKee (1969) propuseram que o vértice inicial fosse um vértice de grau mínimo. Essa escolha nem sempre era a melhor. Posteriormente, Gibbs, Poole e Stockmeyer (1976) constataram que o melhor vértice inicial é um vértice periférico. Um vértice é periférico se sua excentricidade é igual ao diâmetro do grafo, ou seja, máxima.

Quanto maior a excentricidade de um vértice, menos vértices estarão presentes em um nível de uma estrutura de nível desse vértice (CUTHILL; MCKEE, 1969). Cuthill e McKee (1969), também, apontaram que a largura de banda está relacionada com a quantidade de vértices presentes na estrutura de nível com maior número de vértices.

Um vértice periférico é um ótimo vértice inicial para algumas heurísticas de redução de largura de banda (OLIVEIRA; CHAGAS, 2014). Para se encontrar um vértice periférico, pode-se realizar uma busca em largura em todos os vértices do grafo. O custo dessas operações é de $O(|V| \cdot (|V| + |E|))$, que é um custo relativamente alto. Por isso, Gibbs, Poole e Stockmeyer

(1976) apresentaram um algoritmo para se encontrar um vértice cuja excentricidade é próxima ao diâmetro do grafo. Esse vértice é considerado um vértice pseudo-periférico.

Neste trabalho, utilizou-se o algoritmo de George e Liu (1979) para se encontrar um vértice pseudo-periférico utilizado como vértice inicial para a heurística CMr. Esse algoritmo é uma modificação do algoritmo de Gibbs, Poole e Stockmeyer (1976). Uma das melhorias que compõem o algoritmo final de George e Liu (1979) está na utilização de um vértice qualquer para a construção da primeira estrutura de nível enraizada. Outra melhoria proposta foi que apenas o vértice folha de menor grau é utilizado para montar a nova estrutura de nível enraizada.

O algoritmo de George e Liu (1979) inicia por um vértice $v \in V$ qualquer. Gera-se, então, a estrutura de nível enraizada $\mathcal{L}(v)$. Escolhe-se um vértice de menor grau $u \in L_{\ell(v)}(v)$ e gera-se a estrutura de nível enraizada $\mathcal{L}(u)$. Então, verifica-se se $\ell(v) < \ell(u)$; em caso positivo, o vértice u passa a ser o vértice v e o processo é repetido, em caso negativo, v será o vértice pseudo-periférico encontrado. Uma melhor descrição do algoritmo de George e Liu (1979) e de outros algoritmos, para se encontrar um vértice pseudo-periférico, é encontrada em Oliveira e Chagas (2014).

2.6 Heurísticas recentes para o problema de minimização de largura de banda e de *profile*

A seguir, são abordadas três heurísticas recentes para o problema de minimização de largura de banda. Uma dessas heurísticas, pode ser considerada o possível estado da arte no problema de minimização de largura de banda.

A *busca em sistema carregado*, *Charged System Search (CSS)*, é uma meta-heurística desenvolvida por Kaveh e Talatahari (2010a), para a resolução de problemas estruturais. Veja Kaveh e Talatahari (2010b), para detalhes. Os autores utilizaram as leis de Coulomb e Gauss, da eletrostática, e as leis da mecânica de Newton para desenvolver essa meta-heurística. A utilização de tais leis pela meta-heurística CSS provê um modelo que pode ser utilizado em problemas de otimização discretos e contínuos. Kaveh e Sharafi (2012) utilizaram um modelo baseado nessas leis para criar uma proposta para a renumeração dos vértices de um grafo. Com isso, utilizaram essa meta-heurística para a redução de banda de matrizes esparsas.

Boutora et al. (2007) propuseram um método de redução de banda que funciona para renumeração de malhas triangulares. Uma de suas características é que o processo de renumeração é aplicado antes da construção da matriz esparsa. O método apresenta bons resultados e funciona de forma parecida com um algoritmo de busca em largura em grafos, que foi uma das primeiras estratégias utilizadas para a redução de banda.

Outra heurística aplicada no problema de minimização de largura de banda é o *Variable neighbourhood search for bandwidth reduction (VNS-band)*, proposta por Mladenovic et al. (2010). Grosso modo, a heurística *VNS-band* realiza uma busca por um mínimo local para os vértices contidos em um determinado raio de vizinhança. Para isso, realiza a troca de vizinhança entre os vértices e verifica se a solução encontrada é melhor do que a anterior. O raio de vizinhança aumenta durante a execução do algoritmo.

A heurística *VNS-band* pode ser dividida em quatro etapas: inicialização, perturbação, busca local e permutação de vizinhança. Na primeira etapa, constrói-se uma solução inicial de numeração dos vértices de $G(V, E)$,

por meio de uma busca em largura, com base em um vértice $v \in V$ qualquer. Essa busca leva em consideração uma ordem aleatória de vértices, em cada nível, ao contrário da heurística CMr, por exemplo, que considera a ordem crescente de vértices.

Na segunda etapa, procura-se escapar do vale de um mínimo local por meio de perturbação. Define-se $N_k(S)$, que é a vizinhança dentro de um raio k para uma solução S . Uma solução S' pertence à vizinhança $N_k(S)$ se S' e S diferem entre si em $K + 1$ renumerações. Gera-se uma solução $S' \in N_k(S)$ e realiza-se uma troca sistemática das numerações dessa solução que pode ser feita de duas formas diferentes. A primeira consiste em selecionar os vértices com maior largura de banda e trocam-se suas numerações. A segunda forma realiza uma rotação nas numerações dos vértices. Essa rotação consiste em determinar um intervalo $[b, f]$ e trocar as numerações dos vértices que pertencem à esse intervalo deslocando-os m posições. Ou seja, a numeração $f(b)$ será $f(b + m)$, se $b + m \leq f$, em que $1 \leq b < m < f \leq |V|$. No caso de um índice i em que $i + m > f$, a numeração $f(i)$ será $f(b + |f - (i + m)|)$. Considere a função $f : V \rightarrow \{1, 2, \dots, n\}$ que rotula os vértices de $G(V, E)$, em que $|V| = n$. Para definir qual forma de perturbação será utilizada, verifica-se $k \leq k_{max}$ e, em caso afirmativo, utiliza-se a primeira forma de perturbação e, em caso negativo, a segunda forma.

A terceira etapa é a etapa de busca local, na qual o algoritmo tenta encontrar o mínimo local por meio de trocas de numeração em um conjunto definido de possibilidades. Para tal, seleciona-se um conjunto V_c de vértices críticos do grafo. Um vértice $v \in V$ é considerado crítico se pertence a uma aresta $(u, v) \in E_c$, em que E_c é o conjunto de arestas críticas de G . Uma aresta (u, v) é considerada crítica se $|f(u) - f(v) = \beta(f)|$, em que $\beta(f)$ é a largura

de banda de G definida na equação 3. Define-se a vizinhança reduzida de v como $N(u, \mathcal{S}) = \{u \mid (u, v) \in E_c, |\text{med}(v) - f(u)| < |\text{med}(v) - f(v)|\}$, em que $\text{med}(v) = \lfloor \frac{f_{\min}(v) + f_{\max}(v)}{2} \rfloor$. O conjunto de vizinhanças reduzidas, que será utilizado para efetuar a troca de numeração será $N(\mathcal{S}) = \bigcup_{v \in V_c} N(v, \mathcal{S})$. A busca local é realizada em $N(\mathcal{S}')$, por meio da troca de numeração de um vértice crítico v com cada numeração dos vértices $u \in N(v, \mathcal{S}') \subseteq N(\mathcal{S}')$, até que se encontre uma melhora em relação à solução corrente.

Na quarta etapa, verifica-se se a solução encontrada \mathcal{S}' , após a etapa de perturbação é melhor que a solução corrente \mathcal{S} . E em caso afirmativo, \mathcal{S}' passa a ser a solução corrente e k passa a ser k_{\min} , que é o valor inicial de k . Caso a solução encontrada não seja melhor que a solução inicial, incrementa-se o raio de vizinhança em k_{step} . Os valores de k_{\min} , k_{step} e k_{\max} são parâmetros definidos de acordo com o problema.

Mladenovic et al. (2010) compararam os resultados da heurística VNS-*band* com algumas propostas anteriores, que poderiam ser consideradas o estado da arte no problema de minimização de largura de banda. As comparações de resultados de Mladenovic et al. (2010) indicam que a heurística VNS-*band* é uma das possíveis melhores meta-heurísticas para o problema. Veja Oliveira e Chagas (2014) para maiores detalhes sobre a heurística VNS-*band* e outras heurísticas para redução de largura de banda.

Aplicou-se redes neurais artificiais de treinamento supervisionado para auxiliar o algoritmo *profile-front minimization* (PFM) de Hoit e Wilson (1983) na resolução do problema de minimização de *profile*. Mais detalhes sobre essa aplicação serão apresentados na subseção 2.9.4.

A seguir, na seção 2.7, é apresentado o método iterativo de resolução SELs conhecido como método dos gradientes conjugados.

2.7 Método dos gradientes conjugados

Os métodos iterativos são muito utilizados na resolução de sistemas de equações lineares (SEL), pois demandam menos recursos de memória e processamento do que os métodos diretos. Portanto, SELs oriundos de equações diferenciais parciais, por exemplo, que possuem representação matricial esparsa de grande porte são, geralmente, resolvidos por métodos iterativos (BENZI, 2002). O método dos gradientes conjugados (MGC) é um dos melhores métodos iterativos para resolução de sistemas lineares simétricos e positivos-definidos (SAAD, 2003). O algoritmo para o MGC apresentado nesta seção tem propósito didático apenas. A implementação realizada neste trabalho utilizou a versão do projeto computacional de Oliveira, Kischinhevsky e Tavares (2013) que armazena apenas os coeficientes não nulos da matriz A por meio de listas de adjacências.

Considere o SEL simétrico, positivo-definido $Ax = b$, em que $A \in \mathbb{R}^{n \times n}$, x e $b \in \mathbb{R}^n$. Uma matriz A é positiva-definida se é possível obter um vetor x , tal que $x^T Ax > 0$. A matriz A é simétrica se seus elementos $a_{ij} = a_{ji}$, em que i é o índice referente à linha e j é o índice referente à coluna da matriz.

A solução desse SEL $Ax = b$ pode ser encontrada por meio da minimização da função quadrática $f(x)$, em que $x \in \mathbb{R}^n$. Considere

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

então x^* é uma solução do SEL se, e somente se, x^* minimiza f . Deriva-se

f em relação ao vetor x na equação 7 e iguala-se a zero, para obter

$$\nabla f(x^*) = Ax^* - b = 0. \quad (8)$$

Isso significa que o método dos gradientes conjugados soluciona o sistema de equações quando $\nabla f(x) = 0$.

No início do método dos gradientes conjugados, obtém-se uma solução inicial x_0 para $Ax^* = b$. Tem-se, inicialmente, $r_0 = b - Ax_0$ e $d_1 = r_0$, em que r_k e d_k são os vetores resíduo e vetor de direção de busca, respectivamente, na iteração k com $k = 1, 2, \dots, n-1$ e $n = |x|$. Os vetores r e d auxiliam o método dos gradientes conjugados na busca do mínimo da função f fazendo com que a solução x_k se aproxime cada vez mais de x^* . O vetor d satisfaz a condição de *A-ortogonalidade* (ou conjugada) $d_i^T A d_j = 0$ para $i \neq j$. Os vetores de direção d produzem o método de direção conjugada. Os resíduos são ortogonais entre si, ou seja, $r_i^T r_j = 0$ para $i \neq j$. Os vetores resíduo r_i e direção d_j são mutuamente ortogonais, de forma que, $r_i^T d_j = 0$ para $i \neq j$.

Em cada iteração os vetores r_k e d_k são preenchidos por

$$r_k = r_{k-1} - \alpha_k A d_k \quad (9)$$

e

$$d_{k+1} = r_k - \beta_k A d_k, \quad (10)$$

em que α e β são valores escalares obtidos, respectivamente, por

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

e

$$\beta_k = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}. \quad (12)$$

Finalmente, a solução em cada iteração k é obtida por

$$x_k = x_{k-1} + \alpha_k d_k. \quad (13)$$

O algoritmo do método dos gradientes conjugados é apresentado na Figura 3. Na linha 1, é definida a precisão numérica desejada. Os vetores x , r e d são inicializados nas linhas 3, 4 e 5, respectivamente. A variável contadora k é inicializada na linha 5 e a variável *erro* é inicializada na linha 6. Essas variáveis são utilizadas na estrutura de repetição *enquanto* das linhas 7 a 15. Nas linhas 8, 9 e 10 são computadas as fórmulas 11, 13 e 9, respectivamente. O erro de cada iteração é calculado na linha 11 por $\frac{\|x_k - x_{k-1}\|_\infty}{\|x_k\|_\infty}$ em que $\|x_k\|_\infty = \max\{|x_1|, |x_2|, \dots, |x_n|\}$. Incrementa-se em 1 o valor de k na linha 12. As fórmulas 12 e 10 são calculadas nas linhas 13 e 14, respectivamente.

Caso a precisão desejada não seja obtida em $k \leq n$ iterações, o algoritmo é encerrado. Isso significa que o algoritmo dos gradientes conjugados converge para a solução em, no máximo, n iterações. Caso o erro obtido seja menor que a precisão definida o algoritmo encerra.

Na prática, o método dos gradientes conjugados é geralmente utilizado em conjunto com alguma técnica de condicionamento. A seguir, na seção 2.8, são apresentadas algumas técnicas de condicionamento.

Algoritmo: Método dos gradientes conjugados.

```

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

Figura 3 Algoritmo do método dos gradientes conjugados.

2.8 Precondicionamento de sistemas de equações lineares

Apesar da boa fundamentação teórica do método dos gradientes conjugados e outros métodos iterativos, tais métodos geralmente possuem lenta convergência em problemas comuns como dinâmica de fluidos ou simulação de dispositivos eletrônicos. O precondicionamento é o ingrediente chave para o sucesso de métodos de subespaço de Krylov, como o método dos gradientes conjugados, para essas aplicações (SAAD, 2003). Tanto a eficiência quanto a robustez dos métodos iterativos podem ser melhoradas com a utilização do precondicionamento. Saad (2003), também, afirma que, em geral, a confiabilidade dos métodos iterativos, em se tratando de aplicações

variadas, depende da técnica de condicionamento utilizada.

Em termos simples, o condicionamento pode ser definido como uma forma de transformar um SEL em outro que seja mais fácil de resolver por meio de um método iterativo e que possua a mesma solução que o original. Esse SEL resultante poderá demandar menos iterações para convergir do que o SEL original, embora isso não seja garantido (SAAD, 2003).

O primeiro passo para se realizar o condicionamento é encontrar a matriz condicionadora M . Essa matriz pode ser encontrada de diversas formas, porém, precisa respeitar alguns requisitos mínimos. Do ponto de vista prático, é necessário que o custo de se solucionar $Mx = b$ seja mínimo, pois em cada iteração do método iterativo M , precisará ser calculada. Este custo, obviamente, será adicionado ao método de resolução de sistemas lineares aplicado. Outro requisito é que M precisa ser não singular. Isso significa que M deverá admitir inversa. Em termos simples, a matriz M não pode possuir linha ou coluna linearmente dependente.

Existem três formas de se aplicar o condicionador ao SEL. O condicionador pode ser aplicado pela esquerda em que se gera o sistema condicionado

$$M^{-1}Ax = M^{-1}b. \quad (14)$$

Analogamente, pode-se aplicar o condicionado pela direita, da seguinte maneira

$$AM^{-1}u = b, \quad x \equiv M^{-1}u, \quad (15)$$

em que $u = Mx$.

A terceira forma pode ser utilizada, quando o condicionador está disponível na forma fatorada $M = M_L M_R$, em que M_L e M_R são matrizes

triangulares. Nesse caso, o preconditionador pode ser separado em

$$M_L^{-1}AM_R^{-1}u = M_L^{-1}b, \quad x \equiv M_R^{-1}u. \quad (16)$$

Esse tipo de preconditionador é conhecido como *split*.

É necessário preservar a simetria da matriz original, quando a mesma é simétrica, para isso, pode-se utilizar um preconditionador do tipo *split*. Todavia, existem outras formas de preservar a simetria ou, se possível, tirar vantagem da simetria mesmo se M não está disponível na forma fatorada (SAAD, 2003). Esses detalhes são discutidos na subseção 2.8.1 para o método dos gradientes conjugados. Em seguida, na subseção 2.8.2, são apresentadas algumas técnicas de preconditionamento.

2.8.1 Método dos gradientes conjugados preconditionado

Considere as matriz A e a matriz preconditionadora M que aproxima A de certa forma, ambas simétricas positivo-definidas. Pode-se preconditionar o SEL das três formas já apresentadas nas equações 14, 15 e 16. As duas primeiras equações não são simétricas em geral. Para manter a simetria, pode-se utilizar uma das seguintes estratégias.

Se M está disponível na forma de uma fatoração incompleta de Cholesky, ou seja, $M = LL^T$, pode-se preservar a simetria usando um preconditionador do tipo *split* como na equação 16. Tem-se, então, a seguinte forma de aplicar o preconditionador

$$L^{-1}AL^{-T}u = L^{-1}b, \quad x = L^{-T}u. \quad (17)$$

Entretanto, nesse caso, não é necessário utilizar um preconditionador do

tipo *split* para manter a simetria. Pode-se utilizar o gradiente conjugado preconditionado apresentado na Figura 4. Utiliza-se a matriz preconditionadora M e a variável z_k para alterar o algoritmo do gradiente conjugado apresentado no algoritmo . Para mais detalhes sobre o gradiente conjugado preconditionado, veja Saad (2003).

Algoritmo: Método dos gradientes conjugados preconditionado.

