



**CLÁUDIO VINÍCIUS DE CARVALHO**

**UMA REVISÃO DE HEURÍSTICAS PARA RENUMERAÇÃO  
DE VÉRTICES PARA REDUÇÃO DO CUSTO DE EXECUÇÃO  
DO MÉTODO GMRES PRÉ-CONDICIONADO**

**LAVRAS – MG**

**2018**

**CLÁUDIO VINÍCIUS DE CARVALHO**

**UMA REVISÃO DE HEURÍSTICAS PARA RENUMERAÇÃO DE VÉRTICES PARA  
REDUÇÃO DO CUSTO DE EXECUÇÃO DO MÉTODO GMRES  
PRÉ-CONDICIONADO**

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

Prof. DSc. Sanderson L. Gonzaga de Oliveira

Orientador

**LAVRAS – MG**

**2018**

**Ficha catalográfica elaborada pela Coordenadoria de Processos Técnicos  
da Biblioteca Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).**

Carvalho, Cláudio Vinícius de

Uma revisão de heurísticas para renumeração de vértices para redução do custo de execução do método GMRES pré-condicionado / Cláudio Vinícius de Carvalho. – 2018.

126 p. : il.

Dissertação (mestrado acadêmico) –Universidade Federal de Lavras, 2018.

Orientador: Prof. DSc. Sanderson L. Gonzaga de Oliveira.  
Bibliografia.

1. GMRES pré-condicionado. 2. Heurísticas de renumeração de vértices. 3. Iterated Local Search. I. Oliveira, Sanderson L. Gonzaga de. . II. Título.

**CLÁUDIO VINÍCIUS DE CARVALHO**

**UMA REVISÃO DE HEURÍSTICAS PARA RENUMERAÇÃO DE VÉRTICES PARA  
REDUÇÃO DO CUSTO DE EXECUÇÃO DO MÉTODO GMRES  
PRÉ-CONDICIONADO**

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

APROVADA em 19 de Março de 2018.

Prof. DSc. Diego Nunes Brandão                      CEFET/RJ  
Prof. DSc. Antônio Maria Pereira de Resende    UFLA

Prof. DSc. Sanderson L. Gonzaga de Oliveira  
Orientador

**LAVRAS – MG  
2018**

## **AGRADECIMENTOS**

Agradeço a Deus pela força nos momentos mais árduos essa jornada.

Agradeço à minha amada esposa, Karoline, pelo apoio e parceria durante essa longa caminhada. Por todo o amor, paciência e compreensão nas horas difíceis, e pela valiosa companhia, que ajudaram a tornar o fardo mais leve.

Agradeço a meus pais, Eliane e Cláudio, por me estimularem e acreditarem neste projeto desde o início; por serem meu exemplo, por todo o suporte e ensinamentos, que me fizeram chegar até aqui. Agradeço também a meus queridos irmãos pelo carinho.

Agradeço ao professor Sanderson, pelo convite que deu origem a este projeto. Por toda a dedicação, instruções, críticas, correções e, sobretudo, pela paciência que demonstrou desde o início.

Agradeço aos companheiros e amigos de mestrado Guilherme Chagas, Alexandre Abreu e Júnior Assis, pela base de códigos e conhecimentos sem a qual este projeto não seria possível. Sua dedicação e esforços prévios foram fundamentais para o desenvolvimento deste trabalho.

Agradeço aos professores e funcionários do Departamento de Ciência da Computação - UFLA por todo o apoio.

Agradeço aos amigos e colegas do Laboratório de Estudos e Projetos em Manejo Florestal - LEMAF, pelo companheirismo durante esse período. Agradeço à coordenação do LEMAF por todo apoio e incentivo desde o primeiro momento e por permitir que me dedicasse a este projeto.

## RESUMO

Sistemas de equações lineares envolvendo matrizes esparsas de grande porte surgem, geralmente, da discretização de equações diferenciais parciais, comuns em simulações computacionais de várias áreas da ciência. Métodos iterativos, como o *Generalized Minimal Residual* (GMRES) pré-condicionado, são os mais adequados para resolução desses sistemas. Quando se utiliza esses métodos, pode-se obter redução de seu custo computacional ao se aplicar técnicas de redução de largura de banda ou de *profile* nas matrizes envolvidas. Essas técnicas consistem em agrupar os coeficientes não nulos da matriz o mais próximo possível da diagonal principal por meio de permutações de suas linhas e colunas. Neste trabalho, avaliou-se o desempenho de métodos heurísticos no estado da arte para redução de largura de banda ou de *profile* no contexto de resolução de sistemas de equações lineares com o método GMRES pré-condicionado. Ainda, uma heurística baseada na meta-heurística *Iterated Local Search* para os problemas de redução de largura de banda e de *profile* de matrizes foi proposta. Nos testes realizados em 172 instâncias da base *SuiteSparse Matrix Collection* a heurística proposta apresentou bons resultados, principalmente na redução de *profile* de matrizes assimétricas e de banda de matrizes simétricas. Contudo, seu alto tempo de execução não a qualificou como heurística propícia para reduzir o custo computacional do GMRES pré-condicionado. Treze métodos heurísticos foram avaliados nos experimentos para redução do custo de execução do GMRES pré-condicionado. Foram considerados seis pré-condicionadores, baseados em fatoração incompleta (ILUT, ILUC, ILU(k), VBILUT e VBILUK) e em *multigrid* (ARMS) em 20 instâncias de grandes dimensões. As simulações apontaram, em consonância com a literatura, que os melhores resultados na redução do custo computacional de sistemas de equações lineares são obtidos por heurísticas com baixo custo computacional, mesmo que não apresentem grandes reduções de largura de banda ou *profile*. Ainda, constatou-se que, para certas instâncias, nenhuma heurística contribuiu para a redução do custo de resolução dos sistemas com o GMRES pré-condicionado.

**Palavras-chave:** GMRES. Pré-condicionadores. Renumerações de vértices. Heurísticas. *Iterated Local Search*.

## ABSTRACT

Systems of linear equations that involve large sparse matrices arising from the discretization of partial differential equations are commonplace in computational simulations from many scientific fields. Iterative methods such as the preconditioned Generalized Minimal Residual method (GMRES) are the most suitable for solving such systems. When these methods are used, one can achieve computational cost reductions by applying bandwidth and profile reduction techniques on the related matrices. The purpose of these techniques is to group the coefficients of the matrix near to the main diagonal by applying a sequence of permutations of its rows and columns. In this work, the performance of heuristic methods for bandwidth and profile reductions was evaluated when used alongside the preconditioned GMRES method for solving linear systems. Furthermore, we propose a heuristic method for bandwidth and profile reductions based on the metaheuristic Iterated Local Search. In the tests carried in 172 instances from the SuiteSparse Matrix Collection, the proposed algorithm showed good results, especially in reducing the bandwidth of symmetric matrices and reducing the profile of unsymmetric matrices. However, due to its high execution times, it was not considered conducive to reduce the execution time of the preconditioned GMRES. Thirteen heuristic methods were evaluated in the experiments with the preconditioned GMRES. Six preconditioners based on incomplete factorization (ILUT, ILUC, ILU(k), VBILUT and VBILUK) and on multigrid methods (ARMS) were used, in 20 large instances. In line with previous works, heuristic methods with low computational cost obtained the best results in reducing the computational cost of solving linear systems in the simulations conducted, even though the bandwidth and profile reductions they provide are not the best overall. More, it was observed that for certain instances no heuristic was able to help in reducing the computational cost of solving linear systems with preconditioned GMRES.

**Keywords:** GMRES. Preconditioning. Vertices renumbering. Heuristics. Iterated Local Search.

## LISTA DE FIGURAS

|                                                                                                                                                                                                                                                                   |    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 2.1 – Representações de grafos com diferentes numerações de vértices e suas respectivas matrizes de adjacências . . . . .                                                                                                                                  | 17 |
| Figura 4.1 – Comportamento de uma iteração da meta-heurística <i>Iterated Local Search</i> . Dado um mínimo local $s^*$ , aplica-se uma perturbação que leva a $s'$ . Após efetuar a BuscaLocal em $s'$ , obtemos $s^{*'}$ , que será comparado a $s^*$ . . . . . | 32 |
| Figura 5.1 – Fragmento de um grafo destacando o valor “ideal” do vértice de rótulo 1. São tomados o maior e menor rótulos adjacentes, 45 e 25, respectivamente, e sua soma é dividida por 2. . . . .                                                              | 39 |
| Figura 6.1 – Número de melhores resultados em redução de largura de banda das heurísticas selecionadas em 124 instâncias simétricas . . . . .                                                                                                                     | 47 |
| Figura 6.2 – Valor da métrica $\rho$ das heurísticas selecionadas em redução de largura de banda em 124 instâncias simétricas (quanto menor, melhor) . . . . .                                                                                                    | 47 |
| Figura 6.3 – Número de melhores resultados em redução de <i>profile</i> das heurísticas selecionadas em 124 instâncias simétricas . . . . .                                                                                                                       | 48 |
| Figura 6.4 – Valor da métrica $\rho$ das heurísticas selecionadas em redução de <i>profile</i> em 124 instâncias simétricas (quanto menor, melhor) . . . . .                                                                                                      | 49 |
| Figura 6.5 – Número de melhores resultados para tempo de execução das heurísticas selecionadas em 124 instâncias simétricas . . . . .                                                                                                                             | 50 |
| Figura 6.6 – Valor da métrica $\rho$ para tempo de execução das heurísticas selecionadas em 124 instâncias simétricas (quanto menor, melhor) . . . . .                                                                                                            | 51 |
| Figura 6.7 – Número de melhores resultados em redução de largura de banda das heurísticas selecionadas em 48 instâncias assimétricas . . . . .                                                                                                                    | 52 |
| Figura 6.8 – Valor da métrica $\rho$ das heurísticas selecionadas em redução de largura de banda em 48 instâncias assimétricas (quanto menor, melhor) . . . . .                                                                                                   | 52 |
| Figura 6.9 – Número de melhores resultados em redução de <i>profile</i> das heurísticas selecionadas em 48 instâncias assimétricas . . . . .                                                                                                                      | 53 |
| Figura 6.10 – Valor da métrica $\rho$ das heurísticas selecionadas em redução de <i>profile</i> em 48 instâncias assimétricas (quanto menor, melhor) . . . . .                                                                                                    | 54 |
| Figura 6.11 – Número de melhores resultados para tempo de execução das heurísticas selecionadas em 48 instâncias assimétricas . . . . .                                                                                                                           | 54 |

|                                                                                                                                                                                                                            |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 6.12 – Valor da métrica $\rho$ para tempo de execução das heurísticas selecionadas em 48 instâncias assimétricas (quanto menor, melhor) . . . . .                                                                   | 55 |
| Figura 6.13 – Gráfico sobreposto de $\sum \rho$ para largura de banda e para tempo das simulações com 124 instâncias simétricas, ordenadas por tempo (em escala logarítmica). . . . .                                      | 57 |
| Figura 6.14 – Gráfico sobreposto de $\sum \rho$ para <i>profile</i> e para tempo das simulações com 124 instâncias simétricas, ordenadas por tempo (em escala logarítmica). . .                                            | 57 |
| Figura 6.15 – Gráfico sobreposto de $\sum \rho$ para largura de banda e para tempo das simulações com 48 instâncias assimétricas, ordenadas por tempo (em escala logarítmica). . . . .                                     | 58 |
| Figura 6.16 – Gráfico sobreposto de $\sum \rho$ para <i>profile</i> e para tempo das simulações com 48 instâncias assimétricas, ordenadas por tempo (em escala logarítmica). . . .                                         | 58 |
| Figura 7.1 – Visão geral do funcionamento de um método <i>multigrid</i> . Para resolução de SELs, a matriz de entrada é aproximada por uma série de matrizes menores, até que possa ser resolvida. . . . .                 | 70 |
| Figura 8.1 – Tempo de execução do método GMRES pré-condicionado na instância da área <i>Electromagnetics problem</i> com os pré-condicionadores ILUT e ARMS                                                                | 79 |
| Figura 8.2 – Tempo de execução do método GMRES pré-condicionado na instância da área <i>Optimization problem</i> com os pré-condicionadores ILUC e ARMS . .                                                                | 80 |
| Figura 8.3 – Tempo de execução do método GMRES pré-condicionado nas instâncias da área <i>Directed Weighted Graph problem</i> com os pré-condicionadores ILUT, ILUC e ILU(k) . . . . .                                     | 81 |
| Figura 8.4 – Soma dos tempos de execução do método GMRES pré-condicionado nas instâncias da área <i>Computational Fluid Dynamics problem</i> com os pré-condicionadores ILUT, ILUC e ILU(k) . . . . .                      | 82 |
| Figura 8.5 – Somatório da métrica $\rho$ dos tempos de execução do método GMRES pré-condicionado nas instâncias da área <i>Circuit Simulation problem</i> em que houve convergência com o pré-condicionador ILUT . . . . . | 83 |
| Figura 8.6 – Soma dos tempos de execução do método GMRES pré-condicionado nas instâncias da área <i>Circuit Simulation problem</i> em que houve convergência com o pré-condicionador ARMS . . . . .                        | 84 |

## LISTA DE TABELAS

|                                                                                                                                                                                                                                                                         |     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Tabela 6.1 – <i>Ranking</i> das heurísticas para redução de largura de banda e <i>profile</i> , em relação à métrica $\rho$ . . . . .                                                                                                                                   | 56  |
| Tabela 8.1 – 20 instâncias reais e assimétricas de grande porte utilizadas nos testes para resolução de SELs com o método GMRES pré-condicionado. O tamanho da matriz é indicado por $n$ e o número de elementos não nulos, por $nnz$ . . . . .                         | 76  |
| Tabela 8.2 – Parâmetros utilizados nos testes com o GMRES pré-condicionado a cada tentativa sem convergência. Os valores da 4 <sup>a</sup> tentativa foram utilizados apenas nas instâncias em que não houve convergência em nenhuma das tentativas anteriores. . . . . | 77  |
| Tabela 8.3 – Melhores combinações de heurísticas de ordenação e métodos pré-condicionadores, no geral, nos testes realizados com 20 instâncias assimétricas . . . . .                                                                                                   | 85  |
| Tabela 8.4 – Tempos de execução (em segundos) do método GMRES pré-condicionado em instâncias assimétricas do tipo “ <i>Electromagnetics problem</i> ”. . . . .                                                                                                          | 89  |
| Tabela 8.5 – Tempos de execução (em segundos) do método GMRES pré-condicionado em instâncias assimétricas do tipo “ <i>Optimization problem</i> ”. . . . .                                                                                                              | 89  |
| Tabela 8.6 – Tempos de execução (em segundos) do método GMRES pré-condicionado em instâncias assimétricas do tipo “ <i>Directed Weighted Graph</i> ”. . . . .                                                                                                           | 90  |
| Tabela 8.7 – Tempos de execução (em segundos) do método GMRES pré-condicionado em instâncias assimétricas do tipo “ <i>Computational Fluid Dynamics problem</i> ”. . . . .                                                                                              | 91  |
| Tabela 8.8 – Tempos de execução (em segundos) do método GMRES pré-condicionado em instâncias assimétricas do tipo “ <i>Circuit Simulation problem</i> ”. . . . .                                                                                                        | 92  |
| Tabela A1 – Resultados de redução de largura de banda das heurísticas testadas em 124 instâncias simétricas da <i>SuiteSparse Matrix Collection</i> . . . . .                                                                                                           | 108 |
| Tabela A2 – Resultados de redução de <i>profile</i> das heurísticas testadas em 124 instâncias simétricas da <i>SuiteSparse Matrix Collection</i> . . . . .                                                                                                             | 111 |
| Tabela A3 – Resultados de tempo de processamento, em segundos, das heurísticas testadas em 124 instâncias simétricas da <i>SuiteSparse Matrix Collection</i> . . . . .                                                                                                  | 114 |
| Tabela A4 – Resultados de redução de largura de banda das heurísticas testadas em 48 instâncias assimétricas da <i>SuiteSparse Matrix Collection</i> . . . . .                                                                                                          | 117 |
| Tabela A5 – Resultados de redução de <i>profile</i> das heurísticas testadas em 48 instâncias assimétricas da <i>SuiteSparse Matrix Collection</i> . . . . .                                                                                                            | 118 |

|                                                                                                                                                                                                              |     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Tabela A6 – Resultados de tempo de processamento, em segundos, das heurísticas testadas em 48 instâncias assimétricas da <i>SuiteSparse Matrix Collection</i> . . . . .                                      | 119 |
| Tabela B1 – Parâmetros dos pré-condicionadores e respectivos intervalos de valores utilizados nos testes. . . . .                                                                                            | 121 |
| Tabela B2 – Áreas de aplicação de 35 instâncias da <i>SuiteSparse Matrix Collection</i> usadas nos testes. O número de vértices é indicado por $n$ e o número de coeficientes não nulos, por $nnz$ . . . . . | 125 |
| Tabela B3 – Parametrização do método GMRES: valores de $\sum \rho$ para o parâmetro $m$ nas 35 instâncias selecionadas, com todas as combinações de parâmetros dos pré-condicionadores . . . . .             | 126 |
| Tabela B4 – Parametrização do método GMRES( $m=50$ ): valores de $\sum \rho$ para o parâmetro $\tau$ . . . . .                                                                                               | 126 |
| Tabela B5 – Parametrização do método GMRES( $m=50$ ): valores de $\sum \rho$ para o parâmetro $p$ . . . . .                                                                                                  | 126 |
| Tabela B6 – Parametrização do método GMRES( $m=50$ ): valores de $\sum \rho$ para o parâmetro $k$ . . . . .                                                                                                  | 126 |
| Tabela B7 – Parametrização do método GMRES( $m=50$ ): valores de $\sum \rho$ para o parâmetro $num\_niveis$ . . . . .                                                                                        | 126 |
| Tabela B8 – Parametrização do método GMRES( $m=50$ ): valores de $\sum \rho$ para o parâmetro $tol\_dd$ . . . . .                                                                                            | 126 |
| Tabela B9 – Parametrização do método GMRES( $m=50$ ): valores de $\sum \rho$ para o parâmetro $block\_size$ . . . . .                                                                                        | 126 |

## SUMÁRIO

|       |                                                                                                                                                   |    |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1     | Introdução . . . . .                                                                                                                              | 12 |
| 1.1   | Justificativa . . . . .                                                                                                                           | 13 |
| 1.2   | Objetivos . . . . .                                                                                                                               | 15 |
| 2     | Conceitos básicos . . . . .                                                                                                                       | 16 |
| 2.1   | Conceitos básicos de grafos . . . . .                                                                                                             | 16 |
| 2.2   | Estrutura de armazenamento . . . . .                                                                                                              | 19 |
| 3     | Reduções de largura de banda e de <i>profile</i> de matrizes . . . . .                                                                            | 20 |
| 3.1   | Métodos heurísticos aplicadas ao problema de redução de largura de banda                                                                          | 20 |
| 3.2   | Heurísticas recentes baseadas em meta-heurísticas para o PRLB . . . . .                                                                           | 21 |
| 3.2.1 | <i>Fast Node Centroid with Hill Climbing</i> . . . . .                                                                                            | 23 |
| 3.2.2 | <i>Variable Neighborhood Search</i> . . . . .                                                                                                     | 25 |
| 3.3   | Métodos heurísticos aplicadas ao problema de redução de <i>profile</i> . . . . .                                                                  | 27 |
| 3.3.1 | Heurística NSloan . . . . .                                                                                                                       | 27 |
| 3.3.2 | Heurística Sloan-MGPS . . . . .                                                                                                                   | 28 |
| 3.3.3 | Heurística MPG . . . . .                                                                                                                          | 28 |
| 3.4   | Heurísticas baseadas em meta-heurísticas para o problema de redução de <i>profile</i> . . . . .                                                   | 29 |
| 4     | A meta-heurística <i>Iterated Local Search</i> . . . . .                                                                                          | 31 |
| 4.1   | <i>Iterated Local Search</i> . . . . .                                                                                                            | 31 |
| 4.1.1 | Solução inicial . . . . .                                                                                                                         | 32 |
| 4.1.2 | Busca local . . . . .                                                                                                                             | 33 |
| 4.1.3 | Perturbação . . . . .                                                                                                                             | 33 |
| 4.1.4 | Critério de aceitação . . . . .                                                                                                                   | 34 |
| 4.2   | Aplicações da meta-heurística ILS em problemas de otimização combinatória . . . . .                                                               | 34 |
| 5     | Aplicação da meta-heurística <i>Iterated Local Search</i> aos problemas de reduções de largura de banda e de <i>profile</i> de matrizes . . . . . | 36 |
| 5.1   | Desenvolvimento da heurística baseada na meta-heurística <i>Iterated Local Search</i> . . . . .                                                   | 36 |
| 5.1.1 | Solução inicial . . . . .                                                                                                                         | 36 |
| 5.1.2 | Perturbação . . . . .                                                                                                                             | 38 |

|         |                                                                                              |    |
|---------|----------------------------------------------------------------------------------------------|----|
| 5.1.3   | Busca Local . . . . .                                                                        | 40 |
| 5.1.4   | Critério de aceitação . . . . .                                                              | 41 |
| 5.2     | As heurísticas ILS-Band e ILS-Prof . . . . .                                                 | 42 |
| 5.3     | Detalhes de desenvolvimento . . . . .                                                        | 43 |
| 6       | Simulações para redução de largura de banda e de <i>profile</i> . . . . .                    | 44 |
| 6.1     | Instâncias utilizadas . . . . .                                                              | 44 |
| 6.2     | Métricas utilizadas para comparação dos resultados . . . . .                                 | 45 |
| 6.3     | Simulações com 124 instâncias simétricas . . . . .                                           | 46 |
| 6.3.1   | Redução de largura de banda . . . . .                                                        | 46 |
| 6.3.2   | Redução de <i>profile</i> . . . . .                                                          | 48 |
| 6.3.3   | Tempo de execução . . . . .                                                                  | 49 |
| 6.4     | Simulações com 48 instâncias assimétricas . . . . .                                          | 50 |
| 6.4.1   | Redução de largura de banda . . . . .                                                        | 51 |
| 6.4.2   | Redução de <i>profile</i> . . . . .                                                          | 51 |
| 6.4.3   | Tempo de execução . . . . .                                                                  | 53 |
| 6.5     | Discussão sobre as simulações para redução de largura de banda e de <i>profile</i> . . . . . | 55 |
| 6.5.1   | Simulações em instâncias simétricas . . . . .                                                | 55 |
| 6.5.2   | Simulações em instâncias assimétricas . . . . .                                              | 56 |
| 6.5.3   | Considerações finais sobre o desempenho das heurísticas testadas . . . . .                   | 58 |
| 7       | Resolução de sistemas de equações lineares . . . . .                                         | 60 |
| 7.1     | Métodos iterativos para resolução de SELs . . . . .                                          | 60 |
| 7.1.1   | Métodos baseados no subespaço de Krylov . . . . .                                            | 60 |
| 7.1.1.1 | Convergência de métodos iterativos . . . . .                                                 | 61 |
| 7.1.2   | O método GMRES . . . . .                                                                     | 61 |
| 7.1.2.1 | Método GMRES com reinícios . . . . .                                                         | 63 |
| 7.1.3   | Pré-condicionamento . . . . .                                                                | 63 |
| 7.1.4   | Método GMRES pré-condicionado . . . . .                                                      | 64 |
| 7.1.5   | Paralelização do método GMRES . . . . .                                                      | 65 |
| 7.2     | Técnicas de pré-condicionamento . . . . .                                                    | 66 |
| 7.2.1   | ILU(k) . . . . .                                                                             | 67 |
| 7.2.2   | ILUT . . . . .                                                                               | 69 |
| 7.2.3   | ILUC . . . . .                                                                               | 70 |

|       |                                                                                                                                |     |
|-------|--------------------------------------------------------------------------------------------------------------------------------|-----|
| 7.2.4 | ARMS . . . . .                                                                                                                 | 70  |
| 7.2.5 | Pré-condicionadores em bloco . . . . .                                                                                         | 71  |
| 7.2.6 | Pré-condicionamento paralelo . . . . .                                                                                         | 72  |
| 7.3   | Trabalhos recentes com aplicações do método GMRES pré-condicionado .                                                           | 72  |
| 8     | Simulações para resolução de SELs com o método GMRES                                                                           |     |
|       | pré-condicionado . . . . .                                                                                                     | 74  |
| 8.1   | Descrição das instâncias, heurísticas e pré-condicionadores utilizados . . .                                                   | 74  |
| 8.2   | Simulações com 20 instâncias assimétricas . . . . .                                                                            | 77  |
| 8.2.1 | Simulações com o método GMRES pré-condicionado nas instâncias da<br>área <i>Electromagnetics problem</i> . . . . .             | 78  |
| 8.2.2 | Simulações com o método GMRES pré-condicionado nas instâncias da<br>área <i>Optimization problem</i> . . . . .                 | 79  |
| 8.2.3 | Simulações com o método GMRES pré-condicionado nas instâncias da<br>área <i>Directed Weighted Graph problem</i> . . . . .      | 80  |
| 8.2.4 | Simulações com o método GMRES pré-condicionado nas instâncias da<br>área <i>Computational Fluid Dynamics problem</i> . . . . . | 81  |
| 8.2.5 | Simulações com o método GMRES pré-condicionado nas instâncias da<br>área <i>Circuit Simulation problem</i> . . . . .           | 82  |
| 8.3   | Discussões sobre as simulações para resolução de SELs com o método<br>GMRES pré-condicionado . . . . .                         | 84  |
| 8.3.1 | ILUT . . . . .                                                                                                                 | 86  |
| 8.3.2 | ILU(k) . . . . .                                                                                                               | 86  |
| 8.3.3 | ILUC . . . . .                                                                                                                 | 87  |
| 8.3.4 | ARMS . . . . .                                                                                                                 | 87  |
| 8.3.5 | Considerações finais . . . . .                                                                                                 | 87  |
| 9     | Conclusão e trabalhos futuros . . . . .                                                                                        | 95  |
|       | REFERÊNCIAS . . . . .                                                                                                          | 98  |
|       | APENDICE A – Resultados das simulações para redução de largura de<br>banda e de <i>profile</i> . . . . .                       | 107 |
|       | APENDICE B – Parametrização do método GMRES . . . . .                                                                          | 120 |
| .1    | Determinação dos parâmetros para o método GMRES pré-condicionado .                                                             | 120 |
| .2    | Execuções e parâmetros escolhidos . . . . .                                                                                    | 122 |

## 1. INTRODUÇÃO

Uma ampla variedade de fenômenos físicos pode ser descrita por meio de Equações Diferenciais Parciais (EDPs). Fenômenos térmicos, acústicos, eletrodinâmicos, de fluidos, entre outros, podem ser modelados por meio de EDPs que descrevem a taxa de mudança de suas quantidades físicas, como forças, energias, temperaturas, velocidades, etc (SAAD, 2003).

A resolução desse tipo de equação, normalmente, passa pela aplicação de métodos como o dos volumes finitos ou dos elementos finitos. A saída desses métodos são discretizações, ou aproximações, utilizando um número finito de incógnitas. Essas discretizações levam a sistemas de equações lineares (SEs) da forma

$$Ax = b,$$

em que  $A$  é uma matriz de grande porte e esparsa, isto é, possui poucos elementos não nulos. Grande parte do custo de execução associado a simulações computacionais de diversas áreas da ciência e engenharia está, portanto, ligado à resolução desses sistemas de equações lineares de grande porte, conforme explicam (OLIVEIRA; CHAGAS, 2014).

Para a resolução de sistemas de equações lineares podem ser utilizados métodos diretos ou iterativos. Métodos diretos são amplamente utilizados por serem robustos, altamente confiáveis e por requererem uma quantidade previsível de recursos computacionais (BENZI, 2002). Contudo, eles são pouco escaláveis e, à medida que as instâncias envolvidas crescem, os custos de armazenamento e memória podem se tornar impraticáveis. Métodos iterativos compensam sua menor confiabilidade com custos de execução e armazenamento muito menores que os métodos diretos (SAAD, 2003, p. *xix-xx*). Assim, para sistemas esparsos de grandes proporções ou em que a precisão pode ser sacrificada, métodos iterativos são os indicados.