```

1  $\varepsilon \leftarrow 10^{-16}$ ; // Precisão numérica
2  $x_0 \leftarrow 0$ ; // Estimativa inicial
3  $r_0 \leftarrow b - Ax_0$ ;
4  $z_0 = M^{-1}r_0$ ;
5  $d_0 \leftarrow r_0$ ;
6  $k \leftarrow 1$ ;
7  $erro \leftarrow \varepsilon + 1$ ;
  //  $n$  é a dimensão de  $x$ 
8 enquanto (  $(k \leq n) \wedge (erro > \varepsilon)$  ) 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   $z_k = M^{-1}r_k$ ;
13   $erro \leftarrow \frac{\|x_k - x_{k-1}\|_\infty}{\|x_k\|_\infty}$ ;
14   $k \leftarrow k + 1$ ;
15   $\beta_k \leftarrow \frac{r_k^T z_k}{r_{k-1}^T z_{k-1}}$ ;
16   $d_{k+1} \leftarrow z_k + \beta_k d_k$ ;
17 fim-enquanto;
```

Figura 4 Método dos gradientes conjugados preconditionado.

A seguir, são apresentadas algumas técnicas de preconditionamento comumente utilizadas.

2.8.2 Técnicas de condicionamento

Uma técnica muito simples para se efetuar o condicionamento é a matriz diagonal definida por

$$m_{ij} = \begin{cases} \frac{1}{a_{ij}}, & \text{se } i = j \\ 0, & \text{se } i \neq j \end{cases} \quad (18)$$

que é conhecida como condicionador de Jacobi (BENZI, 2002). Pode-se utilizar esta técnica de condicionamento sem que seja necessária a utilização extra de memória. Para isso, basta utilizar o inverso da diagonal principal de A diretamente.

Pode-se obter um condicionador utilizando a fatoração incompleta de Cholesky da matriz A . Esta fatoração baseia-se na decomposição de Cholesky, utilizada na resolução de SELs, decompondo-se a matriz A em $A = LL^T$, em que L é uma matriz triangular inferior.

No processo de decomposição da matriz A , elementos nulos podem tornar-se não nulos, modificando o padrão de distribuição de zeros da matriz. Para contornar essa situação, utiliza-se a fatoração incompleta e garante-se que os elementos nulos na matriz A permaneçam nulos em M . Esse condicionador é conhecido por condicionador incompleto de Cholesky com zero preenchimento ou IC(0) (LEW et al., 2009). Mais detalhes sobre o IC(0) podem ser encontrados em Benzi (2002).

Outra técnica baseada em uma fatoração incompleta é a fatoração incompleta LU com zero preenchimento conhecida como ILU(0). A fatoração incompleta calcula as matrizes triangulares inferior L e superior U , ambas esparsas. Pode-se definir um limiar de tolerância $\tau > 0$ de forma que os elementos $a_{ij} < \tau_i$ são descartados em M . O valor de τ_i é obtido

multiplicando-se τ pela norma da i -ésima linha da matriz original A . Neste trabalho utilizou-se a norma euclidiana $\|A_{i,*}\| = (\sum_{j=1}^n |a_{ij}|^2)^{1/2}$, como indicado em Saad (2003). Ainda, uma outra regra de descarte pode ser aplicada mantendo-se apenas os p maiores elementos nas linhas das matrizes L e R . Com isso, obtém-se a técnica de condicionamento conhecida como ILUT(p, τ).

Para Benzi (2002), existe uma certa dificuldade em se definir o valor do limiar τ . Geralmente, isso pode ser feito por meio de tentativas e erros. Bons resultados podem ser obtidos para valores no intervalo $10^{-4} < \tau < 10^{-2}$, porém o valor ótimo é fortemente atrelado ao problema (BENZI, 2002).

2.9 Redes neurais

Por mais que a computação tenha evoluído a passos largos nas últimas décadas, o cérebro humano, ainda, é o mais fascinante e incompreensível processador. O mesmo é complexo e efetua processamento, naturalmente, paralelo; seu armazenamento é adaptativo; possui controle de processos distribuído e capacidade para efetuar o processamento de, aproximadamente, 10^{11} a 10^{14} operações por meio de mais de 10^4 ligações entre elementos processados (BARRETO, 1997). Em contrapartida, um computador, *a priori*, efetua processamento sequencial; seu armazenamento é estático; possui controle centralizado de processos e capacidade para efetuar o processamento de aproximadamente 10^5 a 10^6 operações por meio de menos de 10 ligações entre os elementos processados.

Apesar de o processo de formação de pensamentos e de interação entre os neurônios não estarem totalmente decifrados, o conexionismo (*connectionism*) introduziu a ideia de que o cérebro é um sistema altamente

distribuído. É baseado nesse conceito que a atividade de formação do pensamento é estabelecida entre os neurônios (RUMELHART; RAMSEY; STICH, 1991). A quantidade de neurônios que compõe tais conexões, em nosso cérebro, são da ordem de 10^9 (BARRETO, 1997). Outro ponto positivo para as redes neurais é a sua robustez e tolerância a falhas, pois a eliminação de neurônios não afeta a sua funcionalidade global (RAUBER, 1998). Pode-se dizer que todas as funções, inclusive o pensamento e movimentos do organismo, estão intimamente relacionados ao funcionamento dessas pequenas células que compõem nosso sistema nervoso.

Os neurônios conectam-se uns aos outros por meio de sinapses e formam uma grande rede, chamada de rede neural. As sinapses transmitem estímulos por meio de diferentes concentrações dos elementos químicos $Na+$ (Sódio) e $K+$ (Potássio). Esse processo pode ser estendido por todo o corpo humano, como exemplo, tem-se a transmissão de um estímulo sensorial quando queimamos a ponta do dedo. A rede neural garante uma fabulosa capacidade de processamento e armazenamento de informação.

O sistema nervoso, formado por um conjunto extremamente complexo de neurônios, realiza comunicação por meio de impulsos. Quando um impulso é recebido, o neurônio o processa e, caso seja atingido um limite de ação, dispara um segundo impulso que produz uma substância neurotransmissora que flui do corpo celular para o axônio, que por sua vez pode, ou não, estar conectado a um dendrito de outra célula. O neurônio que transmite o pulso pode controlar a frequência de pulsos aumentando ou diminuindo a polaridade na membrana pós-sináptica.

Os neurônios e suas conexões, segundo Rumelhart, Hinton e Williams (1986), são os responsáveis pela determinação do funcionamento, com-

portamento e do raciocínio do ser humano. Ao contrário das redes neurais artificiais, redes neurais naturais não transmitem sinais negativos, sua ativação é medida pela frequência com que se emitem os pulsos contínuos e positivos. As redes neurais naturais não são uniformes, exceto em alguns raros pontos do organismo. Em redes neurais naturais, os pulsos não são síncronos nem contínuos, o que as difere de redes neurais artificiais.

Os neurônios são, basicamente, compostos por:

- **dendritos**, que recebem os estímulos transmitidos pelos outros neurônios;
- corpo do neurônio, também chamado de **somma**, que é responsável por coletar e combinar informações vindas de outros neurônios;
- **axônio**, composto de uma fibra tubular que pode alcançar até alguns metros e é responsável por transmitir os estímulos para outras células.

A seguir, é apresentada a técnica de redes neurais artificiais que se baseia no conceito biológico das redes neurais.

2.9.1 Redes neurais artificiais

Uma rede neural artificial (RNA) pode ser entendida como uma técnica computacional que apresenta um modelo matemático baseado na estrutura neural de organismos inteligentes. Formalmente, é definida por Vellasco (2007), da seguinte forma:

Uma rede neural é um sistema computacional constituído por unidades conhecidas como neurônios. Os neurônios são elementos processadores interligados, trabalhando em paralelo para desempenhar uma determinada tarefa. Os modelos RNAs formam uma importante técnica estatística não linear capaz de resolver uma gama de problemas

de grande complexidade. Por isso, são modelos úteis em situações que não é possível definir explicitamente uma lista de regras. Em geral, isso acontece quando o ambiente gerador dos dados muda constantemente. As principais áreas de atuação são para classificação de padrões e previsão.

Uma RNA pode ser definida como uma técnica de aproximação de funções por regressão não linear.

2.9.2 Especificações

O neurônio é a unidade fundamental de uma RNA. Na Figura 5, é apresentado um esquema funcional de um i -ésimo neurônio em uma rede, ou seja, sua estrutura esquemática interior. As entradas do neurônio estão representadas por x_j ($1 \leq j \leq n$), que são as saídas dos neurônios da camada anterior.

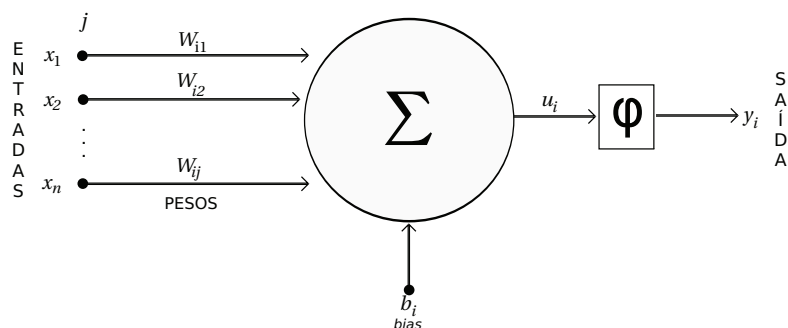


Figura 5 Esquema do i -ésimo neurônio de uma RNA, adaptado de Vellasco (2007).

Os pesos W_{ij} são os parâmetros da rede, em que $1 \leq j \leq n$ e n é a quantidade de entradas do neurônio i . Os parâmetros representam a “memória” da rede, ou seja, a experiência ganha como resultado das apresentações dos padrões à rede. São os pesos que combinam a não linearidade da rede

neural. Na modelagem, recorre-se ao artifício de tratar o *bias* b_i como uma entrada, de modo que, durante o processo de treinamento, a atualização dos pesos aconteça para todos os parâmetros.

u_i representa a combinação das entradas e dos pesos. Corresponde à soma ponderada das entradas e é utilizado como entrada para a função de ativação φ .

y_i é a saída do i -ésimo neurônio que depende do nível de ativação aplicado ao neurônio pela função de ativação. Ressalta-se que a função de ativação refere-se à parte não linear de cada neurônio, sendo o único local onde ocorre tal não linearidade.

Em uma rede neural, os parâmetros a serem estimados são os pesos. Como em cada neurônio chegam todas as entradas, então, o *bias* aparecerá associado a uma entrada fixa $+1$ ou -1 (VELLASCO, 2007).

2.9.3 Treinamento de redes neurais artificiais

Uma característica importante das RNAs é a sua capacidade de generalização. Isso significa que, após o treinamento, a rede é capaz de encontrar bons resultados para entradas que não foram apresentadas à rede durante seu treinamento. A capacidade de generalização da RNA é adquirida durante o seu aprendizado. Os dois principais processos de aprendizagem são: *aprendizado supervisionado* e *aprendizado não supervisionado* (BENINI, 2008).

O aprendizado supervisionado é um dos processos mais comuns de aprendizagem. Utiliza-se o aprendizado supervisionado quando se tem disponíveis as amostras de entrada e o resultado esperado. Isso significa que é possível apresentar à rede um conjunto de entradas e o resultado que se

deseja obter. Dessa forma, pode-se calcular o erro pela diferença entre a saída desejada e o resultado atual da rede. O aprendizado ocorre à medida que os pesos são atualizados por meio do erro obtido na iteração anterior do processo de treinamento da RNA.

O aprendizado não supervisionado pode ser utilizado quando não se pode estabelecer um conjunto de amostras de entrada e o resultado esperado. Nesse caso, o treinamento da RNA não pode ser feito por um algoritmo que utiliza o erro para ajustar os pesos da rede.

Segundo Zuben e Attux (2007), o aprendizado não supervisionado é predominante no cérebro humano. É verificado que as propriedades estruturais e fisiológicas das sinapses no córtex cerebral são influenciadas pelos padrões de atividade que ocorrem nos neurônios sensoriais. No entanto, em essência, nenhuma informação prévia acerca do conteúdo ou significado do fenômeno sensorial está disponível. A implementação de modelos computacionais, para ajuste de pesos sinápticos, via treinamento não supervisionado, deve recorrer apenas aos dados de entrada, tomados como amostras independentes de uma distribuição de probabilidade desconhecida (ZUBEN; ATTUX, 2007).

No aprendizado não supervisionado, os pesos se agrupam de acordo com as características dos subconjuntos dos estímulos apresentados às regiões da camada de entrada (BENINI, 2008). Assim, os pesos se ajustam gradualmente de acordo com as entradas da rede neural. Benini (2008) conclui que quanto mais evidentes forem as particularidades da população de entrada, melhor será o aprendizado da rede, e vice-versa.

Em ambos tipos de treinamento, supervisionado e não supervisionado, existem alguns problemas que precisam ser observados durante o trei-

namento. Um exemplo, conhecido como *overfitting*, ocorre quando a rede é treinada excessivamente e perde sua capacidade de generalização passando a memorizar os dados de entrada (VELLASCO, 2007). Em contrapartida, abortar o treinamento de uma RNA, de forma prematura, pode prejudicar seu desempenho, pois a rede terá valores sensíveis aos valores iniciais aleatórios. Nesse caso, temos uma situação de *underfitting*.

Na subseção 2.9.5, é abordado um tipo de RNA que utiliza o aprendizado não supervisionado, conhecido como mapa auto-organizável de Kohonen ou, simplesmente, mapa auto-organizável.

2.9.4 Redes neurais artificiais e redução de *profile*

Até o presente momento, não se encontrou algum tipo de aplicação direta de RNAs aos problemas de redução de largura de banda e de *profile*. Porém, os autores Hoit, Stoker e Consolazio (1994) utilizaram de RNA para determinar os parâmetros (*ranking parameters*) do algoritmo *profile-front minimization* (PFM) de Hoit e Wilson (1983). Como entrada para a rede, utilizaram de dez parâmetros que representam razões das características estruturais do grafo, por exemplo, número mínimo de vértices conectados dividido pelo número máximo de vértices conectados. Os autores utilizaram uma rede neural de três camadas, também conhecida como *multi-layer perceptron* (MLP), que utiliza a técnica de treinamento supervisionado conhecida como *back propagation*. Por isso, não é capaz de obter resultados satisfatórios para estruturas diferentes daquelas que foram usadas no treinamento. Isso ocorre em virtude da utilização de uma rede neural de treinamento supervisionado.

2.9.5 Mapas auto-organizáveis

Os mapas auto-organizáveis (MAO), ou redes de Kohonen, foram inicialmente desenvolvidos por Kohonen (1981) e consiste de uma rede neural artificial interconectada. Em razão da propriedade de auto-organização, os MAOs podem ser aplicados a problemas de clusterização e de ordenação espacial de dados. Nesse caso, haverá um mapeamento dos dados de entrada para o arranjo de neurônios do MAO (ZUBEN; ATTUX, 2007). Como exemplo, tem-se a aplicação de MAO para solucionar o problema do caixeiro viajante apresentado por Brocki (2007).

Uma rede de Kohonen é um arranjo de neurônios, geralmente restrito a espaços de dimensão 1 ou 2, em que se procura estabelecer e preservar noções de vizinhança (ZUBEN; ATTUX, 2007). No caso de um MAO unidimensional, tem-se uma sequência ordenada de neurônios e o número de pesos é igual ao número de entradas.

Na Figura 6, tem-se um exemplo de topologia de um mapa auto-organizável unidimensional. Os valores de entrada e saída para o MAO são, respectivamente, x_1, x_2, \dots, x_{dim} e y_1, y_2, \dots, y_n . Os pesos dos neurônios são representados por W_{ij} , que é a conexão do peso da entrada x_j com o neurônio i , em que $1 \leq j \leq dim$, $1 \leq i \leq n$, dim é a quantidade de entradas e n é a quantidade de neurônios do MAO.

Uma abordagem competitiva de aprendizado não supervisionado muito utilizada é a *winner takes all* (WTA). Quando um vetor de entrada é passado para o MAO, executa-se o processo de aprendizagem. Somente o neurônio vencedor terá seu peso sináptico atualizado. O neurônio vencedor será o que tiver pesos mais correlacionados aos valores de uma determinada entrada, definida aleatoriamente. Correlação é o produto escalar do vetor

de entrada com os pesos sinápticos considerados (BROCKI, 2007).

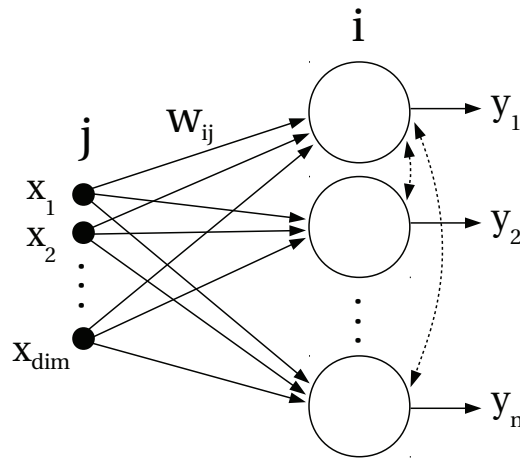


Figura 6 Exemplo de topologia de mapa auto-organizável, adaptado de Zuben e Attux (2007).

O processo de aprendizado não supervisionado para o MAO é descrito por

$$W_i = W_i + \eta(x - W_i). \quad (19)$$

O processo de aprendizado consiste na atualização dos pesos W_i do neurônio vencedor i , com $0 \leq i \leq n$, em que n é a quantidade de neurônios, x é valor de entrada e a taxa de aprendizado é $0 < \eta < 1$. A taxa de aprendizado é um parâmetro do algoritmo.

Uma variação do WTA que, geralmente, apresenta melhor convergência é o *winner takes most* (WTM), pois permite a ativação de neurônios que, raramente, seriam ativados (BROCKI, 2007). Ativação é o processo de modificação dos pesos sinápticos de um neurônio. A diferença entre os dois algoritmos é que no WTM mais neurônios têm seus pesos modificados em

uma única iteração. Nesse caso, o vencedor e seus vizinhos serão ativados.

Uma forma simplificada de implementar o WTM é acrescentar à equação 19 as seguintes equações de ativação:

$$W_{i-1} = W_{i-1} + \eta(x - W_{i-1}), \quad (20)$$

$$W_{i+1} = W_{i+1} + \eta(x - W_{i+1}). \quad (21)$$

O algoritmo presente na Figura 7 é um exemplo dos passos para o treinamento de um MAO. A taxa de aprendizado η é utilizada na linha 7 para modificar os pesos sinápticos do neurônio vencedor e seus vizinhos pelas equações 19, 20 e 21.

Algoritmo: Algoritmo de treinamento de um MAO.

Entrada: vetor de entrada, *iter_max* e taxa de aprendizado η .

Saída: rede neural de Kohonen treinada.

```

1 iter ← 1;
2 atribuem-se pesos aleatórios aos neurônios;
3 enquanto ( iter ≤ iter_max ) faça
4   seleciona-se um valor aleatório do conjunto de
   treinamento;
5   encontra-se o neurônio vencedor;
6   encontram-se os vizinhos do vencedor;
7   modificam-se os pesos sinápticos dos neurônios,
   considerando-se a taxa de aprendizado  $\eta$ ;
8   reduza o valor de  $\eta$ ;
9   iter ← iter + 1;
10 fim-enquanto;
11 retorna rede neural de Kohonen treinada;
```

Figura 7 Pseudocódigo de um algoritmo de treinamento de um MAO.

3 MAPA AUTO-ORGANIZÁVEL APLICADO AOS PROBLEMAS DE REDUÇÃO DE LARGURA DE BANDA E DE *PROFILE*

Neste capítulo, descrevem-se as metodologias de pesquisa, de desenvolvimento e as ferramentas utilizadas para a aplicação dos mapas auto-organizáveis no problema de reduções de largura de banda e de *profile*. Na seção 3.1, é apresentada a metodologia de pesquisa. A metodologia e os detalhes de desenvolvimento são apresentados na seção 3.2.

Desenvolveu-se uma heurística, baseada em mapa auto-organizável, para realizar reduções de largura de banda e de *profile*. Os detalhes de implementação e um exemplo de funcionamento são apresentados na seção 3.3. As ferramentas utilizadas são apresentadas na seção 3.4. A metodologia empregada na realização das simulações é apresentada na seção 3.5.

3.1 Metodologia de pesquisa

A metodologia de pesquisa utilizada neste trabalho iniciou-se com um estudo do problema de reduções largura de banda e de *profile*. Este estudo envolveu a pesquisa e a compreensão dos principais algoritmos e meta-heurísticas utilizados na resolução desses problemas e semelhantes. Um exemplo de problema semelhante é o problema do caixeiro viajante (PCV). Estudou-se uma proposta de soluções para estes problemas envolvendo redes neurais artificiais e mapa auto-organizável. Em específico, estudaram-se técnicas de aprendizado não supervisionado, como redes neurais de Hopfield e Tank (1985) e mapas auto-organizáveis (KOHONEN, 1981).

Optou-se pela utilização de rede neural de treinamento não supervisionado, em virtude de sua capacidade de utilização em instâncias de

estruturas desconhecidas. Isso possibilita que a meta-heurística seja mais genérica e possa ser aplicada em diversos tipos de matrizes, inclusive matrizes assimétricas. Apesar de as redes neurais de Hopfield e Tank (1985) terem sido as primeiras a serem utilizadas para solucionar o PCV, os resultados estavam aquém dos obtidos por outras heurísticas na época (BENINI, 2008). Há uma similaridade entre o problema de redução de largura de banda e o PVC, se for considerado que a renumeração dos vértices no problema de redução de largura de banda é similar à ordenação dos vértices que precisam ser percorridos no PCV. Sendo assim, utilizou-se mapa auto-organizável (MAO), por causa de sua capacidade de ordenação espacial e por ter obtido melhores resultados no PCV do que os outros tipos de redes neurais citados no presente trabalho.

Por último, avaliou-se o possível estado da arte para o problema de redução de largura de banda. Não se encontrou um estudo específico realizado nesse sentido. Essa revisão foi útil para se decidir qual método para redução de largura de banda seria utilizado para as comparações com o método desenvolvido neste trabalho. Escolheu-se o *Cuthill-McKee* reverso (CMr) por ser um método clássico e eficiente e o *variable neighbourhood search for bandwidth reduction* (VNS-band) de Mladenovic et al. (2010), pois, segundo os autores, os resultados indicam que o referido método pode ser o estado da arte no problema de redução de largura de banda.

3.2 Metodologia de desenvolvimento

A implementação da heurística CMr, em linguagem *C++*, foi realizada conforme o algoritmo presente na subseção 2.4. Utilizou-se, também, a implementação em *C++* utilizada em Oliveira e Chagas (2014) do algo-

ritmo de George e Liu (1979) para encontrar um vértice pseudo-periférico. A escolha de um vértice pseudo-periférico foi utilizada para se encontrar o vértice inicial para a heurística CMr. A utilização da heurística CMr, utilizando como vértice inicial um vértice pseudo-periférico, não pode ser considerada um nova heurística. Conforme descrito em Oliveira e Chagas (2014), para simplificar, é utilizada a sigla **CMr-pseudo**, para indicar a utilização da heurística CMr, juntamente com o algoritmo de George e Liu (1979), para encontrar um vértice pseudo-periférico. Foi utilizado o método dos gradientes conjugados preconditionado por Jacobi de Oliveira (2014).

Implementou-se como preconditionadores do método dos gradientes conjugados (MGC), a fatoração incompleta de Cholesky (IC(0)) e a fatoração incompleta LU com limiar (ILUT(p , τ)). Os preconditionadores, assim como o MGC, foram implementados em linguagem *C++*. Os preconditionadores IC(0) e ILUT(p , τ) não foram utilizados neste trabalho porque suas implementações ficaram ineficientes. Devido à utilização de uma estrutura de dados baseada em listas de adjacências, os custos computacionais de se realizar as fatorações incompletas de Cholesky e LU ficaram muito altos se comparados com o custo do MGC. Como exemplo, o tempo, para realizar a fatoração incompleta de Cholesky, em um grafo com 10728 vértices, foi de, aproximadamente, 962 segundos. A resolução do sistema de equações lineares (SEL), representado por esse grafo, pelo MGC sem preconditionamento e sem reduções de largura de banda e de *profile*, demora menos de 26 segundos, em média.

Um esquema de armazenamento que pode ser utilizado é a matriz *skyline* (FELIPPA, 1975), também conhecida como matriz de banda variável ou matriz *profile*. Na matriz *skyline*, armazenam-se apenas as entradas a

partir do primeiro elemento não nulo até a diagonal principal em forma de coluna. Se a matriz é simétrica, apenas a matriz triangular inferior é armazenada.

Um vantagem da matriz *skyline* está no fato de que o acesso pode ser realizado com custo $O(1)$. Esse esquema de armazenamento é importante para métodos diretos e é adequado para as decomposições de Cholesky e LU (DUFF; ERISMAN; REID, 1986). Porém, pode haver uma ocupação desnecessária de memória ao se armazenar os elementos nulos presentes na banda da matriz. Para minimizar esse problema, costuma-se realizar a redução de largura de banda antes de utilizar essa forma de armazenamento.

Neste trabalho, utilizam-se listas de adjacências, pois, em muitas situações práticas, a malha tem sua estrutura modificada constantemente. Computacionalmente, modificar uma estrutura, baseada em listas de adjacências, tem menor custo do que modificar matrizes ou vetores. Em uma estrutura matricial, baseada em vetores simples (*arrays*), ao se inserir ou remover elementos, as estruturas precisam ser devidamente duplicadas e após, a estrutura original deve ser completamente desalocada da memória. O mesmo acontece para estruturas mais avançadas, como o *Vector* na linguagem *C++*. Nas listas de adjacências, essas operações são feitas na estrutura já existente via ponteiros.

Utilizou-se o arquivo executável da implementação da heurística *VNS-band*, originalmente implementado para redução de largura de banda apenas, compatível com sistema operacional *Linux*. Esse executável foi implementado em linguagem *C*. Especificamente, o executável do *VNS-band* foi gentilmente fornecido pelo professor Dr. Dragan Urosevic. Esse executável, enviado por um dos autores da heurística *VNS-band*, só executa com

instâncias de, no máximo, 500 mil vértices. O cálculo do *profile* foi possível, aplicando-se a ordenação encontrada pela heurística *VNS-band*, na matriz original e utilizando a matriz reordenada para efetuar o cálculo, nos experimentos realizados com a base *Harwell-Boeing*, mostrados na seção 4.1, nas páginas 67 e 76.

Nos demais experimentos, envolvendo resolução de sistemas de equações lineares, o projeto computacional principal gerava um arquivo de entrada para o executável da heurística *VNS-band*, com base no grafo original. O executável da heurística *VNS-band* era executado e escrevia a renumeração encontrada em um arquivo de saída. Com base nesse arquivo de saída, o grafo original era renumerado no projeto computacional principal e efetuava-se o cálculo da nova largura de banda e do *profile*.

Os tempos de escrita e leitura de arquivos não foram considerados nos experimentos. A execução do executável da heurística *VNS-band* requer os seguintes parâmetros, na seguinte ordem: arquivo de entrada contendo a matriz de coeficientes, arquivo de saída de resultados, tempo limite para execução e tipo do arquivo de entrada.

3.3 Detalhes da implementação da heurística *MAO-band*

Desenvolveu-se uma heurística, baseada em mapa auto-organizável, para realizar reduções de largura de banda e de *profile*, denominada *MAO-band*. Implementou-se a heurística *MAO-band* em linguagem *C++*. Essa decisão levou em consideração as facilidades, como orientação a objetos, que a linguagem proporciona e o fato de que os resultados serão comparados com os resultados da heurística *VNS-band* que, também, está implementado nessa linguagem. Para facilitar na implementação da heurística *MAO-band*,

utilizou-se o *framework KNNL* (HABDANK-WOJEWODZKI; RYBARSKI, 2006). Esse *framework*, desenvolvido em *C++*, é gratuito e possui diversos métodos que auxiliam na implementação de mapas auto-organizáveis.

A heurística *MAO-band* requer, como parâmetros, um arquivo de entrada, a taxa de aprendizado η e a quantidade máxima de épocas para treinamento da rede neural. O arquivo de entrada deve conter apenas caracteres textuais em que cada linha contém os índices de um valor não nulo da matriz esparsa. Esses índices representam uma aresta do grafo que a matriz representa. Um parâmetro opcional do algoritmo é o tempo máximo de execução. Caso esse parâmetro seja fornecido, o treinamento pode ser encerrado precocemente se o tempo de execução ultrapassar o valor desse parâmetro.

Para cada instância, a heurística *MAO-band* gera um MAO unidimensional de vizinhança simples em que cada neurônio recebe dois valores que representam os índices i e j não nulos da matriz de entrada.

A heurística *MAO-band* foi desenvolvida para que possa ser executada em matrizes simétricas e assimétricas. Caso a matriz represente um grafo não direcionado, cada neurônio representará uma aresta $\{v_i, v_j\}$ do grafo. Assim como nos neurônios de uma rede *Adaptive Linear Element* (ADALINE), proposta por Widrow e Hoff (1960), os neurônios da heurística *MAO-band* não possuem função de ativação. Utilizou-se a abordagem de aprendizado *winner takes most* (WTM), pois possui melhor convergência que o *winner takes all* (WTA).

A Figura 8 apresenta, em alto nível, os passos executados pela heurística *MAO-band*. Seus parâmetros de entrada são um conjunto E de arestas, taxa de aprendizado η , quantidade máxima de épocas de treinamento

max_epocas e o parâmetro opcional de tempo limite de execução $tempo_max$. A saída do algoritmo será composta pelo conjunto de vértices renumerados $vertices_renumerados$, a largura de banda β e o $profile$ final.

Algoritmo: MAO-band.

Entrada: conjunto de arestas E , η , max_epocas , $tempo_max$.
Saída: vértices renumerados, β , $profile$.

```

1  $\beta \leftarrow E.calculaBanda()$ ;
2  $profile \leftarrow E.calculaProfile()$ ;
3  $MAO \leftarrow criarMAOVizinhancaSimples(1,E)$ ; // cria MAO unidimensional
   de vizinhança simples
4  $ce \leftarrow N * 0.025 + 5$ ;
5  $epoca \leftarrow 1$ ;
6  $encerra\_execucao \leftarrow false$ ;
7 enquanto (  $(epoca \leq max\_epocas)$  e  $(encerra\_execucao \neq true)$  ) faça
8    $MAO.wtm(\eta, ce)$ ; // Executa algoritmo de aprendizado WTM
   vezes
9    $\eta = \eta * 0,98$ ; // Reduz  $\eta$  em 2% para refinar o aprendizado
10   $e \leftarrow MAO.retornaArestas()$ ;
11   $\beta\_atual \leftarrow e.calculaBanda()$ ;
12   $profile\_atual \leftarrow e.calculaProfile()$ ;
13  se (  $(\beta\_atual < \beta)$  ou  $(profile\_atual < profile)$  ) então
14     $\beta \leftarrow \beta\_atual$ ;
15     $profile \leftarrow profile\_atual$ ;
16  fim-se;
17  senão
18     $encerra\_execucao \leftarrow true$ ; // Encerra a execução caso não
   haja redução de largura de banda
19  fim-se;
20  se (  $tempo\_max > 0$  ) então
21     $tempo\_exec \leftarrow tempoExecucao()$ ; // Obtém o tempo de execução
   atual
22    se (  $tempo\_exec > tempo\_max$  ) então  $encerra\_execucao \leftarrow true$ ;
   // Encerra a execução caso ultrapasse o tempo máximo
   de execução
23  fim-se;
24   $epoca \leftarrow epoca + ce$ ;
25 fim-enquanto;
26  $vertices\_renumerados \leftarrow MAO.verticesRenumerados()$ ;
27 retorna  $vertices\_renumerados$ ,  $\beta$ ,  $profile$ ;
```

Figura 8 Pseudocódigo da heurística MAO-band.

Nas linhas **1**, **2** calculam-se, respectivamente, a largura de banda β e o *profile* do grafo inicial. Um mapa auto-organizável unidimensional de vizinhança simples é criado na linha **3**. Um conjunto $ce = (N * 0,025) + 5$ de épocas é definidos na linha **4**, em que N é a quantidade de vértices do grafo. O cálculo de ce foi definido empiricamente com o intuito de que seu valor aumente proporcionalmente em relação à quantidade de vértices da instância. Nas linhas **5** e **6** são definidas a variável contadora de épocas e de controle de execução, respectivamente.

O algoritmo executará o laço de repetição das linhas **7–25** enquanto não for atingido o número máximo de épocas ou a variável *encerra_execucao* não tiver seu valor alterado para *true*. Antes de atingir a quantidade máxima de épocas de treinamento, a heurística MAO-*band* pode encerrar seu treinamento caso não haja melhora no resultado após uma quantidade ce de épocas de treinamento ou caso o tempo máximo de execução seja atingido. Para isso, a heurística MAO-*band* executa ce épocas de treinamento não supervisionado WTM com taxa de aprendizado η na linha **8**. O algoritmo **7**, presente na subseção 2.9.5, mostrado na página 48, é um exemplo de algoritmo WTM. Na linha **9**, o valor da taxa de aprendizado η é reduzido em 2% para refinar o aprendizado no decorrer das épocas de treinamento.

Atribui-se à variável e as arestas presentes em E renumerados. Caso os valores obtidos pela rede neural não representem exatamente uma aresta, é feito um ajuste nos valores dos neurônios. Esse ajuste é um arredondamento para os inteiros mais próximos que representem uma aresta, que ainda não possui um neurônio no MAO que a represente. Por exemplo, caso o valor obtido por um neurônio seja $(v_i = 2, v_j = 3,94)$, o seu valor será ajustado para $v_i = 2, v_j = 4$, caso nenhum outro neurônio represente essa aresta.

Os resultados parciais para largura de banda e *profile* encontrados são verificados nas linhas **10** e **11**, respectivamente. Caso a largura de banda atual seja menor que a largura de banda já obtida ou o *profile* encontrado seja menor que o já obtido, atualizam-se os valores de β e *profile* nas linhas **14** e **15**. Caso não ocorra redução de largura de banda ou de *profile*, define-se o valor da variável *encerra_execucao* como *true* na linha **18** para que o treinamento seja finalizado. Caso o parâmetro opcional *tempo_max* seja fornecido e seja maior que 0, a condicional da linha **20** será verdadeira. Nesse caso, obtém-se na linha **21** o tempo atual de execução e na linha **22** verifica-se este é maior do que *tempo_max*. Em caso afirmativo, o valor da variável *encerra_execucao* é definido como *true*, pois se deve encerrar o treinamento. Na linha **24**, incrementa-se *ce* ao valor da variável *epoca*.

Finalmente, na linha **26**, atribui-se à variável *vertices_renumerados* os vértices em sua nova numeração que serão retornados pelo algoritmo junto com os valores de β e *profile* na linha **27**. A nova numeração dos vértices é feita ordenando-se sequencialmente os vértices do primeiro neurônio, depois do segundo e assim por diante até que todos os vértices tenham sido numerados.

A Figura 9 é um exemplo de um MAO-*band* após seu treinamento com as arestas E do respectivo grafo não direcionado $G(V, E)$ e a renumeração final do mesmo. Ainda, na Figura 9, são apresentadas as matrizes de adjacência que representam G antes e depois da renumeração. Seja a função $f : V \rightarrow \{1, 2, \dots, n\}$ que rotula os vértices, em que $|V| = n$. Nesse exemplo, tem-se $f : f(v_1) = 1, f(v_2) = 2, f(v_3) = 3, f(v_4) = 4, f(v_5) = 5$. Considere $E = \{e_1 = (v_1, v_5), e_2 = (v_5, v_2), e_3 = (v_2, v_3), e_4 = (v_3, v_5), e_5 = (v_1, v_4), e_6 = (v_5, v_1), e_7 = (v_2, v_5), e_8 = (v_3, v_2), e_9 = (v_5, v_3), e_{10} = (v_4, v_1)\}$.

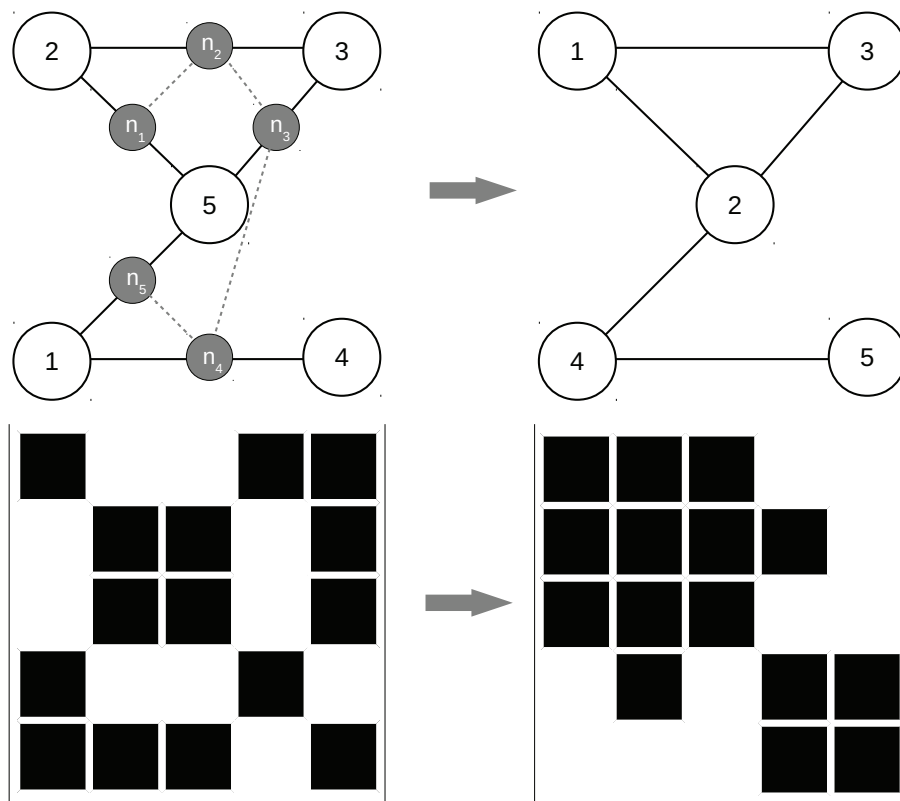


Figura 9 Exemplo de renumeração dos vértices de um grafo e de reordenamento da matriz de adjacência correspondente feito pela heurística MAO-band.

Inicialmente um MAO-band é criado com $|E|/2 = 5$ neurônios, pois para um grafo não direcionado um único neurônio é criado para representar as arestas (v_i, v_j) e (v_j, v_i) . No processo de criação dos neurônios, seus pesos iniciais são aleatórios. Após o seu devido treinamento, sua configuração será semelhante à apresentada do lado esquerdo superior da Figura 9. Cada neurônio, representado por um círculo cinza, estará associado a uma aresta de E .

Nesse exemplo, após o treinamento, os neurônios possuem os seguintes valores: $n_1 = (2, 5)$, $n_2 = (2, 3)$, $n_3 = (3, 5)$, $n_4 = (1, 4)$, $n_5 = (1, 5)$. Então, a nova rotulação dos vértices de G será $f : f(v_2) = 1$, $f(v_5) = 2$, $f(v_3) = 3$, $f(v_1) = 4$, $f(v_4) = 5$. Pois 2, 5, 3, 1 e 4 é a ordem, em que os vértices estão, desconsiderando-se as repetições, na representação das arestas dos neurônios da heurística MAO-*band*. O resultado da nova rotulação de G é apresentado do lado superior direito da Figura 9.

Ainda no exemplo da Figura 9, temos que a largura de banda inicial de G é $\beta = 4$ e o seu *profile* inicial é $profile = 8$. Após a renumeração realizada pela heurística MAO-*band*, a largura de banda de G é $\beta = 2$ e o seu *profile* é $profile = 6$. Houve uma pequena redução na largura de banda e no *profile*.

Esse exemplo, além de apresentar o funcionamento da heurística MAO-*band*, é útil para apontar uma limitação da heurística. A heurística MAO-*band* não leva em consideração informações importantes para o problema de redução de largura de banda, como o grau de cada vértice, a excentricidade do grafo ou mesmo se haverá uma ligação entre vértices com grande diferença em suas numerações finais. Por não considerar esses detalhes, o desempenho da heurística MAO-*band* pode ser prejudicado.

A heurística MAO-*band* tenta encontrar uma ordenação nas arestas, como um caminho no problema do caixeiro viajante. A intenção é fazer com que os vértices adjacentes fiquem com numeração próxima, assim a largura de banda do vértice será pequena. Como os mapas auto-organizáveis trabalham com a organização dos dados, não se encontrou uma forma de utilizar informações como o grau dos vértices.

Caso haja uma ligação entre vértices com grande diferença em suas numerações finais, isso só poderá ser verificado ao término do treinamento da heurística MAO-*band*. Por causa disso, o treinamento é feito em conjuntos menores de épocas para que seja possível avaliar a solução encontrada. Porém, para instâncias grandes o tempo máximo pode ser ultrapassado logo no primeiro conjunto de treinamento.

3.4 Ferramentas utilizadas

Utilizou-se o sistema operacional *Slackware 14.1 64 bits*, o editor de textos *Kate 3.10.5* para auxiliar na escrita dos códigos em linguagem *C++* e o *framework KNNL* (HABDANK-WOJEWODZKI; RYBARSKI, 2006) para auxiliar na criação dos MAOs. Para que fosse possível utilizar maior precisão nos cálculos, utilizou-se a biblioteca *GNU MPFR* com precisão de 256 *bits*. O compilador utilizado foi o *G++ 4.8.2* e para ajudar na compilação utilizou-se o *GNU Make 3.82*.

O computador utilizado na implementação e execução das simulações continha um processador *Intel(R) Core(TM) i5-3570 CPU @ 3.40GHz* com 6144 KB de memória *cache* e 16GB de memória principal.

3.5 Metodologia utilizada nas simulações

Nas simulações com matrizes da base *Harwell-Boeing*, apresentadas na seção 4.1, cada uma das quatro heurísticas CMr, CMr-pseudo, MAO-*band* e VNS-*band* foram executadas dez vezes para todas as instâncias. Para a heurística VNS-*band*, foram utilizados 2 segundos de tempo máximo nas execuções com as 33 instâncias bastante pequenas e 20 segundos para as 80 instâncias pequenas. Para a heurística MAO-*band*, foram utilizadas 500

épocas como limite máximo. A taxa de aprendizado utilizada foi $\eta = 0,9$ pois, no geral, obtiveram-se melhores resultados em vários testes empíricos utilizando esse valor.

Nas simulações com SELs, oriundos da discretização das equações de condução do calor e de Laplace por volumes finitos resolvidos pelo método dos gradientes conjugados preconditionado por Jacobi, apresentadas nas seções 4.2 e 4.3, respectivamente, cada uma das quatro heurísticas CMr, CMr-pseudo, MAO-*band* e VNS-*band* foram executadas de três a dez vezes para todas as instâncias. O parâmetro utilizado para a heurística MAO-*band* e para a heurística VNS-*band* foi o tempo máximo de execução de 1 segundo. Como a heurística CMr executou sempre em tempo menor que 1 segundo, utilizou-se esse valor, pois o parâmetro precisa ser inteiro por causa de uma restrição do executável da heurística VNS-*band*. Utilizou-se a taxa de aprendizado $\eta = 0,9$, definida empiricamente, para a heurística MAO-*band*.

4 RESULTADOS E ANÁLISES

Neste capítulo, descrevem-se os detalhes, resultados e análises das simulações com matrizes da base *Harwell-Boeing* e com três conjuntos de sistemas de equações lineares (SEL) oriundos das discretizações das equações de condução de calor e de Laplace por volumes finitos.

Na seção 4.1, são apresentados os resultados obtidos nas simulações com as heurísticas CMr, CMr-pseudo, VNS-*band* e MAO-*band* realizadas em 113 instâncias da base *Harwell-Boeing* (DUFF; GRIMES; LEWIS, 1989). Essa base contém matrizes muito pequenas, tanto simétricas quanto assimétricas. As instâncias são mostradas em 33 instâncias (bastante pequenas), de 30 a 237 vértices e 80 instâncias (pequenas), de 207 a 1104 vértices.

Na seção 4.2, são apresentados resultados e análises de simulações preliminares na resolução de SELs resultantes da discretização da condução do calor por volumes finitos. Na seção 4.2.2, são apresentados resultados e análises das simulações com SELs oriundos da discretização da condução do calor por volumes finitos resolvidos pelo método dos gradientes conjugados preconditionado por Jacobi. Esses SELs foram gerados pelo projeto computacional de simulações da dissertação de Oliveira (2014).

Na seção 4.3, são apresentados resultados e análises das simulações com SELs em que as linhas das matrizes são numeradas aleatoriamente. Esses SELs são oriundos da discretização da equação de Laplace por volumes finitos e são resolvidos pelo método dos gradientes conjugados preconditionado por Jacobi (OLIVEIRA, 2014). Na seção 4.3.2, são apresentados resultados e análises das simulações com SELs numerados por uma curva de preenchimento de espaço e oriundos da discretização da equação de Laplace por volumes finitos resolvidos pelo método dos gradientes

conjugados preconditionado por Jacobi (OLIVEIRA, 2014).

4.1 Simulações com as heurísticas CMr, CMr-pseudo, VNS-band e MAO-band com matrizes da base *Harwell-Boeing*

Cada uma das quatro heurísticas CMr, CMr-pseudo, MAO-band e VNS-band foram executadas dez vezes com cada uma das 113 instâncias. A variância e o desvio padrão para o tempo de execução, de todas as quatro heurísticas, foram menores do que 0,0081 e 0,0953, respectivamente.

Para a heurística VNS-band, foram utilizados 2 segundos de tempo máximo nas execuções com as 33 instâncias bastante pequenas e 20 segundos para as 80 instâncias pequenas. Para a heurística MAO-band, foram utilizadas 500 épocas como limite máximo; porém, a execução termina antes, caso não haja melhora no resultado, conforme explicado na subseção 3.3, na página 56. A taxa de aprendizado utilizada foi $\eta = 0,9$ para que o aprendizado seja mais rápido no início das iterações e, no decorrer do aprendizado, seu valor é decrementado para refinar o aprendizado, conforme descrito no algoritmo, mostrado na página 55.

Nas tabelas presentes nesta subseção, a dimensão da matriz de entrada será representada por Dim., a menor largura de banda encontrada será representada por β_{min} , a maior largura de banda encontrada será representada por β_{max} e a largura de banda média encontrada será representada por $\bar{\beta}$. Analogamente, o menor *profile* encontrado será representada por $profile_{min}$, o maior *profile* encontrado será representada por $profile_{max}$ e o *profile* médio encontrado será representada por $\overline{profile}$. No caso da heurística CMr, iniciando por um vértice pseudo-periférico ou por um vértice qualquer, o resultado encontrado é sempre o mesmo, pois o vértice inicial

não muda nas execuções com uma mesma instância. Portanto, a largura de banda é representada nas tabelas por β e o *profile* é representado por *profile*. O tempo de execução é apresentado em segundos.

A melhor heurística, em relação às reduções de largura de banda e de *profile*, nesse conjunto de matrizes foi a heurística *VNS-band*. Apesar de não ter executado tão rápido quanto as heurísticas *CMr*, *CMr-pseudo* ou *MAO-band*, a heurística *VNS-band* reduziu mais a largura de banda e o *profile* nessas instâncias da base *Harwell-Boeing*. Em relação ao custo computacional, a heurística *CMr* foi a melhor. As heurísticas *CMr* e *CMr-pseudo* executaram de maneira bastante rápida. Nota-se que estes experimentos são preliminares porque a intenção de se realizar as reduções de largura de banda e de *profile* é obter redução do custo computacional na resolução de sistemas de equações lineares.

A seguir, na subseção 4.1.1, são apresentados os resultados e as análises das simulações com as 33 instâncias bastante pequenas. Os resultados e as análises das simulações com as 80 instâncias pequenas são apresentados na subseção 4.1.2.

4.1.1 Resultados e análises das 33 instâncias bastante pequenas

Nas Tabelas 1, 2, 3 e 4, são apresentados os resultados das execuções das heurísticas *CMr*, *CMr-pseudo*, *VNS-band* e *MAO-band*, respectivamente, com as 33 instâncias bastante pequenas da base *Harwell-Boeing*.

Tabela 1 Simulações com a heurística CMr para as instâncias bastante pequenas.

Instância	Dim.	β	<i>profile</i>	Tempo (s)
arc130	130	121	4812	0,00005996
ash85	85	10	589	0,00003017
bcpwr01	39	8	161	0,00003751
bcpwr02	49	13	240	0,00007858
bcpwr03	118	21	805	0,00004184
bcsstk01	48	26	672	0,00023370
bcsstk04	132	65	4278	0,00013490
bcsstk05	153	24	2254	0,00007824
bcsstk22	138	14	873	0,00011299
can_144	144	18	1074	0,00008968
can_161	161	30	3085	0,00004933
curtis54	54	14	360	0,00007208
dwt_234	234	21	856	0,00016224
fs_183_1	183	155	7357	0,00001778
gent113	113	27	1406	0,00006491
gre_115	115	36	2387	0,00009222
gre_185	185	29	3898	0,00003787
ibm32	32	15	265	0,00005367
impcol_b	59	34	972	0,00009279
impcol_c	137	47	2813	0,00007885
lns_131	131	39	1447	0,00012903
lund_a	147	23	2303	0,00015350
lund_b	147	23	2299	0,00009696
mcca	180	85	6559	0,00004147
nos1	237	3	78	0,00002856
nos4	100	16	918	0,00007117
pores_1	30	9	179	0,00003264
steam3	80	7	424	0,00005881
west0132	132	61	3521	0,00011363
west0156	156	72	5613	0,00008969
west0167	167	93	7394	0,00011741
will199	199	115	11386	0,00016854
will57	57	14	279	0,00004452

Tabela 2 Simulações com a heurística CMr-pseudo para as instâncias bastante pequenas.

Instância	Dim.	β	<i>profile</i>	Tempo (s)
arc130	130	119	6898	0,000700176
ash85	85	10	589	0,000284660
bccpwr01	39	8	131	0,000166511
bccpwr02	49	12	222	0,000201434
bccpwr03	118	20	866	0,000391971
bccstk01	48	26	672	0,000233132
bccstk04	132	54	3717	0,001531624
bccstk05	153	24	2290	0,000875630
bccstk22	138	14	838	0,000481941
can_144	144	18	1076	0,000618853
can_161	161	34	3651	0,000606810
curtis54	54	14	360	0,000217049
dwt_234	234	25	871	0,000346127
fs_183_1	183	155	7903	0,000789483
gent113	113	27	1406	0,000291095
gre_115	115	36	2385	0,000373544
gre_185	185	27	3697	0,000506343
ibm32	32	16	264	0,000163571
impcol_b	59	38	1071	0,000426192
impcol_c	137	46	3014	0,000579822
lms_131	131	32	1226	0,000458155
lund_a	147	23	2303	0,000966124
lund_b	147	23	2303	0,001005767
mcca	180	3	6	0,000572762
nos1	237	3	78	0,000253914
nos4	100	22	1012	0,000317033
pores_1	30	9	179	0,000149436
steam3	80	7	424	0,000311363
west0132	132	59	3262	0,000589935
west0156	156	74	5741	0,000719277
west0167	167	66	5414	0,000696500
will199	199	107	11038	0,000756329
will57	57	14	295	0,000226325

Tabela 3 Simulações com a heurística VNS-*band* para as instâncias bastante pequenas.

Instância	Dim.	β_{min}	β_{max}	$\bar{\beta}$	$profile_{min}$	$profile_{max}$	$profile$	Tempo (s)