O método dos gradientes conjugados (MGC) (HESTENES; STIEFEL, 1952; LANCZOS, 1952) é um exemplo de método iterativo. Esse método, baseado no subespaço de Krylov, pode ser utilizado nas situações em que as matrizes envolvidas são simétricas e positivas definidas. Para casos de matrizes não simétricas, ou não positivas definidas, pode-se recorrer a métodos como o *Generalized Minimal Residual Method* (GMRES) (SAAD; SCHULTZ, 1986), ou o *Biconjugate Gradient Stabilized Method* (Bi-CGSTAB) (VORST, 1992), por exemplo. Recomenda-se a leitura de Saad (2003) para um profundo detalhamento de métodos iterativos para a resolução de sistemas de equações lineares.

Apesar de serem os mais recomendados para instâncias de grande porte, em algumas situações, métodos iterativos podem levar muito tempo para encontrarem uma solução. Contudo, é possível reduzir o tempo de execução de métodos iterativos para resolução de sistemas de equações lineares ao se aplicar técnicas de pré-processamento nas matrizes envolvidas (SAAD, 2003, p. 275) (OLIVEIRA; CHAGAS, 2014).

Uma forma de pré-processamento que ajuda a acelerar a convergência de métodos iterativos para resolução de SELs é o uso de pré-condicionadores. No trabalho de Benzi (2002) é apresentado um longo histórico de técnicas de pré-condicionamento aplicadas a métodos iterativos. As reduções de largura de banda e de *profile* de matrizes são outras dessas técnicas de pré-processamento. O objetivo dessas técnicas é, sumariamente, encontrar uma permutação das linhas e colunas da matriz de entrada que faça com que seus elementos não nulos fiquem o mais próximo possível da diagonal principal.

Este projeto está organizado como descrito a seguir. Nas seções 1.1 e 1.2 são apresentadas as justificativas e objetivos deste trabalho, respectivamente. No Capítulo 2 são apresentados conceitos básicos para o entendimento deste texto. O Capítulo 3 é dedicado a explicar os problemas de reduções de largura de banda e de *profile* de matrizes, além dos principais métodos desenvolvidos para esse fim. No Capítulo 4 explica-se o funcionamento do método meta-heurístico *Iterated Local Search*. No Capítulo 5 encontra-se a metodologia utilizada no desenvolvimento e implementação do método para redução de largura de banda e *profile* baseado na meta-heurística *Iterated Local Search*. Em seguida, no Capítulo 6, são apresentados os resultados das simulações com reduções de largura de banda e de *profile*. Métodos para resolução de sistemas de equações lineares são abordados no Capítulo 7. No Capítulo 8 são apresentados os resultados das simulações com resolução de SELs. Por fim, no Capítulo 9, apresentam-se as conclusões sobre este trabalho e proposições de trabalhos futuros.

## 1.1 Justificativa

Como descrito, a resolução de sistemas de equações lineares é responsável por uma grande parcela do custo computacional de simulações numéricas em muitas áreas da ciência e das engenharias. Nessas aplicações, métodos iterativos utilizados para a resolução desses sistemas podem se beneficiar de etapas de pré-processamento dos dados, de forma que fiquem mais estáveis e se tornem menos custosos de serem resolvidos computacionalmente (DUFF; MEURANT, 1989; DUTTO, 1993).

O uso de pré-condicionadores, “é o ingrediente mais crítico no desenvolvimento de resolutores eficientes para problemas desafiadores da computação científica, e a importância do pré-condicionamento está destinada a aumentar ainda mais.”, como explica Benzi (2002, p. 419). Os pré-condicionadores, como os baseados na fatoração LU incompleta, por exemplo, são sensíveis à ordem das incógnitas e equações. Logo, métodos de reordenação de matrizes podem ser grandes aliados na construção de pré-condicionadores eficientes (SAAD, 2003).

Duff e Meurant (1989) compararam diversas técnicas de reordenação de matrizes oriundas de SELs. Os autores verificaram que técnicas de redução de largura de banda, em especial, o algoritmo de Cuthill-McKee (CUTHILL; MCKEE, 1969), causaram uma considerável redução no custo computacional do método dos gradientes conjugados. Diferentes métodos de ordenação foram aplicados com resultados similares por D’Azevedo, Forsyth e Tang (1992), Dutto (1993), Benzi, Haws e Tuma (2000), Duff e Koster (2001), Oliveira, Bernardes e Chagas (2017).

Ao se aplicar métodos de redução de largura de banda ou de *profile* nas matrizes envolvidas, obtém-se, pela reordenação de linhas e colunas, uma representação mais compacta dos dados de entrada, sem que se altere a informação subjacente. Essa representação, além de possibilitar a diminuição no custo de armazenamento, permite que se tire maior proveito da forma com que os dados são organizados na memória do dispositivo, com um aumento a taxa de *cache hits*, de acordo com Das et al. (1994), Burgess e Giles (1997).

O número de permutações de linhas e colunas possíveis em uma matriz  $A$ , de dimensões  $n \times n$ , é da ordem de  $n!$ . O problema de minimização de largura de banda de matrizes pertence à classe de problemas NP-Difícil, como mostrado por Papadimitriou (1976), assim como o problema de minimização de *profile* de matrizes, provado por Lin e Yuan (1994). Pela sua importância prática, diversos métodos exatos e heurísticos têm sido propostos para esse problema desde a década de 1960. Mais recentemente, métodos meta-heurísticos também têm sido aplicados nesse problema (CHAGAS; OLIVEIRA, 2015).

O GMRES (SAAD; SCHULTZ, 1986) é um dos métodos de escolha para resolução de sistemas de equações lineares, em especial, os sistemas assimétricos ou não positivos definidos (SAAD, 2003). Neste trabalho, avaliam-se as ordenações produzidas por métodos heurísticos para redução de largura de banda e de *profile* de matrizes considerados no estado da arte enquanto ferramentas para redução do custo de execução do método GMRES pré-condicionado. Além disso, apresenta-se uma proposta de aplicação da meta-heurística *Iterated Local Search*

(ILS) nos problemas de reduções de largura de banda e de *profile*. Essa escolha é amparada pelos bons resultados obtidos pela aplicação dessa meta-heurística em problemas de otimização combinatória similares (APPLEGATE; COOK; ROHE, 2003; STÜTZLE; HOOS, 2002; STÜTZLE, 2006; PAQUETE; STÜTZLE, 2002; SCHIAVINOTTO; STÜTZLE, 2004; VILLANUEVA et al., 2010; VALDEZ; MEDINA, 2012; SAKURABA et al., 2015). Deste modo, investiga-se também a viabilidade da aplicação da heurística proposta com o intuito de reduzir o custo de resolução de sistemas de equações lineares com o método GMRES.

## 1.2 Objetivos

O objetivo geral deste projeto é avaliar as reduções no custo de execução do método GMRES pré-condicionado obtidas pelas ordenações produzidas por métodos no estado da arte para os problemas de reduções de largura de banda e de *profile*. Pretende-se também investigar o potencial de aplicação da meta-heurística *Iterated Local Search* nos problemas de reduções de largura de banda e de *profile* de matrizes, e se sua utilização pode contribuir com a diminuição do custo de execução do método GMRES. Para tanto, é preciso que os seguintes objetivos específicos sejam alcançados.

- Desenvolvimento de um algoritmo baseado na meta-heurística ILS para redução de largura de banda e *profile* de matrizes simétricas e assimétricas.
- Experimentações com resolução de SELs com o método GMRES, para as diversas combinações de pré-condicionadores e métodos de redução de largura de banda e de *profile* de matrizes no estado da arte, além do algoritmo proposto.
- Avaliação da possível redução do custo computacional de resolução dos SELs decorrente da aplicação de cada método selecionado.
- Identificação, se pertinente, das melhores combinações entre pré-condicionadores e métodos de redução de largura de banda ou de *profile* para diferentes áreas de aplicação das instâncias que levem a menores tempos de resolução de SELs com o método GMRES pré-condicionado.

## 2. CONCEITOS BÁSICOS

Neste capítulo, apresentam-se conceitos básicos para melhor entendimento do trabalho. Na Seção 2.1 são explicadas noções básicas de grafos. Estruturas de armazenamento para grafos esparsos são explicadas na Seção 2.2.

### 2.1 Conceitos básicos de grafos

Para fins de revisão, considere uma matriz  $A$ ,  $n \times n$ , e um vetor  $b$  de tamanho  $n$ , ambos de valores reais. Deseja-se encontrar um vetor  $x$ ,  $X \in \mathbb{R}^n$ , que satisfaça o sistema linear  $Ax = b$ , tal que

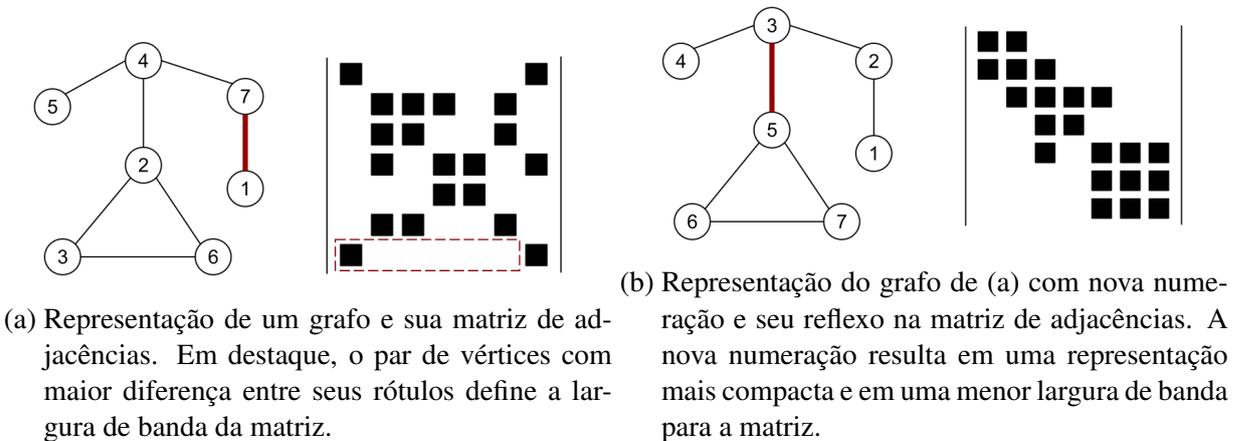
$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}.$$

Durante a resolução computacional de sistemas como o acima, a maior parte da atenção é comumente voltada para as características da matriz de coeficientes  $A$ , mais do que para o vetor de incógnitas  $x$  ou o de termos independentes  $b$ . Isso porque a eficiência de métodos resolutores diretos e iterativos depende das características da matriz  $A$ . Para este texto, considera-se que as matrizes envolvidas são de grande porte e esparsas.

Uma matriz de coeficientes  $A$  pode ser vista como uma matriz de adjacências e, por consequência, como um grafo. Uma matriz  $A$ ,  $n \times n$ , equivale a um grafo  $G = (V, E)$ , em que  $V = \{v_1, v_2, \dots, v_n\}$  é um conjunto de vértices e  $E = \{e_1, e_2, \dots, e_n\}$ . Considere o exemplo da Figura 2.1(a). Nela, são representados um grafo não-ponderado e não-direcionado e sua respectiva matriz de adjacências.

Para o grafo representado na Figura 2.1(b) foi utilizada uma numeração diferente para os vértices. Perceba que os efeitos dessa nova numeração se traduzem uma estrutura mais “compacta” da matriz de adjacências. De maneira análoga, permutações aplicadas às linhas e colunas da matriz de coeficientes alteram a numeração dos vértices em  $V$ . Em um sistema de equações lineares, as reordenações de linhas ou colunas da matriz  $A$  também causam a reordenação do vetor de termos independentes ou do vetor de incógnitas, respectivamente (SAAD, 2003, p.99-103). Considere agora as seguintes definições.

Figura 2.1 – Representações de grafos com diferentes numerações de vértices e suas respectivas matrizes de adjacências



**Definição 1 (largura de banda)** A largura de banda  $\beta$  de uma matriz  $A$ , de dimensões  $n \times n$  pode ser definida como a maior distância entre um elemento não nulo de uma linha  $i$  e a diagonal principal de  $A$ :

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

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

em que  $\beta_i(A)$  é a largura de banda da  $i$ -ésima linha de  $A$ . Isto é, a largura de banda da matriz  $A$  equivale à maior largura de banda entre as  $n$  linhas de  $A$ .

No exemplo da Figura 2.1(a), a maior distância entre um elemento não nulo e a diagonal principal da matriz ocorre na linha  $i=1$ . Assim, a largura de banda  $\beta(A) = \beta_7(A) = 7 - 1 = 6$ . Na matriz da Figura 2.1(b), a largura de banda da matriz é  $\beta(A) = \beta_5(A) = 5 - 3 = 2$ .

Do ponto de vista da teoria dos grafos, o problema de minimização de largura de banda consiste em encontrar a numeração dos vértices que minimize a máxima diferença absoluta entre os rótulos de vértices adjacentes (OLIVEIRA; CHAGAS, 2014). Nos grafos das Figuras 2.1(a) e 2.1(b), as arestas que ligam os vértices com maior diferença absoluta aparecem destacadas. Na Figura 2.1(b), a largura de banda é mínima para essa matriz.

**Definição 2 (profile)** O profile de uma matriz simétrica corresponde à soma das larguras de banda de cada uma de suas linhas. Seja  $A$  uma matriz simétrica  $n \times n$ , então:

$$p(A) = \sum_{i=1}^n \beta_i(A).$$

Para o cálculo de profile de matrizes assimétricas, considere  $L$  e  $U$  como partições de  $A$  em matrizes triangulares inferior e superior, respectivamente. O profile de  $A$  é definido como  $p(A) = \sum_{i=1}^N (\beta_i(L) + \beta_i(U))$ .

Dado um grafo  $G = (V, E)$ , considere também as definições extraídas de Oliveira e Chagas (2014, p.9-11).

**Definição 3 (adjacência)** *Dois vértices são adjacentes se existe uma aresta entre eles. Nesse caso, o conjunto de vértices adjacentes ao vértice  $v$  é dado por  $adj(G, v) = \{u \in V : \{v, u\} \in E\}$*

**Definição 4 (grau de um vértice)** *O grau de um vértice  $v$  é o número de vértices adjacentes a  $v$ , isto é,  $grau(G, v) = |u \in V : \{v, u\} \in E|$ .*

**Definição 5 (caminho)** *Um caminho é uma sequência de vértices, de modo que há aresta ligando um vértice ao próximo da sequência, até o último. O tamanho desse caminho é seu número de arestas.*

**Definição 6 (distância)** *A distância  $d(u, v)$  entre dois vértices  $u, v \in V$  é o tamanho do menor caminho entre eles.*

**Definição 7 (excentricidade)** *A excentricidade  $\ell : V \rightarrow \mathbb{N}$  de um vértice  $v \in V$  é definida como  $\ell(v) = \max_{u \in V} (d(v, u))$ . Isto é, a maior distância entre  $v$  e algum outro vértice  $u \in V$ .*

**Definição 8 (diâmetro)** *O diâmetro  $\Phi(G) = \max_{v \in V} (\ell(v))$  é a maior excentricidade de um grafo  $G = (V, E)$ .*

**Definição 9 (vértice periférico)** *Um vértice é dito periférico se  $\ell(v) = \Phi(G)$ , isto é, sua excentricidade é igual ao diâmetro do grafo. Um vértice  $v$  é considerado pseudo-periférico se sua excentricidade  $\ell(v)$  for próxima ao diâmetro  $\Phi(G)$  do grafo.*

**Definição 10 (estrutura de nível enraizada)** *A estrutura de nível enraizada em  $v$  de profundidade  $\ell(v)$  é o particionamento  $\mathcal{L}(v) = \{L_0(v), L_1(v), \dots, L_{\ell(v)}(v)\}$ , em que  $L_0(v) = \{v\}$  e  $L_i(v) = adj(L_{i-1}(v)) - \cup_{j=0}^{i-1} L_j(v)$ , para  $i = 1, 2, \dots, \ell(v)$ . Em outras palavras, cada nível da estrutura enraizada contém vértices adjacentes aos vértices no nível anterior que ainda não pertencem à estrutura.*

**Definição 11 (largura de nível)** *A largura de nível  $b(\mathcal{L}(v))$  é o número de vértices do nível com mais vértices na estrutura, isto é,  $b(\mathcal{L}(v)) = \max_{0 \leq i \leq \ell(v)} (|L_i(v)|)$ .*

## 2.2 Estrutura de armazenamento

As matrizes envolvidas nos sistemas de equações da forma  $Ax = b$  consideradas nesse trabalho costumam ser esparsas e de grandes dimensões. Deste modo, sua representação por meio de matrizes de adjacências é inviável, devido ao alto custo de armazenamento. É possível representar essas matrizes em formatos mais compactos, mas que retenham características de acesso rápido aos dados. Os métodos *Compressed Row Storage* (CRS) e *Compressed Column Storage* (CCS), por exemplo, mantêm vetores com referências apenas aos valores não nulos e seus índices na matriz de entrada. Considere a matriz assimétrica 5 x 5 abaixo, com 10 elementos não nulos ( $nnz = 10$ ).

$$\begin{pmatrix} 3 & 0 & 0 & -1 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ -2 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 3 & 5 \end{pmatrix}$$

A forma de representar essa matriz no formato *Compressed Row Storage* utiliza três vetores. O vetor  $val$ ,  $|val| = nnz$ , armazena os coeficientes não nulos da matriz, percorrida linha a linha. O vetor  $colInd$ ,  $|colInd| = nnz$ , armazena o índice das colunas dos elementos armazenados no vetor  $val$ . O vetor  $rowPtr$ ,  $|rowPtr| = n + 1$ , armazena os índices dos elementos em  $val$  que iniciam uma linha. Por convenção,  $rowPtr[n + 1] = nnz + 1$ . A representação da matriz acima no formato CRS é a seguinte:

$$val = [3, -1, 1, 2, -2, 4, 2, 1, 3, 5],$$

$$colInd = [1, 4, 2, 3, 1, 3, 4, 1, 4, 5],$$

$$rowPtr = [1, 3, 5, 7, 8, 11]$$

Outra estrutura para armazenamento compacto de matrizes é a CRS-SSS, uma combinação das estruturas de dados CRS e *Skyline Storage Scheme* (SSS) (FELIPPA, 1975). Nessa estrutura, indicada para matrizes simétricas, os coeficientes da diagonal principal são armazenados em um quarto vetor e os coeficientes não nulos da matriz triangular inferior são armazenados como no formato CRS, aliando baixo custo de armazenamento a acesso rápido aos dados por meio de índices. Para mais detalhes sobre formas de armazenamento de matrizes, recomenda-se a leitura de Oliveira (2015).

### 3. REDUÇÕES DE LARGURA DE BANDA E DE *PROFILE* DE MATRIZES

Dada a natureza NP-Difícil dos problemas de reduções de largura de banda e de *profile* de matrizes (PAPADIMITRIOU, 1976; LIN; YUAN, 1994) e sua importância prática, uma variedade de métodos exatos e heurísticos têm sido propostos para esses problemas desde a década de 1960. Métodos exatos, como os empregados por Corso e Manzini (1999) e por Martí, Campos e Piñana (2008), por exemplo, vasculham o conjunto de  $n!$  permutações das linhas e colunas da matriz utilizando alguma forma de *backtracking*. Em problemas práticos nos quais as matrizes envolvidas têm grandes dimensões, como neste trabalho, eles são preteridos em favor de métodos heurísticos.

Neste capítulo, apresentam-se três métodos heurísticos simples mas bastante utilizados para redução de largura de banda: o método de Cuthill-McKee, Cuthill-McKee reverso e GPS, na seção 3.1. Uma breve revisão dos principais métodos baseados em meta-heurísticas é apresentada na seção 3.2. Alguns métodos heurísticos proeminentes para redução de *profile* de matrizes são introduzidos na seção 3.3. Métodos meta-heurísticos aplicados no problema de redução de *profile* são mencionados na seção 3.4.

#### 3.1 Métodos heurísticos aplicadas ao problema de redução de largura de banda

Uma das heurísticas clássicas para o problema de redução de largura de banda (PRLB) de matrizes foi proposta por Cuthill e McKee (CM) (CUTHILL; MCKEE, 1969). Essa heurística, aplicável a matrizes simétricas, é muito semelhante à busca em largura. O que a difere da busca em largura é a ordem na qual os vértices do grafo são visitados. Na heurística CM, dado um vértice inicial, seus vértices adjacentes são visitados por ordem crescente de grau. Como explicam Oliveira e Chagas (2014, p.16), essa ordenação faz com que os vértices de maior grau fiquem posicionados, na medida do possível, nas linhas centrais da matriz, o que resulta em uma boa configuração com relação à largura de banda. O pseudocódigo da heurística CM é apresentado no Algoritmo 3.1.

Uma melhoria da heurística CM, a heurística Cuthill-McKee reverso (RCM) foi apresentada por George (GEORGE, 1971). O autor constatou que a simples inversão da numeração obtida pela heurística CM resultava em um *profile* pelo menos tão bom quanto o obtido pela numeração original, sem alterar sua largura de banda.

Após analisarem a heurística de CM, Gibbs, Poole Jr e Stockmeyer (1976) verificaram que ela não era adequada em situações em que há um custo alto em se encontrar o vértice inicial.

---

**Algoritmo 3.1** Pseudocódigo do algoritmo de Cuthill-McKee
 

---

```

1: CUTHILL-MCKEE
2:   Entrada: um grafo  $G(V,E)$  e um vértice inicial  $v \in V$ .
3:   Saída: renumeração  $p$  com  $|V|$  elementos;
4:    $p(1) = v;$  // Índice do vértice atual
5:    $i = 1;$  // Índice do vértice atual na lista renumerada
6:    $j = 1;$ 
7:   enquanto  $i < |V|$  faça
8:     para vértice  $u \in adj(G, p(j))$  não renumerado, em ordem crescente de grau faça
9:        $i = i + 1$ 
10:       $p(i) = u;$ 
11:       $j = j + 1$ 
   retorne  $p$ 

```

---

Isso porque, no algoritmo de Cuthill-McKee original, são geradas estruturas de nível de todos os vértices com grau mínimo e, então, aquele vértice cuja estrutura possui a menor largura é escolhido como vértice inicial (CUTHILL; MCKEE, 1969). A heurística proposta de Gibbs, Poole e Stockmeyer (GPS) busca remediar essa situação.

A heurística GPS pode ser dividida em três etapas. Na primeira etapa, são escolhidos os vértices iniciais. Em vez de selecionar todos os vértices de grau mínimo, a heurística inicia por um par de vértices pseudo-periféricos. As estruturas de nível desses vértices são utilizadas como ponto de partida. Como a excentricidade de um vértice pseudo-periférico é próxima do diâmetro do grafo, sua estrutura de níveis enraizada terá mais vértices e, conseqüentemente, menor largura de nível do que a de um vértice de menor excentricidade.

Na segunda etapa da heurística GPS, as estruturas de nível do passo anterior são combinadas. Tenta-se reduzir a largura de nível combinada, de modo que ela tenha, no máximo, a menor largura de nível entre as estruturas de nível enraizadas de  $v$  e de  $u$ , como explicam Oliveira e Chagas (2014).

Na última etapa da heurística, acontece a renumeração da estrutura de nível da etapa anterior, em ordem inversa dos níveis. Para uma revisão aprofundada sobre métodos heurísticos para o problema de redução de largura de banda de matrizes, recomenda-se a leitura de Oliveira e Chagas (2014).

### 3.2 Heurísticas recentes baseadas em meta-heurísticas para o PRLB

Em problemas de otimização combinatória, como os problemas de reduções de largura de banda ou de *profile* de matrizes, métodos meta-heurísticos são constantemente empregados.

Uma meta-heurística, como explica Blum e Roli (2003), pode ser vista como arcabouço algorítmico genérico que pode ser aplicado a diferentes problemas de otimização, e que exige relativamente poucas modificações para torná-la adaptada a um problema específico. Como exemplos de meta-heurísticas, podem ser citados Algoritmos Genéticos, *Simulated Annealing*, GRASP, *Ant Colony Optimization*, *Particle Swarm Optimization*, entre várias outras.

Um dos primeiros trabalhos relevantes a aplicar um método baseado em meta-heurísticas no PRLB foi o de Martí et al. (2001), com a meta-heurística *Tabu Search* (TS-band). Apesar de ter custo computacional muito mais alto em comparação ao algoritmo GPS (GIBBS; POOLE JR; STOCKMEYER, 1976), as larguras de banda obtidas pelo TS-Band (MARTÍ et al., 2001) foram levemente superiores.

A meta-heurística GRASP com *Path Relinking* (GRASP-PR) foi utilizada no trabalho de Piñana et al. (2004), no qual se obtiveram soluções com qualidade comparável às encontradas pela heurística TS-band. Contudo, a técnica de *Path Relinking* usada para melhorar as soluções encontradas aumenta ainda mais o tempo de processamento, deixando a heurística mais lenta que as heurísticas TS-band (MARTÍ et al., 2001) e GPS (GIBBS; POOLE JR; STOCKMEYER, 1976).

Implementações de heurísticas baseadas em meta-heurísticas incorporam, em geral, alguma forma de busca local. Nos trabalhos com algoritmos genéticos (GA) de Lim, Brian e Xiao (2003) e Lim, Rodrigues e Xiao (2006), com *Particle Swarm Optimization* (PSO) de Lim, Lin e Xiao (2007) e com *Ant Colony Optimization* (ACO) de Lim et al. (2006), a busca local *Hill Climbing* (HC) é utilizada para essa finalidade. As três heurísticas propostas conseguiram soluções melhores quando comparados às heurísticas GRASP-PR, TS-band e GPS, sendo mais lentas apenas que esta última.

Alguns autores propuseram soluções variadas para o PRLB utilizando programação genética (KOOHESTANI; POLI, 2010), autômatos inteligentes (MAMAGHANI; MEYBODI, 2011), *Scatter Search* (SS) (CAMPOS; PIÑANA; MARTÍ, 2011), abordagem evolucionária (ISAZADEH; IZADKHAH; MOKARRAM, 2012) e algoritmos genéticos (POP; MATEI; COMES, 2014). Essas abordagens não se destacam na resolução do PRLB, seja por apresentarem altos tempos de execução nos testes mostrados nessas publicações ou por serem comparados a métodos já superados na literatura, como mostrado em Chagas e Oliveira (2015).

Algoritmos baseados em colônias de formigas foram utilizados por Pinteá, Crişan e Chira (2010), Pinteá, Crişan e Chira (2012) e Czibula et al. (2013). Esses trabalhos não tiveram

destaque porque os autores não compararam seus resultados com os de outras heurísticas disponíveis para o PRLB ((PINTEA; CRIŞAN; CHIRA, 2010; PINTEA; CRIŞAN; CHIRA, 2012)) ou utilizaram uma quantidade muito pequena de instâncias ((CZIBULA et al., 2013)) nos testes apresentados.

A meta-heurística *Simulated Annealing* (SA) foi utilizada no PRLB por Rodriguez-Tello, Hao e Torres-Jimenez (2008). A heurística proposta, SA- $\sigma$ , obteve resultados comparáveis aos das heurísticas TS-band e GRASP-PR, mas foi superada pelas heurísticas propostas nos trabalhos de (MLADENOVIC et al., 2010) e (LIM; RODRIGUES; XIAO, 2006), como mostram (CHAGAS; OLIVEIRA, 2015). Recentemente, uma nova abordagem baseada na meta-heurística SA, denominada *Dual Representation Simulated Annealing* (DRSA) (TORRES-JIMENEZ et al., 2015), alcançou reduções de largura de banda melhores dos que as das heurísticas com que foi comparada ((PIÑANA et al., 2004; RODRIGUEZ-TELLO; HAO; TORRES-JIMENEZ, 2008; MLADENOVIC et al., 2010)).

No trabalho de Koohestani e Poli (2011), é apresentada uma abordagem baseada em hiper-heurísticas com programação genética (KP-Band-GL) para a solução do PRLB. A heurística KP-Band-GL é uma variação da busca em largura, em que os vértices são renumerados de acordo com uma fórmula específica gerada por programação genética. Apesar de os testes apresentados serem menos extensos que os apresentados nos trabalhos que utilizam as meta-heurísticas VNS (MLADENOVIC et al., 2010) e FNC-HC (LIM; RODRIGUES; XIAO, 2003; LIM; RODRIGUES; XIAO, 2004; LIM; RODRIGUES; XIAO, 2007), o custo computacional alcançado foi razoavelmente baixo nas instâncias testadas, com qualidade de solução também razoável. Em Oliveira, Bernardes e Chagas (2016), a heurística KP-Band-GL é apontada como uma das melhores heurísticas de baixo custo computacional para redução de largura de banda.