arc130	130	63	63	63	2607	2607	2607	2,021671
ash85	85	14	14	14	542	542	542	2,147214
bcsplr01	39	7	7	7	94	94	94	2,153311
bcsplr02	49	9	9	9	170	170	170	2,232932
bcsplr03	118	10	10	10	304	304	304	2,024756
bcsstk01	48	24	24	24	592	592	592	2,340587
bcsstk04	132	23	23	23	1024	1024	1024	2,024362
bcsstk05	153	34	34	34	1787	1787	1787	2,025727
bcsstk22	138	52	52	52	1375	1375	1375	2,013647
can_144	144	13	13	13	1043	1043	1043	2,016136
can_161	161	18	18	18	1791	1791	1791	2,016942
curtis54	54	16	16	16	514	514	514	2,218309
dwt_234	234	28	28	28	712	712	712	2,024691
fs_183_1	183	81	81	81	6534	6534	6534	2,019108
gent113	113	27	27	27	1406	1406	1406	2,020292
gre_115	115	23	23	23	1502	1502	1502	2,021816
gre_185	185	21	21	21	2822	2822	2822	2,024613
ibm32	32	17	17	17	240	240	240	2,157805
impcolb	59	25	25	25	694	694	694	2,445159
impcolc	137	30	30	30	1684	1684	1684	2,021948
lms_131	131	20	20	20	1149	1149	1149	2,020778
lund_a	147	16	16	16	921	921	921	2,022756
lund_b	147	23	23	23	2532	2532	2532	2,033406
mcca	180	85	85	85	6559	6559	6559	2,022809
nos1	237	5	5	5	11	11	11	2,025822
nos4	100	10	10	10	639	639	639	2,027258
pores_1	30	7	7	7	149	149	149	2,376064
steam3	80	11	11	11	600	600	600	2,372114
west0132	132	34	34	34	2214	2214	2214	2,022329
west0156	156	57	57	57	4122	4122	4122	2,021465
west0167	167	48	48	48	4006	4006	4006	2,020496
will199	199	65	65	65	7972	7972	7972	2,025605
will57	57	9	9	9	265	265	265	2,182894

Tabela 4 Simulações com a heurística MAO-*band* para as instâncias bastante pequenas.

Instância	Dim.	β_{min}	β_{max}	$\bar{\beta}$	$profile_{min}$	$profile_{max}$	$profile$	Tempo (s)
arc130	130	107	115	107	6806	7714	6934	0,593
ash85	85	53	69	59	1465	1911	1642	0,073
bcsplr01	39	24	30	25	283	395	305	0,008
bcsplr02	49	27	38	29	419	461	441	0,015
bcsplr03	118	87	101	89	3179	3474	3267	0,105
bcsstk01	48	37	40	38	845	876	859	0,030
bcsstk04	132	111	115	112	7764	8037	7860	0,777
bcsstk05	153	133	135	133	9673	10268	9841	0,586
bcsstk22	138	108	112	108	5304	5781	5590	0,161
can_144	144	117	124	118	7983	8531	8299	0,328
can_161	161	131	134	131	9482	10261	9546	0,384
curtis54	54	37	45	37	805	1033	814	0,044
dwt_234	234	186	207	190	13095	13920	13111	0,435
fs_183_1	183	152	163	153	9582	12714	10106	0,688
gent113	113	89	97	92	4181	4332	4221	0,222
gre_115	115	89	96	89	4339	4753	4371	0,147
gre_185	185	152	159	154	12483	12830	12525	0,633
ibm32	32	22	24	22	341	370	356	0,011
impcolb	59	44	49	44	1184	1299	1204	0,054
impcolc	137	107	117	107	5546	5958	5767	0,182
lms_131	131	106	109	107	4474	4884	4546	0,229
lund_a	147	125	126	125	9243	9507	9354	0,542
lund_b	147	120	127	124	9068	9508	9144	0,601
mcca	180	143	159	146	11732	12710	12026	1,570
nos1	237	196	198	196	15791	15950	15825	0,505
nos4	100	75	85	76	2931	3466	2966	0,096
pores_1	30	21	23	21	333	349	345	0,014
steam3	80	63	71	64	2452	2600	2474	0,226
west0132	132	105	110	106	6120	6529	6204	0,166
west0156	156	124	136	125	7652	8790	7917	0,180
west0167	167	139	144	139	9660	10467	9772	0,278
will199	199	167	174	167	15088	15517	15184	0,482
will57	57	41	48	42	721	868	756	0,048

As execuções das heurísticas CMr e CMr-pseudo foram bastante rápidas para todas as 33 instâncias bastante pequenas. Em média, a heurística CMr executou em um tempo 6 vezes menor do que a heurística CMr-pseudo. A heurística VNS-*band* foi a heurística mais demorada para essas instâncias. Se comparados os tempos de execução totais das heurísticas VNS-*band* e MAO-*band*, a heurística MAO-*band* executou, em média, 6,64 vezes mais rápido.

Na subseção 4.1.1.1, são apresentadas algumas análises dos resultados em relação à redução de largura de banda. Na subseção 4.1.1.2, são apresentadas algumas análises dos resultados em relação à redução de *profile*.

4.1.1.1 Redução de largura de banda

No geral, a heurística VNS-*band* obteve resultados melhores para as reduções de largura de banda e de *profile* se comparado às heurísticas CMr, CMr-pseudo e MAO-*band*. Para redução de largura de banda, a heurística VNS-*band* obteve resultados melhores em 22 instâncias e empatou em 3 instâncias com a heurística CMr. A heurística CMr foi melhor que a heurística VNS-*band* em 8 instâncias.

Se comparado à heurística VNS-*band*, a heurística CMr-pseudo obteve mais redução de largura de banda em 9 instâncias. A heurística VNS-*band* foi melhor que a heurística CMr-pseudo em 22 instâncias e houve empate em 2 instâncias, nas reduções de largura de banda.

A heurística CMr-pseudo obteve maior redução de largura de banda em 11 instâncias e empatou em 16 instâncias, se comparado com a heurística CMr. A heurística CMr reduziu mais a largura de banda em 6 instâncias,

se comparado com a heurística CMr-pseudo.

A heurística MAO-*band* obteve melhores reduções de largura de banda em apenas 2 instâncias se comparado com a heurística CMr, que obteve melhores resultados nas outras 31 instâncias. A heurística MAO-*band* foi melhor do que a heurística CMr-pseudo em apenas 2 instâncias, em termos de redução de largura de banda. A heurística VNS-*band* foi melhor do que a heurística MAO-*band* em todas as 33 instâncias bastante pequenas.

Em média, a heurística VNS-*band* reduziu 28,2% mais a largura de banda se comparado à heurística CMr, 20,16% se comparado à heurística CMr-pseudo e 71,7% se comparado à heurística MAO-*band*. A heurística CMr reduziu 60,7% mais largura de banda se comparado à heurística MAO-*band*. No geral, a heurística CMr-pseudo reduziu 9,5% mais largura de banda do que a heurística CMr. Em média, as reduções de largura de banda obtidas pela heurística CMr-pseudo foram 64,6% melhores do que as reduções obtidas pela heurística MAO-*band*.

4.1.1.2 Redução de *profile*

A heurística VNS-*band* conseguiu maior redução de *profile* em 29 instâncias se comparado à heurística CMr. Este obteve maior redução que a heurística VNS-*band*, em apenas 2 instâncias e houve empate das duas heurísticas em 2 instâncias. Em relação à redução de *profile*, a heurística CMr-pseudo obteve resultados melhores do que a heurística VNS-*band* em 5 instâncias e empatou em 1 instância. A heurística VNS-*band* reduziu mais o *profile* em 27 instâncias, se comparado com a heurística CMr-pseudo.

A heurística CMr-pseudo obteve maiores reduções de *profile* em 12 instâncias e empatou com a heurística CMr em 8 instâncias. A heurística CMr foi melhor do que a heurística CMr-pseudo em 13 instâncias, em termos de redução de *profile*. Todas as heurísticas reduziram mais *profile* do que a heurística MAO-band em todas as instâncias bastante pequenas.

A heurística VNS-band reduziu, em média, 28,1% mais *profile* se comparado à heurística CMr, 22,1% se comparado à heurística CMr-pseudo e 71,2% se comparado à heurística MAO-band. A heurística CMr reduziu, em média, 60% mais *profile* se comparado à heurística MAO-band. No geral, a heurística CMr-pseudo reduziu 7,8% mais *profile* do que a heurística CMr. Em média, a redução de *profile* obtida pela heurística CMr-pseudo foi 63% melhor do que a obtida pela heurística MAO-band.

4.1.2 Resultados e análises das 80 instâncias pequenas

Nas Tabelas 5, 6, 7 e 8, são apresentados os resultados das execução das heurísticas CMr, CMr-pseudo, VNS-band e MAO-band, respectivamente, com as 80 instâncias pequenas da base *Harwell-Boeing*. As heurísticas CMr e CMr-pseudo executaram em menor tempo para todas essas instâncias, se comparadas com as heurísticas VNS-band e MAO-band. Em média, a heurística CMr executou em tempo 5,2 vezes menor do que a heurística CMr-pseudo. A heurística VNS-band foi a heurística mais demorada para essas instâncias. Se comparados os tempos de execução totais das heurísticas VNS-band e MAO-band, a heurística MAO-band executou, em média, 2 vezes mais rápido.

Tabela 5 Simulações com a heurística CMr para as instâncias pequenas.

Instância	Dim.	β	<i>profile</i>	Tempo (s)
494_bus	494	85	16107	0,0002216
662_bus	662	112	29950	0,0003203
685_bus	685	93	24192	0,0003738
ash292	292	40	6451	0,0001925
bcpwr04	274	55	4224	0,0001849
bcpwr05	443	83	13377	0,0002461
bcsstk06	420	50	13238	0,0004548
bcsstk19	817	22	9594	0,0005239
bcsstk20	485	17	4725	0,0002377
bcsstm07	420	58	14151	0,0004110
bp_0	822	406	170541	0,0007090
bp_200	822	441	188910	0,0008221
bp_400	822	530	200178	0,0008941
bp_600	822	541	206356	0,0008987
bp_800	822	525	210650	0,0009944
bp_1000	822	532	214324	0,0010590
bp_1200	822	547	217056	0,0010710
bp_1400	822	536	217101	0,0010690
bp_1600	822	533	217735	0,0010960
can_292	292	60	9573	0,0001795
can_445	445	92	23808	0,0002653
can_715	715	145	48632	0,0004203
can_838	838	184	40402	0,0008816
dwt_209	209	42	4063	0,0001340
dwt_221	221	27	3334	0,0001261
dwt_245	245	40	3929	0,0001440
dwt_310	310	24	4408	0,0001494
dwt_361	361	25	5139	0,0001578
dwt_419	419	55	12666	0,0002771
dwt_503	503	64	14816	0,0003704
dwt_592	592	42	10983	0,0002849
dwt_878	878	37	21163	0,0004367
dwt_918	918	72	24088	0,0004973
dwt_992	992	63	36296	0,0017700
fs_541_1	541	535	123121	0,0005737
fs_680_1	680	24	5128	0,0003239
fs_760_1	760	62	27899	0,0005719
gr_30_30	900	59	33872	0,0004065
gre_343	343	28	7186	0,0001628
gre_512	512	36	13820	0,0002348

Tabela 5 Conclusão.

Instância	Dim.	β	<i>profile</i>	Tempo (s)
gre_216a	216	21	3383	0,0001109
hor_131	434	101	22686	0,0002895
impcoLa	207	69	6911	0,0001662
impcoLd	425	70	13648	0,0002413
impcoLe	225	78	7562	0,0001743
jagmesh1	936	27	21817	0,0003332
jpwh_991	991	177	82508	0,0005904
lnsp_511	511	59	12529	0,0002916
mbeacxc	496	480	112372	0,0286300
mbeaflw	496	480	112372	0,0273800
mbeause	496	472	107456	0,0160200
mcfе	765	16	136	0,0010393
nnc261	261	38	5070	0,0001732
nnc666	666	63	21910	0,0004142
nos2	957	3	1908	0,0001133
nos3	960	79	46171	0,0015790
nos5	468	93	26094	0,0003441
nos6	675	16	9305	0,0001970
nos7	729	65	34110	0,0002998
orsirr_2	886	160	73678	0,0003875
plat362	362	52	13366	0,0003380
plskz362	362	28	5374	0,0001525
pores_3	532	13	5319	0,0000835
saylr1	238	14	2499	0,0000788
saylr3	1000	81	33979	0,0000503
sherman1	1000	56	28970	0,0000548
sherman4	1104	36	16885	0,0000638
shl_0	663	422	126626	0,0004442
shl_200	663	437	125118	0,0004889
shl_400	663	432	132136	0,0004814
steam1	240	50	7395	0,0002331
steam2	600	63	30852	0,0011490
str_0	363	183	35286	0,0003560
str_200	363	224	36030	0,0004364
str_600	363	224	39113	0,0005010
west0381	381	264	49724	0,0003525
west0479	479	236	48635	0,0003678
west0497	497	256	39916	0,0003530
west0655	655	342	102151	0,0005457
west0989	989	475	203984	0,0007728

Tabela 6 Simulações com a heurística CMr-pseudo para as instâncias pequenas.

Instância	Dim.	β	<i>profile</i>	Tempo (s)
494_bus	494	68	12442	0,015911160
662_bus	662	100	29000	0,001965659
685_bus	685	66	19678	0,002426655
ash292	292	34	4659	0,001266698
bcsprw04	274	57	4418	0,001134755
bcsprw05	443	77	13018	0,001268915
bcsstk06	420	49	13116	0,002820024
bcsstk19	817	32	16347	0,003671967
bcsstk20	485	20	5036	0,001229999
bcsstm07	420	50	13310	0,002527275
bp_0	822	470	174018	0,005489178
bp_200	822	512	177399	0,005809424
bp_400	822	499	196398	0,006853037
bp_600	822	517	195803	0,007150194
bp_800	822	473	200110	0,007669320
bp_1000	822	526	215552	0,007592031
bp_1200	822	545	221021	0,007560934
bp_1400	822	545	223213	0,007679831
bp_1600	822	550	222402	0,007677593
can_292	292	65	9706	0,001243380
can_445	445	98	23651	0,001810912
can_715	715	148	50407	0,003015681
can_838	838	163	41395	0,004452894
dwt_209	209	43	4605	0,000921899
dwt_221	221	27	2479	0,000940857
dwt_245	245	57	5626	0,000915688
dwt_310	310	13	2695	0,000953513
dwt_361	361	15	4714	0,001064964
dwt_419	419	73	15441	0,001943841
dwt_503	503	64	14816	0,002462600
dwt_592	592	42	10952	0,001913828
dwt_878	878	63	35024	0,003901760
dwt_918	918	72	24214	0,004722281
dwt_992	992	63	36296	0,006201804
fs_541_1	541	533	124225	0,002881185
fs_680_1	680	23	4059	0,001620790
fs_760_1	760	63	23004	0,004050759
gr_30_30	900	59	33872	0,003535679
gre_343	343	28	7186	0,000874914
gre_512	512	36	13820	0,001353702

Tabela 6 Conclusão.

Instância	Dim.	β	<i>profile</i>	Tempo (s)
gre_216a	216	21	3383	0,000555402
hor_131	434	101	23394	0,003167495
impcol_a	207	59	6033	0,000824221
impcol_d	425	70	13686	0,001595097
impcol_e	225	70	6454	0,001134907
jagmesh1	936	27	21817	0,003899640
jpwh_991	991	173	83954	0,003664203
linsp_511	511	59	12537	0,001531486
mbeacxc	496	418	90587	0,117503400
mbeaflw	496	418	90587	0,114604400
mbeause	496	445	89130	0,080851780
mcfе	765	16	136	0,005855626
nnc261	261	38	5121	0,000915924
nnc666	666	66	22058	0,002870332
nos2	957	3	320	0,001197334
nos3	960	79	46168	0,006488021
nos5	468	94	26186	0,002696224
nos6	675	30	13799	0,001352586
nos7	729	65	34110	0,002117143
orsirr_2	886	160	73678	0,002695676
plat362	362	39	9520	0,002257130
plskz362	362	226	18240	0,000647885
pores_3	532	13	5319	0,000758541
saylr1	238	15	2538	0,000404804
saylr3	1000	62	26751	0,002200409
sherman1	1000	62	26751	0,002182426
sherman4	1104	36	16885	0,000721162
shl_0	663	423	131969	0,002511814
shl_200	663	445	129665	0,002842154
shl_400	663	435	130515	0,002797392
steam1	240	50	7395	0,001546063
steam2	600	63	30852	0,004242573
str_0	363	221	35345	0,002721867
str_200	363	228	40413	0,003113790
str_600	363	234	39482	0,003799179
west0381	381	268	48362	0,002277668
west0479	479	254	54035	0,002519756
west0497	497	258	40351	0,002031894
west0655	655	324	102748	0,003887034
west0989	989	476	204468	0,005692289

Tabela 7 Simulações com a heurística VNS-*band* para as instâncias pequenas.

Instância	Dim.	β_{min}	β_{max}	$\bar{\beta}$	$profile_{min}$	$profile_{max}$	$profile$	Tempo (s)
494_bus	494	30	35	34	4654	5613	5421	20,05741
662_bus	662	51	51	51	12526	12526	12526	20,06107
685_bus	685	39	39	39	10335	10335	10335	20,04037
ash292	292	19	19	19	2159	2159	2159	20,05758
bcpwr04	274	24	25	24	2405	2710	2527	20,06426
bcpwr05	443	29	29	29	4580	4580	4580	20,06072
bcsstk06	420	45	45	45	14078	14078	14078	20,06570
bcsstk19	817	14	14	14	5768	5768	5768	20,10831
bcsstk20	485	13	13	13	1314	1314	1314	20,07043
bcsstm07	420	45	45	45	13842	13842	13842	20,05539
bp_0	822	254	254	254	103522	103522	103522	20,07008
bp_200	822	273	273	273	117117	117117	117117	20,28890
bp_400	822	292	292	292	127701	127701	127701	20,06734
bp_600	822	342	342	342	146178	146178	146178	20,05442
bp_800	822	290	290	290	132325	132325	132325	20,06576
bp_1000	822	308	308	308	131267	131267	131267	20,07419
bp_1200	822	296	296	296	122734	122734	122734	20,08248
bp_1400	822	320	320	320	140033	140033	140033	20,20999
bp_1600	822	362	362	362	151668	151668	151668	20,07004
can_292	292	39	39	39	7911	7911	7911	20,05900
can_445	445	58	58	58	16523	16523	16523	20,06586
can_715	715	108	108	108	29481	29481	29481	20,07167
can_838	838	91	91	91	41679	41679	41679	20,08732
dwt_209	209	23	23	23	2126	2126	2126	20,05763
dwt_221	221	13	13	13	1833	1833	1833	20,04896
dwt_245	245	21	21	21	1527	1527	1527	20,07825
dwt_310	310	10	10	10	203	203	203	20,07183
dwt_361	361	16	16	16	4670	4670	4670	20,05744
dwt_419	419	28	28	28	5743	5743	5743	20,05848
dwt_503	503	48	48	48	14445	14445	14445	20,04280
dwt_592	592	29	29	29	8995	8995	8995	20,04082
dwt_878	878	27	27	27	2471	2471	2471	20,07279
dwt_918	918	37	37	37	11316	11316	11316	20,06850
dwt_992	992	73	73	73	60682	60682	60682	20,07939
fs_541_1	541	270	270	270	82803	82803	82803	20,08319
fs_680_1	680	18	18	18	10126	10126	10126	20,06852
fs_760_1	760	38	38	38	23996	23996	23996	20,06188
gr_30_30	900	59	59	59	36319	36319	36319	20,06512
gre_343	343	28	28	28	6850	6850	6850	20,05589
gre_512	512	38	38	38	14058	14058	14058	20,06809

Tabela 7 Conclusão.

Instância	Dim.	β_{min}	β_{max}	β	$profile_{min}$	$profile_{max}$	$profile$	Tempo (s)
gre_216a	216	21	21	21	3223	3223	3223	20,05474
hor_131	434	55	55	55	18909	18909	18909	20,05672
impcol_a	207	32	32	32	3264	3264	3264	20,03790
impcol_d	425	40	40	40	10491	10491	10491	20,04532
impcol_e	225	42	42	42	4133	4133	4133	20,08011
jagmesh1	936	31	31	31	11451	11451	11451	20,05716
jpwh_991	991	226	226	226	122646	122646	122646	20,08159
lnsp_511	511	45	45	45	12808	12808	12808	20,06099
mbeacxc	496	437	437	437	85845	85845	85845	20,12536
mbeaffw	496	437	437	437	85845	85845	85845	20,10534
mbeause	496	417	417	417	80434	80434	80434	20,09914
mcfе	765	385	385	385	150623	150623	150623	20,05816
nnc261	261	24	24	24	4233	4233	4233	20,05256
nnc666	666	56	56	56	26099	26099	26099	20,06738
nos2	957	3	3	3	1908	1908	1908	146,3017
nos3	960	79	79	79	46502	46502	46502	20,06550
nos5	468	64	64	64	23567	23567	23567	20,67920
nos6	675	20	20	20	8956	8956	8956	20,05350
nos7	729	78	78	78	14789	14789	14789	20,04841
orsirr_2	886	95	95	95	67152	67152	67152	20,08723
plat362	362	36	36	36	5510	5510	5510	20,07728
plskz362	362	239	239	239	27491	27491	27491	20,07023
pores_3	532	13	13	13	5319	5319	5319	20,05096
saylr1	238	14	14	14	2977	2977	2977	20,05687
saylr3	1000	56	56	56	28970	28970	28970	20,07593
sherman1	1000	56	56	56	28970	28970	28970	20,05967
sherman4	1104	36	36	36	16885	16885	16885	25,02572
shl_0	663	231	231	231	56453	56453	56453	20,06478
shl_200	663	240	240	240	56999	56999	56999	20,03162
shl_400	663	237	237	237	57962	57962	57962	20,03568
steam1	240	47	47	47	8904	8904	8904	20,07940
steam2	600	63	63	63	31884	31884	31884	20,06241
str_0	363	120	120	120	20957	20957	20957	20,05703
str_200	363	127	127	127	25582	25582	25582	20,05792
str_600	363	133	133	133	25466	25466	25466	20,06190
west0381	381	153	153	153	35645	35645	35645	20,03501
west0479	479	124	124	124	30177	30177	30177	20,06934
west0497	497	86	86	86	22312	22312	22312	20,05587
west0655	655	253	253	253	82617	82617	82617	20,04390
west0989	989	335	335	335	172238	172238	172238	20,03900

Tabela 8 Simulações com a heurística MAO-*band* para as instâncias pequenas.

Instância	Dim.	β_{min}	β_{max}	$\bar{\beta}$	$profile_{min}$	$profile_{max}$	$profile$	Tempo (s)
494_bus	494	406	434	415	56460	61624	57899	4,23
662_bus	662	553	579	568	114491	115393	115302	10,10
685_bus	685	585	607	587	136583	143048	137229	12,50
ash292	292	245	245	245	30269	30269	30269	2,35