### **3.2.1 *Fast Node Centroid with Hill Climbing***

A busca local *Hill Climbing* foi utilizada com resultados notáveis na heurística *Fast Node Centroid with Hill Climbing* (FNC-HC) (LIM; RODRIGUES; XIAO, 2003; LIM; RODRIGUES; XIAO, 2004; LIM; RODRIGUES; XIAO, 2007). A heurística FNC-HC é uma variação da heurística *Node Centroid with Hill Climbing* (NC-HC), dos mesmos autores (LIM; RODRIGUES; XIAO, 2003), em que certos parâmetros do algoritmo são ajustados automaticamente de acordo com a matriz de entrada.

A abordagem utilizada pelo FNC-HC consiste em aplicar uma estratégia global de ajustes de vértices, ou nós, que tenta aproximar os vértices com maior largura de banda dos vértices na sua vizinhança. Essa estratégia é aliada a uma busca com *Hill Climbing*, para se encontrar mínimos locais. O algoritmo pode ser dividido em três etapas: inicialização, ajustes de nós e busca local.

A etapa de inicialização da heurística FNC-HC utiliza uma busca em largura partindo de um vértice aleatório. Essa abordagem gera uma estrutura de níveis enraizada que, embora não seja tão boa quanto a ordenação de Cuthill-McKee, por exemplo, é boa o bastante para garantir um bom ponto de partida.

Para o entendimento da etapa de ajustes de nós da heurística FNC-HC, é preciso definir o conceito de *vértice crítico*. A heurística define o diâmetro de um vértice  $v \in V$  como  $diam_{(u,v) \in E}(s(v)) = \max |s(u) - s(v)|$ , em que  $s(v)$  é a numeração do vértice  $v$  na solução atual. Um vértice  $v$  é denominado crítico se  $diam(s(v)) = \beta_i(G)$ , isto é, se seu diâmetro é igual à largura de banda do grafo. Ainda, um vértice  $v$  é denominado  $\lambda$ -crítico se  $diam(s(v)) = \lambda \beta_i(G)$ , para valores de  $\lambda \in [0, 1]$  próximos de 1.

A etapa de ajuste de nós tenta reduzir o diâmetro dos vértices  $\lambda$ -críticos aproximando-os dos vértices de seu feixe. O feixe  $F_\lambda(v)$  de um vértice  $v$  é  $F_\lambda(v) = adj(G, v) \cap \{u : (\{u, v \in E\}) |s(u) - s(v)| > \lambda \beta_i(G)\} \cup v$ . Para realizar a aproximação dos vértices  $\lambda$ -críticos do centróide do seu feixe, aplica-se um peso  $w(v)$  a cada vértice, tal que:

$$w(v) = \sum_{u \in F_\lambda(v)} s(u) / F_\lambda(v).$$

Na etapa de busca local, com *Hill Climbing*, a heurística FNC-HC identifica os vértices críticos e realiza trocas entre a numeração do vértice crítico atual e a de um vértice na vizinhança, na esperança de melhorar a qualidade da solução. Os autores da heurística consideraram que houve melhora na solução se houve redução na largura de banda do grafo ou diminuição do número de vértices críticos.

A busca local pode ser um processo demorado caso inclua muitos vértices candidatos. É preciso, assim, definir uma vizinhança limitada na qual realizar as trocas. Em Martí et al. (2001) são apresentados os conceitos de numeração mínima e máxima de um vértice como

$$min(v) = \min\{s(u) | u, v \in E\},$$

$$\max(v) = \max\{s(u) | u, v \in E\},$$

respectivamente. Isto é, eles representam a maior e a menor numeração entre aquelas dos vértices vizinhos a  $v$ . Assim, a numeração ideal  $best_s(v)$  de um vértice é:

$$best_s(v) = \lfloor \max(v) + \min(v) / 2 \rfloor.$$

Na heurística FNC-HC, a vizinhança reduzida de um vértice  $v \in V$  na qual se realizam as buscas é definida como:

$$N'(v) = \{u : ||best_s(v) - s(u)| < |best_s(v) - s(v)||\}, \quad (3.1)$$

isto é, ela considera todos os vértices  $u$  cuja numeração  $s(u)$  esteja mais próxima de  $best_s(v)$  do que  $s(v)$ .

Os autores da heurística FNC-HC a compararam às heurísticas GPS (GIBBS; POOLE JR; STOCKMEYER, 1976), GRASP-PR (PIÑANA et al., 2004), TS-band (MARTÍ et al., 2001), PSO (LIM; LIN; XIAO, 2007), *Node Shifting* (NS) (LIM; RODRIGUES; XIAO, 2006) e GA-HC (LIM; BRIAN; XIAO, 2003), obtendo maiores reduções de banda que todas essas. Além disso, os autores indicam que a heurística FNC-HC foi muito mais rápida que as heurísticas baseadas em meta-heurísticas citadas e quase tão rápida quanto o algoritmo GPS (GIBBS; POOLE JR; STOCKMEYER, 1976). Esse último resultado, em especial, coloca-a numa categoria que torna seu uso interessante em aplicações reais, por apresentar um bom equilíbrio entre velocidade de execução e redução de largura de banda, como notam Chagas e Oliveira (2015).

### 3.2.2 *Variable Neighborhood Search*

Um dos trabalhos que mais se destaca na redução de largura de banda de matrizes utiliza a meta-heurística *Variable Neighborhood Search* (VNS) (MLADENOVIC et al., 2010). A ideia básica do método é explorar a vizinhança de um mínimo local aumentando-a de forma sistemática, até que uma solução melhor seja encontrada. A cada vez que a vizinhança  $N_k(x)$  de uma solução é expandida, vários pontos em seu interior são considerados como pontos de partida para uma busca local. Uma nova solução na vizinhança é escolhida se for melhor que a solução atual, conforme explicam Mladenovic et al. (2010).

O laço principal da meta-heurística VNS pode ser dividido em três etapas: perturbação, busca local e mudança de vizinhança. Para a inicialização do algoritmo, os autores utilizam

uma estrutura de nível enraizada a partir de um vértice  $v$  escolhido aleatoriamente. Os vértices de cada nível da estrutura são numerados de maneira aleatória. Para controlar o tamanho da vizinhança, utiliza-se um parâmetro  $k$ , inicialmente definido como  $k_{min}$ .

A etapa de perturbação no algoritmo VNS leva de uma solução  $S$  para uma solução  $S'$  a partir da troca da numeração de vértices. Os autores apresentam duas formas de perturbação da solução, além de um procedimento que determina qual delas será aplicada de acordo com o estado do algoritmo. Na primeira forma de perturbação, um vértice aleatório é selecionado no conjunto  $K \in V$ , tal que  $K = \{v | \beta(f, v) \geq B'\} \geq k$ , em que  $f$  é a numeração, ou solução, atual. O parâmetro  $B'$  é definido de forma que o conjunto  $K$  contenha apenas os vértices com maior largura de banda. Um vértice  $u \in V$  é selecionado, bem como um vértice  $v$  crítico adjacente a  $u$ , tal que  $(|f(u) - f(v)| = \beta(G))$ . Então, é encontrado um vértice  $w \in V$  para ser trocado, de modo que a diferença máxima entre  $u$  e  $v$  seja mínima.

A segunda forma de perturbação do VNS utiliza um procedimento de rotação de duas vizinhanças proposto por Rodriguez-Tello, Hao e Torres-Jimenez (2008). Nesse procedimento, são definidos índices  $b$  e  $e$ , e um índice intermediário  $m$  (de *begin*, *end* e *middle*). As numerações do vértices no intervalo  $[b, e]$  são deslocadas  $m$  posições. Como exemplo, a numeração  $s(b)$  passará a ser a posição  $s(b + m)$ . O procedimento de perturbação a ser utilizado pela meta-heurística VNS depende do valor do parâmetro  $k$ . Se  $k < k'_{max}$ , isto é, se o tamanho da vizinhança for menor que o máximo definido, aplica-se o primeiro procedimento, do contrário, aplica-se o segundo. Segundo os autores, o segundo procedimento é necessário pois, em algumas situações, o algoritmo pode ficar preso em um mínimo local mais profundo.

Para o procedimento de busca local, a meta-heurística VNS lança mão do mesmo procedimento utilizado pela heurística FNC-HC, realizando trocas na vizinhança reduzida definida na equação 3.1. No VNS, os vértices no conjunto de candidatos  $N'(v)$  são trocados um a um em ordem crescente de  $|mid(v) - f(u)|$  até que a solução melhore.

Por fim, deve-se decidir se a solução  $S''$  obtida após a busca local será aceita como solução atual ou não. No VNS, a solução  $S''$  é aceita se  $\beta(depois) < \beta(antes)$ . Caso  $\beta(depois) = \beta(antes)$ , a solução  $S''$  é aceita se o número de vértices críticos em  $S''$  for menor que antes.

Os autores do trabalho compararam a abordagem proposta com as heurísticas TS-band (MARTÍ et al., 2001), GRASP-PR (PIÑANA et al., 2004), SS (CAMPOS; PIÑANA; MARTÍ, 2011), GA (LIM; BRIAN; XIAO, 2003), NS (LIM; RODRIGUES; XIAO, 2006) e SA- $\sigma$  (RODRIGUEZ-TELLO; HAO; TORRES-JIMENEZ, 2008), obtendo resultados melhores que

todas elas. Além disso, o algoritmo baseado na meta-heurística VNS proposto (VNS-Band) apresentou custo computacional muito mais baixo que as outras heurísticas. Por aliar velocidade e qualidade de solução, a heurística VNS-Band poderia ser considerada como possível estado da arte para a resolução do PRLB, segundo Chagas e Oliveira (2015).

### 3.3 Métodos heurísticos aplicadas ao problema de redução de *profile*

Nesta seção, apresentam-se as heurísticas para redução de *profile* NSloan e MPG. Diversas outras heurísticas foram propostas para o problema, como levantado por Bernardes e Oliveira (2015). Porém, como apontado por Oliveira, Bernardes e Chagas (2016), estas duas heurísticas destacam-se por serem as heurísticas de baixo custo mais promissoras para redução de *profile*.

#### 3.3.1 Heurística NSloan

Uma modificação da heurística de Sloan (SLOAN, 1986), a heurística NSloan (*Normalized Sloan*), foi proposta por Kumfert e Pothen (1997). A heurística de Sloan original para redução de *profile* pode ser dividida em duas etapas. Inicialmente, dois vértices pseudo-periféricos  $s, e \in V$  são encontrados. O algoritmo mantém controle sobre apenas um vértice de cada grau entre os encontrados, limitando a montagem de estruturas de nível. Em seguida, dada uma função de priorização, os vértices do grafo são renumerados partindo do vértice inicial  $s$ . Os demais vértices são renumerados de acordo com uma prioridade definida em relação a seu grau distância do vértice final  $e$ .

Uma das diferenças introduzidas a heurística NSloan foi a utilização do algoritmo de Duff, Reid e Scott (DUFF; REID; SCOTT, 1989) para se encontrar os vértices pseudo-periféricos na etapa inicial do algoritmo. Outra modificação é que o algoritmo NSloan utiliza uma normalização para a função de priorização originalmente utilizada no algoritmo de Sloan. Uma terceira modificação proposta foi na estrutura de dados que armazena os vértices candidatos a renumeração em cada iteração. Kumfert e Pothen (1997) propuseram a utilização de uma heap binária e mostraram que essa estrutura apresenta um ganho considerável no tempo de execução em comparação à estrutura de lista não ordenada utilizada no algoritmo de Sloan.

Os autores da heurística NSloan mostraram que seu algoritmo apresenta melhores resultados que as heurísticas RCM e Sloan. Além disso, Oliveira, Bernardes e Chagas (2016) comparou-a com diversas outras heurísticas e apontou a heurística NSloan como uma das me-

lhores heurísticas de baixo custo para redução de *profile* tanto para matrizes simétricas quanto para assimétricas.

### 3.3.2 Heurística Sloan-MGPS

Outra variação da heurística de Sloan (SLOAN, 1986) foi proposta por Reid e Scott (1999), chamada de Sloan-MGPS. Assim como na heurística NSloan (KUMFERT; POTHEN, 1997), as alterações na heurística original envolvem o método utilizado para se encontrar vértices pseudo-periféricos e a estrutura de dados que armazena os vértices a serem renumerados.

Durante a etapa de seleção de vértices no último nível da estrutura de nível enraizada atual, Reid e Scott (1999) propuseram a escolha de, no máximo, cinco vértices no último nível, de maneira a limitar a quantidade de estruturas de níveis enraizadas a serem construídas no passo seguinte. Após a escolha dos vértices iniciais  $s$  e  $e$ , os autores propuseram que o vértice de maior excentricidade entre eles fosse eleito o vértice inicial  $s$ . Devido à semelhança com o algoritmo utilizado na primeira etapa da heurística GPS (GIBBS; POOLE JR; STOCKMEYER, 1976), o método proposto por Reid e Scott (1999) foi chamado MGPS (*modified Gibbs-Poole-Stockmeyer*).

Assim como sugerido por Kumfert e Pothen (1997), a heurística Sloan-MGPS propõe a utilização de uma *heap* binária para armazenar os vértices candidatos a renumeração em cada iteração. Reid e Scott (1999) observam que, para instâncias grandes, como as utilizadas neste trabalho, essa estrutura é mais eficiente que a estrutura de listas empregada na heurística de Sloan (SLOAN, 1986).

### 3.3.3 Heurística MPG

A heurística de Medeiros, Pimenta e Goldenberg (MPG) (MEDEIROS; PIMENTA; GOLDENBERG, 1993) para redução de *profile* utiliza a primeira fase da heurística de Sloan para encontrar dois vértices pseudo-periféricos. Em seguida, na etapa de renumeração, são mantidas duas filas de prioridades,  $t$  e  $q$ . A fila  $t$  contém vértices candidatos a renumeração e a fila  $q$  contém os vértices que estão em  $t$  ou que são elegíveis para inserção em  $t$ . O grau de um vértice  $v$  é definido como o número de vértices adjacentes a  $v$  que ainda não foram renumerados e que não pertençam à fila de prioridades  $q$ . Um vértice  $v$  é inserido em  $q$  de acordo com uma função de priorização específica. A cada iteração, os vértices com grau atual maior que 1

são inseridos na fila  $t$  e os vértices com grau igual 0 são removidos e renumerados em ordem crescente de grau.

Os autores da heurística MPG a compararam com as heurísticas RCM, GPS e Sloan, conseguindo melhores resultados que estas, no geral. Ainda, a heurística MPG foi apontada com uma das melhores heurísticas de baixo custo para redução de *profile* para matrizes simétricas e a melhor heurística para redução de *profile* para matrizes assimétricas por Oliveira, Bernardes e Chagas (2016), em comparação com diversas outras.

### 3.4 Heurísticas baseadas em meta-heurísticas para o problema de redução de *profile*

Por se tratar de um problema extremamente relacionado, algoritmos baseados em meta-heurísticas são também aplicados, com frequência, no problema de redução de *profile*, como mostra o trabalho de Bernardes e Oliveira (2015). Como exemplos de aplicações de meta-heurísticas para redução de *profile* pode-se citar a utilização da meta-heurística *Ant Colony Optimization* nos trabalhos de Kaveh e Sharafi (2007), Kaveh e Sharafi (2008), da meta-heurística *Charged System Search* em Kaveh e Sharafi (2012), da meta-heurística *Scatter Search* no trabalho de Sánchez-Oro et al. (2015) e de métodos baseados em hiper-heurísticas com programação genética nos trabalhos de Koohestani e Poli (2012a), Koohestani e Poli (2012b), Koohestani e Poli (2014), para redução de *profile*.

Entretanto, os algoritmos baseados em meta-heurísticas citados não se encontram entre os possíveis melhores métodos heurísticos para o problema de redução de *profile*, conforme apontam Bernardes e Oliveira (2015). Alguns desses trabalhos foram superados por outros trabalhos que já não figuram entre os estados da arte, como é o caso dos trabalhos de Koohestani e Poli (2012b) (superado pelo algoritmo de (SLOAN, 1986)), de Kaveh e Sharafi (2007) (superado pelos algoritmos de (KING, 1970) e de Sloan (1986)), de Kaveh e Sharafi (2008) (superado pela heurística proposta por Kaveh e Sharafi (2012)) e Kaveh e Sharafi (2012) (superado pelo algoritmo de Gibbs, Poole Jr e Stockmeyer (1976)) (BERNARDES; OLIVEIRA, 2015). Outros autores testaram as heurísticas propostas apenas contra métodos já superados, como no trabalho de Koohestani e Poli (2012a) (comparado com o método Cuthill-McKee reverso George (1971) e com o algoritmo de Gibbs (1976)), e Sánchez-Oro et al. (2015) (comparado com o método Cuthill-McKee reverso (GEORGE, 1971) e com a heurística proposta por Lewis (1994)) (BERNARDES; OLIVEIRA, 2015). Ainda, o trabalho de Koohestani e Poli (2014) apresenta tempos

de execução proibitivos na prática e, por isso, não poderia ser considerado um dos melhores métodos, na prática.

## 4. A META-HEURÍSTICA *ITERATED LOCAL SEARCH*

A meta-heurística *Iterated Local Search* (ILS) (LOURENÇO; MARTIN; STÜTZLE, 2003) é um método simples e versátil. Como o nome sugere, essa meta-heurística utiliza um procedimento de busca local que, iterado de forma sistemática, permite que o espaço de soluções do problema de otimização seja explorado criteriosamente. Neste capítulo, os conceitos básicos do funcionamento dessa meta-heurística são explicados, na Seção 4.1. Na Seção 4.2, apresenta-se um breve resumo de problemas de otimização nos quais essa meta-heurística tem sido aplicada com sucesso.

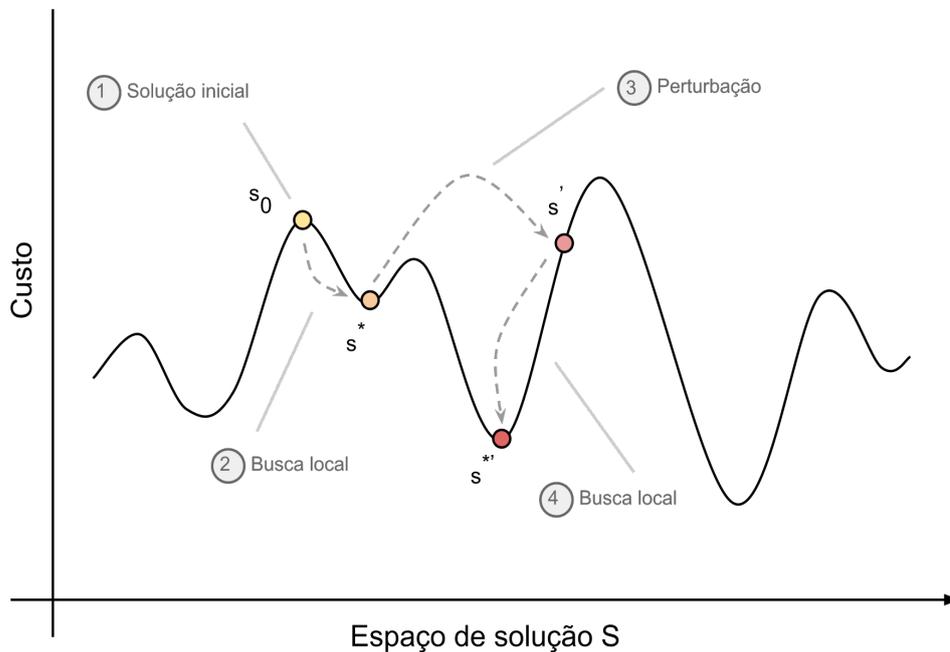
### 4.1 *Iterated Local Search*

Considere um procedimento de busca local para um problema de otimização de interesse. Seja  $C$  a função custo do problema de otimização a ser *minimizada*. O conjunto de todas as soluções candidatas  $s$  é denotado por  $S$ . O procedimento de busca local define um mapeamento de muitos para um do conjunto  $S$  para o conjunto  $S^*$  de soluções ótimas locais  $s^*$ . Após a aplicação da busca local em  $S$ , a questão que surge é: como repetir a busca local e melhorar a solução atual  $s^*$ ? A alternativa mais simples é aplicar a busca a partir de um novo ponto de partida aleatório. Porém, essa abordagem tem chances cada vez menores de dar certo, à medida que instâncias maiores são utilizadas.

A forma da meta-heurística ILS de escolher uma nova região para explorar consiste em aplicar uma perturbação à solução atual  $s^*$ , levando-a a um estado intermediário  $s'$ . Em seguida, a busca local é aplicada à solução  $s'$ , da qual se obtém uma solução  $s^{*'}$  em  $S^*$ . Se  $s^{*'}$  for aprovada em um teste de aceitação, ela se torna a solução corrente no próximo passo da busca por uma solução melhor em  $S^*$ ; do contrário,  $s^*$  é mantida como solução corrente. Desse modo, a perturbação em  $s^*$  leva a busca a uma região do espaço de busca diferente da anterior, mas que ainda retém características da região em que foi encontrada a solução corrente. As perturbações realizadas são também um mecanismo da meta-heurística ILS para evitar que o algoritmo fique preso a ótimos locais. A noção de uma iteração da meta-heurística ILS é mostrada na figura 4.1. Uma visão de alto nível da estrutura básica da meta-heurística *Iterated Local Search* é mostrada no Algoritmo 4.1.

Como visto no Algoritmo 4.1, as etapas da meta-heurística ILS são bem definidas em blocos ou módulos distintos: geração da solução inicial, procedimento de busca local, perturbação e critério de aceitação. Esses módulos podem ser trocados ou otimizados independente-

Figura 4.1 – Comportamento de uma iteração da meta-heurística *Iterated Local Search*. Dado um mínimo local  $s^*$ , aplica-se uma perturbação que leva a  $s'$ . Após efetuar a BuscaLocal em  $s'$ , obtemos  $s^{*'}$ , que será comparado a  $s^*$ .



Fonte: do autor

---

**Algoritmo 4.1** Visão alto-nível do *Iterated Local Search*

---

```

1: ITERATED LOCAL SEARCH
2:    $s_0 = \text{SoluçãoInicial}();$ 
3:    $s^* = \text{BuscaLocal}(s_0);$ 
4:   repeat
5:      $s' = \text{Perturbação}(s^*, \text{histórico});$ 
6:      $s^{*'} = \text{BuscaLocal}(s');$ 
7:      $s^* = \text{CritérioDeAceitação}(s^*, s^{*'}, \text{histórico});$ 
8:   until condição de parada atingida

```

---

mente, de acordo com as características do problema em que a meta-heurística ILS é aplicada. A seguir, cada um desses módulos é examinado.

#### 4.1.1 Solução inicial

O primeiro módulo a se considerar no desenvolvimento de uma heurística baseada na meta-heurística *Iterated Local Search* é o de geração da solução inicial  $s_0$ . Em casos nos quais o tempo de execução não é um fator limitante, a solução inicial  $s_0$  pode ser gerada de forma aleatória. Contudo, claramente, quanto melhor a solução inicial, menos passos do procedimento de busca local são necessários para se encontrar boas soluções.

As formas mais comuns de se gerar a solução inicial são a geração aleatória e a baseada em alguma heurística gulosa. Segundo Lourenço, Martin e Stützle (2003), são dois os pontos

em que uma solução  $s_0$  gerada de forma gulosa supera uma gerada de forma aleatória. Primeiro, a busca local aplicada sobre uma solução gerada de maneira gulosa gera melhores soluções  $s_0^*$ . Além disso, em média, as soluções derivadas de métodos gulosos necessitam de menos passos de busca local para convergirem e, em consequência disso, retornam em menos tempo. Para longos processamentos, Lourenço, Martin e Stützle (2003) observam que a dependência de  $s_0$  tende a ser bastante pequena, visto que em determinado momento se perderá a memória da solução inicial.

#### 4.1.2 Busca local

Em seguida, a escolha de um procedimento de busca local equilibrado é crucial para o sucesso da meta-heurística ILS. Em princípio, qualquer algoritmo de busca local disponível pode ser usado, como afirma Stützle (2006). Como esse procedimento será muito requisitado, ele deverá ser rápido e retornar boas soluções.

Apesar de geralmente ser verdade que, quanto melhor a busca local, melhor a implementação do ILS resultante, em alguns casos isso pode não acontecer. Há casos, por exemplo, em que o aumento do custo computacional é tão grande em comparação à solução fornecida que pode ser melhor aceitar procedimentos de busca local que gerem soluções de qualidade inferior, mas que possam ser iterados mais vezes, dado o mesmo tempo de execução. Em qualquer caso, deve ser avaliado o balanço entre a qualidade e a velocidade do ótimo local gerado pelo procedimento escolhido, como sugerem Lourenço, Martin e Stützle (2003).

#### 4.1.3 Perturbação

Aplicar perturbações à solução atual é a forma utilizada pelo ILS e por outras meta-heurísticas de busca local como o *Simulated Annealing* e o VNS para escapar de ótimos locais, nos quais o algoritmo pode eventualmente ficar preso. A perturbação é *complementar* à busca local e também tem a função de alterar a “vizinhança” atual e guiar a amostragem do espaço de soluções.

A *intensidade* da perturbação refere-se ao número de componentes da solução alterados pela perturbação, conforme Besten, Stützle e Dorigo (2001). Se a perturbação for muito forte, o ILS terá comportamento similar a reinícios aleatórios: pouco será mantido da solução atual e a probabilidade de se encontrar soluções melhores será pequena. Se, por outro lado, ela for

muito fraca, corre-se o risco de voltar a ótimos locais já visitados. Uma boa perturbação tem como característica não poder ser desfeita facilmente pela busca local.

Nem sempre, contudo, é possível aplicar uma perturbação de intensidade fixa. Nesses casos, a perturbação precisa ser adaptativa, ou deve incorporar alguma forma de memória à perturbação, de forma similar ao que é feito na Busca Tabu. Ainda é possível mudar a intensidade da perturbação de forma determinística durante a busca, segundo Lourenço, Martin e Stützle (2003). O comportamento do ILS neste último caso é análogo ao do VNS, em que a intensidade (vizinhança) é incrementada até que uma solução melhor seja encontrada. Em qualquer um dos casos, é preciso experimentar com valores diferentes e analisar as características do problema para se encontrar a forma perturbação que dê melhores resultados.

#### 4.1.4 Critério de aceitação

Por fim, o critério de aceitação atua, juntamente com a perturbação, no estabelecimento do equilíbrio entre diversificação e intensificação da busca. Cabe ao critério de aceitação determinar se  $s^{*}$  será aceita ou não como solução atual.

Uma abordagem comum é aceitar qualquer solução  $s^{*}$  que seja melhor que a atual  $s^*$ . Conforme explicam Lourenço, Martin e Stützle (2003), essa abordagem leva a uma grande *intensificação* da busca. O extremo oposto seria sempre aceitar a solução intermediária  $s^{*}$ , a despeito de seu custo. Essa estratégia é equivalente a um passeio aleatório em  $S$ , que faz com que a perturbação seja sempre aplicada na solução mais recente.

Técnicas híbridas para o critério de aceitação são comuns. Em Lourenço, Martin e Stützle (2003), é possível encontrar referências ao uso de esquemas similares ao da busca tabu, que tornam o critério de aceitação sensível a soluções previamente visitadas, e ao *Simulated Annealing*, que permitem que a intensificação ou diversificação seja balanceada de maneira mais inteligente durante o ciclo de execução do algoritmo.

## 4.2 Aplicações da meta-heurística ILS em problemas de otimização combinatória

Como ressaltam Lourenço, Martin e Stützle (2003), apesar da simplicidade conceitual, o ILS tem levado a resultados que são estado da arte em problemas de otimização combinatória. Há exemplos de trabalhos que aplicam a meta-heurística ao problema do caixeiro viajante (AP- PLEGATE; COOK; ROHE, 2003; STÜTZLE; HOOS, 2002), coloração de grafos (PAQUETE; STÜTZLE, 2002), particionamento de grafos (MARTIN; OTTO, 1995), entre outros ((SCHI-

AVINOTTO; STÜTZLE, 2004; VILLANUEVA et al., 2010; VALDEZ; MEDINA, 2012; SAKURABA et al., 2015)). Em particular, a meta-heurística ILS tem sido aplicada com êxito em problemas de alocação de recursos (BESTEN; STÜTZLE; DORIGO, 2001; CONGRAM; POTTS; VELDE, 2002; STÜTZLE, 2006; VANSTEENWEGEN et al., 2009) e roteamento de veículos (PENNA; SUBRAMANIAN; OCHI, 2013; IBARAKI et al., 2008; HASHIMOTO; YAGIURA; IBARAKI, 2008). Recomenda-se a leitura de Lourenço, Martin e Stützle (2010) discussões sobre diversas outras aplicações da meta-heurística em problemas de otimização.

## 5. APLICAÇÃO DA META-HEURÍSTICA *ITERATED LOCAL SEARCH* AOS PROBLEMAS DE REDUÇÕES DE LARGURA DE BANDA E DE *PROFILE* DE MATRIZES

Neste capítulo é explicada a metodologia utilizada no desenvolvimento da heurística baseada na meta-heurística *Iterated Local Search* para os problemas de reduções de largura de banda e de *profile* de matrizes proposta neste trabalho. A estrutura básica da meta-heurística ILS consiste de quatro módulos: busca local, solução inicial, perturbação e critério de aceitação. Na Seção 5.1, é explicado como cada um desses módulos foi desenvolvido na heurística proposta e a Seção 5.2 apresenta uma visão geral do algoritmo com a integração de todos os módulos. A Seção 5.3 fecha este capítulo incluindo detalhes de implementação do algoritmo.

### 5.1 Desenvolvimento da heurística baseada na meta-heurística *Iterated Local Search*

O desenvolvimento da heurística baseada na meta-heurística ILS para redução de largura de banda e de *profile* proposta neste trabalho iniciou-se com a identificação das técnicas já aplicadas na literatura que poderiam ser incorporadas a cada uma dessas etapas. Como resultado, a solução desenvolvida utiliza elementos consolidados em outras heurísticas aplicadas no problema, expandindo-as para outras utilizações e introduzindo novas ideias.

No algoritmo proposto, a ordenação inicial é gerada por uma busca em largura partindo de um vértice pseudo-periférico. Um procedimento de busca local baseado no *HillClimbing* de Lim, Rodrigues e Xiao (2006), Lim, Rodrigues e Xiao (2007) é aplicado. A perturbação na solução atual é alcançada por i) uma estratégia de rotular os vértices do grafo de acordo com seu "valor ideal", baseado no conceito introduzido por Martí et al. (2001) no contexto de buscas locais e ii) outra que tenta tornar mais próximas as numerações dos vértices com maior largura de banda e seus adjacentes. Todo o processo é repetido até que um número máximo de iterações seja atingido ou até que seja identificado que a solução tenha estagnado em um ótimo local. Nas seções seguintes, cada etapa do algoritmo proposto é explicada com mais detalhes.

#### 5.1.1 Solução inicial

Na etapa de inicialização, deseja-se gerar uma ordenação que seja um bom ponto de partida para o algoritmo. Em geral, quanto melhor a solução inicial  $s_0$ , melhores serão os resultados obtidos pelo algoritmo, ou ainda, menor o tempo necessário para sua convergência,

como afirmam Lourenço, Martin e Stützle (2003, p.11). Neste trabalho, a solução inicial é obtida por uma meio de busca em largura no grafo. Essa abordagem é amplamente consolidada na literatura (CUTHILL; MCKEE, 1969; GIBBS; POOLE JR; STOCKMEYER, 1976; LIM; LIN; XIAO, 2007; LIM; RODRIGUES; XIAO, 2007; MLADENOVIC et al., 2010) e baseia-se na observação de que, em uma boa numeração, os vértices adjacentes devem ter rótulos próximos.

Em geral, a escolha de um vértice aleatório como ponto de partida para a busca em largura fornece resultados satisfatórios, ou pelo menos melhores do que a ordenação original. Como observado em Gibbs, Poole Jr e Stockmeyer (1976), a escolha de um vértice periférico é ainda melhor como ponto de partida. Isso porque a estrutura de nível correspondente tende a ter mais níveis de profundidade e um menor número de vértices em um mesmo nível. Isso implica em vértices de níveis posteriores recebendo rótulos mais próximos aos dos vértices do nível anterior e, conseqüentemente, em menores larguras de banda. Algoritmos para se encontrar vértices periféricos são caros e, na prática, estes são substituídos por vértices pseudo-periféricos, que podem ser encontrados com custo menor.

Neste trabalho, empregou-se o algoritmo de George e Liu (GEORGE; LIU, 1979) para se encontrar um vértice pseudo-periférico  $v$ . O algoritmo original de George e Liu (GEORGE; LIU, 1979) inicia-se com a escolha de um vértice  $v \in V$  qualquer. Em seguida, a estrutura de nível enraizada  $\mathcal{L}(v)$  é gerada. Um vértice  $u$  de grau mínimo no último nível da estrutura  $\mathcal{L}(v)$  é selecionado. Se a excentricidade de  $\mathcal{L}(u)$  for maior que a de  $\mathcal{L}(v)$ , o vértice  $u$  passa a ser o atual e o processo é repetido. Do contrário, o algoritmo termina com o vértice  $v$  como o vértice pseudo-periférico de saída.

Ainda, foi proposta uma variação do algoritmo de George e Liu em que considera-se o vértice  $u$  como atual se a excentricidade de  $\mathcal{L}(u)$  for maior que a de  $\mathcal{L}(v)$ , ou se a excentricidade de  $\mathcal{L}(u)$  for igual a de  $\mathcal{L}(v)$ , mas o grau de  $u$  for menor que o grau de  $v$ . Na solução inicial utilizada no algoritmo desenvolvido, são geradas numerações com vértices iniciais dados pelo algoritmo de George e Liu e pela variação proposta. A numeração escolhida como solução inicial é aquela que apresentar menor largura de banda ou *profile*, dependendo do que se deseja reduzir. Ainda que seja mais custoso que a escolha de um vértice aleatório, observou-se que, em geral, o tempo gasto na solução inicial é ínfimo perto do tempo de execução total da heurística e que o custo extra é compensado por soluções melhores.

### 5.1.2 Perturbação

No trabalho utilizando a heurística TS-Band de Martí et al. (2001), dada uma numeração  $f$ , os seguintes valores são definidos:

$$f_{max}(v) = \max\{f(u) | (u, v) \in E\},$$

$$f_{min}(v) = \min\{f(u) | (u, v) \in E\}.$$

Isto é,  $f_{max}(v)$  e  $f_{min}(v)$  são os rótulos dos vértices adjacentes a um vértice  $v$  com maior e menor numeração, respectivamente. O valor  $mid(v)$  é dado por

$$mid(v) = \left\lfloor \frac{f_{max}(v) + f_{min}(v)}{2} \right\rfloor. \quad (5.1)$$

Em trabalhos anteriores (MARTÍ et al., 2001; LIM; RODRIGUES; XIAO, 2007; MLADENOVIC et al., 2010), o valor  $mid(v)$  tem sido utilizado para se encontrar um conjunto de vértices candidatos a serem permutados durante um procedimento de busca local *HillClimbing*. No contexto de reduções de largura de banda e de *profile*,  $mid(v)$  pode ser visto como o valor "ideal" de rótulo que  $v$  pode assumir, dada uma numeração  $f$ , que minimize a diferença entre seu rótulo e os rótulos de seus vértices adjacentes.

O procedimento de perturbação base do algoritmo proposto,  $midOrdering(f)$ , se utiliza desse princípio, como mostrado no Algoritmo 5.1. Para cada vértice  $v$  do grafo,  $v.mid$  é calculado como definido em 5.1, sem a função *piso*. A adição de um valor aleatório  $r$  entre  $-k_{rand}$  e  $k_{rand}$  acrescenta variação entre as execuções e torna o procedimento menos propenso a estagnar após execuções seguidas. Em seguida, os vértices são ordenados em ordem crescente em  $v.mid$ . Os empates são resolvidos de forma arbitrária. Note que  $v.mid$  consiste de um valor real e, dessa forma, o número de empates entre rótulos tende a ser menor.

---

#### Algoritmo 5.1 Pseudocódigo da função de perturbação $midOrdering(f)$

---

```

MIDORDERING(F)
  Entrada: int  $k_{rand}$ 
  para  $v = 1$  to  $|V|$  faça
     $r = random(-k_{rand}, k_{rand});$ 
     $v.mid = ((v.f_{max} - v.f_{min})/2) + r;$ 
  Ordene  $v \in V$  de acordo com  $v.mid$ 

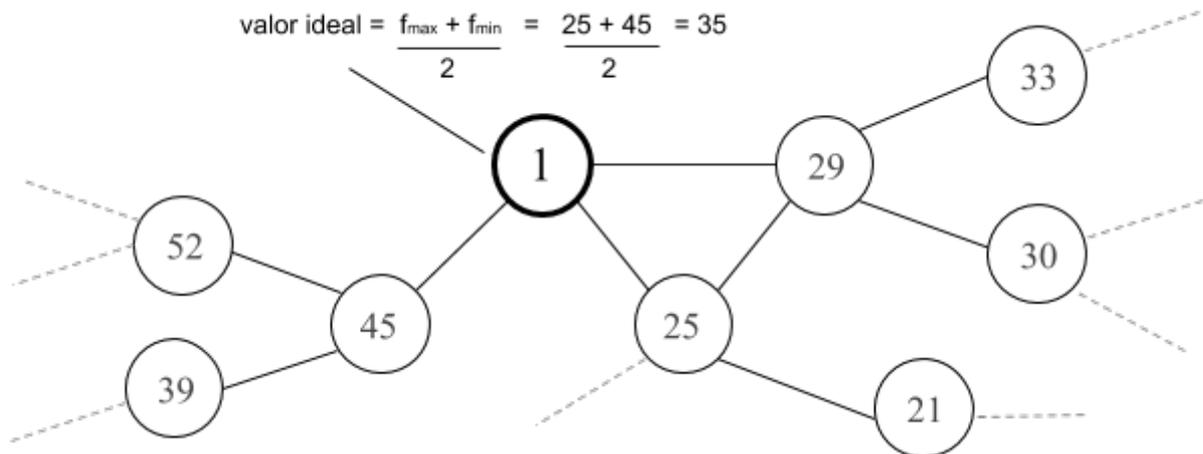
```

---

Como exemplo, tome o cenário representado na Figura 5.1, em que o vértice em destaque possui um rótulo distante dos de seus adjacentes, bem como dos demais vértices na vi-

zinhança. O valor ideal para o vértice de rótulo 1, em destaque, é mostrado, de acordo com a equação 5.1. No exemplo, os valores de  $v.mid$  calculados pelo procedimento  $midOrdering(f)$  para os vértices mostrados serão possivelmente próximos ao do vértice em destaque. Dessa forma, após a ordenação dos vértices (linha 6 do Algoritmo 5.1), a diferença entre seus rótulos será menor e, possivelmente, a largura de banda ou *profile* do grafo.

Figura 5.1 – Fragmento de um grafo destacando o valor “ideal” do vértice de rótulo 1. São tomados o maior e menor rótulos adjacentes, 45 e 25, respectivamente, e sua soma é dividida por 2.



A ideia do procedimento  $midOrdering()$  é aproveitar das características do problema para causar uma perturbação na solução intermediária  $s'$  enquanto a leva para uma nova solução  $s''$  que seja possivelmente melhor que  $s'$ . Como pode ser presumido, nas primeiras chamadas a  $midOrdering()$ , a perturbação resultante pode ser bastante intensa. Contudo, após um certo número de iterações, esse efeito tende a ser cada vez menor, à medida que os vértices se "asentam", ou se "estabelecem" em torno de certas regiões do espaço de soluções. Após cada chamada a  $midOrdering()$ , os valores de largura de banda e *profile* do grafo são atualizados.

Nos testes preliminares, contudo, foi verificado que para algumas instâncias o procedimento de perturbação  $midOrdering()$  pode não produzir melhorias em comparação à ordenação original. Em outros casos, os ganhos são bastante significativos num primeiro momento, mas tendem a estagnar após certo número de iterações. Para contornar essas situações, um outro procedimento de perturbação é utilizado.

Assim como o procedimento  $midOrdering()$ , o segundo procedimento procura perturbar a solução atual enquanto a leva para uma solução intermediária possivelmente melhor. Considere as seguintes definições. Um vértice  $v$  é dito crítico se  $B_f(v) = B_f(G)$ , onde  $B_f(v)$  é a largura de banda de  $v$  e  $B_f(G)$  é a largura de banda do grafo, dada uma ordenação (ou solução)

$f$ . Um vértice  $v$  é dito quase-crítico, ou  $\lambda$ -crítico, se  $B_f(v) \geq \alpha_{crit} * B_f(G)$ , e  $0 < \alpha_{crit} < 1$ . O procedimento `groupCritical()` tenta diminuir a distância entre os vértices  $\lambda$ -críticos e seus vértices adjacentes, aproximando seus rótulos como explicado no Algoritmo 5.2.

---

**Algoritmo 5.2** Pseudocódigo da função de perturbação `groupCritical(f)`

---

```

1: GROUPCRITICAL(MULT)
2:   Entrada: conjunto de rótulos  $\lambda$ -críticos, parâmetro multiplicador  $mult$ 
3:   para  $c \in \lambda - criticos$  faça
4:     para  $a \in adj(c)$  faça
5:        $inc = abs(c.degree - a.degree) * (a.label - c.label) / (c.degree + a.degree)$ ;
6:       se  $a.mid < crit.mid$  então
7:          $inc = inc * -1$ 
8:          $inc = inc * mult$ 
9:          $crit.mid = crit.mid + inc$ ;
10:  Ordene  $v \in V$  de acordo com  $v.mid$ 

```

---

O valor  $inc$  da linha 5 define um incremento que é adicionado ao valor  $crit.mid$ . Idealmente, esse incremento vai tornar o rótulo do vértice  $v$  mais próximo dos rótulos dos vértices  $\lambda$ -críticos adjacentes a  $v$ . Em seguida, os vértices são ordenados em ordem crescente de  $mid$  e renumerados. O parâmetro entrada  $mult$  é utilizado para controlar a intensidade da perturbação aplicada. Por padrão, utiliza-se  $mult = 1$ .

O conjunto de vértices  $\lambda$ -críticos de uma numeração  $f$  contém vértices que não influenciam na largura de banda  $B_f(G)$  da solução atual, mas que podem contribuir para diminuir o *profile* da nova ordenação. Ainda, como o conjunto de vértices  $\lambda$ -críticos inclui o de vértices críticos, é razoável que aproximar a numeração de seus vértices adjacentes melhore a largura de banda atual. O Algoritmo 5.3 mostra como os dois procedimentos são combinados no algoritmo proposto.

---

**Algoritmo 5.3** Pseudocódigo da função de perturbação combinada

---

```

PERTURBAÇÃO()
  Entrada: solução intermediária  $s'$ 
   $s'' = midOrdering()$ ;
   $melhorouSolucao = (s'' < s')$ ;
  se  $melhorouSolucao == false$  então
    groupCritical(1);

```

---

### 5.1.3 Busca Local

O procedimento de busca local é o núcleo da meta-heurística ILS. O procedimento utilizado no algoritmo proposto é baseado no *HillClimbing* sugerido em Lim, Rodrigues e Xiao

(2006), Lim, Rodrigues e Xiao (2007). Nesse procedimento, é definido um conjunto  $N'(v)$  com os vértices  $u$ , tal que

$$N'(v) = \{u : (v.mid - f(u)) < (v.mid - f(v))\},$$

em que  $f(v)$  é o rótulo de  $v$  na numeração  $f$ .

Em outras palavras  $N'(v)$  contém os vértices  $u$  tal que  $f(u)$  está mais próximo de  $mid(v)$  do que  $f(v)$ . Esses vértices são candidatos a troca com vértices  $\lambda$ -críticos  $v$  durante o *HillClimbing*.

Uma desvantagem desse método de *HillClimbing* é que ele tende a ser computacionalmente caro. Por esse motivo, chamá-la em todas as iterações do algoritmo, como sugere o framework clássico da meta-heurística ILS, pode não ser adequado. Além disso, foi percebido que a aplicação em sequência do procedimento de perturbação `midOrdering()` pode apresentar ganhos mais significativos do que sua aplicação alternada com a busca local. Nessa situação, a busca local seria executada desnecessariamente. Por isso, optou-se por realizar a busca local ao fim da etapa de perturbações, caso a solução  $s'$  seja melhor que a solução atual.

#### 5.1.4 Critério de aceitação

O critério de aceitação é o que determina se a solução intermediária  $s'$  será aceita como solução atual  $s^*$ . Uma estratégia comum é aceitar uma nova solução apenas se ela for melhor que a solução atual. O algoritmo proposto permite que soluções melhores ou iguais à solução atual possam ser aceitas como nova solução atual.

De modo a evitar que ocorra estagnação na busca, uma perturbação intensa é aplicada caso nenhuma melhora na solução seja identificada por um número definido de iterações, armazenado na variável `maxItersNoImprovement`, no Algoritmo 5.4. Optou-se por utilizar uma adaptação do procedimento `groupCritical()` em que o parâmetro `mult` foi definido com um valor mais alto. O valor escolhido para `mult` nessa perturbação intensa foi o da variável `numberOfResets/2`. A variável `numberOfResets` armazena o número de vezes que a perturbação intensa foi aplicada. Assim, quanto mais vezes for detectada a estagnação, mais intensa é a perturbação que busca contorná-la.

Quando a perturbação intensa é aplicada, o valor da variável  $\alpha_{crit}$  é diminuído. Isso implica num número possivelmente maior de vértices  $\lambda$ -críticos nas próximas iterações. Essa alteração afeta tanto a `buscaLocal()` quanto o procedimento `groupCritical()`, atuando assim

no aumento da intensificação e diversificação da busca, respectivamente. Além disso, a variável *maxItersNoImprovement* é reiniciada com 0 e a variável de controle *numberOfResets* é incrementada em 1.

## 5.2 As heurísticas ILS-Band e ILS-Prof

Até esse ponto, o algoritmo proposto é independente da métrica que se deseja minimizar, largura de banda ou *profile*. Adaptar o algoritmo para uma delas consiste basicamente em adaptar o critério de aceitação de forma correspondente. Deste modo, definimos aqui duas variações da heurística proposta. Uma versão com foco em redução de largura de banda, é nomeada **ILS-Band**. Nessa versão, dada a nova solução  $s'$ , o critério de aceitação considera que houve melhoria na solução atual  $s^*$  se  $\beta(s') \leq \beta(s^*)$ . A outra versão, com foco em redução de *profile*, é nomeada **ILS-Prof**. Analogamente, seu critério de aceitação considera que houve melhora na solução  $s^*$  se  $Profile(s') \leq Profile(s^*)$ .

O pseudocódigo das heurísticas é apresentado no Algoritmo 5.4. Note que a diferenciação entre as duas versões, ILS-Band e ILS-Prof, acontece na implementação do critério de aceitação, na linha 10.

---

### Algoritmo 5.4 Pseudocódigo genérico das heurísticas ILS-Band e ILS-Prof

---

```

1: ILS-GENERIC()
2:  Entrada:  maxIterations,  maxResets,  maxItersNoImprovement,  vértice pseudo-
   periférico  $v_{pseudo}$ 
3:   $iteration = 0$ ;
4:   $itersNoImprovement = 0$ ;
5:   $s^* = SolucaoInicial(v_{pseudo})$ ;
6:  repeat
7:     $s' = Perturbacao(s^*)$ ;
8:    se  $s' < s^*$  então
9:       $s' = BuscaLocal()$ ;
10:    $s^* = Aceitacao(s', s^*)$ ;
11:   se Não houve melhora na solução então
12:      $itersNoImprovement++$ ;
13:     se  $itersNoImprovement == maxItersNoImprovement$  então
14:        $groupCritical(numberOfResets / 2)$ ; // Perturbação intensa
15:        $numberResets++$ ;
16:        $itersNoImprovement = 0$ ;
17:      $iteration = iteration + 1$ ;
18:  until  $iteration \geq maxIterations$  ou  $numberResets > maxResets$ ; retorne  $s^*$ 

```

---

A variável *itersNoImprovement* armazena o número de iterações sem melhora na solução  $s^*$ . Se não houver melhoras por *maxItersNoImprovement* iterações, uma perturbação intensa é realizada, como descrita na Seção 5.1.2 (linhas 11 a 16 do Algoritmo 5.4). A cada perturbação intensa, a variável *numberResets* é incrementada. A variável *maxResets* controla o número máximo de vezes que a solução atual sofre a perturbação intensa. Juntamente com a variável *maxIterations*, a variável *maxResets* funciona como critério de parada para o algoritmo.

Para os testes realizados neste trabalho, a variável *maxIterations* foi definida em 300, *maxResets* em 30 e *maxItersNoImprovement* em 15. O valor utilizado para a variável  $k_{rand}$ , usada no procedimento *midOrdering()* foi 6. A variável  $\alpha_{crit}$  é inicializada com 0.95 e decrementada em 0.013, a cada perturbação intensa.

### 5.3 Detalhes de desenvolvimento

Para a implementação das heurísticas propostas foi utilizada a linguagem C++. Internamente, os vértices são mantidos em uma estrutura de armazenamento CRS, descrita no Capítulo 2. A representação compacta obtida por meio do CRS é importante quando se lida com matrizes de grandes dimensões, que demandariam alto custo de armazenamento com o uso de listas de adjacências. Além disso, é possível realizar a permutação de linhas e colunas na estrutura CRS com custo baixo. Essa característica é altamente desejada, já que o algoritmo proposto se utiliza exaustivamente dessa operação.

Para a implementação dos algoritmos propostos utilizou-se a IDE CLion 2017.2 e o editor de textos *Sublime Text 2*. O compilador utilizado foi o g++ versão 5.4.0 com auxílio do cmake 3.10.1. Todos os testes foram executados por meio do sistema computacional, também escrito em C++, utilizado no trabalho de Oliveira, Bernardes e Chagas (2016). Esse sistema é responsável por implementar e gerenciar a execução das heurísticas comparadas com a heurística proposta. À exceção da heurística VNS-Band, todas as outras estão implementadas nesse sistema.

## 6. SIMULAÇÕES PARA REDUÇÃO DE LARGURA DE BANDA E DE *PROFILE*

Neste capítulo encontram-se os resultados das simulações para reduções de largura de banda e de *profile* comparando as heurísticas ILS-Band e ILS-Prof propostas a heurísticas relevantes na literatura para esses problemas. A descrição das instâncias utilizadas e métricas empregadas nas simulações deste capítulo encontram-se nas Seções 6.1 e 6.2. As simulações conduzidas foram divididas entre instâncias simétricas e assimétricas. Os resultados da simulações com instâncias simétricas são apresentados na Seção 6.3. Os resultados da simulações com instâncias assimétricas são apresentados na Seção 6.4.

### 6.1 Instâncias utilizadas

No trabalho de Martí et al. (2001), as simulações para redução de largura de banda foram realizadas sobre um conjunto de 113 instâncias da base Harwell-Boeing (DUFF; GRIMES; LEWIS, 1992), com dimensões variando entre 30 e 1104 vértices. Em diversos trabalhos seguintes e de anos recentes, o conjunto de instâncias utilizadas foi o mesmo (PIÑANA et al., 2004; LIM; RODRIGUES; XIAO, 2003; LIM; RODRIGUES; XIAO, 2004; LIM; RODRIGUES; XIAO, 2006; LIM et al., 2006; LIM; LIN; XIAO, 2007; LIM; RODRIGUES; XIAO, 2007; RODRIGUEZ-TELLO; HAO; TORRES-JIMENEZ, 2008; MLADENOVIC et al., 2010; CAMPOS; PIÑANA; MARTÍ, 2011; ISAZADEH; IZADKHAH; MOKARRAM, 2012; POP; MATEI; COMES, 2014; TORRES-JIMENEZ et al., 2015).

Para este trabalho, considerou-se que as dimensões das matrizes utilizadas nos trabalhos citados já não são satisfatórias para os padrões atuais e, por isso, foi necessário utilizar instâncias maiores. Assim, as simulações para redução de largura de banda e *profile* de matrizes deste trabalho utilizaram um conjunto de 172 instâncias quadradas da base de matrizes esparsas da *Suite Sparse Matrix Collection* (DAVIS; HU, 2011), divididas em 124 instâncias simétricas e 48 assimétricas. As dimensões das matrizes variam entre 5940 e 112985 mil vértices.

As heurísticas propostas, ILS-Band e ILS-Prof, foram comparadas às heurísticas RCM-GL, RBFS-GL, GPS, KP-Band-GL, FNC-HC e VNS-Band para redução de largura de banda e Sloan, Sloan-MGPS, NSloan e MPG para redução de *profile*. As heurísticas FNC-HC e GPS para redução de largura de banda e NSloan e MPG para redução de *profile* foram elencadas por Oliveira, Bernardes e Chagas (2016) como as possíveis melhores heurísticas de baixo custo para seus respectivos problemas. Além dessas, os autores destacam as heurísticas RCM-GL e RBF-GL por seu baixo custo e processamento e a de Sloan por se sobressair em problemas

de áreas específicas. As heurísticas VNS-Band e KP-Band-GL são apontadas por Chagas e Oliveira (2015) como possíveis heurísticas no estado da arte para redução de largura de banda de matrizes simétricas.

As implementações utilizadas das heurísticas RCM-GL, RBFS-GL, GPS, KP-Band-GL, Sloan, Sloan-MGPS, NSloan e MPG foram as utilizadas por Oliveira, Bernardes e Chagas (2016). Recomenda-se consultar o referido trabalho para detalhes de implementação e calibração dessas heurísticas. O código-fonte da heurística FNC-HC foi gentilmente cedido pelo professor Dr. Fei Xiao. Os arquivos executáveis versões 32 e 64 bits da heurística VNS-Band foram gentilmente cedidos pelo professor Dr. Dragan Urosevic.

A heurística VNS-Band utiliza um parâmetro de tempo máximo de execução para encerrar o algoritmo. O tempo limite utilizado nos testes foi a média de tempos do método baseado em meta-heurística com maior tempo de execução. Como poderá ser observado nos testes nas seções seguintes, as médias dos tempos utilizadas foram as da heurística FNC-HC. Os valores foram 60 segundos para instâncias simétricas e 50 segundos (valor aproximado para cima) para instâncias assimétricas.

Nas simulações computacionais nas subseções seguintes, cada heurística foi executada 10 vezes com cada uma das 172 instâncias selecionadas, divididas entre simétricas e assimétricas. Os valores de largura de banda, *profile* e tempo de execução nas tabelas das subseções seguintes correspondem à média das 10 execuções.

Para os testes computacionais realizados nas instâncias da base *SuiteSparse Matrix Collection* utilizou-se um computador com processador Intel Core i7-2600 CPU 3.40GHz com 8MB de cache L3 e 8GB de memória RAM DDR3 1333MHz. O sistema operacional utilizado foi o Linux Mint 18.2 com 64 bits e kernel 4.10.0-40-generic.

## 6.2 Métricas utilizadas para comparação dos resultados

Para cada heurística, o número de instâncias nas quais ela obteve o melhor resultado entre todas as outras é indicado nas tabelas por “Num melhores”. Apesar de este ser um bom indicativo do desempenho do algoritmo, essa métrica, por si só, não é adequada. Isso porque uma heurística que conseguisse resultados próximos dos melhores na maioria das instâncias sem, contudo, conseguir o melhor resultado em nenhuma delas, seria prejudicada. Por esse motivo, o trabalho de Oliveira, Bernardes e Chagas (2016) aplica uma métrica  $\rho$  que considera

a distância relativa entre a solução encontrada por uma heurística e a melhor solução encontrada naquela instância. Seja uma heurística  $h$ , a métrica  $\rho$  é dada por

$$\rho_p = \frac{\text{profile}_h - \text{profile}_{min}}{\text{profile}_{min}}, \quad (6.1)$$

em que  $\text{profile}_h$  é o *profile* obtido pela heurística e  $\text{profile}_{min}$  é o menor *profile* obtido por uma heurística nessa instância. O mesmo vale para a largura de banda e tempo de execução. Quanto menor o valor de  $\rho$ , melhor o desempenho geral da heurística naquele conjunto de instâncias. O somatório de  $\rho$  para cada um desses atributos também é indicado nas tabelas seguintes.