bcpwr04	274	234	242	234	23304	23449	23405	1,86
bcpwr05	443	361	369	364	50212	51561	50603	4,07
bcsstk06	420	372	372	372	73219	73552	73252	13,00
bcsstk19	817	718	724	718	244985	251012	246118	28,00
bcsstk20	485	412	428	415	75927	81131	77889	6,91
bcsstm07	420	368	372	370	72943	75216	73620	12,50
bp_0	822	708	734	715	243865	253150	244793	25,40
bp_200	822	725	734	727	254985	263961	257677	30,90
bp_400	822	722	735	725	258853	261434	259111	31,30
bp_600	822	721	722	721	258426	275762	260982	32,80
bp_800	822	711	732	715	266339	283989	272423	35,60
bp_1000	822	712	727	716	265417	277897	266665	36,40
bp_1200	822	720	729	723	268313	272304	270308	36,90
bp_1400	822	720	730	723	273355	281906	276805	34,10
bp_1600	822	721	729	722	271571	278094	274349	37,80
can_292	292	253	257	253	31522	31522	31522	2,96
can_445	445	385	396	386	74766	76722	74961	7,91
can_715	715	626	627	626	192899	202552	196219	24,10
can_838	838	738	748	739	287338	293679	288783	37,80
dwt_209	209	175	180	176	16368	16591	16390	1,41
dwt_221	221	189	196	193	17053	18000	17235	1,27
dwt_245	245	207	208	207	18878	19737	18963	1,56
dwt_310	310	262	262	262	34322	36542	34800	3,07
dwt_361	361	303	317	308	48218	48600	48396	4,15
dwt_419	419	354	364	358	65390	68086	66402	6,46
dwt_503	503	438	441	438	101074	101403	101238	13,50
dwt_592	592	513	517	516	131105	132578	131671	14,40
dwt_878	878	765	781	770	290660	292215	291292	35,20
dwt_918	918	798	811	799	312033	318123	312642	38,10
dwt_992	992	875	882	878	427840	433546	430145	89,30
fs_541_1	541	471	483	474	110566	114507	111553	20,00
fs_680_1	680	592	606	594	141849	145013	143747	16,70
fs_760_1	760	668	675	669	231571	234797	232219	44,30
gr_30_30	900	791	804	792	303349	308550	304703	40,00
gre_343	343	288	300	292	41735	43185	42167	3,53
gre_512	512	446	454	448	94134	96623	94382	8,59

Tabela 8 Conclusão.

Instância	Dim.	β_{min}	β_{max}	$\bar{\beta}$	$profile_{min}$	$profile_{max}$	$profile$	Tempo (s)
gre_216a	216	176	176	176	16061	16061	16061	1,21
hor_131	434	374	375	374	73660	75377	74228	16,30
impcoLa	207	175	177	175	14388	14821	14547	0,86
impcoLd	425	363	366	364	61230	63353	61601	4,53
impcoLe	225	188	198	194	19632	19698	19665	2,10
jagmesh1	936	803	828	810	304899	315938	308106	29,90
jpwh_991	991	869	881	872	332098	346453	334206	56,60
lnsp_511	511	442	442	442	80405	81970	80689	12,00
mbeacxc	496	441	444	441	117204	117204	117204	206,00
mbeaflw	496	443	444	443	117462	118981	117932	194,00
mbeause	496	441	445	441	118382	121801	118723	160,00
mcfe	765	673	686	674	264899	264899	264899	167,00
nnc261	261	223	225	224	22691	22746	22729	3,24
nnc666	666	582	584	583	149606	157308	150376	22,30
nos2	957	823	823	823	268394	268865	268817	22,00
nos3	960	837	856	843	391071	394046	392558	82,20
nos5	468	411	416	412	86300	88769	86862	9,10
nos6	675	568	598	574	138814	142724	139748	12,80
nos7	729	631	635	631	184603	190029	187567	16,00
orsirr_2	886	756	787	765	273696	277702	274096	48,50
plat362	362	315	321	316	55681	58288	56541	8,47
plskz362	362	309	309	309	43361	45864	43904	2,42
pores_3	532	452	452	452	100168	102349	100618	14,80
saylr1	238	202	207	202	17074	17562	17178	1,81
saylr3	1000	869	870	869	213129	222639	215214	38,40
sherman1	1000	877	880	877	211879	216197	213523	38,60
sherman4	1104	936	949	945	198926	209078	200956	48,10
shl_0	663	568	592	572	135349	189368	146468	9,71
shl_200	663	569	590	578	139972	143038	140585	9,95
shl_400	663	568	593	576	143320	187801	149969	9,91
steam1	240	206	206	206	24663	25131	24731	6,78
steam2	600	529	538	530	161680	163992	162323	67,90
str_0	363	315	319	316	48960	48960	48960	6,29
str_200	363	317	320	317	52166	53964	52525	8,73
str_600	363	312	319	315	54450	54554	54543	8,80
west0381	381	333	339	334	56986	57845	57626	6,36
west0479	479	416	428	418	84049	84559	84202	7,04
west0497	497	418	441	429	84586	91906	87586	7,10
west0655	655	567	567	567	159756	163496	160130	14,50
west0989	989	855	882	861	347936	347936	347936	34,60

Na subseção 4.1.2.1, são apresentadas algumas análises dos resultados em relação à redução de largura de banda. Na subseção 4.1.2.2, são apresentadas algumas análises dos resultados em relação à redução de *profile*.

4.1.2.1 Redução de largura de banda

Novamente, a heurística *VNS-band* obteve resultados melhores do que as heurísticas *CMr*, *CMr-pseudo* e *MAO-band*. A heurística *CMr* foi melhor que a heurística *VNS-band* em apenas 8 instâncias em termos de redução de largura de banda. A heurística *VNS-band* obteve maior redução da largura de banda em 15 instâncias e empatou com a heurística *CMr* em 10 instâncias.

A heurística *VNS-band* foi melhor que a heurística *CMr-pseudo* em 61 instâncias, nas reduções de largura de banda. A heurística *CMr-pseudo* obteve mais redução de largura de banda em 11 instâncias e empatou em 8 instâncias, se comparado à heurística *VNS-band*.

A heurística *CMr-pseudo* obteve mais redução de largura de banda em 26 instâncias e empatou em 23 instâncias, se comparado à heurística *CMr*. A heurística *CMr* foi melhor que a heurística *CMr-pseudo* em 31 instâncias.

A heurística *MAO-band* reduziu mais largura de banda, se comparado à heurística *CMr*, em 5 instâncias. A heurística *MAO-band* reduziu mais a largura de banda do que a heurística *CMr-pseudo* em apenas 1 instância e houve empate em 1 instância. A heurística *VNS-band* reduziu mais largura de banda do que a heurística *MAO-band* em todas as instâncias pequenas.

Em média, a heurística *VNS-band* reduziu 30,4% mais largura de banda se comparado à heurística *CMr*, 30,9% se comparado à heurística *CMr-pseudo* e 77,4% se comparado à heurística *MAO-band*. A heurística *CMr* reduziu 0,8% mais largura de banda do que a heurística *CMr-pseudo*, de uma forma geral. Os resultados médios para a redução de largura de banda obtidos pela heurística *CMr-pseudo* foram 67,3% melhores do que os obtidos pela heurística *MAO-band*.

4.1.2.2 Redução de *profile*

A heurística *VNS-band* reduziu mais *profile* do que a heurística *CMr* em 63 instâncias. A heurística *CMr* obteve menor *profile* em 13 instâncias e empatou com a heurística *VNS-band* em 4 instâncias. Em relação à redução de *profile*, a heurística *CMr-pseudo* obteve melhores resultados do que a heurística *CMr* em 30 instância e empatou em 14 instâncias. A heurística *CMr* obteve maior redução de *profile* em 36 instâncias, se comparado com a heurística *CMr-pseudo*.

A heurística *MAO-band* reduziu mais *profile* do que a heurística *CMr-pseudo* em apenas 1 instância. Os valores de *profile*, obtidos pela heurística *MAO-band*, foram maiores do que os obtidos pela heurística *VNS-band* e pela heurística *CMr* em todas as instâncias pequenas.

Em média, a heurística *VNS-band* reduziu 25,9% mais *profile* se comparado à heurística *CMr*, 25% se comparado à heurística *CMr-Pseudo* e 72,9% se comparado à heurística *MAO-band*. A heurística *CMr-pseudo* reduziu 1,13% mais *profile* do que a heurística *CMr*, de uma forma geral. Os resultados médios para a redução de *profile* obtidos pela heurística *CMr-pseudo* foram 63,9% melhores do que os obtidos pela heurística *MAO-band*.

4.2 Simulações com SELs oriundos da discretização da condução do calor

As simulações, a seguir, foram aplicadas em SELs resultantes da discretização da condução do calor por volumes finitos, com malhas geradas pelo refinamento de *Delaunay* e, conseqüentemente, diagramas de *Voronoi* são utilizados. Esses SELs foram gerados pelo projeto computacional de Oliveira (2014). A equação de condução do calor é uma equação diferencial parcial parabólica e pode ser definida, de forma generalizada, por

$$\frac{\partial}{\partial t}(\rho\phi) = \nabla \cdot \left(\frac{k}{c_p} \nabla \phi \right),$$

em que ϕ é a temperatura, ρ é a massa volumétrica de um corpo sólido, c_p é o seu calor específico, k é a condutibilidade térmica e t é o tempo. As condições de contorno utilizadas foram as condições de contorno de Dirichlet, conforme descrito em Oliveira (2014). 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.

O parâmetro utilizado para a heurística MAO-*band* e para a heurística VNS-*band* foi o tempo máximo de execução de 1 segundo. Utilizou-se a taxa de aprendizado $\eta = 0,9$ para a heurística MAO-*band*.

Na subseção 4.2.1, apresentam-se as simulações com resoluções dos SELs pelo método dos gradientes conjugados sem condicionamento. Na subseção 4.2.2, apresentam-se as simulações com resoluções dos SELs pelo método dos gradientes conjugados condicionado por Jacobi. Em ambas subseções, são apresentadas algumas análises sobre os resultados obtidos. As ocupações de memória do projeto computacional desenvolvido e das heurís-

ticas de reduções de largura de banda e de *profile* utilizadas são apresentadas e analisadas na subseção 4.2.3.

4.2.1 SELs resolvidos pelo MGC sem condicionamento

Para as simulações mostradas nesta subseção, os SELs foram resolvidos pelo método dos gradientes conjugados (MGC) sem condicionamento. Estas simulações são preliminares, uma vez que o método dos gradientes conjugados raramente é utilizado sem um condicionador. Por outro lado, estas simulações servem para que as ocupações de memória das heurísticas para reduções de largura de banda e de *profile* possam ser estudadas.

Foram executados experimentos sem redução de largura de banda e com as heurísticas CMr, VNS-*band* e MAO-*band* dez vezes para cada instância. A variância e o desvio padrão para o tempo de execução foram menores do que 0,0088 e 0,0941, respectivamente.

Na Tabela 9, apresentam-se os resultados da resolução dos SELs sem reduções de largura de banda e de *profile*. Nas Tabelas 10, 11 e 12, apresentam-se os resultados da resolução dos SELs com reduções de largura de banda e de *profile* realizado pelas heurísticas CMr, VNS-*band* e MAO-*band*, respectivamente. A coluna **Iterações** apresenta a quantidade de iterações do MGC. Nas últimas colunas dessas tabelas, apresentam-se as reduções dos tempos totais de execução da heurística correspondente, mais os tempos de execução do MGC, em relação ao tempo de execução do MGCPJ sem reduções de largura de banda e de *profile*. Os percentuais positivos significam redução do custo computacional e os percentuais negativos significam aumento do custo computacional. Os tempos de execuções

das heurísticas e do MGC são apresentados em segundos.

Tabela 9 Resolução dos SELs sem reduções de largura de banda e de *profile*.

Vértices	β_0	<i>profile</i> ₀	MGC (s)	Iterações
942	918	337600	0,405	158
2165	2120	1818156	1,841	239
4846	4769	9116750	7,497	341
10728	10626	45314579	25,855	504
23367	23167	216212086	100,537	740
50592	50461	1020411959	383,359	1067
108683	108216	4725435534	1252,040	1547
232052	231672	21652820640	3802,335	2248
492853	492100	97893937993	9650,400	8832

Tabela 10 Resolução dos SELs com reduções de largura de banda e de *profile* pela heurística CMr.

Vértices	β	<i>profile</i>	CMr (s)	MGC (s)	Iterações	Redução
942	60	34396	0,0018	0,433	158	-6,91%
2165	93	126101	0,0020	1,954	239	-6,25%
4846	161	468110	0,0020	7,517	341	-0,29%
10728	243	1653567	0,0070	24,882	504	3,74%
23367	420	5570921	0,0156	80,647	739	19,77%
50592	729	19268973	0,0440	313,347	1065	18,25%
108683	911	58863605	0,1479	1217,346	1545	2,76%
232052	1248	180479814	0,2381	3072,977	2245	19,18%
492853	2094	604703562	0,3004	7647,917	3256	20,75%

A heurística CMr obteve uma redução média de 97,6% de largura de banda e 96,3% de *profile*. As reduções de largura de banda e de *profile* com a heurística VNS-*band*, na média, foram de 92,1% e 95,2%, respectivamente.

A heurística MAO-*band* reduziu, em média, apenas 5,8% de *profile* e a largura de banda aumentou, em média, 0,2%. Além das limitações citadas na seção 3.2, o treinamento pode ter sido interrompido precocemente, em razão do tempo máximo de execução, que pode causar *underfitting* e prejudicar o desempenho da heurística MAO-*band*.

Tabela 11 Resolução dos SELs com reduções de largura de banda e de *profile* pela heurística VNS-*band*.

Vértices	β	<i>profile</i>	VNS- <i>band</i> (s)	MGC (s)	Iterações	Redução
942	65	39395	1,032484	0,407	158	-255,43%
2165	94	143168	1,021301	1,532	239	-38,69%
4846	153	496101	1,045590	6,131	341	4,27%
10728	246	1490854	1,105813	24,664	504	0,33%
23367	3425	16625796	1,381177	101,781	736	-2,61%
50592	578	18984571	2,350518	355,387	1068	6,68%
108683	8354	75297775	7,821070	1263,160	1546	-1,51%
232052	28057	445824204	32,764600	3439,130	2253	8,69%
492853	89719	1226201509	112,504000	7814,150	3253	17,86%

Tabela 12 Resolução dos SELs com reduções de largura de banda e de *profile* pela heurística MAO-*band*.

Vértices	β	<i>profile</i>	MAO- <i>band</i> (s)	MGC (s)	Iterações	Redução
942	913	319688	0,0081613	0,405	158	-2,02%
2165	2123	1722120	0,0340908	1,466	239	18,52%
4846	4804	8713715	0,1680248	5,245	341	27,80%
10728	10649	42911616	0,8604821	31,905	504	-26,73%
23367	23275	203965392	4,6147470	128,590	737	-32,49%
50592	50454	958821317	27,0875600	454,813	1067	-25,70%
108683	108567	4425760105	115,3464000	1379,734	1544	-19,41%
232052	231738	20191150501	541,6842000	4026,048	2247	-20,13%
492853	492677	91114577926	2489,565000	9810,660	3237	-27,46%

Para as duas menores instâncias, as heurísticas CMr e VNS-*band* não conseguiram reduzir o tempo de execução do método dos gradientes conjugados. Apenas com a heurística MAO-*band*, houve redução de 18,5% para a instância de 2165 vértices. Para a instância de 4846 vértices, a heurística MAO-*band* obteve a maior redução, considerando-se o tempo de execução da heurística mais o tempo do MGC.

Para as instâncias com mais de 10 mil vértices, houve redução no tempo de execução do método dos gradientes conjugados, reduzindo-se a largura de banda e o *profile* pelas heurísticas CMr e VNS-*band*. Com ex-

ção da instância com 4846 vértices, a heurística CMr obteve os melhores resultados em termos de redução do tempo necessário para solucionar os SELs pelo método dos gradientes conjugados.

4.2.2 SELs resolvidos pelo MGC preconditionado por Jacobi

Para as simulações mostradas nesta subseção, os SELs foram resolvidos pelo método dos gradientes conjugados preconditionado por Jacobi (MGCPJ). Foram executados experimentos sem reduções de largura de banda e de *profile* e com as heurísticas CMr, CMr-pseudo, VNS-*band* e MAO-*band* dez vezes para cada instância. A variância e o desvio padrão para o tempo de execução foram menores do que 0,0804 e 0,2835, respectivamente.

Na Tabela 13, apresentam-se os resultados da resolução dos SELs sem reduções de largura de banda e de *profile* pelo MGCPJ. Nas Tabelas 14, 15, 16 e 17, apresentam-se os resultados da resolução dos SELs com reduções de largura de banda e de *profile* realizadas pelas heurísticas CMr, CMr-pseudo, VNS-*band* e MAO-*band*, respectivamente. Nas últimas colunas dessas tabelas, apresentam-se as reduções dos tempos totais de execução da heurística correspondente, mais os tempos de execução do MGCPJ, em relação ao tempo de execução do MGCPJ sem reduções de largura de banda e de *profile*. Os percentuais positivos significam redução do custo computacional e os percentuais negativos significam aumento do custo computacional. Os tempos de execuções das heurísticas e do MGCPJ são apresentados em segundos.

Tabela 13 Resolução dos SELs preconditionados por Jacobi sem reduções de largura de banda e de *profile*.

Vértices	β_0	<i>profile</i> ₀	MGCPJ (s)	Iterações
942	918	337600	0,447	150
2165	2120	1818156	1,580	221
4846	4769	9116750	5,897	319
10728	10626	45314579	22,571	463
23367	23167	216212086	84,405	666
50592	50461	1020411959	280,458	973
108683	108216	4725435534	883,092	1411
232052	231672	21652820640	2913,121	2030
492853	492100	97893937993	8325,150	2942

Tabela 14 Resolução dos SELs preconditionados por Jacobi com reduções de largura de banda e de *profile* pela heurística CMr.

Vértices	β	<i>profile</i>	CMr (s)	MGCPJ (s)	Iterações	Redução
942	60	34396	0,0001000	0,481	150	-7,63%
2165	93	126101	0,0003000	1,867	221	-18,18%
4846	161	468110	0,0050000	8,070	319	-36,93%
10728	243	1653567	0,0100000	30,479	463	-35,08%
23367	420	5570921	0,0199968	95,495	666	-13,16%
50592	729	19268973	0,0380032	285,577	973	-1,84%
108683	911	58863605	0,0697856	773,950	1410	12,35%
232052	1248	180479814	0,1642950	2835,922	2036	2,64%
492853	2094	604703562	0,3140270	7999,050	2952	3,91%

Tabela 15 Resolução dos SELs preconditionados por Jacobi com reduções de largura de banda e de *profile* pela heurística CMr-pseudo.

Vértices	β	<i>profile</i>	CMr-pseudo (s)	MGCPJ (s)	Iterações	Redução
942	60	37967	0,002000	0,414	150	6,94%
2165	98	142742	0,002000	1,480	221	6,20%
4846	158	548857	0,002000	4,908	319	16,74%
10728	300	1967041	0,008000	20,700	463	8,25%
23367	386	5727030	0,024006	73,398	666	13,01%
50592	560	18693169	0,079974	231,078	973	17,58%
108683	866	60896766	0,140032	713,633	1409	19,17%
232052	1197	190319855	0,343381	2195,710	2034	24,62%
492853	1990	680032614	0,636245	6811,613	2941	18,17%

Tabela 16 Resolução dos SELs preconditionados por Jacobi com reduções de largura de banda e de *profile* pela heurística VNS-*band*.

Vértices	β	<i>profile</i>	VNS- <i>band</i> (s)	MGCPJ (s)	Iterações	Redução
942	65	39395	1,05103	0,450	150	-235,80%
2165	94	143168	1,01666	1,621	221	-66,94%
4846	153	496101	1,03874	6,380	319	-25,81%
10728	246	1490854	1,37914	23,889	463	-11,95%
23367	3425	16625796	1,85177	84,683	666	-2,52%
50592	578	18984571	3,04496	263,354	973	5,01%
108683	8354	75297775	8,31570	792,340	1412	9,33%
108683	8354	75297775	36,93923	792,440	1412	6,08%
108683	8354	75297775	44,91960	792,313	1412	5,19%
232052	28057	445824204	31,06000	2848,496	2043	1,15%
232052	28057	445824204	52,37122	2848,420	2043	0,42%
492853	89719	1226201509	90,93720	6829,813	2946	16,87%

Tabela 17 Resolução dos SELs preconditionados por Jacobi com reduções de largura de banda e de *profile* pela heurística MAO-*band*.

Vértices	β	<i>profile</i>	MAO- <i>band</i> (s)	MGCPJ (s)	Iterações	Redução
942	929	325811	0,0113573	0,442	150	-1,42%
2165	2141	1737751	0,0404324	1,749	221	-13,26%
4846	4797	8727741	0,1905796	6,442	319	-12,47%
10728	10664	42851180	1,1421635	30,266	463	-39,15%
23367	23276	204059064	6,0100250	103,434	666	-29,67%
50592	50491	957689576	26,0013500	335,666	973	-28,96%
108683	108536	4425630101	117,7785000	1019,436	1408	-28,78%
232052	231829	20200695713	533,7337500	3354,255	2034	-33,46%
492853	492663	91172829064	2315,1150000	9432,782	2948	-41,11%

A heurística *CMr-pseudo* obteve redução do custo computacional para todas as instâncias desse conjunto de teste. Para instâncias de até 50592 vértices, não houve redução no tempo computacional para a heurística *CMr*. Para a heurística *VNS-band*, não houve redução no tempo computacional para instâncias de até 23367 vértices. Não se obteve redução no custo computacional da heurística *MAO-band* mais o gradiente conjugado preconditionado por Jacobi nesse conjunto de testes.

Na simulação com a instância de 50592 vértices, após a execução da heurística *VNS-band*, houve redução de 5% no custo computacional do *MGCPJ*. Após a execução da heurística *CMr*, para a mesma instância de 50592 vértices, houve um aumento de 1,84% no custo computacional da heurística mais o *MGCPJ*. As reduções de largura de banda e de *profile* da heurística *VNS-band* foram 26% e 1,5% menores do que as reduções realizadas pela heurística *CMr*, respectivamente, nessa instância. Com a heurística *CMr-pseudo*, obteve-se uma redução de 17,58% no custo computacional e a largura de banda e *profile* obtidos foram 3,11% e 1,53% menores do que os obtidos pela heurística *VNS-band*, respectivamente. Portanto, com a instância de 50592 vértices, a heurística *VNS-band* foi melhor que a heurística *CMr*, mas pior do que a heurística *CMr-pseudo*.

Aumentou-se o tempo máximo de execução da heurística *VNS-band* para verificar se haveria maiores reduções de largura de banda e de *profile* e, consequentemente, redução no custo computacional do *MGCPJ*. Para um tempo máximo de 35 segundos e 44 segundos, não se obtiveram as reduções desejadas para a instância de 108683 vértices. Para a instância de 232052 vértices, aumentou-se o tempo máximo de execução da heurística *VNS-band* para 50 segundos. O custo da heurística mais o custo do *MGCPJ* aumentou

em 21,2 segundos. Para essas instâncias, a heurística CMr-pseudo foi a heurística que mais reduziu largura de banda e o tempo total necessário para se resolver os SELs.

Para a instância de 492853 vértices, a heurística CMr obteve 99,5% de redução de largura de banda e 99,4% de redução de *profile*. A heurística VNS-*band* reduziu 81,8% e 98,7% de largura de banda e de *profile*, respectivamente, para a mesma instância. Entretanto, o número de iterações do MGCPJ executou 8 iterações a menos na execução com a heurística VNS-*band* do que com a heurística CMr. Ainda, o custo computacional com a heurística VNS-*band* foi, aproximadamente, 13,5% menor do que o custo computacional com a heurística CMr. Para essa mesma instância de 492853 vértice, a heurística CMr-pseudo executou 5 iterações a menos do que a heurística VNS-*band*. Novamente, a heurística CMr-pseudo foi a melhor heurística em termos de redução do custo computacional e obteve uma redução de 1,6% em relação à heurística VNS-*band*.

Nas simulações com as instâncias de 2165, 4846 e 10728 vértices, a heurística VNS-*band* obteve mais redução de largura de banda do que a heurística CMr-pseudo. Nas simulações com as instâncias de 4846 e 10728 vértices, a heurística VNS-*band* obteve mais redução de *profile* do que a heurística CMr-pseudo. Mesmo assim, nas simulações com as instâncias de 2165, 4846 e 10728 vértices, a heurística CMr-pseudo obteve custos computacionais menores do que os obtidos pela heurística VNS-*band*. Percebe-se que as reduções de largura de banda e de *profile* não são diretamente proporcionais à redução do custo computacional do MGCPJ. Por isso, pode ser melhor utilizar um método extremamente rápido, como a heurística CMr-pseudo e que reduza, razoavelmente, a largura de banda e o *profile* do que

utilizar uma heurística baseada em meta-heurística, que demande maior custo computacional para encontrar uma solução próxima à solução ótima, como é o caso da heurística *VNS-band*.

A heurística CMr-pseudo reduziu mais largura de banda do que a heurística CMr em quase todas as instâncias. As exceções foram as instâncias de 2165 e de 10728 vértices, em que a heurística CMr reduziu mais 5% e 19% do que a heurística CMr-pseudo, respectivamente. No geral, a heurística CMr reduziu mais *profile* do que a heurística CMr-pseudo. Como na heurística CMr o vértice inicial pode ser qualquer vértice do grafo, caso esse vértice inicial possua menor largura de nível em sua estrutura de nível, a largura de banda poderá ser menor do que a largura de banda encontrada iniciando-se por um vértice pseudo-periférico. Essa constatação foi feita por Cuthill e McKee (1969).

Conclui-se que, no geral, a heurística CMr-pseudo reduziu mais largura de banda e a heurística CMr reduziu mais o *profile* do que as outras heurísticas. A heurística CMr-pseudo foi a única heurística que obteve redução do custo computacional de resolução dos SELs para todas as instâncias. Portanto, para esse conjunto de testes a heurística CMr-pseudo foi a heurística que obteve os melhores resultados.

4.2.3 Ocupações de memória

Em relação às ocupações de memória das execuções mostradas nas subseções 4.2.1 e 4.2.2, a heurística *VNS-band* ocupou, aproximadamente, 129,5MB, 371,1MB, 967,1MB, 2,24GB, 4,93GB e 10,59GB para as instâncias com 10728, 23367, 50592, 108683, 232052 e 492853 vértices, respectivamente. Percebe-se que a ocupação de memória esperada da heurística

VNS-*band* é linear em relação ao número de vértices do grafo.

A heurística MAO-*band* ocupou 141,4 MB de memória para a instância de 232052 vértices e 296,4MB de memória para a instância de 492853 vértices. As instâncias com 10728, 23367, 50592 e 108683 vértices, ocuparam 7,3MB, 16,1MB, 30,7MB e 65,7MB, respectivamente. Para instâncias menores, a ocupação de memória da heurística MAO-*band* foi relativamente pequena. A ocupação de memória esperada da heurística MAO-*band*, também, é linear em relação ao número de vértices do grafo, mas a ocupação de memória da heurística MAO-*band* foi de 17,7 a 36,6 vezes menor do que a ocupação da heurística VNS-*band*. Essas informações são mostradas na Tabela 18, cuja última coluna apresenta a razão entre a ocupação de memória da heurística VNS-*band* e a ocupação de memória da heurística MAO-*band*. Nas Figuras 10 e 11, pode-se observar as ocupações de memória da heurística VNS-*band* e da heurística MAO-*band*, respectivamente.

Tabela 18 Ocupações de memória das heurísticas VNS-*band* e MAO-*band* com as instâncias resultantes da discretização da equação de condução do calor por volumes finitos.

Vértices	VNS- <i>band</i> (MB)	MAO- <i>band</i> (MB)	VNS- <i>band</i> /MAO- <i>band</i>
10728	129,50	7,30	17,74
23367	371,10	16,10	23,05
50592	967,10	30,70	31,50
108683	2293,76	65,70	34,91
232052	5048,32	141,40	35,70
492853	10844,16	296,40	36,59

A ocupação de memória da heurística CMr é bastante pequena, pois nenhuma estrutura de armazenamento é utilizada na renumeração dos vértices que é feita via manipulação de ponteiros no próprio grafo, isto é, o método executa *in place*.

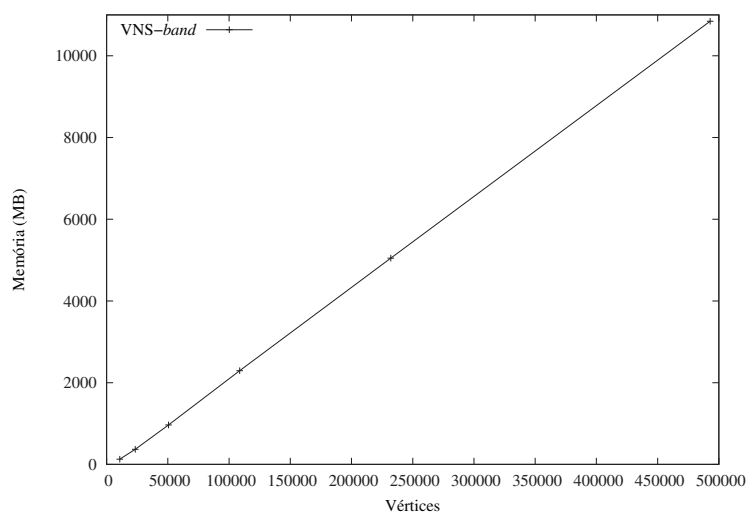


Figura 10 Gráfico das ocupações de memória da heurística VNS-band com as instâncias resultantes da discretização da equação de condução do calor por volumes finitos.

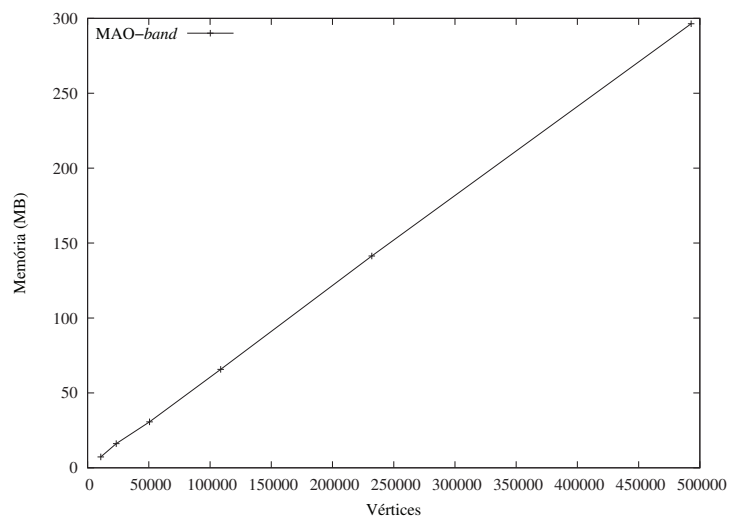


Figura 11 Gráfico das ocupações de memória da heurística MAO-band com as instâncias resultantes da discretização da equação de condução do calor por volumes finitos.

Implementou-se um projeto computacional principal responsável por instanciar o grafo, realizar o cálculo de largura de banda e de *profile*, executar as heurísticas de reduções de largura de banda e de *profile*, o método dos gradientes conjugados, entre outros. A ocupação desse projeto computacional foi de aproximadamente 1,2GB de memória para a instância de 492853 vértices. Para se obter o consumo total de memória, é necessário somar o consumo da heurística utilizada e o consumo do projeto computacional principal.

As simulações descritas nas subseções 4.2.1 e 4.2.2 diferem apenas na utilização do MGCPJ, como mostrado na subseção 4.2.2, e da utilização do MGC sem condicionamento, como mostrado na subseção 4.2.1. Como a etapa de reduções de largura de banda e de *profile* é realizada antes da resolução do SEL pelo MGC ou MGCPJ, as ocupações de memória das heurísticas para reduções de largura de banda e de *profile* são semelhantes nos experimentos apresentados nas subseções 4.2.1 e 4.2.2.

4.3 Simulações com SELs oriundos da discretização da equação de Laplace

As simulações a seguir foram aplicadas em SELs resultantes da discretização da equação de Laplace por volumes finitos (OLIVEIRA; KISCHINHEVSKY; TAVARES, 2013). A equação de Laplace é uma equação diferencial parcial elíptica. Considere o problema de Dirichlet representado pela equação de Laplace e definido por

$$\nabla^2 \phi = 0 \text{ em } \Omega \in \mathbb{R}^2, \phi = f \text{ em } \partial\Omega,$$

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 $\partial\Omega$. As condições de contorno foram definidas nos lados superior, inferior e esquerdo do quadrado unitário com um valor de fronteira definido único f e no lado direito com um valor diferente (OLIVEIRA; KISCHINHEVSKY; TAVARES, 2013).

Na subseção 4.3.1, apresentam-se as simulações com SELs numerados aleatoriamente. Na subseção 4.3.2, apresentam-se as simulações com SELs numerados por uma curva de preenchimento de espaço.

4.3.1 Simulações com SELs numerados aleatoriamente

Nesta subseção, são descritas simulações com SELs oriundos da discretização da equação de Laplace e resolvidos pelo método dos gradientes conjugados preconditionado por Jacobi (MGCPJ). Entretanto, antes das reduções de largura de banda e de *profile* e resolução do SEL pelo MGCPJ, essas instâncias tiveram seus vértices numerados aleatoriamente.

Foram executados experimentos sem redução de largura de banda e com as heurísticas CMr, CMr-pseudo, VNS-*band* e MAO-*band* cinco vezes para cada instância. A variância e o desvio padrão para o tempo de execução foram menores do que 0,00116 e 0,0334, respectivamente.

Utilizou-se a taxa de aprendizado $\eta = 0,9$ para a heurística MAO-*band*. Os tempos máximos de execução utilizados para as heurísticas MAO-*band* e VNS-*band* foram de 1 segundo. Para a heurística VNS-*band*, mesmo aumentando-se o tempo máximo de execução para três vezes o tempo de execução de cada instância presente na Tabela 22, não se obteve diferença na renumeração dos vértices encontrada pela heurística para instâncias até

75542 vértices. Para as demais instâncias, o resultado da heurística não modificou aumentando-se o tempo de execução para o dobro do valor de cada instância presente na Tabela 22. Essas verificações foram feitas diretamente com o executável da heurística *VNS-band* e os arquivos de entrada de cada instância gerados pelo projeto computacional principal.

A seguir, na subseção 4.3.1.1, são apresentados os resultados dos experimentos realizados e algumas análises em relação às reduções de largura de banda e de *profile* e resoluções dos SELs pelo MGCPJ. Na subseção 4.3.1.2, são apresentadas e analisadas as ocupações de memória do projeto computacional desenvolvido e das heurísticas de reduções de largura de banda e de *profile* utilizadas.

4.3.1.1 Reduções de largura de banda e de *profile* e resoluções dos SELs pelo MGCPJ

Na Tabela 19, apresentam-se os resultados da resolução dos SELs numerados aleatoriamente sem reduções de largura de banda e de *profile* pelo MGCPJ. Nas Tabelas 20, 21, 23 e 22 apresentam-se os resultados da resolução dos SELs numerados aleatoriamente com reduções de largura de banda e de *profile* realizadas pelas heurísticas CMr, CMr-pseudo, *VNS-band* e *MAO-band*, respectivamente. Nas últimas colunas dessas tabelas, apresentam-se as reduções dos tempos totais de execução da heurística correspondente, mais os tempos de execução do MGCPJ, em relação ao tempo de execução do MGCPJ sem reduções de largura de banda e de *profile*. Os percentuais positivos significam redução do custo computacional e os percentuais negativos significam aumento do custo computacional. Os tempos de execuções das heurísticas e do MGCPJ são apresentados em segundos.

Tabela 19 Resolução dos SELs numerados aleatoriamente pelo MGCPJ sem reduções de largura de banda e de *profile*.

Vértices	β_0	<i>profile</i> ₀	MGCPJ (s)	Iterações
1970	1960	1144370	1,182	257
7322	7248	16083808	12,264	498
15944	15902	76482022	49,246	745
34238	34059	357518296	156,072	1069
75542	75490	1744941733	497,228	1540
101780	101583	3169282786	797,200	2173
192056	191738	11329772559	1652,863	2383
277118	277019	23512579029	2831,510	2771

Tabela 20 Resolução dos SELs numerados aleatoriamente pelo MGCPJ com reduções de largura de banda e de *profile* pela heurística CMr.

Vértices	β	<i>profile</i>	CMr (s)	MGCPJ (s)	Iterações	Redução
1970	42	54969	0,004000	1,118	257	5,08%
7322	78	396904	0,004000	9,018	498	26,44%
15944	118	1146108	0,006000	39,080	745	20,63%
34238	191	3413246	0,013994	135,492	1069	13,18%
75542	276	12096451	0,027965	434,972	1540	12,51%
101780	405	21403315	0,045960	827,164	2173	-3,76%
192056	360	42574333	0,076715	1729,330	2383	-4,63%
277118	424	74626367	0,106670	2862,997	2771	-1,12%

Tabela 21 Resolução dos SELs numerados aleatoriamente pelo MGCPJ com reduções de largura de banda e de *profile* pela heurística CMr-pseudo.

Vértices	β	<i>profile</i>	CMr-pseudo (s)	MGCPJ (s)	Iterações	Redução
1970	41	57016	0,00010	1,104	257	6,60%
7322	79	419510	0,00200	8,798	498	28,25%
15944	147	1744124	0,00800	40,664	745	17,41%
34238	268	5944221	0,02601	138,076	1069	11,51%
75542	354	16473744	0,07401	450,500	1540	9,38%
101780	413	28716475	0,11006	830,350	2173	-4,17%
192056	463	55315711	0,20668	1715,582	2383	-3,81%
277118	564	93997033	0,28979	2851,740	2771	-0,72%

Tabela 22 Resolução dos SELs numerados aleatoriamente pelo MGCPJ com reduções de largura de banda e de *profile* pela heurística VNS-band.

Vértices	$\bar{\beta}$	<i>profile</i>	VNS-band (s)	MGCPJ (s)	Iterações	Redução
1970	258	112853	1,0362	1,180	257	-87,50%
7322	1341	1111620	1,0608	10,910	498	2,39%
15944	1242	3021502	1,1754	44,946	745	6,34%
34238	5760	24187972	1,6368	143,136	1069	7,24%
75542	20916	44314478	3,5275	441,110	1540	10,58%
101780	4628	25638453	5,2884	1011,217	2173	-27,51%
192056	7107	55753142	15,1769	1979,810	2383	-20,70%
277118	1452	160565391	28,4754	2876,810	2771	-2,61%

Tabela 23 Resolução dos SELs numerados aleatoriamente pelo MGCPJ com reduções de largura de banda e de *profile* pela heurística MAO-band.

Vértices	$\bar{\beta}$	<i>profile</i>	MAO-band (s)	MGCPJ (s)	Iterações	Redução
1970	1943	1168845	0,0328	1,232	257	-7,01%
7322	7253	16069995	0,4450	11,570	498	2,03%
15944	15832	77060810	2,1334	48,608	745	-3,04%
34238	34101	358010103	10,2024	169,860	1069	-15,37%
75542	75369	1745768985	52,0152	543,308	1540	-19,73%
101780	101658	3163501571	97,1522	1067,300	2173	-46,07%
192056	191803	11332807570	387,6390	2755,270	2383	-90,15%
277118	276424	23509884357	712,8293	3549,482	2771	-50,53%

Não se obteve redução do custo computacional para instâncias com mais de 101780 vértices reduzindo-se a largura de banda e o *profile* pelas heurísticas CMr, CMr-pseudo, VNS-*band* ou MAO-*band*. As heurísticas CMr e CMr-pseudo foram as que mais reduziram o custo computacional para as instâncias com até 75542 vértices. A heurística CMr-pseudo foi a melhor heurística para as instâncias de 1970 e 7322 vértices. Para as instâncias de 15944, 34238 e 75542 vértices, a heurística CMr reduziu mais o custo computacional de resolução dos SELs pelo MGCPJ do que as demais heurísticas.

As heurísticas VNS-*band* e MAO-*band* não foram melhores do que as heurísticas CMr e CMr-pseudo. Com a heurística VNS-*band*, obteve-se o pior custo computacional para a instância menor de 1970 vértice. Para essa instância, houve um aumento de 87,5% no custo computacional se comparado com a execução do MGCPJ sem reduções de largura de banda e de *profile*. A heurística MAO-*band* obteve redução do custo computacional apenas para a instância de 7322 vértices; porém, as outras heurísticas obtiveram melhores resultados para essa instância.

4.3.1.2 Ocupações de memória

As ocupações de memória da heurística VNS-*band* para as instâncias de 1970, 7322, 15944, 34238, 75542, 101780, 192056, 277118, foram de 29,9MB, 75,9MB, 196,7MB, 490,5MB, 1,17GB, 1,6GB, 3,12GB e 4,51GB, respectivamente. As ocupações de memória da heurística MAO-*band* para as instâncias de 1970, 7322, 15944, 34238, 75542, 101780, 192056, 277118 vértices, foram de 1MB, 3,1MB, 6,4MB, 18,1MB, 36,6MB, 47MB, 88,3MB, 141,2MB, respectivamente. Essas informações são mostradas na Tabela 24,

cuja última coluna apresenta a razão entre a ocupação de memória da heurística *VNS-band* e a ocupação de memória da heurística *MAO-band*.

Tabela 24 Ocupações de memória das heurísticas *VNS-band* e *MAO-band* com as instâncias resultantes da discretização da equação de Laplace por volumes finitos numeradas aleatoriamente.

Vértices	<i>VNS-band</i> (MB)	<i>MAO-band</i> (MB)	<i>VNS-band/MAO-band</i>
1970	29,9	1,0	29,90
7322	75,9	3,1	24,48
15944	196,7	6,4	30,73
34238	490,5	18,1	27,10
75542	1198,1	36,6	32,73
101780	1638,4	47,0	34,86
192056	3194,9	88,3	36,18
277118	4618,2	141,2	32,71