### 6.3 Simulações com 124 instâncias simétricas

Nas Tabelas A1, A2 e A3 presentes no Apêndice A são apresentados os resultados de redução de largura de banda, de *profile* e o tempo de execução, respectivamente, das 10 heurísticas testadas em 124 instâncias simétricas. Uma compilação dessas tabelas é apresentada na forma das figuras 6.1 a 6.6. A análise dos resultados de cada métrica é feita a seguir.

#### 6.3.1 Redução de largura de banda

Como visto na Tabela A1, a heurística que obteve o maior número de melhores resultados em redução de largura de banda foi a heurística FNC-HC, em 66 das 124 instâncias simétricas (Figura 6.1). Esse número corresponde a mais da metade das instâncias testadas. Em seguida, a heurística ILS-Band proposta obteve 34 melhores resultados nesse conjunto (pouco mais de um quarto das instâncias), acompanhada da heurística GPS, com 21 melhores resultados. O ranking ocupado por essas três heurísticas com relação à métrica  $\rho$  (vide Figura 6.2) coincide com a quantidade de melhores resultados encontrados por cada uma.

Esses resultados corroboram os obtidos pelo trabalho de Oliveira, Bernardes e Chagas (2016), no qual as heurísticas FNC-HC e GPS foram consideradas as melhores heurísticas para redução de largura de banda de baixo custo. Nesse contexto, notamos como os resultados obtidos pela heurística ILS-Band proposta foram relevantes, se posicionando entre essas duas heurísticas. Além disso, o valor da métrica  $\rho$  para a heurística ILS-Band (25,9) ficou mais próximo do valor da heurística FNC-HC (17,4) do que do valor da heurística GPS (39,5).

Figura 6.1 – Número de melhores resultados em redução de largura de banda das heurísticas selecionadas em 124 instâncias simétricas

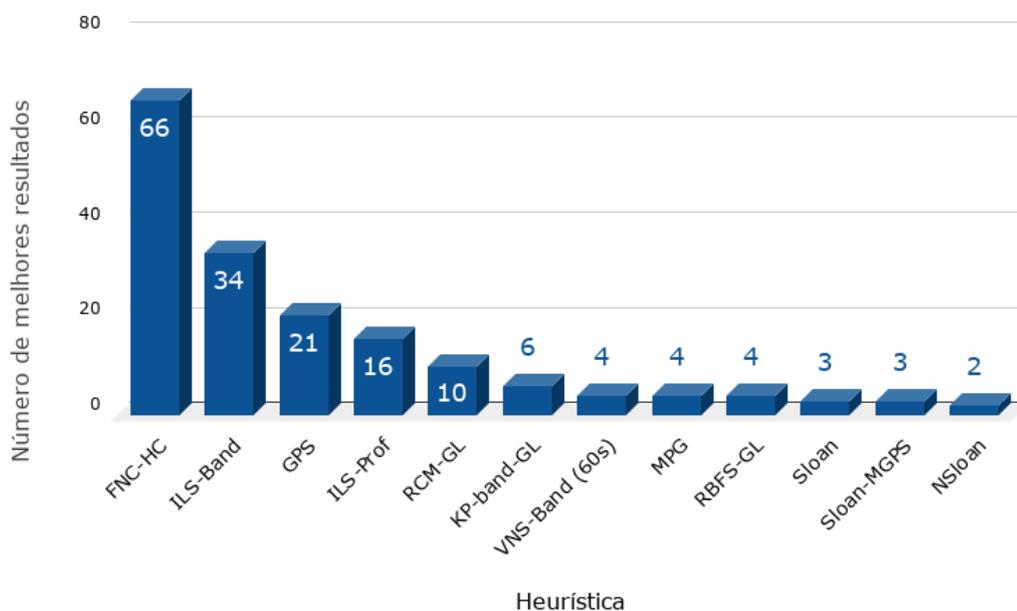
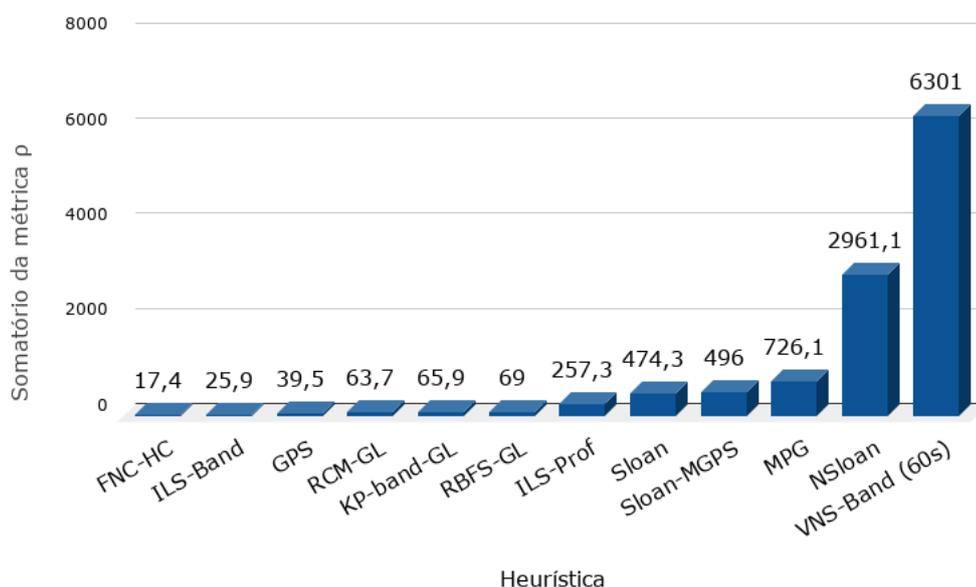


Figura 6.2 – Valor da métrica  $\rho$  das heurísticas selecionadas em redução de largura de banda em 124 instâncias simétricas (quanto menor, melhor)



Apesar de poder ser considerada uma das heurísticas no estado da arte para o problema segundo Mladenovic et al. (2010), Oliveira, Bernardes e Chagas (2016), a heurística VNS-Band obteve apenas 4 melhores resultados nesse conjunto de instâncias (Figura 6.1). Isso pode ser explicado pela limitação de tempo imposta, de 60 segundos. Para comparação, os autores da heurística VNS-Band aplicaram-na sobre instâncias muito menores (até 1000 vértices) com o

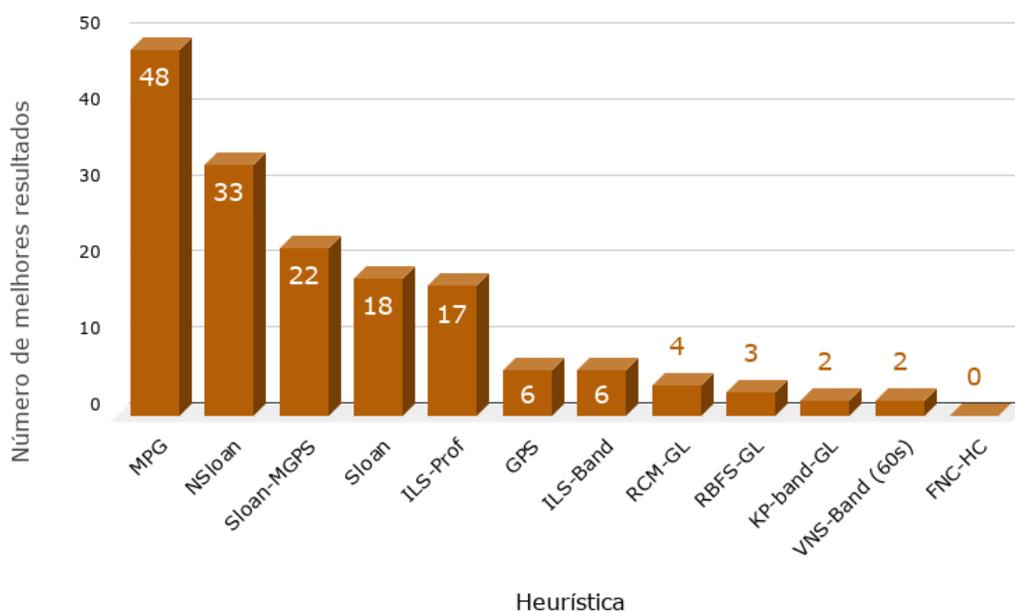
tempo máximo de 500 segundos (MLADENOVIC et al., 2010). Além disso, a heurística VNS-Band obteve o pior desempenho em relação à métrica  $\rho$ , muito acima do segundo pior resultado (heurística NSloan), como pode ser visto na Figura 6.2.

A heurística RCM-GL conseguiu 10 melhores resultados mas foi uma das mais rápidas nesse conjunto, como será discutido adiante, na seção 6.3.3. As heurísticas RBFS-GL, Sloan e Sloan-MGPS conseguiram, respectivamente, 4, 3 e 3 melhores resultados e foram as mais rápidas nos testes. As heurísticas para redução de *profile* MPG e NSloan obtiveram, respectivamente 4 e 2 melhores resultados de redução de largura de banda.

### 6.3.2 Redução de *profile*

Nos testes de redução de *profile* de instâncias simétricas da Tabela A2, a heurística MPG obteve 48 melhores resultados e a melhor posição no ranking da métrica  $\rho$  (Figuras 6.3 e 6.4). Em seguida, a heurística NSloan obteve 33 melhores resultados. Juntas, as duas heurísticas somaram pouco mais de 65% os melhores resultados para redução de *profile* nas instâncias simétricas.

Figura 6.3 – Número de melhores resultados em redução de *profile* das heurísticas selecionadas em 124 instâncias simétricas

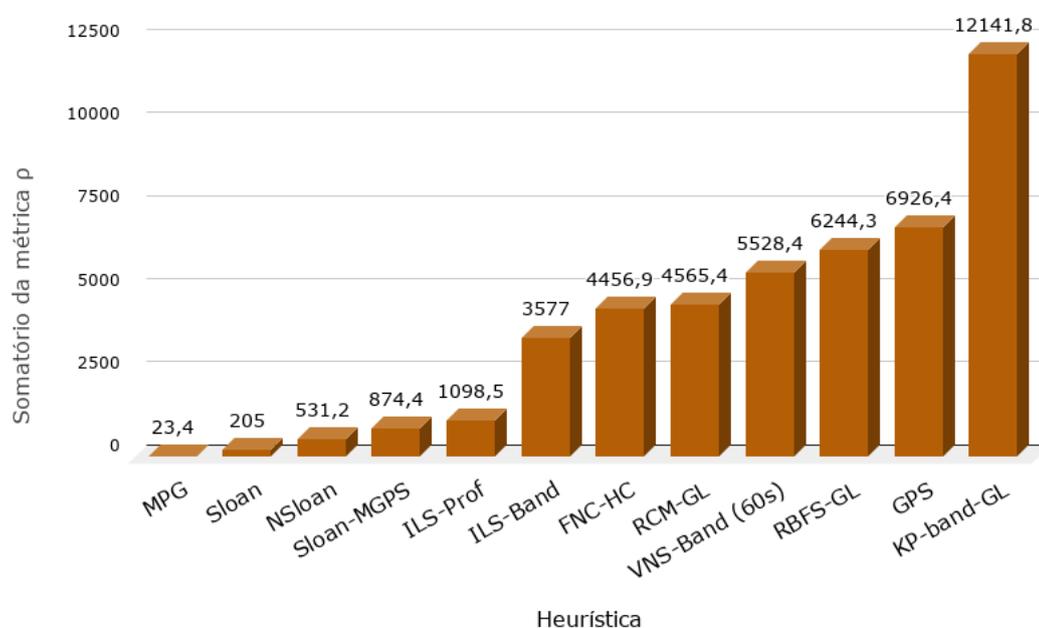


O algoritmo ILS-Prof proposto obteve 17 melhores resultados em redução de *profile* nesse conjunto, atrás da heurística de Sloan, com 18, e da heurística Sloan-MGPS, com 22. O algoritmo ILS-Band conseguiu apenas 6 melhores resultados para redução de *profile*, assim

como a heurística GPS. A heurística VNS-Band conseguiu 2 melhores resultados nesse conjunto, e a heurística FNC-HC, nenhum melhor resultado.

Quanto à métrica  $\rho$ , as heurísticas para redução de *profile* MPG, Sloan, NSloan e Sloan-MGPS ocuparam as primeiras posições. Em seguida, em quinto e sexto lugar, respectivamente, ficaram as heurísticas propostas, ILS-Prof e ILS-Band, esta já com grande margem em relação às primeiras colocadas. A heurística KP-Band-GL obteve o pior desempenho na métrica  $\rho$  para redução de *profile* nas instâncias assimétricas, com ampla diferença com relação à segunda pior nesse quesito, a heurística GPS.

Figura 6.4 – Valor da métrica  $\rho$  das heurísticas selecionadas em redução de *profile* em 124 instâncias simétricas (quanto menor, melhor)

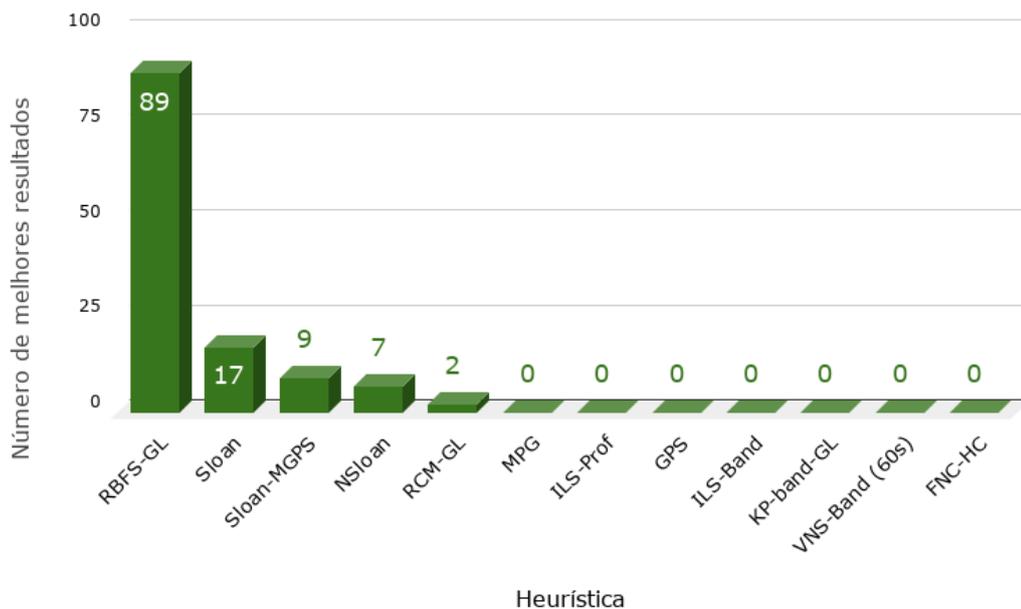


### 6.3.3 Tempo de execução

Os tempos de execução das heurísticas nos testes com instâncias simétricas são mostrados na Tabela A3 e Figura 6.5. A heurística RBFS-GL foi mais rápida em 89 das 124 instâncias simétricas testadas. Esse resultado é natural, visto que se trata apenas do custo de se encontrar um vértice pseudo-periférico somado ao custo de uma busca em largura ordinária. A implementação da heurística de Sloan conseguiu os menores tempos de execução em 17 instâncias, a heurística Sloan-MGPS em 9 delas, e a heurística NSloan, em 7. Em 2 instâncias simétricas, a heurística RCM-GL obteve os menores tempos de execução.

Em relação à métrica  $\rho$ , os piores resultados foram obtidos pelas heurísticas baseadas em meta-heurísticas para redução de largura de banda, VNS-Band e FNC-HC (vide Figura 6.6). O valor mais alto foi o da meta-heurística VNS-Band, com o tempo fixo de 60 segundos por instância, seguida da heurística FNC-HC. Entre as heurísticas para redução de *profile*, a heurística ILS-Prof foi muito mais lenta, no geral, que as heurísticas Sloan, NSloan e Sloan-MGPS, de acordo com a métrica  $\rho$ , mas mais rápida que a heurística MPG. Entre as heurísticas para redução de largura de banda, a meta-heurística ILS-Band foi mais lenta que as heurísticas RBFS-GL, RCM-GL e KP-Band-GL, mas consideravelmente mais rápida que as heurísticas FNC-HC e VNS-Band.

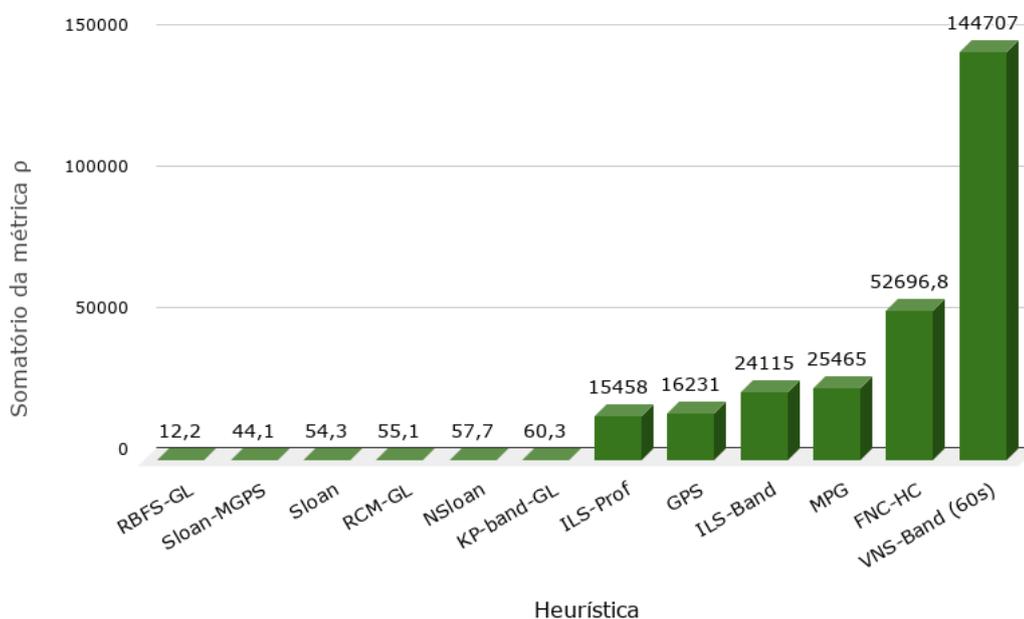
Figura 6.5 – Número de melhores resultados para tempo de execução das heurísticas selecionadas em 124 instâncias simétricas



#### 6.4 Simulações com 48 instâncias assimétricas

Nas Tabelas A4, A5 e A6 presentes no Apêndice A são apresentados os resultados de redução de largura de banda, de *profile* e o tempo de execução, respectivamente, das 12 heurísticas testadas comparadas sobre 48 instâncias simétricas. Uma compilação dessas tabelas é apresentada na forma das figuras 6.7 a 6.12. A análise dos resultados desses testes é feita a seguir.

Figura 6.6 – Valor da métrica  $\rho$  para tempo de execução das heurísticas selecionadas em 124 instâncias simétricas (quanto menor, melhor)



#### 6.4.1 Redução de largura de banda

Como mostrado na Tabela A4, a heurística que obteve o maior número de melhores resultados em redução de largura de banda nas instâncias assimétricas testadas foi a FNC-HC, em 24 das 48 instâncias (Figura 6.7). As heurísticas ILS-Band e ILS-Prof obtiveram, respectivamente, 12 e 10 melhores resultados cada, seguidas da heurística GPS, com 7 melhores resultados. As heurísticas para redução de *profile* MPG, Sloan, NSloan e Sloan-MGPS não obtiveram nenhum melhor resultado em redução de largura de banda nas instâncias assimétricas.

Em relação à métrica  $\rho$ , a heurística FNC-HC também teve o melhor desempenho (Figura 6.8). A heurística proposta ILS-Prof foi apenas a terceira melhor nesse quesito, atrás da heurística GPS por uma diferença considerável. As heurísticas RCM-GL, KP-Band-GL e RBFS-GL ocuparam as últimas posições quanto à essa métrica. Isto é, elas foram ainda piores, no geral, que as heurísticas para redução de *profile* MPG, Sloan, NSloan e Sloan-MGPS nas instâncias assimétricas.

#### 6.4.2 Redução de *profile*

Os resultados de redução de *profile* em 48 instâncias assimétricas da Tabela A5 mostram que a heurística proposta ILS-Prof obteve o maior número de melhores resultados (17)

Figura 6.7 – Número de melhores resultados em redução de largura de banda das heurísticas selecionadas em 48 instâncias assimétricas

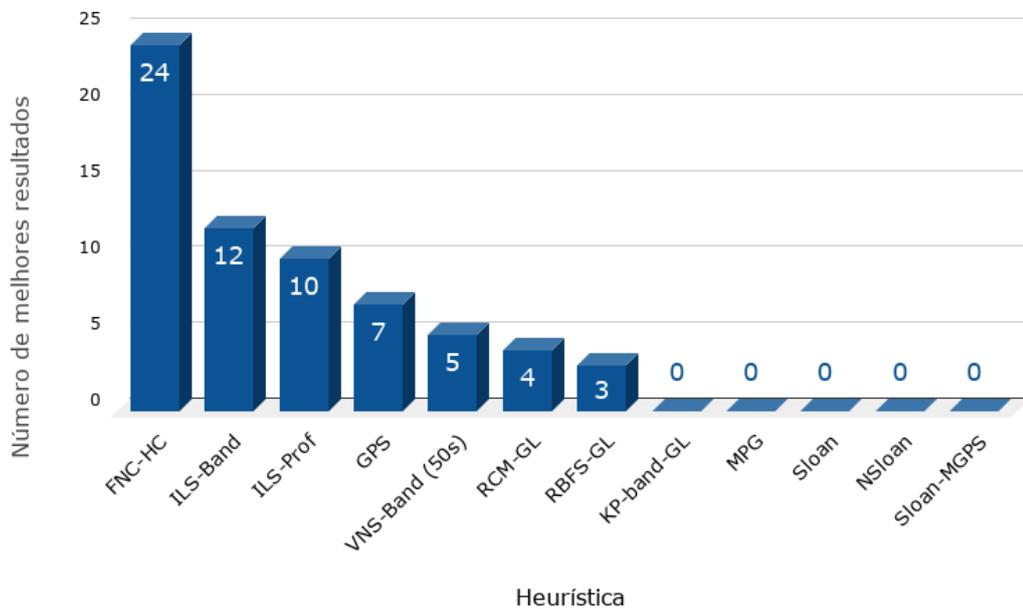
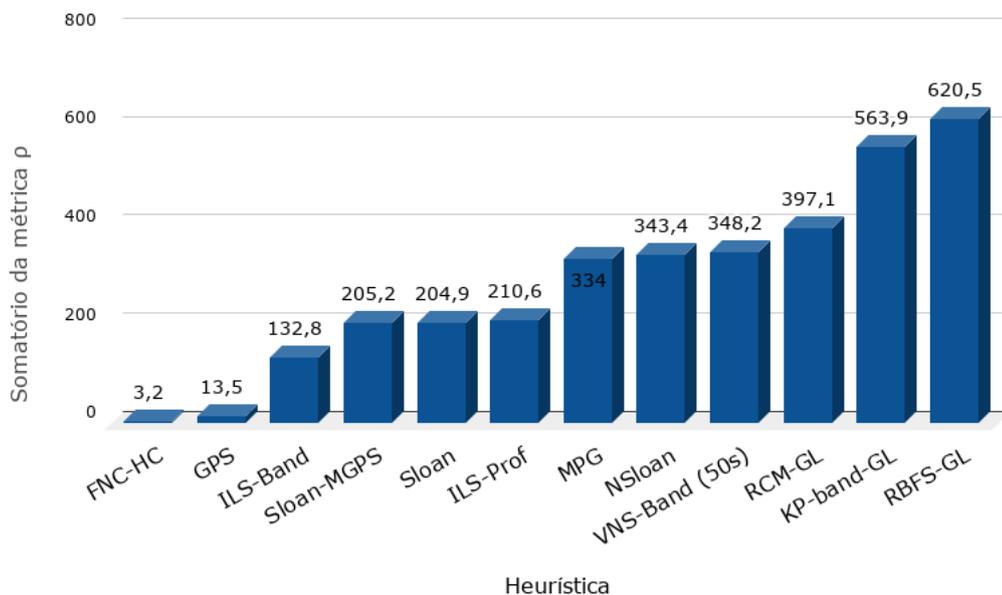
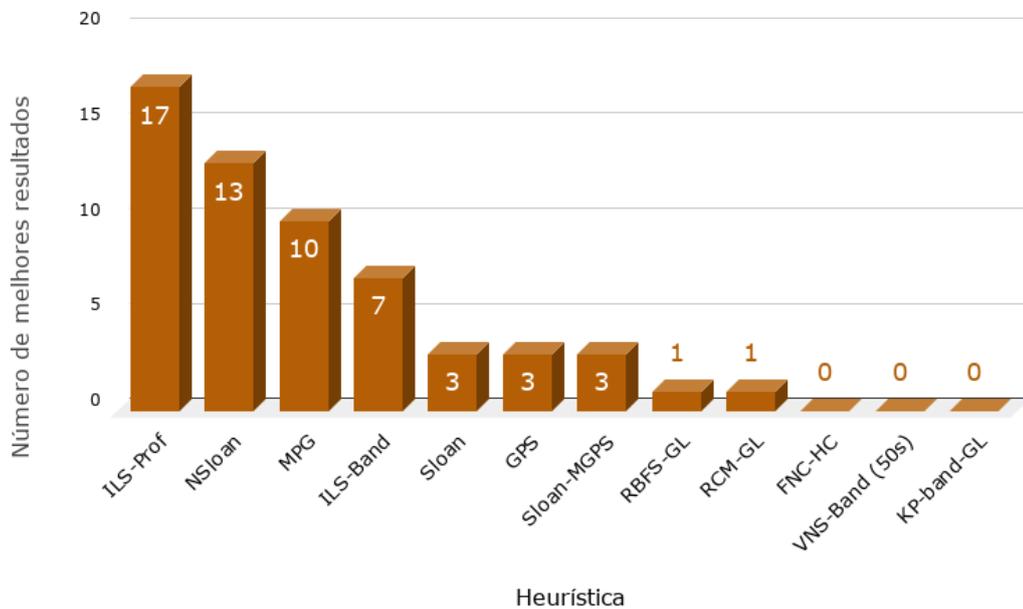


Figura 6.8 – Valor da métrica  $\rho$  das heurísticas selecionadas em redução de largura de banda em 48 instâncias assimétricas (quanto menor, melhor)



nessas instâncias (Figura 6.9). A heurística NSloan foi a segunda melhor nesse quesito, com 13 melhores resultados, seguida da heurística MPG, com 10 melhores resultados. A heurística ILS-Band foi a quarta melhor em número de melhores resultados, superando as heurísticas de Sloan e Sloan-MGPS.

Figura 6.9 – Número de melhores resultados em redução de *profile* das heurísticas selecionadas em 48 instâncias assimétricas



As heurísticas que obtiveram o melhor desempenho quanto à métrica  $\rho$  para redução de *profile* foram MPG e NSloan, em consonância com o trabalho de Oliveira, Bernardes e Chagas (2016). A heurística de Sloan foi a terceira melhor no somatório de  $\rho$  para redução de *profile* e a heurística Sloan-MGPS, a quarta. A heurística ILS-Prof, na quinta posição, ficou mais próxima do valor de  $\rho$  das quatro melhores heurísticas do que da sexta colocada, a heurística GPS, como mostra a Figura 6.10. Assim como nos testes de redução de largura de banda da seção 6.4.1, as heurísticas RCM-GL, KP-Band-GL e RBFS-GL ocuparam as últimas posições quanto à essa métrica.

### 6.4.3 Tempo de execução

Os tempos de execução das heurísticas nas 48 instâncias assimétricas são mostrados na Tabela A6. Assim como para as instâncias simétricas, a heurística RBFS-GL foi a mais rápida na maioria das instâncias assimétricas testadas (44 de 48 instâncias). As únicas outras heurísticas a conseguirem melhores resultados foram as heurísticas de Sloan, em 2 instâncias, e suas variações, NSloan e Sloan-MGPS, em 1 instância cada (Figura 6.11).