As ocupações de memória esperadas das heurísticas *VNS-band* e *MAO-band* são lineares em relação ao número de vértices do grafo. Todavia, o consumo de memória foi de 24,5 a 36,2 vezes menor com a heurística *MAO-band* do que com a heurística *VNS-band*, para esse conjunto de instâncias. Por sua vez, as heurísticas *CMr* e *CMr-pseudo* executam *in place*. Nas Figuras 12 e 13, podem-se observar as ocupações de memória das heurísticas *VNS-band* e *MAO-band*, respectivamente.

Para esse conjunto de testes, a heurística *CMr* foi a heurística que obteve melhores resultados para instâncias de 15944 até 75542 vértices. Para instâncias até 7322 vértices, a heurística *CMr-pseudo* obteve maior redução no tempo de resolução dos SELs pelo MGCPJ. Para instâncias com mais de 101780 vértices, não foi vantajoso realizar as reduções de largura de banda e de *profile*.

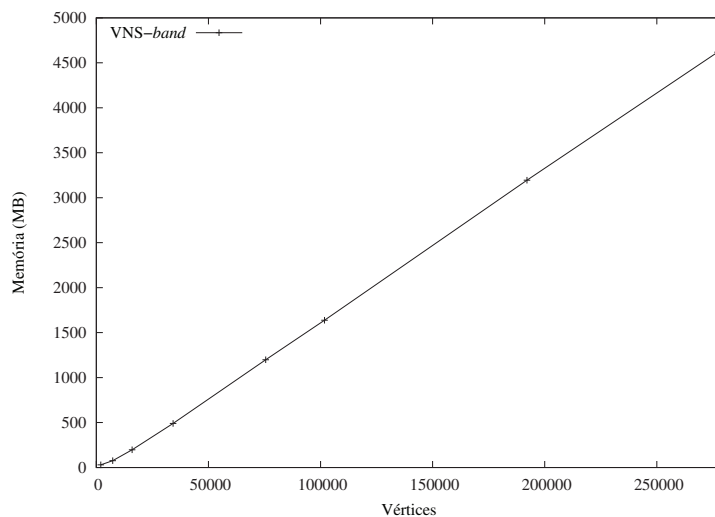


Figura 12 Gráfico das ocupações de memória da heurística VNS-band com as instâncias resultantes da discretização da equação de Laplace por volumes finitos numeradas aleatoriamente.

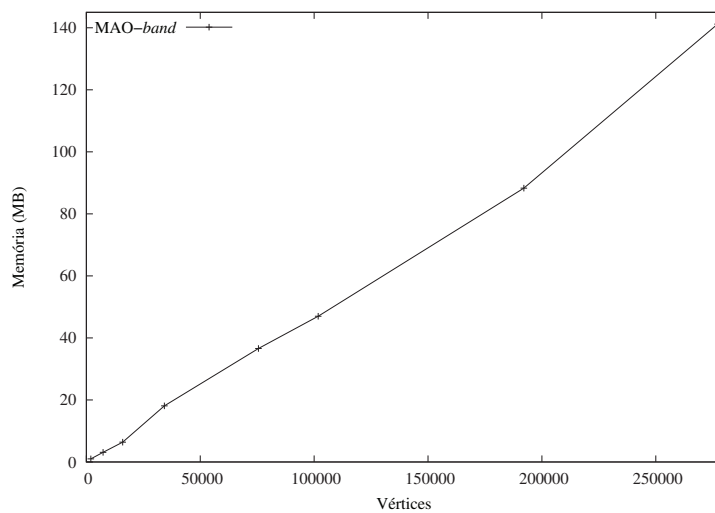


Figura 13 Gráfico das ocupações de memória da heurística MAO-band com as instâncias resultantes da discretização da equação de Laplace por volumes finitos numeradas aleatoriamente.

4.3.2 Simulações com SELs numerados por uma curva de preenchimento de espaço

Nesta subseção, são descritas simulações com SELs oriundos da discretização da equação de Laplace e resolvidos pelo MGCPJ. Essas instâncias foram numeradas pela curva de Sierpiński modificada (VELHO; FIGUEIREDO; GOMES, 1999; OLIVEIRA; KISCHINHEVSKY, 2008).

Foram executados experimentos sem redução de largura de banda e com as heurísticas CMr, CMr-pseudo, VNS-*band* e MAO-*band* cinco vezes para cada instância até 192056 vértices. Para as demais instâncias, os experimentos foram executados três vezes. A variância e o desvio padrão para o tempo de execução foram no máximo 0,82 e 0,9, respectivamente.

Utilizou-se a taxa de aprendizado $\eta = 0,9$ para a heurística MAO-*band*. Os tempos máximos de execução utilizados para as heurísticas MAO-*band* e VNS-*band* foram de 1 segundo. Para a heurística VNS-*band*, mesmo aumentando-se o tempo máximo de execução para cinco vezes o tempo de execução de cada instância presente na Tabela 29, não se obteve diferença na renumeração dos vértices encontrada pela heurística para instâncias até 39716 vértices. Para as instâncias de 68414 e 105764 vértices, o resultado permaneceu inalterado para um tempo máximo de execução de até três vezes o tempo de cada instância presente na Tabela 29. Para as demais instâncias, o resultado da heurística não modificou, mesmo aumentando-se o tempo de execução para o dobro do valor de cada instância presente na Tabela 29. Essas verificações foram feitas diretamente com o executável da heurística VNS-*band* e os arquivos de entrada de cada instância gerados pelo projeto computacional principal.

Os resultados, obtidos nesse conjunto de testes, indicam que a heurística CMr-pseudo é, entre as heurísticas utilizadas para reduções de largura de banda e de *profile*, a heurística que pode obter maiores reduções no custo computacional de resolução dos SELs pelo MGCPJ. Ainda, os resultados apontam que há indícios de que pode não compensar realizar reduções de largura de banda e de *profile* em SELs numerados por uma curva de preenchimento de espaço, como nas simulações mostradas por Oliveira, Kischinevsky e Tavares (2013), e resolvê-los pelo MGCPJ.

A seguir, na subseção 4.3.2.1, são apresentados os resultados dos experimentos realizados e algumas análises em relação às reduções de largura de banda e de *profile* e resoluções dos SELs pelo MGCPJ. Na subseção 4.3.2.2, são apresentadas e analisadas as ocupações de memória do projeto computacional desenvolvido e das heurísticas de reduções de largura de banda e de *profile* utilizadas.

4.3.2.1 Reduções de largura de banda e de *profile* e resoluções dos SELs pelo MGCPJ

Na Tabela 25, apresentam-se os resultados da resolução dos SELs sem reduções de largura de banda e de *profile* pelo MGCPJ. Nas Tabelas 26, 27, 28 e 29 apresentam-se os resultados da resolução dos SELs com reduções de largura de banda e de *profile* realizadas pelas heurísticas CMr, CMr-pseudo, MAO-band e VNS-band, respectivamente. Nas últimas colunas dessas tabelas, apresentam-se as reduções dos tempos totais de execução da heurística correspondente, mais os tempos de execução do MGCPJ, em relação ao tempo de execução do MGCPJ sem reduções de largura de banda e de *profile*. Os percentuais positivos significam redução do custo computacional.

onal e os percentuais negativos significam aumento do custo computacional. Os tempos de execuções das heurísticas e do MGCPJ são apresentados em segundos.

Tabela 25 Resolução dos SELs numerados por uma curva de preenchimento de espaço pelo MGCPJ sem reduções de largura de banda e de *profile*.

Vértices	β_0	<i>profile</i> ₀	MGCPJ (s)	Iterações
1874	1873	57765	1,0840	257
5882	5881	333971	7,7900	454
16922	16921	1710910	46,1420	767
39716	39715	6309342	166,3220	1144
68414	68413	14882117	375,8966	1514
105764	105763	29560801	705,5030	1846
192056	192055	80822936	1687,8002	2383
281954	281953	156196014	2946,8490	2812
373796	373795	256465457	4576,0367	3150
489068	489067	387216428	6380,9000	3519
740288	740287	838516571	11597,7233	4270
1015004	1015003	1580908606	17095,1000	4569
1485410	1485409	3181425530	28794,2000	5270

Tabela 26 Resolução dos SELs numerados por uma curva de preenchimento de espaço pelo MGCPJ com reduções de largura de banda e de *profile* pela heurística CMr.

Vértices	β	<i>profile</i>	CMr (s)	MGCPJ (s)	Iterações	Redução
1874	41	51767	0,00001	1,0500	257	3,14%
5882	71	275020	0,00200	6,1880	454	20,54%
16922	115	1252527	0,00600	45,3800	767	1,64%
39716	195	4376986	0,01401	168,9060	1144	-1,56%
68414	238	9598308	0,02201	386,7980	1514	-2,91%
105764	311	18180951	0,04000	729,6400	1846	-3,43%
192056	360	42574333	0,08013	1734,5600	2383	-2,78%
281954	407	69777828	0,10991	2967,7833	2812	-0,71%
373796	427	99244903	0,17032	4652,7500	3150	-1,68%
489068	469	149352050	0,18910	6395,3200	3519	-0,23%
740288	469	236208613	0,29082	11746,5000	4270	-1,29%
1015004	462	316647386	0,40414	17256,3000	4569	-0,95%
1485410	469	491372577	0,57617	29445,4333	5270	-2,26%

Tabela 27 Resolução dos SELs numerados por uma curva de preenchimento de espaço pelo MGCPJ com reduções de largura de banda e de *profile* pela heurística CMr-pseudo.

Vértices	β	<i>profile</i>	CMr-pseudo (s)	MGCPJ (s)	Iterações	Redução
1874	38	51860	0,00001	1,044	257	3,69%
5882	76	325097	0,00001	6,262	454	19,61%
16922	152	1872378	0,00800	45,876	767	0,56%
39716	286	7557126	0,03400	168,812	1144	-1,52%
68414	397	16275773	0,06601	384,414	1514	-2,28%
105764	432	26637451	0,11334	734,037	1846	-4,06%
192056	463	55315711	0,19669	1699,760	2383	-0,72%
281954	565	96091953	0,28979	2949,933	2812	-0,11%
373796	570	130555143	0,38673	4429,383	3150	3,20%
489068	587	179902601	0,50100	6404,116	3519	-0,37%
740288	587	278024983	0,76732	11815,867	4270	-1,89%
1015004	462	363280959	1,03900	17458,400	4569	-2,13%
1485410	521	561183705	1,50733	29568,167	5270	-2,69%

Tabela 28 Resolução dos SELs numerados por uma curva de preenchimento de espaço pelo MGCPJ com reduções de largura de banda e de *profile* pela heurística MAO-band.

Vértices	β	<i>profile</i>	MAO-band (s)	MGCPJ (s)	Iterações	Redução
1874	1844	1010425	0,0275	1,0680	257	-1,06%
5882	5821	10060410	0,2742	6,5260	454	12,71%
16922	16870	83773254	2,2759	48,0440	767	-9,05%
39716	39549	465224178	13,3085	200,5520	1144	-28,58%
68414	68267	1381895033	40,5791	470,3960	1514	-35,94%
105764	105497	3313239522	99,3210	906,4600	1846	-42,56%
192056	191661	10915862166	324,8073	2117,3200	2383	-44,69%
281954	281736	23513746884	702,4636	3691,4667	2812	-49,11%
373796	373512	41365091373	1245,7633	5651,6631	3150	-50,73%
489068	488681	70667989807	2134,4400	8052,7000	3519	-59,65%
740288	739318	161947149989	4622,6700	14639,5000	4270	-66,09%
1015004	1014083	304420871105	8071,1300	21463,7000	4569	-72,77%
1485410	1483781	652546694835	19135,0333	36480,9000	5270	-93,15%

Tabela 29 Resolução dos SELs numerados por uma curva de preenchimento de espaço pelo MGCPJ com reduções de largura de banda e de *profile* pela heurística VNS-*band*.

Vértices	β	<i>profile</i>	VNS- <i>band</i> (s)	MGCPJ (s)	Iterações	Redução
1874	363	105250	1,01259	1,1080	257	-95,63%
5882	887	529491	1,03764	7,6920	454	-12,06%
16922	4765	2328301	1,16345	47,1760	767	-4,76%
39716	8043	13261308	1,66940	172,3940	1144	-4,65%
68414	2542	17876119	2,78124	401,4400	1514	-7,54%
105764	15174	22202516	5,08892	732,4900	1846	-4,55%
192056	2888	71935018	14,22770	1869,6033	2383	-11,61%
281954	20891	109731059	28,40410	3067,0733	2812	-5,04%
373796	25580	145246396	49,23200	5079,4266	3150	-12,08%
489068	84473	531636561	84,66560	6441,1200	3519	-2,27%

As heurísticas VNS-*band* e MAO-*band* não foram melhores que a heurística CMr ou a heurística CMr-pseudo nesse conjunto de instâncias, ao utilizar o MGCPJ para resolver os SELs. Nas cinco instâncias maiores de 373796, 489068, 740288, 1015004 e 1485410 vértices, a heurística MAO-*band* aumentou o *profile* em 61%, 83%, 93%, 92% e 105%, respectivamente. Como consequência, o custo computacional aumentou em mais de 50% para essas cinco instâncias.

Para a instância de 489068 vértices, a heurística VNS-*band* aumentou o *profile* em 37,3%. Não se obteve redução no custo de resolução dos SELs pelo MGCPJ para nenhuma instância desse conjunto de testes ao realizar reduções de largura de banda e de *profile* pela heurística VNS-*band*.

A heurística CMr reduziu mais o *profile* do que as demais heurísticas, em todas as instâncias. A heurística CMr-pseudo obteve maior redução de largura de banda com a instância de 1874 vértices do que as demais heurísticas. Com a instância de 1015004 vértices, as heurísticas CMr e

CMr-pseudo empataram e reduziram mais a largura de banda do que as demais heurísticas. Em todas as demais instâncias, a heurística CMr obteve maiores reduções de largura de banda do que as demais heurísticas.

A heurística CMr-pseudo foi a única heurística que obteve redução do custo computacional para uma instância com mais de 16922 vértices. Além de obter a maior redução para a menor instância de 1874 vértices, a heurística CMr-pseudo obteve uma redução de 3,2% no tempo de resolução do SELs pelo MGCPJ para a instância de 373796 vértices. A heurística CMr foi a heurística que obteve melhores resultados em termos de redução do custo computacional para as instâncias de 5882 e 16922 vértices.

4.3.2.2 Ocupações de memória

As ocupações de memória da heurística *VNS-band* para as instâncias presentes na Tabela 29 foram 29,5MB, 56,6MB, 144,7MB, 327,2MB, 557,4MB, 857,2MB, 1,5421GB, 2,26GB, 3,9GB, 5,01GB, nessa ordem. As ocupações de memória da heurística *MAO-band* para as instâncias presentes na Tabela 28 foram 0,8MB, 2,6MB, 9,3MB, 20,3MB, 36,4MB, 46,9MB, 87,1MB, 141,7MB, 172,9MB, 203,8MB, 336,9MB, 423,3MB e 678,8MB, nessa ordem. Essas informações são mostradas na Tabela 30, cuja última coluna apresenta a razão entre a ocupação de memória da heurística *VNS-band* e a ocupação de memória da heurística *MAO-band*.

As ocupações de memória esperadas de ambas as heurísticas são lineares em relação ao número de vértices do grafo. Entretanto, o consumo de memória da heurística *MAO-band* foi de 15,3 a 36,9 vezes menor do que o consumo da heurística *VNS-band*, para esse conjunto de instâncias. Por sua vez, as heurísticas CMr e CMr-pseudo executam *in place*. Nas Figuras 14 e

15, podem-se observar as ocupações de memória das heurísticas *VNS-band* e *MAO-band*, respectivamente.

Tabela 30 Ocupações de memória das heurísticas *VNS-band* e *MAO-band* com as instâncias resultantes da discretização da equação de Laplace por volumes finitos numeradas por uma curva de preenchimento de espaço.

Vértices	<i>VNS-band</i> (MB)	<i>MAO-band</i> (MB)	<i>VNS-band/MAO-band</i>
1874	29,5	0,8	36,88
5882	56,6	2,6	21,77
16922	144,7	9,3	15,56
39716	327,2	20,3	16,12
68414	557,4	36,4	15,31
105764	857,2	46,9	18,28
192056	1579,1	87,1	18,13
281954	2317,1	141,7	16,35
373796	3993,6	172,9	23,10
489068	5130,2	203,8	25,17
740288	--	336,9	--
1015004	--	423,3	--
1485410	--	678,8	--

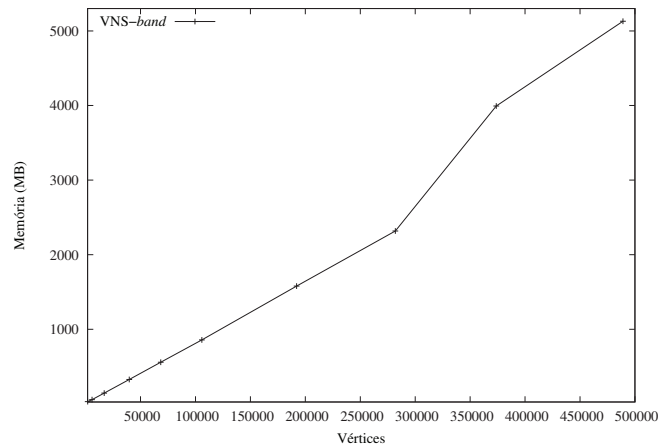


Figura 14 Gráfico das ocupações de memória da heurística *VNS-band* com as instâncias resultantes da discretização da equação de Laplace por volumes finitos numeradas por uma curva de preenchimento de espaço.

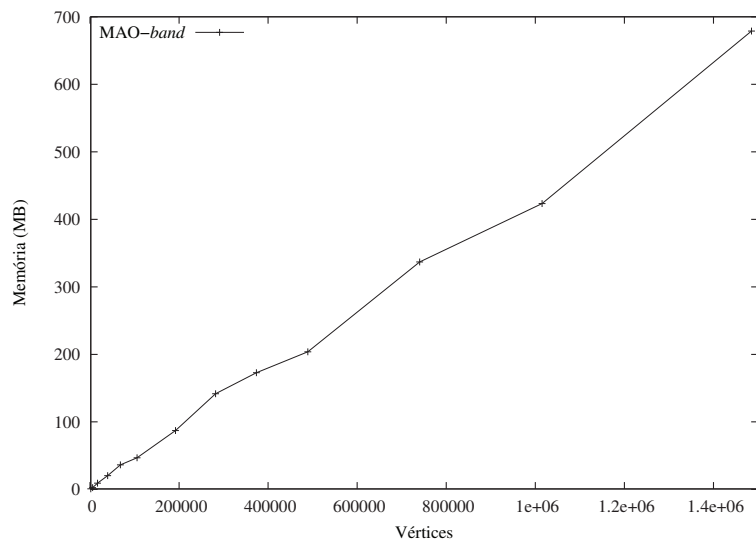


Figura 15 Gráfico das ocupações de memória da heurística MAO-band com as instâncias resultantes da discretização da equação de Laplace por volumes finitos numeradas por uma curva de preenchimento de espaço.

A ocupação de memória do projeto computacional principal foi de, aproximadamente, 2,96GB, para a instância de 1485410 vértices. Como descrito, a ocupação de memória total é obtida somando-se a ocupação do projeto computacional principal e a ocupação da heurística utilizada.

5 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, realizou-se uma verificação de desempenho de heurísticas para as reduções de largura de banda e de *profile* para a redução do custo computacional de resolução de sistemas de equações lineares. Especificamente, foi utilizado o método dos gradientes conjugados preconditionado por Jacobi.

Uma das heurísticas avaliadas foi proposta e implementada utilizando mapa auto-organizável. A heurística MAO-*band* foi desenvolvida para que possa ser executada em matrizes simétricas e assimétricas.

Foram avaliadas, também, as heurísticas CMr, CMr-pseudo e VNS-*band*. A heurística CMr é, possivelmente, a heurística mais clássica para reduções de largura de banda e de *profile*. A heurística CMr-pseudo é o método de reduções de largura de banda e de *profile* disponível no MATLAB (The MathWorks Inc, 1994-2014). A heurística VNS-*band* pode ser considerada o estado da arte para redução de largura de banda. Portanto, comparou-se a heurística proposta, MAO-*band*, com heurísticas importantes para os problemas de redução de largura de banda e de redução de *profile*.

Foram executados experimentos com cinco conjuntos de instâncias. Foram utilizadas instâncias de diferentes tamanhos, que variaram de algumas dezenas de vértices até quase um milhão e meio de vértices. Em particular, não se encontraram testes publicados com essas heurísticas, para reduções de largura de banda e de *profile*, com instâncias desse porte.

A seguir, na subseção 5.1, são feitas algumas considerações finais. Na subseção 5.2, são apresentadas propostas de trabalhos futuros.

5.1 Considerações finais

Nesta subseção, são enfatizadas as conclusões mais relevantes. Na subseção 5.1.1, são feitas algumas considerações sobre a heurística proposta, baseada em mapa auto-organizável, para as reduções de largura de banda e de *profile* de matrizes esparsas simétricas e assimétricas.

Na subseção 5.1.2, são apresentadas algumas considerações sobre os experimentos com instâncias da base *Harwell-Boeing*. Considerações sobre as ocupações de memória do projeto computacional desenvolvido e das heurísticas para reduções de largura de banda e de *profile* utilizadas são apresentadas na subseção 5.1.3. Na subseção 5.1.4, são sumarizadas algumas considerações sobre as simulações de reduções de largura de banda e de *profile* com resolução do SEL pelo MGCPJ.

5.1.1 MAO-band: proposta de uma heurística para reduções de largura de banda e de *profile* por mapa auto-organizável

Apesar de ter executado de forma rápida nos dois conjuntos de instâncias pequenas da base *Harwell-Boeing*, a nova heurística, MAO-band, não obteve resultados competitivos com os resultados obtidos pelas heurísticas CMr, CMr-pseudo e VNS-band. Isso foi constatado tanto na aproximação da solução do problema, quanto em relação ao custo computacional para apresentar uma solução aproximada.

Nas execuções com as instâncias da base *Harwell-Boeing* e com SELs até, aproximadamente, onze mil vértices, a heurística MAO-band executou mais rápido do que a heurística VNS-band. Entretanto, de uma forma geral, a heurística MAO-band não conseguiu bons resultados nas simulações envolvendo resolução de sistemas de equações lineares. Nessas simulações,

o desempenho da heurística MAO-*band* pode ter sido prejudicado pelo *underfitting* causado pelo tempo máximo de execução estabelecido. Ainda, a heurística MAO-*band* não considera informações importantes para as reduções de largura de banda e de *profile*, como o grau de cada vértice, a excentricidade ou a estrutura de nível do vértice inicial da renumeração, como exemplos.

5.1.2 Experimentos com instâncias da base *Harwell-Boeing*

Nos experimentos preliminares realizados com instâncias da base *Harwell-Boeing*, a heurística VNS-*band* foi a que apresentou as maiores reduções de largura de banda e de *profile*. No geral, a heurística VNS-*band* obteve maiores reduções de largura de banda e de *profile*, apesar dos bons resultados e das execuções extremamente rápidas das heurísticas CMr e CMr-pseudo nessas instâncias. A heurística CMr foi a que apresentou os menores custos computacionais nas reduções de largura de banda e de *profile* com matrizes da base *Harwell-Boeing*, em comparação com as demais heurísticas. A heurística CMr-pseudo, também, obteve custos computacionais bastante reduzidos se comparada com as heurísticas VNS-*band* e MAO-*band*.

Esses experimentos com instâncias da base *Harwell-Boeing* foram simulações preliminares porque o objetivo principal de se realizar as reduções de largura de banda e de *profile* foi para se obter redução do custo computacional na resolução de SELs. Não se pode concluir qual é a melhor heurística a ser utilizada, em situações práticas, ao se analisar apenas as reduções de largura de banda e de *profile*, pois a finalidade é a redução do custo computacional de resolução dos SELs. Devem-se analisar as reduções

de largura de banda e de *profile* e o custo computacional da heurística em conjunto com o custo computacional do método resolutor do SEL.

5.1.3 Ocupações de memória

Os crescimentos das ocupações de memória esperadas das heurísticas MAO-*band* e VNS-*band* são lineares no número de vértices da instância. Nos experimentos realizados, a ocupação de memória da heurística VNS-*band* foi cerca de 36 vezes maior do que a ocupação de memória da heurística MAO-*band*. Esse fator pode ser maior para instâncias maiores. Como exemplo, para a instância oriunda da discretização da equação de condução do calor por volumes finitos com quase 500 mil vértices, a ocupação do projeto computacional, em conjunto com a heurística VNS-*band*, foi de aproximadamente 12GB. Com isso, por exemplo, em um computador pessoal com 16GB de memória principal, poderia ser executada uma instância com, no máximo, 670 mil vértices. Entretanto, o executável do VNS-*band*, gentilmente fornecido pelo professor Dr. Dragan Urosevic, só executa com instâncias de até 500 mil vértices. É importante notar que, atualmente, não é comum que computadores pessoais em empresas contenham 16GB de memória principal. Atualmente, pode-se afirmar que é comum que os computadores pessoais em empresas tenham 4GB de memória principal. Ainda, há de se mencionar que foram utilizadas listas de adjacências para representar o grafo, para que se tivesse baixo custo de armazenamento. Caso fosse utilizado um esquema *skyline* (FELIPPA, 1975), por exemplo, o custo de armazenamento poderia ser ainda maior.

A heurística CMr, com ou sem a utilização do método de George e Liu (1979) para encontrar um vértice pseudo-periférico, tem a vantagem de executar *in place*. Portanto, as heurísticas CMr e CMr-pseudo são mais vantajosas do que as heurísticas VNS-*band* e MAO-*band* em relação à ocupação de memória.

5.1.4 Simulações de reduções de largura de banda e de *profile* com resolução do SEL pelo MGCPJ

Nas simulações com os sistemas de equações lineares (SEL) oriundos da discretização da equação de condução de calor por volumes finitos, a heurística CMr-pseudo foi, entre as heurísticas utilizadas nas reduções de largura de banda e de *profile*, a que obteve maiores reduções no custo computacional do método dos gradientes conjugados preconditionado por Jacobi (MGCPJ). A heurística CMr-pseudo obteve até 24,6% de redução no custo computacional de resolução desses SELs pelo MGCPJ.

Ainda, a heurística CMr, iniciado por um vértice pseudo-periférico ou por um vértice qualquer, foi a heurística que obteve melhores resultados nas simulações com SELs numerados aleatoriamente oriundos da discretização da equação de Laplace por volumes finitos; apesar de que, nas simulações realizadas com essas instâncias, só foi vantajoso realizar reduções de largura de banda e de *profile* com grafos até cem mil vértices. Isso porque as reduções de largura de banda e de *profile* realizadas pelas heurísticas CMr, CMr-pseudo, VNS-*band* e MAO-*band* não resultaram em reduções dos custos computacionais de resolução dos SELs pelo MGCPJ, para as instâncias com mais de cem mil vértices. Verificou-se, nessas instâncias, que as reduções de largura de banda e de *profile* não foram diretamente proporcionais

à redução do custo computacional do MGCPJ.

Em alguns dos SELs oriundos de discretizações de equações diferenciais parciais por volumes finitos, as reduções de largura de banda e de *profile* não resultaram em redução do custo computacional necessário para resolução do SEL pelo MGCPJ. Como descrito, as reduções de largura de banda e de *profile* não são diretamente proporcionais à redução do custo computacional do MGCPJ. Como exemplos, veja as simulações com as heurísticas CMr-pseudo e VNS-band com as instâncias de 4846 e 10728 vértices, nas Tabelas 15 e 16, mostradas na subseção 4.2.2, na página 87.

Por último, os resultados das simulações com SELs oriundos da discretização da equação de Laplace por volumes finitos, presentes na subseção 4.3.2, apresentam indícios de que pode não compensar realizar reduções de largura de banda e de *profile* com grafos numerados por uma curva de preenchimento de espaço e resolvê-los pelo método dos gradientes conjugados preconditionado por Jacobi. Com isso, podem não ser necessárias as reduções de largura de banda e de *profile* quando os politopos da malha são numerados por uma curva de preenchimento de espaço, como a curva de Sierpiński modificada (OLIVEIRA; KISCHINHEVSKY, 2008). Isso porque uma curva de preenchimento de espaço já pode fornecer uma numeração adequada dos vértices do grafo que representa a malha, de forma que o *profile* da matriz de adjacências seja pequeno.

Em geral, a heurística CMr-pseudo foi a que apresentou os melhores resultados na redução do custo computacional da simulação completa, em comparação com as demais heurísticas. Conclui-se que, nas simulações apresentadas, a melhor heurística para reduções de largura de banda e de *profile* foi a heurística CMr-pseudo, pois apresentou melhor relação entre

as reduções de largura de banda e de *profile* e o custo computacional da heurística, em conjunto com o custo computacional de resolução do SEL pelo MGCPJ.

Os experimentos mostrados neste trabalho não garantem qual das heurísticas testadas é a melhor em reduções de largura de banda e de *profile*. Isso porque há uma quantidade enorme de tipos de SELs, oriundos das mais diversas aplicações. Entretanto, com esses experimentos, pode-se afirmar que há indícios de que a heurística VNS-*band* não seja a heurística no estado da arte no problema de redução de largura de banda, principalmente ao se utilizar listas de adjacências para representar a discretização do problema. Isso porque utilizar uma heurística que reduz razoavelmente a largura de banda e o *profile* da matriz, com custo computacional muito baixo, como é o caso da heurística CMr-pseudo, pode ser melhor do que utilizar uma heurística que reduza muito a largura de banda e o *profile* da matriz, mas com custo computacional alto. Por isso, os experimentos realizados apresentam indícios de que a heurística CMr-pseudo (GEORGE; LIU, 1981) seja a heurística no estado da arte nos problemas de redução de largura banda e de *profile*.

5.2 Trabalhos futuros

Como trabalho futuro, pretende-se implementar e avaliar o *multi-grid* algébrico como preconditionador do método dos gradientes conjugados e analisar seu desempenho em conjunto com as reduções de largura de banda e de *profile*. Pretende-se também, implementar e avaliar os métodos de preconditionamento *Successive Over-Relaxation* (SOR), *Symmetric Successive Over-Relaxation* (SSOR) e Gauss-Seidel. Essas técnicas serão utilizadas

como preconditionador do método dos gradientes conjugados para avaliar seu desempenho computacional em conjunto com as reduções de largura de banda e de *profile*.

Futuramente, pretende-se estudar e implementar um método eficiente de inversão de matrizes representadas por uma estrutura de dados baseada em listas de adjacências. É necessário inverter a matriz preconditionadora, de forma eficiente, antes de utilizá-la no método dos gradientes conjugados.

Pretende-se implementar um método de busca local para a melhoria do resultado encontrado pela heurística CMr-pseudo. Esse método deverá apresentar custo computacional reduzido, pois seu desempenho será avaliado em conjunto com o MGCPJ para a resolução de SELs. Como exemplos, podem ser usadas as técnicas de *hill climbing* ou *k-opt*.

Pretende-se, ainda, implementar uma estrutura de dados baseada no esquema *skyline*. Pode-se utilizar a estrutura *Vector* para armazenar ponteiros para outras estruturas do tipo *Vector*, contendo ponteiros para objetos que representem as adjacências. Um *Vector*, em linguagem *C++*, é um *container* para vetores (*array*) que podem ser alocados dinamicamente. Assim como um *array*, um objeto do tipo *Vector* possui uma alocação contínua de memória; com isso, pode-se utilizar otimizações referentes à localidade espacial de memória e o acesso aos elementos pode ser feito com custo $O(1)$. Com as reduções de largura de banda e de *profile*, o tamanho dos vetores de adjacências e a quantidade de ponteiros nulos serão pequenas. Dessa forma, o consumo desnecessário de memória será minimizado.

Com essa nova estrutura de dados, será necessário avaliar o desempenho dos métodos de condicionamento supracitados, inclusive os condicionadores $IC(0)$ e $ILUT(p, \tau)$, em conjunto com o método dos gradientes conjugados. Será necessário também, avaliar a escalabilidade do custo de processamento computacional e da ocupação de memória dessa estrutura. Hipoteticamente, essa estrutura de dados deverá ser mais eficiente em um cenário em que o grafo é estático. Por outro lado, o custo de se inserir, ordenar e remover elementos nessa estrutura baseada em *Vectors*, é maior do que em uma estrutura baseada em listas de adjacências. Essas operações são feitas via ponteiros em listas de adjacências. Via de regra, inserir e remover elementos em um *Vector* requer alocar um novo *Vector* de tamanho adequado, mover todos os elementos para o novo *Vector* em suas respectivas posições e destruir a estrutura original. Portanto, essa estrutura de dados pode não ser vantajosa em uma simulação em que a malha do método numérico para a solução da equação diferencial parcial é adaptada repetidamente.

Em suma, fica em aberto qual conjunto de técnicas demanda menor custo computacional em simulações para a discretização de equações diferenciais parciais por volumes finitos com refinamento adaptativo da malha:

1. reduções de largura de banda e de *profile* em um esquema *skyline* implementado com *Vector*, mais o método de condicionamento $IC(0)$ ou $ILUT(p, \tau)$ ou SSOR ou SOR ou *multigrid* algébrico, e um método de inversão de matriz triangular, em conjunto com o método dos gradientes conjugados;
2. reduções de largura de banda e de *profile* em uma estrutura de dados baseada em listas de adjacências e resolução dos SELs pelo método

dos gradientes conjugados preconditionado por Jacobi.

Em particular, na primeira opção, uma heurística como *VNS-band*, que reduz bastante a largura de banda, pode ser vantajosa em relação às demais. Isso porque uma redução maior de largura de banda fará com que haja menor custo de armazenamento no esquema *skyline* e, consequentemente, menor custo computacional ao se percorrer essa estrutura de dados implementada com *Vector*. A redução de largura de banda não influencia na ocupação de memória dos SELs representados por listas de adjacências, porque os elementos nulos não são armazenados nessa estrutura de dados.

REFERÊNCIAS

BARRETO, J. M. Introdução às redes neurais artificiais. In: ESCOLA REGIONAL DE INFORMÁTICA DA SBC REGIONAL SUL, 5., 1997. **Anais...** Florianópolis: Editora da SBC, 1997. p. 41-71.

BENINI, F. A. V. **Rede neural recorrente com perturbação simultânea aplicada no problema do caixeiro viajante**. 2008. 90 p. Dissertação (Mestrado em Engenharia Elétrica) - Universidade de São Paulo, São Carlos, 2008.

BENZI, M. Preconditioning techniques for large linear systems: a survey. **Journal of Computational Physics**, San Diego, v. 182, n. 2, p. 418-477, Nov. 2002.

BOUATORA, Y. et al. A new method for minimizing the bandwidth and profile of square matrices for triangular finite elements mesh. **IEEE Transactions on Magnetics**, New York, v. 43, n. 4, p. 1513-1516, Apr. 2007.

BROCKI, L. Kohonen self-organizing map for the traveling salesperson problem. **Recent Advances in Mechatronics**, Berlin, v. 18, n. 1, p. 116-120, Set. 2007.

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.

DUFF, I. S.; ERISMAN, A. M.; REID, J. K. **Direct methods for sparse matrices**. New York: Oxford University Press, 1986.

DUFF, I. S.; GRIMES, R. G.; LEWIS, J. G. **Harwell-Boeing sparse matrix collection**. [S.l.: s.n], 1989. Disponível em: <<http://math.nist.gov/MatrixMarket/data/Harwell-Boeing/>>. Acesso em: 13 maio 2014.

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

GEORGE, A.; LIU, J. **Computer solution of large sparse positive definite systems**. New Jersey: Prentice-Hall, 1981.

GEORGE, 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, Sept. 1979.

GEORGE, J. A. **Computer implementation of the finite element method**. 1971. 227 p. Tese (Doutorado em Ciência da Computação) - Stanford University, Stanford, 1971.

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, Apr. 1976.

HABDANK-WOJEWODZKI, S.; RYBARSKI, J. **KNNL, C++ Kohonen Neural Network Library**. [S.l.: s.n], 2006. Disponível em: <<http://knnl.sourceforge.net/>>. Acesso em: 8 set. 2013.

HOIT, M.; STOKER, D.; CONSOLAZIO, G. Neural networks for equation renumbering. **Computers and Structures**, Elmsford, v. 52, n. 5, p. 1011-1021, Jan. 1994.

HOIT, M.; WILSON, E. L. An equation numbering algorithm based on a minimum front criteria. **Computers and Structures**, Pergamon, v. 16, n. 1, p. 225-239, Dec. 1983.

HOPFIELD, J. J.; TANK, D. W. Neural computation of decisions in optimization problems. **Biological Cybernetics**, New York, v. 52, n. 3, p. 141-152, Jul. 1985.

KAVEH, A. **Structural mechanics: graph and matrix methods**. Michigan: Research Studies Press, 2004.

KAVEH, A.; SHARAFI, P. Ordering for bandwidth and profile minimization problems via charged system search algorithm. **Transactions of Civil Engineering**, Shiraz, v. 36, n. 1, p. 39-52, May 2012.

KAVEH, A.; TALATAHARI, S. A novel heuristic optimization method: charged system search. **Acta Mechanica**, New York, v. 213, n. 3, p. 267-286, Jan. 2010a.

KAVEH, A.; TALATAHARI, S. Optimum design of skeletal structures using imperialist competitive algorithm. **Computers and Structures**, Elmsford, v. 88, n. 21-22, p. 1220-1229, Nov. 2010b.

KOHONEN, T. **Construction of similarity diagrams for phonemes by a selforganizing algorithm**. Helsinki: Helsinki University of Technology, 1981. (Technical Report TKK-F-A463).

LEW, S. et al. Accuracy and run-time comparison for different potential approaches and iterative solvers in finite element method based EEG source analysis. **Applied Numerical Mathematics**, Amsterdam, v. 59, n. 8, p. 1970-1988, Aug. 2009.

LIM, A. et al. Ant colony optimization with hill climbing for the bandwidth minimization problem. **Applied Soft Computing**, Amsterdam, v. 6, n. 2, p. 180-188, Jan 2006.

LIM, A.; RODRIGUES, B.; XIAO, F. Heuristics for matrix bandwidth reduction. **European Journal of Operational Research**, Amsterdam, v. 174, n. 1, p. 69-91, Oct. 2006.

LIU, J. W.-H. **On Reducing the Profile of Sparse Symmetric Matrices**. 1976. 208 p. Tese (Doutorado em Ciência da Computação) - University of Waterloo, Waterloo, 1976.

MATLAB THE LANGUAGE OF TECHNICAL COMPUTING. Disponível em: <<http://www.mathworks.com/products/matlab/>>. Acesso em 30 out. 2014.

MLADENOVIC, N. et al. Variable neighbourhood search for bandwidth reduction. **European Journal of Operational Research**, Amsterdam, v. 200, n. 1, p. 14-27, Jan. 2010.

OLIVEIRA, F. S. de. **Uma avaliação de movimento de malhas baseado na formulação laplaciana na resolução da equação do calor por discretizações de volumes finitos com refinamento de delaunay**. 2014. 124 p. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Lavras, Lavras, 2014.

OLIVEIRA, S. L. G. de. **Simulações do método Cuthill-McKee reverso e em matrizes de sistemas de equações lineares resolvidos pelo método dos gradientes conjugados pré-condicionado por Jacobi**. Lavras: Editora da UFLA, 2014. (Relatório técnico RT2014-01).

OLIVEIRA, S. L. G. de; CHAGAS, G. O. **Introdução a heurísticas para redução de largura de banda de matrizes**. São Carlos: Sociedade Brasileira de Matemática Aplicada e Computacional, 2014.

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

OLIVEIRA, S. L. G. de; KISCHINHEVSKY, M.; TAVARES, J. M. R. S. Novel graph-based adaptive triangular mesh refinement for finite-volume discretizations. **Computer Modeling in Engineering & Science CMES**, Palmdale, v. 95, n. 2, p. 119-141, Dez. 2013.

PAPADIMITRIOU, C. H. The np-completeness of the bandwidth minimization problem. **Computing**, Saint Louis, v. 16, n. 3, p. 263-270, Jul. 1976.

RAUBER, T. W. Redes neurais artificiais. In: ENCONTRO REGIONAL DE INFORMÁTICA, SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 1998, Nova Friburgo. **Anais...** Nova Friburgo: SBC, 1998. p. 201-228.

RUMELHART, D.; HINTON, G.; WILLIAMS, R. Learning representations by back-propagation errors. **Nature**, London, v. 2, n. 323, p. 533-536, Oct. 1986.

RUMELHART, D.; RAMSEY, W.; STICH, S. **Philosophy and connectionist theory**: volume 1. Lawrence: Erlbaum, 1991.

SAAD, Y. **Iterative methods for sparse linear systems**. 2. ed. Philadelphia: Society for Industrial and Applied Mathematics, 2003.

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

VELLASCO, M. M. B. R. **Redes Neurais Artificiais**. [S.l.: s.n], 2007. Disponível em: <www.ica.ele.pucrio.br/files/download.rails?fileId=132>. Acesso em: 15 abr. de 2013.

WIDROW, B.; HOFF, M. E. Adaptive switching circuits. Stanford: Western Electronic Show and Convention, 1960. p. 96-104.

ZUBEN, F. J. V.; ATTUX, R. R. F. Rede Neural de Kohonen e Aprendizado Não-Supervisionado. São Paulo: Editora da Unicamp, 2007. Disponível em: <ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia353_1s07/topico8_07.pdf>. Acesso em 5 abr. 2013.