Assim como para as instâncias simétricas, os piores tempos de execução nesse conjunto de instâncias foram obtidos pelas heurísticas baseadas em meta-heurísticas. O valor mais alto da métrica  $\rho$  foi o da heurística VNS-Band com tempo fixo de 50 segundos, com grande margem

Figura 6.10 – Valor da métrica  $\rho$  das heurísticas selecionadas em redução de *profile* em 48 instâncias assimétricas (quanto menor, melhor)

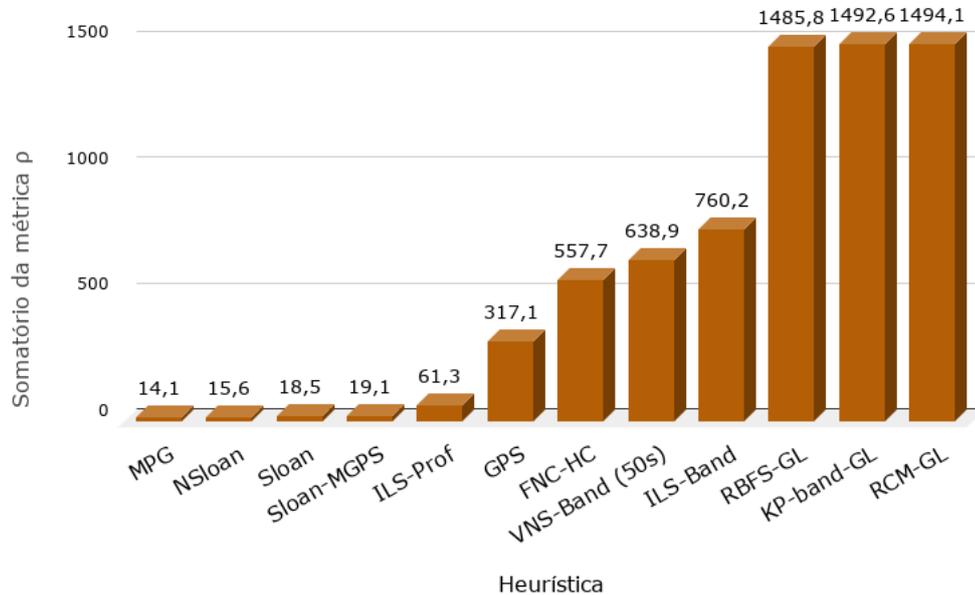
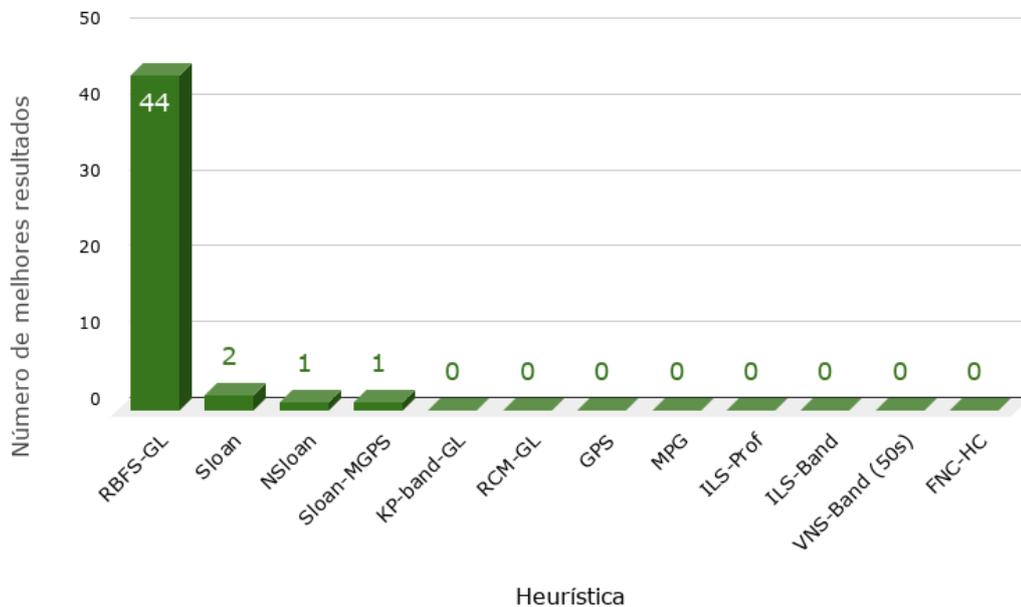


Figura 6.11 – Número de melhores resultados para tempo de execução das heurísticas selecionadas em 48 instâncias assimétricas

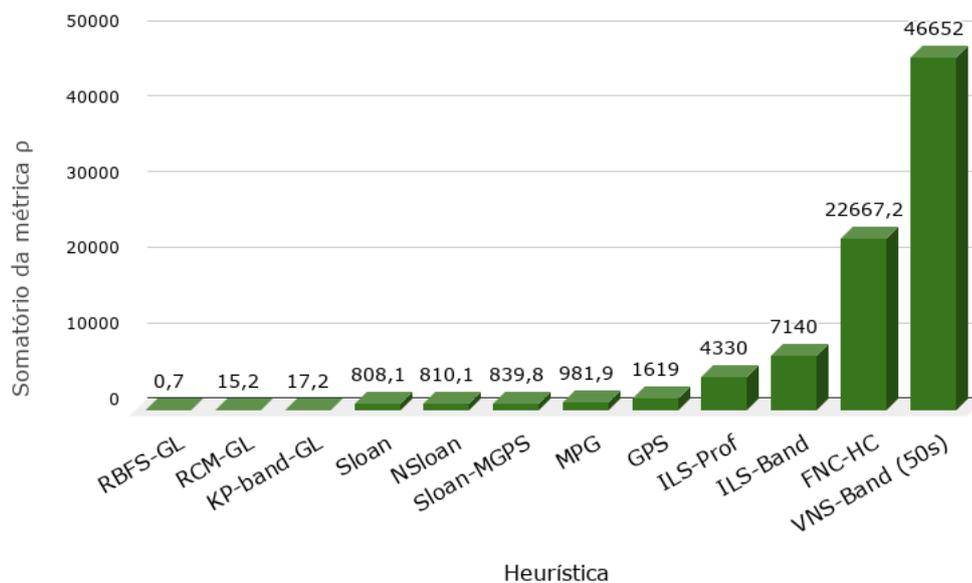


em relação à segunda pior, a heurística FNC-HC. As heurísticas propostas, ILS-Prof e ILS-Band, obtiveram o nono e décimo melhor tempo, respectivamente.

Em contrapartida ao desempenho ruim quanto a redução de largura de banda (Seção 6.4.1) e de *profile* (seção 6.4.2) nas instâncias assimétricas, as heurísticas redução de largura de banda RCM-GL, RBFS-GL e KP-Band-GL obtiveram os menores tempo de execução. As

heurísticas Sloan, NSloan, Sloan-MGPS e MPG ocuparam da quarta à sétima posições em relação à métrica  $\rho$  nesse conjunto de instâncias, respectivamente.

Figura 6.12 – Valor da métrica  $\rho$  para tempo de execução das heurísticas selecionadas em 48 instâncias assimétricas (quanto menor, melhor)



## 6.5 Discussão sobre as simulações para redução de largura de banda e de *profile*

Neste capítulo, foram apresentados os resultados das simulações com 172 instâncias da *Suite Sparse Matrix Collection* (DAVIS; HU, 2011), divididas em 124 instâncias simétricas e 48 instâncias assimétricas. Dez heurísticas para reduções de largura de banda e de *profile* na literatura foram comparadas às heurísticas ILS-Band e ILS-Prof propostas. O *ranking* das heurísticas testadas com relação à métrica  $\rho$  é mostrado na Tabela 6.1.

### 6.5.1 Simulações em instâncias simétricas

Nos testes com 124 instâncias simétricas, a heurística proposta ILS-Band obteve o segundo melhor desempenho geral em redução de largura de banda de acordo com a métrica  $\rho$ , atrás apenas da heurística FNC-HC. Nos testes para redução de *profile*, a heurística proposta ILS-Prof foi a quinta melhor, atrás das heurísticas MPG, NSloan, Sloan-MGPS e de Sloan.

As Figuras 6.13 e 6.14 ilustram o desempenho relativo das heurísticas nos testes realizados com instâncias simétricas em redução de largura de banda e de *profile*, respectivamente. Por meio de uma análise da Figura 6.13 constata-se que, enquanto as heurísticas FNC-HC e

Tabela 6.1 – *Ranking* das heurísticas para redução de largura de banda e *profile*, em relação à métrica  $\rho$ 

| Heurística | Simétricas |           |           | Assimétricas |           |           |
|------------|------------|-----------|-----------|--------------|-----------|-----------|
|            | $\beta$    | profile   | t (s)     | $\beta$      | profile   | t (s)     |
| ILS-Band   | 2°         | 6°        | 9°        | 3°           | 9°        | 9°        |
| ILS-Prof   | 7°         | 5°        | 7°        | 6°           | 5°        | 10°       |
| RCM-GL     | 4°         | 7°        | 4°        | 10°          | 12°       | 2°        |
| RBFS-GL    | 6°         | 10°       | <b>1°</b> | 12°          | 10°       | <b>1°</b> |
| KP-Band-GL | 5°         | 12°       | 6°        | 11°          | 11°       | 3°        |
| GPS        | 3°         | 11°       | 8°        | 2°           | 6°        | 9°        |
| FNC-HC     | <b>1°</b>  | 8°        | 11°       | <b>1°</b>    | 7°        | 12°       |
| VNS-Band   | 12°        | 9°        | 12°       | 9°           | 8°        | 11°       |
| Sloan      | 8°         | 2°        | 3°        | 5°           | 3°        | 4°        |
| NSloan     | 9°         | 3°        | 5°        | 8°           | 2°        | 5°        |
| Sloan-MGPS | 11°        | 4°        | 2°        | 4°           | 4°        | 6°        |
| MPG        | 10°        | <b>1°</b> | 10°       | 7°           | <b>1°</b> | 7°        |

ILS-Band produzem soluções de qualidade superior, o melhor equilíbrio entre qualidade e velocidade de solução pode ser conseguido com a heurística RBFS-GL. De maneira análoga, uma análise da Figura 6.14 sugere que, enquanto a heurística MPG é claramente melhor em redução de *profile* de instâncias simétricas, um melhor equilíbrio entre qualidade e velocidade de solução é encontrado na heurística de Sloan.

### 6.5.2 Simulações em instâncias assimétricas

Nos testes com 48 instâncias assimétricas, a heurística proposta ILS-Band obteve a terceira colocação em redução de largura de banda de acordo com a métrica  $\rho$ , atrás das heurísticas FNC-HC e GPS. Assim como nas instâncias simétricas, a heurística proposta ILS-Prof obteve a quinta posição, atrás das quatro heurísticas para redução de *profile* testadas.

As Figuras 6.15 e 6.16 ilustram o desempenho relativo das heurísticas nos testes realizados com instâncias assimétricas em redução de largura de banda e de *profile*, respectivamente. Por meio de uma análise desses resultados não é tão simples determinar com precisão quais heurísticas apresentam o melhor equilíbrio entre qualidade e velocidade da solução no conjunto de instâncias assimétricas. Esses resultados sugerem que a escolha de qual heurística utilizar em problemas práticos envolvendo instâncias assimétricas deve levar em consideração o quanto se deseja sacrificar em qualidade da solução ou em velocidade de processamento.

Figura 6.13 – Gráfico sobreposto de  $\sum \rho$  para largura de banda e para tempo das simulações com 124 instâncias simétricas, ordenadas por tempo (em escala logarítmica).

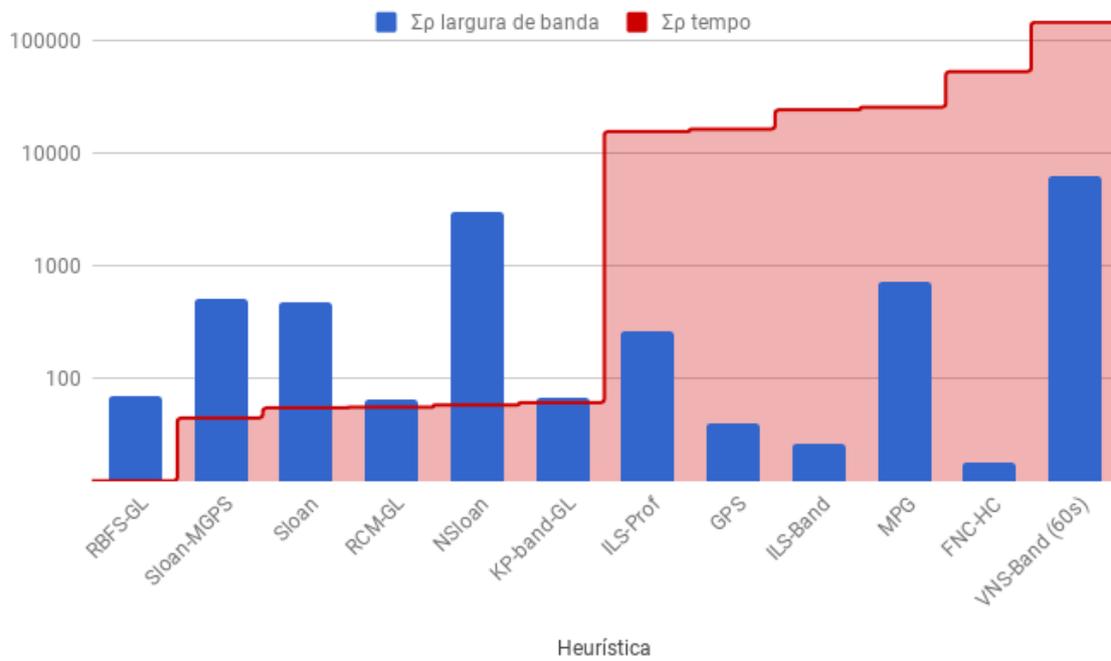


Figura 6.14 – Gráfico sobreposto de  $\sum \rho$  para *profile* e para tempo das simulações com 124 instâncias simétricas, ordenadas por tempo (em escala logarítmica).

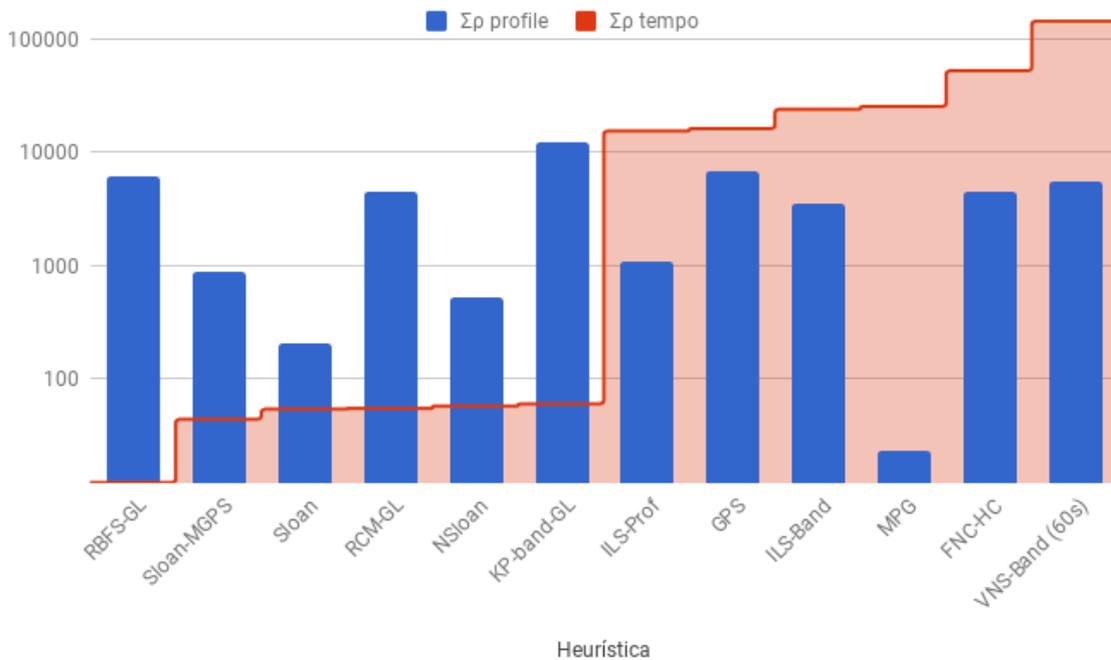


Figura 6.15 – Gráfico sobreposto de  $\sum \rho$  para largura de banda e para tempo das simulações com 48 instâncias assimétricas, ordenadas por tempo (em escala logarítmica).

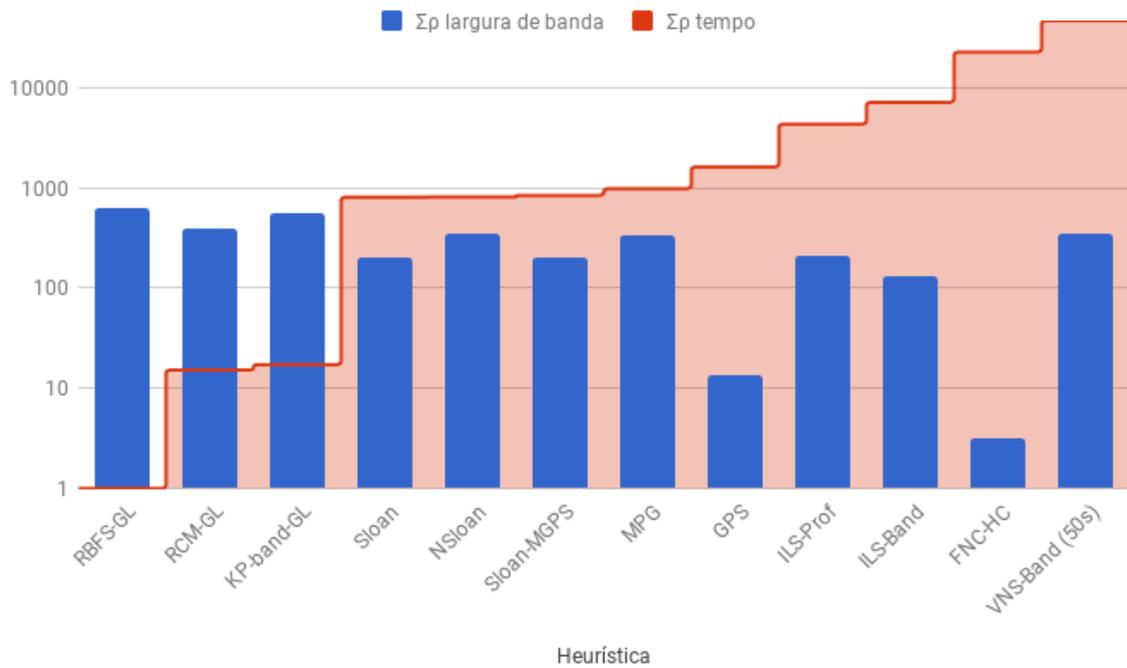
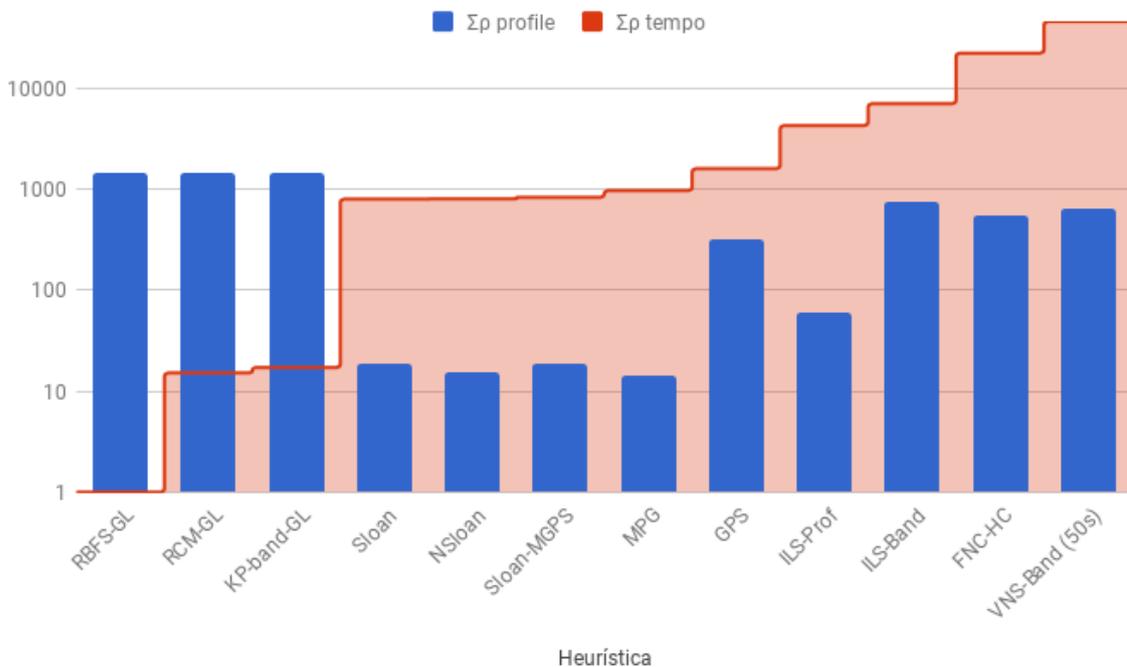


Figura 6.16 – Gráfico sobreposto de  $\sum \rho$  para *profile* e para tempo das simulações com 48 instâncias assimétricas, ordenadas por tempo (em escala logarítmica).



### 6.5.3 Considerações finais sobre o desempenho das heurísticas testadas

Nos testes com instâncias simétricas e assimétricas deste capítulo, a heurística ILS-Band obteve o oitavo melhor tempo, entre as doze heurísticas testadas. A heurística ILS-Prof obteve

o sétimo melhor tempo nas instâncias simétricas e apenas o nono melhor tempo nas instâncias assimétricas. Em outras palavras, as heurísticas propostas ficaram entre as cinco mais lentas nos dois conjuntos de instâncias. Levando em consideração os resultados desses testes, não é possível classificar as heurísticas propostas, ILS-Band e ILS-Prof, como heurísticas de baixo custo para redução de largura de banda e de *profile*. Os tempos de execução das heurísticas propostas sugerem, portanto, que seu uso *não é apropriado* para problemas práticos em que o tempo de execução é um fator limitante, como o problema de redução do custo do método GMRES pré-condicionado.

No trabalho de Oliveira, Bernardes e Chagas (2016), a heurística NSloan foi considerada a mais promissora heurística de baixo custo para redução de *profile* em instâncias simétricas, e a heurística MPG, para instâncias simétricas. Neste trabalho, no entanto, a heurística MPG foi a melhor heurística em redução de *profile* tanto para instâncias simétricas quanto para instâncias assimétricas, em relação à métrica  $\rho$ .

Por fim, lembramos que a heurística VNS-Band (MLADENOVIC et al., 2010) pode ser considerada o estado da arte para redução de largura de banda quanto à qualidade das soluções. Nos testes conduzidos neste capítulo com instâncias simétricas, a heurística VNS-Band foi a pior em redução de largura de banda e também a mais lenta entre as doze heurísticas testadas. Nos testes conduzidos com instâncias assimétricas, a heurística VNS-Band foi a oitava em redução de largura de banda e a quarta mais lenta. As heurísticas RBFS-GL, KP-Band-GL e RCM-GL, que foram piores que a heurística VNS-Band em redução de largura de banda nas instâncias assimétricas foram, no entanto, as mais rápidas nesse conjunto de instâncias. Uma explicação para esse desempenho ruim é o tempo fixado nos testes deste trabalho, de 60 segundos para instâncias simétricas e 50 segundos para instâncias assimétricas. No trabalho original, a heurística VNS-Band foi testada sobre instâncias com até 1000 vértices apenas, e tempo máximo de 500 segundos (MLADENOVIC et al., 2010). Com base nos testes deste capítulo, não se pode afirmar que a heurística VNS-Band não está no estado da arte quanto à qualidade das soluções obtidas em redução de largura de banda, mas, definitivamente, que ela não deve ser considerada apropriada para problemas práticos em que o tempo de execução é uma fator crucial.

## 7. RESOLUÇÃO DE SISTEMAS DE EQUAÇÕES LINEARES

A redução do custo de execução de métodos para resolução de sistemas de equações lineares é o foco deste trabalho. Neste capítulo, são descritos métodos iterativos para resolução de sistemas de equações lineares, com ênfase em métodos baseados no subespaço de Krylov, na Seção 7.1. Na Seção 7.2 são introduzidas técnicas para pré-condicionamento. Na Seção 7.3 apresenta-se uma breve compilação de trabalhos recentes que utilizaram o método GMRES pré-condicionado.

### 7.1 Métodos iterativos para resolução de SELs

Os métodos para resolução de sistemas de equações lineares podem ser divididos entre métodos diretos e iterativos. Para muitos problemas práticos, métodos diretos podem levar muito tempo para convergir ou divergir, como afirma Benzi (2002). Para esses casos, em que computar a solução exata do sistema de equações lineares é uma tarefa inviável, são utilizados métodos iterativos.

Dada uma solução inicial aproximada, métodos iterativos para resolução de SELs tentam, a cada passo, alterar alguns componentes dessa aproximação, até que a convergência seja alcançada, como explica Saad (2003, p.105). O objetivo de cada iteração é tentar eliminar componentes do vetor residual  $r_0 = b - Ax$ .

Entre os métodos iterativos, podem-se citar os métodos clássicos de Jacobi, Gauss-Seidel e *Successive Over-Relaxation* (SOR), métodos baseados no subespaço de Krylov e métodos multinível. Na seção a seguir, são apresentados métodos baseados no subespaço de Krylov, com foco no *Generalized Minimum Residual Method* (GMRES), utilizado nos testes deste trabalho.

#### 7.1.1 Métodos baseados no subespaço de Krylov

Métodos como o GMRES, BI-CGSTAB, MINRES e o MGC, entre outros, são baseados no subespaço de Krylov. Dada uma matriz  $A \in \mathbb{C}^{m \times m}$  e  $b \in \mathbb{C}^m$ ,

$$K_m = \text{span}\{b, Ab, A^2b, \dots, A^{m-1}b\}$$

é o subespaço de Krylov de dimensão  $m$ . Um método iterativo do subespaço de Krylov é um método para o qual o subespaço  $K_m$  é

$$K_m(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, \dots, A^{m-1}r_0\},$$

em que  $r_0 = b - Ax_0$ .

Os métodos iterativos do subespaço de Krylov variam, basicamente, na forma de construir uma aproximação para  $Ax = b$ , por meio de um vetor  $x_m \in K_m$ . Como explica Saad (2003), as aproximações obtidas por um método do subespaço de Krylov, para  $x_0 = 0$ , são da forma  $A^{-1}b \approx q_{m-1}(A)b$ , em que  $q_{m-1}$  é um polinômio de grau  $m - 1$ . Dessa forma, podemos dizer que a solução  $x = A^{-1}b$  é aproximada por  $q_{m-1}(A)b$ .

### 7.1.1.1 Convergência de métodos iterativos

O comportamento de convergência dos métodos do subespaço de Krylov depende do **número de condição** e da distribuição dos **autovalores** e da matriz de coeficientes. Como explica Saad (2003), autovalores agrupados em torno de um único ponto, distante da origem, implicam em convergência rápida, enquanto autovalores espalhados, especialmente em torno da origem, fazem com que a convergência seja lenta.

O número de condição  $\kappa(A)$  da matriz de coeficientes também desempenha um papel fundamental na convergência de métodos iterativos. O número de condição mede o quão sensível é a solução do problema em função de perturbações nos dados, como explicam Pyzara, Bylina e Bylina (2011). Em outras palavras, o número de condição está fortemente relacionado à precisão da solução aproximada do sistema fornecida por métodos iterativos. Diz-se que uma matriz  $A$  é *bem-condicionada* se seu número de condição  $\kappa(A)$  não é muito grande. Nesse caso, o resultado obtido por um método iterativo terá uma boa precisão. Do contrário, em matrizes *mal-condicionadas*, a solução obtida por um métodos iterativos, como os do subespaço de Krylov, irá gerar aproximações com erros grandes, em comparação à solução exata.

### 7.1.2 O método GMRES

Em 1952, Lanczos (1952) e Hestenes e Stiefel (1952) desenvolveram independentemente o métodos dos gradientes conjugados (MGC) para resolução de sistemas de equações lineares com matrizes simétricas e positivas definidas, como explica Benzi (2002). Posteriormente, iniciou-se uma busca por uma extensão desses métodos que pudesse ser aplicada a sistemas assimétricos e indefinidos. Essas pesquisas culminaram com o desenvolvimento do *Generalized Minimum Residual* (GMRES), por Saad e Schultz (1986). Como outros métodos

do subespaço de Krylov, o GMRES tenta aproximar a solução de um sistema  $Ax = b$  por meio de um vetor  $x_m \in K_m$ . Nesse caso, um vetor  $x_m$  que minimize o resíduo  $\|b - Ax_m\|_2$ .

Uma visão geral do método método GMRES é dada no Algoritmo 7.1. O algoritmo recebe um sistema de equações da forma  $Ax = b$ , o número máximo de iterações e a dimensão  $m$  do subespaço de Krylov. Nas linhas 3 e 4 são computados o vetor residual inicial e sua norma. Na linha 5, cria-se o vetor de Krylov inicial.

No laço de iteração das linhas 6 a 12 acontece a expansão do subespaço de Krylov e a projeção do sistema de equações original nesse espaço. Na base desse laço está um método chamado de Iteração de Arnoldi (ARNOLDI, 1951). Como explica Saad (2003), esse é um procedimento eficiente para aproximar autovalores de matrizes esparsas de grande porte. A Iteração de Arnoldi constrói uma projeção ortogonal do subespaço de Krylov  $K_m$ . A cada iteração, nas linhas 8 a 10, o processo de ortogonalização de Gram-Schmidt é utilizado para gerar uma sequência de vetores de Arnoldi ortonormais  $w$ .

Ao fim do laço interno, na linha 13, tem-se uma matriz de Hessenberg do tipo superior  $\bar{H}$ . Uma matriz deste tipo é quase triangular, com a diferença de possuir também elementos na subdiagonal imediatamente abaixo da diagonal principal. Na linha 14, encontra-se  $y_m$  que minimize  $\|\beta e_{-1} - \bar{H}_m y\|_2$ . Conforme explica Saad (2003), essa operação envolve um problema dos mínimos quadrados de ordem  $(m + 1) \times m$  em que  $m$  é tipicamente pequeno. Na linha 16, o algoritmo retorna a solução aproximada obtida.

---

**Algoritmo 7.1** GMRES - Fonte: Saad (2003)

---

```

1: GMRES
2:   Entrada: Sistema de equações lineares  $Ax = b$ , palpite inicial  $x_0$ , número máximo de
   iterações externas  $n_{outer}$ , dimensão do subespaço de Krylov  $m$ 
3:    $r_0 = b - Ax_0$  // Resíduo inicial
4:    $\beta = \|r_0\|_2$  // Norma residual inicial
5:    $v_1 = r_0/\beta$  // Define o primeiro vetor de Krylov
6:   para  $j = 1, \dots, m$  faça // Iteração interna do GMRES
7:      $w_j = Av_j$  // Produto matriz-vetor
8:     para  $i = 1, \dots, j$  faça // Ortogonalização de Gram-Schmidt
9:        $h_{i,j} = (w_j, v_i)$ 
10:       $w_j = w_j - h_{i,j}v_i$ 
11:      $h_{j+1,j} = \|w_j\|_2$ 
12:      $v_{j+1} = w_j/h_{j+1,j}$  // Define o próximo vetor de Krylov
13:    $\bar{H} = \{h_{i,j}\}$  // Define uma matriz reduzida do sistema
14:    $y = \operatorname{argmin}_{\hat{y}} \|\beta e_1 - \bar{H}\hat{y}\|_2$  // Resolução do sistema reduzido
15:    $x_m = x_0 + \sum_{i=1}^m y_i z_i$  // Formação da solução aproximada
16:   retorne  $x_m$ 

```

---

### 7.1.2.1 Método GMRES com reinícios

Um dos problemas com o algoritmo GMRES clássico é que o subespaço de Krylov é expandido continuamente até que a convergência seja alcançada. Isso eleva linearmente os custos de memória e pode tornar o algoritmo inviável. Para contornar esses problemas, certas implementações do GMRES utilizam soluções baseadas em reinícios. No Algoritmo 7.2, que ilustra o GMRES com reinícios, se a dimensão definida do subespaço de Krylov  $m$  não for suficiente, o algoritmo é reiniciado partindo da solução intermediária  $x$  já encontrada (linha 21).

Embora seja fundamental para limitar o consumo de recursos computacionais, o uso de reinícios não garante a convergência do método GMRES. Ainda, é consenso que o número de iterações  $m$  deve ser escolhido a partir de experimentações: se for muito pequeno, o algoritmo pode demorar a convergir; se for muito maior que o necessário, implica nos problemas de consumo de recursos já mencionados. O uso do método GMRES com pré-condicionadores, explicados a seguir, pode ajudar a acelerar sua convergência.

### 7.1.3 Pré-condicionamento

Embora métodos iterativos sejam mais atrativos do que métodos diretos para resolução de SELs de grande porte, eles não possuem tanta robustez quanto esses últimos. O uso de pré-condicionamento pode contribuir na melhoria da robustez e da eficiência (isto é, na convergência) de métodos iterativos (SAAD, 2003, p.275).

Pré-condicionar um sistema de equações significa, em suma, transformá-lo em outro sistema, mais fácil de resolver por meio de um método iterativo, como explica Ferronato (2012). Isso envolve encontrar uma matriz  $M$  que será aplicada no sistema de equações original.

Segundo Benzi (2002), Saad (2003), a matriz  $M$  deve atender aos seguintes requisitos: ela deve ser fácil de construir e de ser aplicada, e deve tornar o sistema pré-condicionado  $Mx = b$  mais simples de ser resolvido. Além disso, deve ser não singular (deve possuir inversa) e ser próxima de  $A$ , de algum modo.

Uma matriz pré-condicionante  $M$  pode ser aplicada de três formas. No pré-condicionamento à esquerda, temos:

$$M^{-1}Ax = M^{-1}b.$$

O pré-condicionamento à direita é da forma:

$$AM^{-1}u = b, x \equiv M^{-1}u.$$

Finalmente, o pré-condicionador utilizado pode estar disponível na forma  $M = M_L M_R$ , em que  $M_L$  e  $M_R$  são, em geral, matrizes triangulares, como explica Saad (2003, p.276). Nesse caso, o pré-condicionamento é dito *dividido* (*split preconditioning*):

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

A escolha da forma de pré-condicionamento depende da aplicação, método iterativo utilizado e características do problema. Note que, em qualquer um dos casos, o sistema de equações pré-condicionado possui a mesma solução do sistema  $Ax = b$  original.

#### 7.1.4 Método GMRES pré-condicionado

As três formas de pré-condicionamento apresentadas podem ser utilizadas no método GMRES. Como explica Saad (2003, p.285-287), na maioria das situações práticas, não existe diferença significativa na convergência do método ao se escolher uma ou outra. A exceção se dá em casos em que  $M$  é mal-condicionada. Nessas situações, como explicado, a solução do sistema está propensa a grandes variações, dadas pequenas alterações em etapas intermediárias.

Entretanto, ainda segundo Saad (2003), o uso do do pré-condicionador à direita é preferível por possibilitar a aplicação de uma variante "flexível" do GMRES, em que o pré-condicionador pode variar a cada passo. Essa flexibilidade permite também que qualquer técnica iterativa possa ser usada como pré-condicionador, como o SSOR, *multigrid*, etc., e mesmo outros métodos iterativos como o próprio GMRES e o método dos gradientes conjugados.

Na versão do método GMRES pré-condicionado à direita, a base ortogonal do subespaço de Krylov construída pela iteração de Arnoldi assume a forma

$$\text{span}\{r_0, AM^{-1}r_0, \dots, (AM^{-1})^{m-1}r_0\}.$$

O Algoritmo 7.2 apresenta a versão flexível do GMRES (FGMRES - *Flexible GMRES*) com pré-condicionamento à direita. A adaptação do Algoritmo 7.1 para esta versão envolve definir o passo pré-condicionante  $z_j = M_j^{-1}v_j$  (linha 8) e substituir o produto  $w = Av_j$  (linha

9) por  $w = Az_j$ . O cálculo da solução aproximada, na linha 17, passa a ser definido por  $x = x_0 + \sum_{i=1}^m y_i z_i$ . Isto é, com essas modificações, as operações passam a ser feitas sobre a matriz pré-condicionada.

---

**Algoritmo 7.2** *Flexible GMRES* com pré-condicionamento à direita - Fonte: Saad (2003)

---

```

1: FGMRES
2:   Entrada: Sistema de equações lineares  $Ax = b$ , palpite inicial  $x_0$ , pré-condicionador  $M$ ,
   número máximo de iterações externas  $n_{outer}$ , dimensão do subespaço de Krylov  $m$ 
3:   para  $l = 1, \dots, n_{outer}$  faça
4:      $r_0 = b - Ax_0$  // Resíduo inicial
5:      $\beta = \|r_0\|_2$  // Norma residual inicial
6:      $v_1 = r_0/\beta$  // Define o primeiro vetor de Krylov
7:     para  $j = 1, \dots, m$  faça // Iteração interna do GMRES
8:        $z_j = M_j^{-1}v_j$  // Passo pré-condicionador
9:        $w = Az_j$  // Produto matriz-vetor
10:      para  $i = 1, \dots, j$  faça // Ortogonalização de Gram-Schmidt
11:         $h_{i,j} = (w, v_i)$ 
12:         $w = w - h_{i,j}v_i$ 
13:       $h_{j+1,j} = \|w\|_2$ 
14:       $v_{j+1} = w/h_{j+1,j}$  // Define o próximo vetor de Krylov
15:       $\tilde{H} = \{h_{i,j}\}$  // Define uma matriz reduzida do sistema
16:       $y = \underset{\hat{y}}{\operatorname{argmin}} \|\beta e_1 - \tilde{H}\hat{y}\|_2$  // Resolução do sistema reduzido
17:       $x = x_0 + \sum_{i=1}^m y_i z_i$  // Formação da solução aproximada
18:      se  $\|\beta e_1 - \tilde{H}\hat{y}\|_2 < \varepsilon$  então // Verificação de convergência
19:        retorne
20:      senão
21:         $x_0 = x$  // Solução reinicializada para próxima iteração

```

---

### 7.1.5 Paralelização do método GMRES

A resolução de sistemas de equações lineares de grande porte com métodos iterativos, principalmente que envolvem matrizes mal-condicionadas, pode demandar muito tempo. Diversos estudos têm se preocupado em desenvolver versões paralelas do método GMRES, aproveitando-se do fato de que muitas das operações envolvidas no algoritmo (produtos vetoriais, escalares e normas, por exemplo) são altamente paralelizáveis. Como exemplos de trabalhos que propuseram versões paralelas do método GMRES, podem ser citados os de Brozolo e Robert (1989), Cunha e Hopkins (1994), Erhel (1995).

Mais recentemente, a resolução de SELs utilizando GPUs (*Graphics Processing Unit*) tem sido o foco de pesquisas da área. O esforço principal desses estudos têm sido em desenvolver rotinas, ou *kernels*, para acelerar operações de produtos vetoriais envolvendo matrizes

esparças (*sparse matrix-vector product* - SpMV), requisitadas exaustivamente em resolutores iterativos.

No trabalho de Couturier e Domas (2012), uma implementação em GPUs do método GMRES com reinícios é proposta. Mesmo utilizando uma forma simples de pré-condicionamento, em que  $M$  é igual à diagonal de  $A$ , os autores obtiveram melhoras de até 23 vezes no tempo de processamento em comparação à versão sequencial do resolutor.

*Clusters* de GPUs também foram estudados por Khodja et al. (2014). Os autores apontam que a resolução de sistemas de equações lineares de grande porte é mais eficiente em *clusters* de GPUs do que em *clusters* de CPUs, desde que a localidade espacial e temporal dos dados seja bem gerenciada. Além disso, os autores utilizam técnicas de reordenação das submatrizes para minimizar o *overhead* de comunicação entre os GPUs.

O trabalho de DeVries et al. (2013) inclui versões paralelas do *Flexible* GMRES (FGMRES) para diversas arquiteturas, como GPUs, sistemas multi-CPU e multi-GPU. Os autores apontam *speedup* significativo, principalmente na versão híbrida multi-GPU com OpenMP + CUDA.

He et al. (2016) propõem uma versão do GMRES pré-condicionado chamada GPU-GMRES, que utiliza uma abordagem híbrida CPU-GPU, com unidades gráficas dispostas em *clusters*. A fatoração LU incompleta implementada no GPU-GMRES permite o ajuste do *fill-in* de maneira a melhor aproveitar a eficiência das GPUs, segundo os autores. Os resultados obtidos em instâncias provenientes de problemas com redes de energia apontam que a abordagem proposta propicia um significativo *speedup* sobre a versão sequencial do mesmo resolutor.

Na seção a seguir, são apresentadas técnicas de pré-condicionamento comumente utilizadas em conjunto com o método GMRES. Apresentam-se técnicas baseadas na fatoração incompleta da matriz  $A$  bem como uma técnica baseada em *multigrid*.

## 7.2 Técnicas de pré-condicionamento

De maneira geral, um pré-condicionador consiste numa forma de modificação, ou aproximação, de um sistema de equações que o torne mais fácil de ser resolvido por um método iterativo. A quantidade de maneiras pelas quais se pode aplicar pré-condicionamento em um sistema de equações lineares é, deste modo, praticamente ilimitada. Como nota Saad (2003, p.297), “encontrar um bom pré-condicionador para resolver um dado sistema linear esparso é, frequentemente, visto como combinação de arte e ciência.”

Apesar disso, quando se trata de métodos iterativos baseados no subespaço de Krylov, uma das famílias de pré-condicionadores mais utilizada é a dos baseados na fatoração incompleta da matriz de coeficientes  $A$ . Como explica Dutto (1993), esse tipo de pré-condicionador fornece uma boa aceleração para métodos iterativos como o MGC e o GMRES, além de ser fácil de gerar e usar.

A fatoração LU incompleta consiste em decompor a matriz  $A$  como  $A = LU - R$ :

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix} - \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix},$$

em que  $L$  é uma matriz diagonal inferior,  $U$  é uma matriz diagonal superior, e  $R$  é a matriz de resíduo, ou erro, da fatoração. A matriz  $R$  contém os elementos descartados durante o processo de fatoração incompleta.

Diz-se que a fatoração LU de uma matriz  $A$  apresenta *fill-in* quando posições que eram nulas em  $A$  passam a ser não-nulas nos fatores LU. Isso significa que, tanto a matriz triangular inferior  $L$  (*lower*) quanto a matriz triangular superior  $U$  (*upper*), podem ser menos esparsos (ou mais densos) que  $A$ . Vários critérios podem ser utilizados para descartar os elementos em  $L$  e  $U$  e controlar o *fill-in* da fatoração. Nos métodos considerados a seguir, esses critérios são a posição dos elementos em  $A$ , seu valor ou uma combinação de ambos.

### 7.2.1 ILU(k)

A posição dos fatores na matriz pode ser um critério utilizado para descartar o *fill-in* nas etapas de fatoração LU incompleta. Seja  $P$  o padrão de elementos nulos de  $A$ , tal que:  $P \subset \{(i, j) | i \neq j; 1 \leq i, j \leq n\}$ . A fatoração LU incompleta sem *fill-in*, ILU(0), pode ser definida como é mostrado no Algoritmo 7.3.

Esse tipo de fatoração é dita sem *fill-in* porque o padrão de elementos nulos  $P$  dos fatores  $LU$  coincide com as posições nulas de  $A$ . Como afirma Benzi (2002), a fatoração ILU(0) é bastante simples de se implementar e pode ser eficiente para certas aplicações. Contudo, na maior parte dos problemas reais, a aproximação de  $A$  que ela fornece pode não ser precisa o suficiente para garantir uma boa taxa de convergência do resolutor.

---

**Algoritmo 7.3** ILU(0) - Fonte: Saad (2003)
 

---

```

1: ILU(0)
2:   para  $i = 2, \dots, n$  faça
3:     para  $k = 1, \dots, i - 1$  se  $(i, k) \in P$  faça
4:        $a_{ik} = a_{ik}/a_{kk}$ 
5:     para  $j = k + 1, \dots, n$  se  $(i, j) \in P$  faça
6:        $a_{ij} = a_{ij} - a_{ik}a_{kj}$ 

```

---

Uma maneira de remediar a falta de precisão do ILU(0) é permitir algum *fill-in* durante a fatoração. Para tanto, define-se uma hierarquia de pré-condicionadores baseada no conceito de “níveis de *fill-in*”, chamada ILU(k). Inicialmente, o nível de *fill-in*  $lev_{ij}$  de um elemento não nulo é definido como 1 e o de um elemento nulo,  $\infty$ . A cada modificação sofrida pelo elemento durante a fatoração, seu nível de *fill-in* é atualizado de acordo com:

$$lev_{ij} = \min\{lev_{ij}, lev_{ik} + lev_{kj} + 1\}. \quad (7.1)$$

O Algoritmo 7.4 mostra o pseudocódigo da fatoração ILU(k), onde  $k$  corresponde aos níveis de *fill-in*. No ILU(k),  $k \geq 0$ , todos os elementos cujo nível de *fill-in* forem maiores que  $k$  serão descartados. Quanto maior o valor de  $k$ , menores tendem a ser os valores dos elementos remanescentes. Contudo, valores de  $k$  altos podem elevar demais o custo de geração do pré-condicionador.

---

**Algoritmo 7.4** ILU(k) - Fonte: Saad (2003)
 

---

```

1: ILU(k)
2:   Para todo elemento não nulo  $a_{ij}$  defina  $lev(a_{ij}) = 0$ 
3:   para  $i = 2, \dots, n$  faça
4:     para  $j = 1, \dots, i - 1$  e para  $lev(a_{ij}) \leq k$  faça
5:       Compute  $a_{ij} = a_{ij}/a_{jj}$ 
6:       Compute  $a_{i*} = a_{i*} - a_{ij}a_{j*}$ 
7:       Atualize os níveis de fill-in dos elementos  $a_{ij}$  não-nulos de acordo com (7.1)
8:     Substitua qualquer elemento na linha  $i$  com  $lev(a_{ij}) > k$  para zero

```

---

Segundo constata Saad (2003), as desvantagens dessa abordagem são o custo de atualização dos níveis e o fato de se desconhecer, de antemão, a quantidade de *fill-in* gerada e o custo computacional envolvido na fatoração.

## 7.2.2 ILUT

Fatorações como o ILU(0), em que os elementos são descartados com base apenas na estrutura de  $A$  podem resultar em pré-condicionadores ineficientes em muitas situações práticas, como aponta Saad (2003, p.321). Pré-condicionadores mais robustos podem ser construídos ao se levar em conta a magnitude dos elementos, em vez de suas posições. Dessa forma, apenas os *fill-in* com magnitude que possam contribuir significativamente para a qualidade do pré-condicionador são mantidos.

A fatorização ILUT( $\tau, p$ ) (*Incomplete LU factorization with Threshold*) (SAAD, 1994) utiliza dois critérios para controlar o *fill-in*. A cada iteração, são descartados todos os elementos que forem menores que a tolerância de descarte  $\tau$  multiplicada pela norma-2 da linha. Dos elementos restantes, apenas os  $p$  maiores em magnitude são mantidos. Uma visão geral do ILUT é dada no Algoritmo 7.5.

---

### Algoritmo 7.5 ILUT - Fonte: Saad (2003)

---

```

1: ILUT
2:   para  $i = 1, \dots, n$  faça
3:      $w = a_{i*}$ 
4:     para  $k = 1, \dots, i - 1$  se  $w_k \neq 0$  faça
5:        $w_k = w_k / a_{kk}$ 
6:       Aplica-se a regra de descarte  $\tau$  a  $w_k$ 
7:       se  $w_k \neq 0$  então
8:          $w = w - w_k * u_{k*}$ 
9:       Aplica-se a regra de descarte  $p$  à linha  $w$ 
10:       $l_{i,j} = w_j$  para  $j = 1, \dots, i - 1$ 
11:       $u_{i,j} = w_j$  para  $j = i, \dots, n$ 
12:       $w = 0$ 

```

---

A utilização dos parâmetros  $\tau$  e  $p$  possibilita que se controle de antemão o *fill-in* máximo ( $p$ ) e que se garanta que os elementos remanescentes possuam magnitude significativa ( $\tau$ ). Nas situações em que o ILUT falha como pré-condicionador, em geral, pode-se obter convergência diminuindo o valor de  $\tau$  e aumentando o de  $p$ , permitindo-se, assim, mais *fill-in* na fatoração.

Uma dificuldade encontrada com a fatoração ILUT é a escolha dos parâmetros. Como aponta Benzi (2002), essa escolha depende fortemente das características do problema e, em geral, é feita por tentativa e erro.

### 7.2.3 ILUC

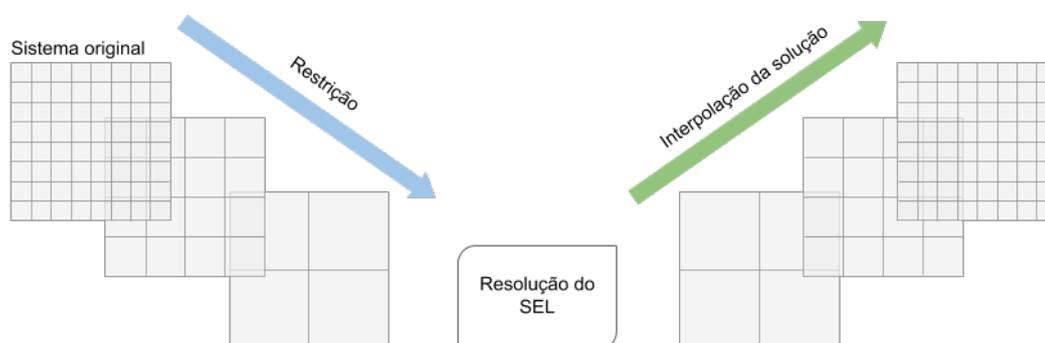
Algoritmos de fatoração baseados na eliminação Gaussiana precisam acessar os elementos em  $L$  e  $U$  por ordem das colunas. Contudo, os valores na linha atual são dinamicamente modificados pela fatoração. Quando a quantidade de *fill-in* é pequena, a busca pelo elemento desejado não incorre em muito custo. Do contrário, o custo dessa busca pode dominar o custo da fatoração em si.

A formulação Crout da fatoração ILU, ou ILUC (LI; SAAD; CHOW, 2003), fornece uma maneira mais eficiente de se percorrer e descartar os elementos de *fill-in*. Essa eficiência é conseguida ao se postergar a atualização dos elementos durante a fatoração. Como apontam os autores, o ILUC permite que estratégias mais robustas de escolhas de pivôs sejam incorporadas, bem como estratégias de descarte de valores baseadas em *threshold*, como a do ILUT.

### 7.2.4 ARMS

Na análise numérica, métodos *multigrid* são algoritmos para resolução de SEL usando uma hierarquia de aproximações (BRIGGS; MCCORMICK; HENSON, 2000). A ideia desses métodos é, a cada etapa, construir uma versão mais “grosseira” da matriz original. O processo é repetido recursivamente, até que atinja um ponto em que pode ser resolvido com custo muito mais baixo do que o problema original. A solução encontrada é, então, interpolada de volta até a versão original. A Figura 7.1 ilustra esse processo.

Figura 7.1 – Visão geral do funcionamento de um método *multigrid*. Para resolução de SELs, a matriz de entrada é aproximada por uma série de matrizes menores, até que possa ser resolvida.



O método Multigrid Algébrico (AMG - *Algebraic MultiGrid*), proposto originalmente por Ruge e Stüben (1987), é um exemplo de método *multigrid*. Sua vantagem sobre os métodos existentes até então era obter as matrizes grosseiras de forma automática, sem necessidade

de ajustes de vários parâmetros. Posteriormente, o AMG serviu de base para a criação de pré-condicionadores que combinam a fatoração LU incompleta com alguma forma de método *multigrid*, como explicam Saad, Soulaïmani e Touihri (2004).

Uma das estratégias que alia a fatoração ILU com o AMG é o *Algebraic Recursive Multilevel Solver* (ARMS), proposto por Saad e Suchomel (2002). Para criar a hierarquia de aproximações da matriz  $A$  original, o pré-condicionador ARMS utiliza o conceito de ordenações por conjuntos independentes (*independent set orderings*). Uma ordenação por conjunto independente transforma o sistema de equações lineares original em:

$$\begin{pmatrix} B & F \\ E & C \end{pmatrix} = \begin{pmatrix} I & 0 \\ EB^{-1} & I \end{pmatrix} \begin{pmatrix} B & F \\ 0 & S \end{pmatrix},$$

em que o bloco  $B$  é bloco diagonal. O *complemento de Schur* desse novo sistema de equações é uma matriz esparsa  $\mathfrak{S}$ , tal que  $\mathfrak{S} = C - EB^{-1}F$ . A ideia do ARMS é empregar essa fatoração de forma recursiva. A cada passo, são aplicadas regras para se limitar o *fill-in* de  $\mathfrak{S}$  e o sistema resultante passa por um novo processo de ordenação por conjuntos independentes. O processo é repetido por um número de níveis até que o sistema seja pequeno o bastante, ou até que o número máximo de níveis seja atingido. Então, o sistema é resolvido por algum outro método padrão, como o ILUT aliado ao método GMRES.

### 7.2.5 Pré-condicionadores em bloco

O pré-condicionamento em blocos é uma técnica comumente aplicada a matrizes provenientes da discretização de problemas elípticos, que pode ser generalizada a outras matrizes esparsas, como explica Saad (2003). Um pré-condicionador em bloco é definido como  $M = (L + \Delta)\Delta^{-1}(\Delta + U)$ , onde  $\Delta$  é uma matriz bloco diagonal.

Um dos problemas com esse tipo de pré-condicionadores é encontrar a estrutura em bloco da matriz de interesse, isto é, agrupar as linhas e colunas da matriz em blocos. Uma das abordagens para resolver esse problema foi proposta por (SAAD, 2002). Chamada de *Variable Block ILU*, essa abordagem identifica automaticamente esses blocos, utilizando métricas de similaridade entre os valores das linhas e colunas. A utilização dessa abordagem com *threshold* e níveis de *fill-in* dá origem, respectivamente, aos pré-condicionadores **VBILUT** e **VBILU(k)**.

### 7.2.6 Pré-condicionamento paralelo

Versões paralelas do método GMRES pré-condicionado devem fazer uso, preferencialmente, de implementações paralelas dos métodos pré-condicionadores. Os trabalhos de Zhang (2002) e de Ghysels et al. (2017) apresentam algumas abordagens para pré-condicionamento paralelo. Pré-condicionadores paralelos baseados em fatoração incompleta foram propostos por Karypis e Kumar (1997) e Hysom e Pothen (2001). Um pré-condicionador paralelo baseado em *multigrid* é proposto por Li, Saad e Sosonkina (2003). Ainda, versões de pré-condicionadores para GPUs são propostos no trabalho de Li e Saad (2013).

### 7.3 Trabalhos recentes com aplicações do método GMRES pré-condicionado

A escolha do melhor algoritmo de ordenação de matrizes para acelerar a convergência de métodos iterativos baseados no subespaço de Krylov tem sido assunto de vários estudos nas últimas décadas. Duff e Meurant (1989) analisam diversos algoritmos de ordenação na resolução de SEL com GC pré-condicionado pela fatoração incompleta de Cholesky. Para problemas como discretizações da equação de Laplace pelo método dos elementos finitos testados, os autores recomendam o uso da heurística CMR. O trabalho de Dutto (1993), compara alguns algoritmos de ordenação quando utilizados com o método GMRES pré-condicionado com ILU(p) em equações de Navier-Stokes. As ordenações aplicadas no trabalho, incluindo a de Cuthill-McKee reversa, melhoraram, em muito, a convergência do GMRES. Resultados similares foram reportados por Benzi, Szyld e Duin (1999). Os autores avaliaram os efeitos da reordenação da matriz de coeficientes em conjunto com o método GMRES pré-condicionado com as fatorações incompletas ILU(p) e ILUT. Os resultados mostraram que a heurística CMR foi, em geral, superior à heurística CM e a ordenação de grau mínimo.

O trabalho de Catabriga et al. (2006) investiga a aplicação do CMR em conjunto com GMRES e BICGSTAB em sistemas de equações assimétricos oriundos da discretização de problemas de fluxo e transportes pelo método dos elementos finitos. Nos testes envolvendo fatoração LU incompleta e refinamento adaptativo da malha, os melhores tempos de execução foram obtidos com a fatoração ILU(1). Guidetti et al. (2010), Lugon e Catabriga (2013a) analisam os efeitos da aplicação dos métodos de Sloan, GPS, grau mínimo aproximado, espectral, e variações do CMR e dissecação aninhada, na convergência do método GMRES pré-condicionado com fatoração LU incompleta. Os autores concluem que, em geral, a ordenação de Cuthill-McKee

reverso é a mais vantajosa para redução do custo computacional. Pessanha, Portugal e Saavedra (2009) propõem um pré-condicionador baseado no método CMR para o método GMRES e conseguem, com isso, sanar instabilidades da fatoração ILU em problemas de fluxo de carga.

## 8. SIMULAÇÕES PARA RESOLUÇÃO DE SELS COM O MÉTODO GMRES PRÉ-CONDICIONADO

Neste capítulo, mostram-se os resultados dos testes para resolução de sistemas de equações lineares com o método GMRES pré-condicionado com heurísticas para redução de largura de banda ou de *profile* selecionadas. A descrição dos métodos e instâncias utilizadas nos testes é apresentada na Seção 8.1. Os resultados das simulações para redução do custo computacional do método GMRES pré-condicionado encontram-se na Seção 8.2. Discussões sobre os resultados dos testes são apresentados na Seção 8.3.

### 8.1 Descrição das instâncias, heurísticas e pré-condicionadores utilizados

O objetivo dos testes para resolução de SELs deste capítulo é identificar possíveis heurísticas que produzam ordenações que façam com que o método GMRES pré-condicionado convirja mais rapidamente do que se executado sobre a ordenação original da matriz. Para que isso possa acontecer, além de produzir boas reduções de largura de banda ou de *profile*, a heurística utilizada deve ter um baixo custo de execução.

No trabalho de Oliveira, Bernardes e Chagas (2016), as heurísticas RCM-GL, KP-Band-GL, NSloan, Sloan-MGPS, MPG, GPS e FNC-HC foram identificadas como as mais promissoras heurísticas de baixo custo para redução de largura de banda ou de *profile* em, pelo menos, uma área de aplicação. As heurísticas RCM-GL, KP-Band-GL, NSloan, MPG, Sloan e RBFS-GL obtiveram os melhores resultados na redução do custo de execução do método dos gradientes conjugados pré-condicionado pela fatoração incompleta de Cholesky no trabalho de Oliveira, Bernardes e Chagas (2017). Por esse motivo, essas heurísticas foram selecionadas para os testes deste capítulo com o método GMRES pré-condicionado. As exceções foram as heurísticas GPS e FNC-HC, que apesar dos bons resultados em redução de largura de banda, não apresentaram custo de execução tão baixos nos testes realizados no Capítulo 6. Ainda, foram consideradas também variações do algoritmos de Cuthill-Mckee (CM) e busca em largura (BFS) sem inversão da solução gerada (BFS-GL e CM-GL, respectivamente), sem escolha de vértice pseudo-periférico como ponto de partida (RBFS e RCM, respectivamente) e sem ambas (BFS e CM, respectivamente).

Durante a seleção de instâncias, considerou-se que instâncias muito pequenas não seriam adequadas para se avaliar o desempenho dos métodos escolhidos. Além disso, o método GMRES é comumente utilizado em situações em que o método dos gradientes conjugados não

pode ser aplicado, isto é, instâncias assimétricas ou não positivas-definidas. Por esse motivo, foram selecionadas instâncias com mais de 400 mil vértices, assimétricas e com valores reais da *Suite Sparse Matrix Collection*. Essa triagem levou a 20 instâncias de cinco áreas de aplicação diferentes, mostradas na Tabela 8.1.

Os pré-condicionadores utilizados nos testes foram ILUT, ILU(k), ILUC, ARMS, VBI-LUT e VBILU(k), descritos no Capítulo 7. As implementações utilizadas do método GMRES, bem como a dos pré-condicionadores, foram as disponíveis na biblioteca ITSOL v2.0 (SAAD, 2017), de Yousef Saad, co-criador do método GMRES. A versão do GMRES disponível na biblioteca ITSOL v2.0 é o *Flexible* GMRES com pré-condicionamento à direita. A biblioteca ITSOL v2.0 implementa versões sequenciais dos métodos citados, e é escrita em C e Fortran. Por esse motivo, foram criados *wrappers* no projeto computacional em C++ para incluir a biblioteca e permitir que as soluções reordenadas das heurísticas testadas servissem de entrada para os resolutores então disponíveis.

O número de condição da matriz tem grande influência na velocidade de convergência e na precisão da solução obtida pelo GMRES. Se o sistema pré-condicionado gerado possuir um número de condição alto, é mais vantajoso abortar a execução do que permitir que ela leve muito tempo para convergir e ainda forneça uma solução ruim. Infelizmente, o custo de se calcular o número de condição de uma matriz de grande porte é extremamente alto. A implementação do método GMRES da biblioteca ITSOL fornece uma boa aproximação do número de condição. Denotada *condest*, ela consiste em calcular  $\|(LU)^{-1}e\|_{\infty}$ , onde  $e$  é um vetor de tamanho  $n$  com todos os elementos iguais a 1. Como afirma Benzi (2002), essa aproximação tem sido um bom indicativo, na prática, de que o pré-condicionamento aplicado não foi bem-sucedido e não é confiável. Para as simulações realizadas neste trabalho, considerou-se que um valor de *condest* maior que  $10^{15}$  seria o limite para abortar a execução do GMRES. Essa escolha foi amparada pelo valor escolhido por Chow e Saad (1997), onde foi usado  $condest = 10^{10}$  como limite, levando-se em conta que as instâncias utilizadas neste trabalho possuem maiores dimensões. Quando o vetor  $b$  não foi fornecido, utilizou-se um vetor de valores aleatórios entre -1 e 1.

Durante testes preliminares identificou-se que, com os pré-condicionadores escolhidos, os piores tempos de execução não passavam de alguns segundos. Assim, e permitindo alguma margem para instâncias mais complexas, estabeleceu-se o tempo máximo de execução 1200 segundos, ou 20 minutos. Após esse tempo, as execuções foram abortadas. Além disso, foi definido que o GMRES seria encerrado quando a norma do vetor de erro residual fosse menor

Tabela 8.1 – 20 instâncias reais e assimétricas de grande porte utilizadas nos testes para resolução de SELs com o método GMRES pré-condicionado. O tamanho da matriz é indicado por  $n$  e o número de elementos não nulos, por  $nnz$

| Instância    | $n$       | $nnz$      | Área de aplicação                           |
|--------------|-----------|------------|---------------------------------------------|
| rajat21      | 411.676   | 1.876.011  | <i>Circuit Simulation Problem</i>           |
| largebasis   | 440.02    | 5.240.084  | <i>Optimization Problem</i>                 |
| cage13       | 445.315   | 7.479.343  | <i>Directed Weighted Graph</i>              |
| rajat30      | 643.994   | 6.175.244  | <i>Circuit Simulation Problem</i>           |
| rajat29      | 643.994   | 3.760.246  | <i>Circuit Simulation Problem</i>           |
| pre2         | 659.033   | 5.834.044  | <i>Circuit Simulation Problem</i>           |
| ASIC_680ks   | 682.712   | 1.693.767  | <i>Circuit Simulation Problem</i>           |
| ASIC_680k    | 682.862   | 2.638.997  | <i>Circuit Simulation Problem</i>           |
| tmt_unsym    | 917.825   | 4.584.801  | <i>Electromagnetics Problem</i>             |
| webbase-1M   | 1.000.005 | 3.105.536  | <i>Directed Weighted Graph</i>              |
| atmosmodj    | 1.270.432 | 8.814.880  | <i>Computational Fluid Dynamics Problem</i> |
| atmosmodd    | 1.270.432 | 8.814.880  | <i>Computational Fluid Dynamics Problem</i> |
| Hamrle3      | 1.447.360 | 5.514.242  | <i>Circuit Simulation Problem</i>           |
| atmosmodl    | 1.489.752 | 10.319.760 | <i>Computational Fluid Dynamics Problem</i> |
| cage14       | 1.505.785 | 27.130.349 | <i>Directed Weighted Graph</i>              |
| memchip      | 2.707.524 | 13.343.948 | <i>Circuit Simulation Problem</i>           |
| FullChip     | 2.987.012 | 26.621.983 | <i>Circuit Simulation Problem</i>           |
| Freescall    | 3.428.755 | 17.052.626 | <i>Circuit Simulation Problem</i>           |
| circuit5M_dc | 3.523.317 | 14.865.409 | <i>Circuit Simulation Problem</i>           |
| rajat31      | 4.690.002 | 20.316.253 | <i>Circuit Simulation Problem</i>           |

que  $10^{-8}$  ou quando se alcançasse um número de iterações igual a  $|n|$ , isto é, igual ao tamanho da matriz.

As implementações dos pré-condicionadores mencionados envolve um certo número de parâmetros que precisam ser definidos, e dos quais depende o desempenho do método GMRES. A fim de se realizar a escolha desses parâmetros da forma mais adequada, realizou-se uma série de simulações cuja descrição se encontra no Apêndice B.

Os valores encontrados nos testes descritos no Apêndice B encontram-se na coluna "1ª tentativa" na Tabela 8.2. Quando a execução do método GMRES com determinado pré-condicionador não convergiu com os parâmetros listados nessa coluna, foram realizadas novas tentativas com o conjunto de valores seguinte, até os valores listados na coluna "3ª tentativa". Os valores dos parâmetros para essas novas tentativas foram escolhidos de forma a permitir mais

*fill-in* na fatoração e aumentar as chances e convergência do método GMRES, como descrito no Capítulo 7.

Nas instâncias em que, após execuções com os três primeiros conjuntos de valores descritos na Tabela 8.2, não houve convergência do método GMRES com nenhuma combinação de pré-condicionador e heurística de renumeração de vértices, uma nova tentativa foi feita. Os valores utilizados nesta situação são os apresentados na coluna "4ª tentativa". Como o *fill-in* esperado nesta tentativa tende a ser maior, optou-se por aumentar o tempo limite dessas execuções para 3600 segundos, ou 1 hora.

Tabela 8.2 – Parâmetros utilizados nos testes com o GMRES pré-condicionado a cada tentativa sem convergência. Os valores da 4ª tentativa foram utilizados apenas nas instâncias em que não houve convergência em nenhuma das tentativas anteriores.

| Pré-condicionador  | 1ª tentativa                   | 2ª tentativa                   | 3ª tentativa                   | 4ª tentativa                   |
|--------------------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|
| ILUT, VBILUT, ILUC | $\tau = 10^{-3}$<br>p = 50     | $\tau = 10^{-6}$<br>p = 75     | $\tau = 10^{-9}$<br>p = 100    | $\tau = 10^{-15}$<br>p = 250   |
| ILU(k), VBILU(k)   | k = 1                          | k = 2                          | k = 4                          | k = 12                         |
| ARMS               | tol_dd = 0.1<br>num_niveis = 3 | tol_dd = 0.4<br>num_niveis = 3 | tol_dd = 0.7<br>num_niveis = 3 | tol_dd = 0.7<br>num_niveis = 5 |

Para os testes computacionais realizados nas 14 menores instâncias (em número de vértice) assimétricas da base *SuiteSparse Matrix Collection* utilizou-se um computador com processador Intel Core i7-4770 CPU 3.40GHz com 8MB de cache L3 e 8GB de memória RAM DDR3 1333MHz. O sistema operacional utilizado foi o Ubuntu 16.04.3 LTS com 64 bits e kernel 4.13.0-39-generic. As 6 maiores instâncias em número de vértices (*cake14*, *memchip*, *FullChip*, *Freescale1*, *circuit5M\_dc* e *rajat31*) foram executadas em máquinas similares, porém com 16GB de memória RAM DDR3 1333MHz.

## 8.2 Simulações com 20 instâncias assimétricas

Nas tabelas 8.4 a 8.8 no fim este capítulo são mostrados os tempos totais de execução do método GMRES pré-condicionado por ILUT, ILU(k), ILUC, ARMS, VBILUT e VBILU(k) com ordenações produzidas pelas heurísticas RBFS-GL, RCM-GL, RBFS, RCM, BFS-GL, CM-GL, BFS, CM, NSloan, Sloan, Sloan-MGPS e MPG. As instâncias foram agrupadas por área de atuação. Em cada uma dessas tabelas, a primeira coluna mostra o pré-condicionador utilizado.

Na segunda coluna encontra-se o tempo de resolução da instância com sua numeração original, isto é, sem utilização de heurísticas de redução de largura de banda ou de *profile*. O símbolo \* indica que a simulação foi abortada por alguma inconsistência durante a etapa de pré-condicionamento, como diagonais ou pivôs nulos, por exemplo. Os casos em que a execução foi abortada porque o número de condição estimado (*condest*) foi acima do limite estabelecido de  $10^{15}$  são indicados por “cond”. O símbolo † indica que a execução do método GMRES foi abortada por não convergir dentro do limite estabelecido de 1200 segundos. Nas situações em que houve convergência após a primeira tentativa, um índice sobrescrito indica em qual tentativa a convergência ocorreu. Por meio dele é possível identificar o valor dos parâmetros utilizado nas execuções. Nas instâncias em que foi realizada a tentativa extra com o quarto conjunto de parâmetros e com 3600 segundos de limite, o símbolo ‡ indica que a execução foi abortada por exceder o tempo limite. Os melhores tempos de execução por pré-condicionador são destacados em cinza claro. Os melhores resultados para cada instância, entre todos os pré-condicionadores, é indicado em cinza escuro. Nas seções a seguir, os resultados das simulações para cada área de atuação são analisados.

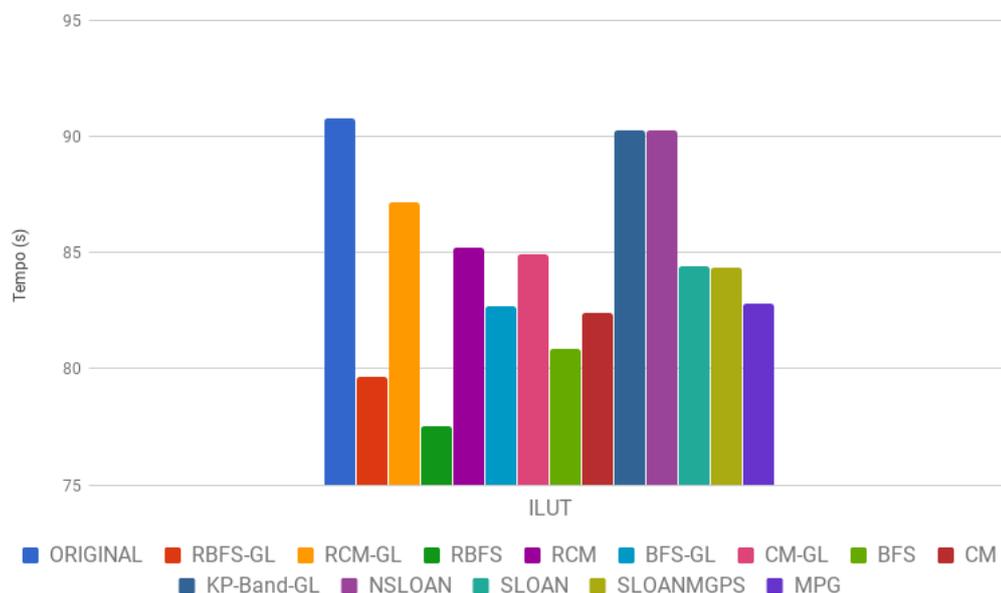
### **8.2.1 Simulações com o método GMRES pré-condicionado nas instâncias da área *Electromagnetics problem***

Os resultados das simulações com o método GMRES pré-condicionado nas instâncias da área *Electromagnetics problem* encontram-se na Tabela 8.4. Apenas uma instância que atendessem aos critérios estabelecidos foi encontrada para essa área de atuação (*tmt\_unsym*).

Nessa instância, o método GMRES convergiu apenas com o pré-condicionador ILUT. Os tempos por heurística são mostrados na Figura 8.1. O menor tempo de convergência foi observado no método GMRES pré-condicionado por ILUT com a numeração produzida pela heurística RBFS.

Nos outros pré-condicionadores testados, ILUC, ILU(k), ARMS, VBILUT e VBILU(k), as simulações não convergiram no tempo máximo estabelecido de 20 minutos. Nota-se também que não houve execuções abortadas por inconsistências na fatoração ou por estimativa de condição alta nessa instância.

Figura 8.1 – Tempo de execução do método GMRES pré-condicionado na instância da área *Electromagnetics problem* com os pré-condicionadores ILUT e ARMS



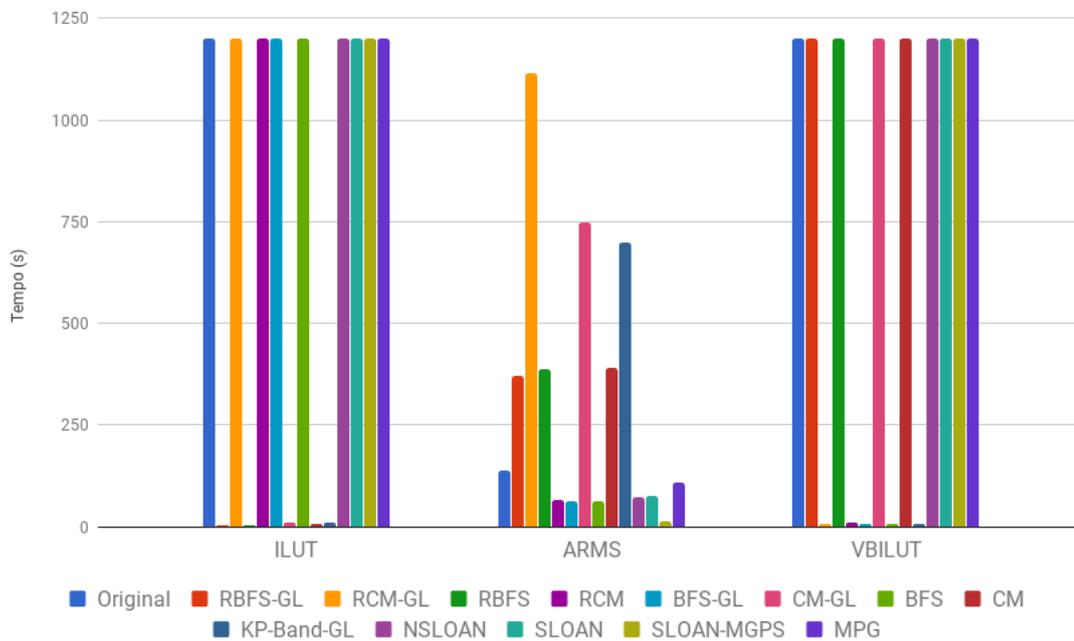
### 8.2.2 Simulações com o método GMRES pré-condicionado nas instâncias da área *Optimization problem*

Os resultados das simulações com o método GMRES pré-condicionado nas instâncias da área *Optimization problem* encontram-se na Tabela 8.5. Apenas uma instância que atendesse aos critérios estabelecidos foi encontrada para essa área de atuação (*largebasis*).

Dos seis os pré-condicionadores testados, em três houve convergência, ILUT, VBILUT e ARMS, mesmo que em poucas combinações de heurísticas e pré-condicionador. Na Figura 8.2, os valores altos de tempos de execução indicam que o método GMRES não convergiu com aquela heurística. No geral, a convergência mais rápida foi observada com o GMRES pré-condicionado com ILUT utilizando com a ordenação produzida pela heurística RBFS (3,85 segundos). Contudo, nota-se que o pré-condicionamento com ILUT e VBILUT sofreu de instabilidades com muitas das heurísticas de renumeração de vértices testadas na instância dessa área.

O método GMRES pré-condicionado com ARMS foi o mais estável nos testes com essa instância, convergindo com todas as heurísticas testadas. O menor tempo de execução nas simulações com o pré-condicionador ARMS foi com a heurística Sloan-MGPS.

Figura 8.2 – Tempo de execução do método GMRES pré-condicionado na instância da área *Optimization problem* com os pré-condicionadores ILUC e ARMS



### 8.2.3 Simulações com o método GMRES pré-condicionado nas instâncias da área *Directed Weighted Graph problem*

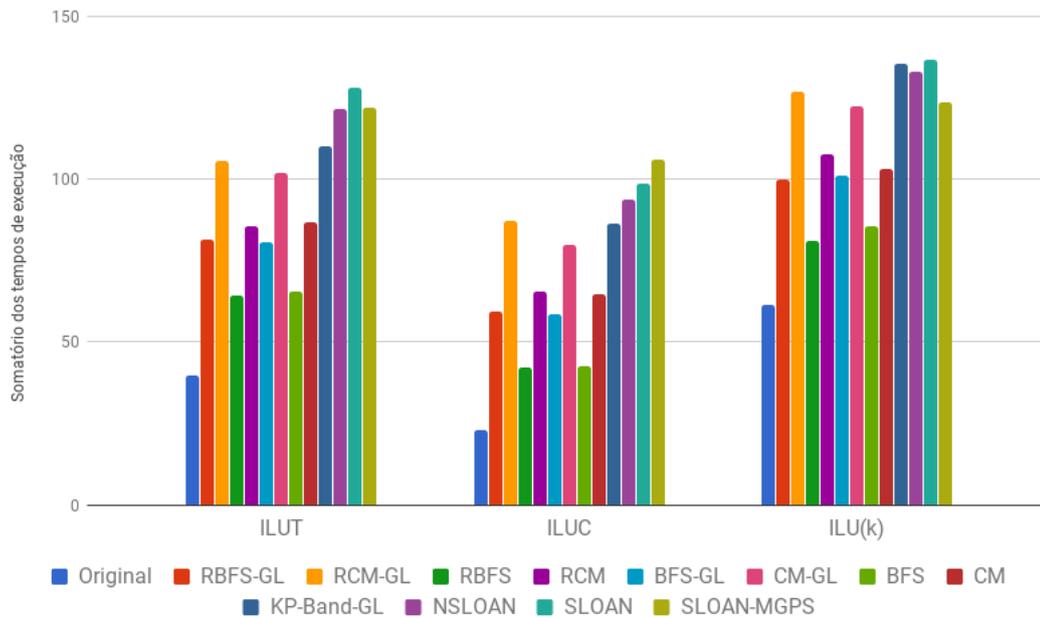
Os resultados das simulações com o método GMRES pré-condicionado nas três instâncias da área *Directed Weighted Graph problem* encontram-se na Tabela 8.6. À exceção da instância *webbase-1M*, nas outras instâncias houve convergência em, pelo menos, uma combinação de pré-condicionador e heurística de redução de largura de banda ou de *profile*.

Nas instâncias *cage13* e *cage14*, a ordenação original da matriz produziu o maior número de melhores soluções. Além disso, em ambas as instâncias, o método GMRES pré-condicionado com ILUC convergiu nos menores tempos entre os pré-condicionadores testados. Os pré-condicionadores ILUC, ILUC e ILU(k) levaram à convergência do método GMRES com todas as heurísticas de reordenação de vértices nas duas instâncias, com exceção da heurística MPG na instância *cage14*. Os tempos de execução do método GMRES com esses pré-condicionadores são ilustrados na Figura 8.3. Em particular, o pré-condicionador ARMS obteve resultados ruins em instâncias dessa área, conseguindo convergência apenas com a ordenação original na instância *cage13*.

Nenhuma combinação de pré-condicionador e heurística de renumeração de vértices levou à convergência do método GMRES na instância *webbase-1M*. Essa instância foi a sub-

metida à execução com o quarto conjunto de parâmetros indicado na Tabela 8.2. Ainda assim, não houve convergência do método GMRES nessa instância.

Figura 8.3 – Tempo de execução do método GMRES pré-condicionado nas instâncias da área *Directed Weighted Graph problem* com os pré-condicionadores ILUT, ILUC e ILU(k)



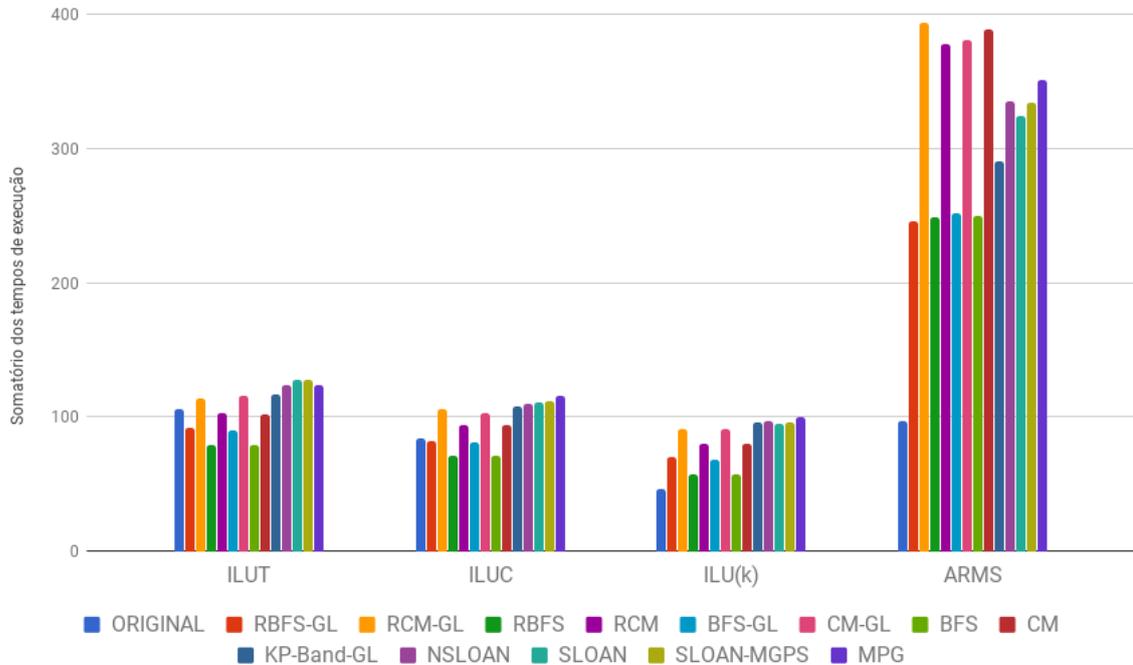
#### 8.2.4 Simulações com o método GMRES pré-condicionado nas instâncias da área *Computational Fluid Dynamics problem*

Os resultados das simulações com o método GMRES pré-condicionado nas quatro instâncias da área *Computational Fluid Dynamics problem* encontram-se na Tabela 8.7. Nas três instâncias dessa área, houve convergência do método GMRES com todos os pré-condicionadores, com exceção dos baseados em bloco, VBILUT e VBILU(k). Além disso, o método GMRES convergiu com as numerações produzidas por todas as heurísticas testadas quando pré-condicionado com ILUT, ILUC, ILU(k) e ARMS. A soma dos tempos de execução do método GMRES pré-condicionado nessas instâncias é mostrada na Figura 8.4.

Em geral, os menores tempos de execução foram obtidos com o pré-condicionador ILU(k). O menor tempo de execução em cada instância dessa área foi obtido com a numeração original da matriz em combinação com o método GMRES pré-condicionado com ILU(k). Notamos ainda que, entre os métodos heurísticos, os baseados em variações de busca em largura, BFS, RBFS e BFS-GL foram responsáveis pelo maior número de melhores resultados

nas instâncias dessa área com os pré-condicionadores testados, como pode ser observado na Tabela 8.7.

Figura 8.4 – Soma dos tempos de execução do método GMRES pré-condicionado nas instâncias da área *Computational Fluid Dynamics problem* com os pré-condicionadores ILUT, ILUC e ILU(k)



### 8.2.5 Simulações com o método GMRES pré-condicionado nas instâncias da área *Circuit Simulation problem*

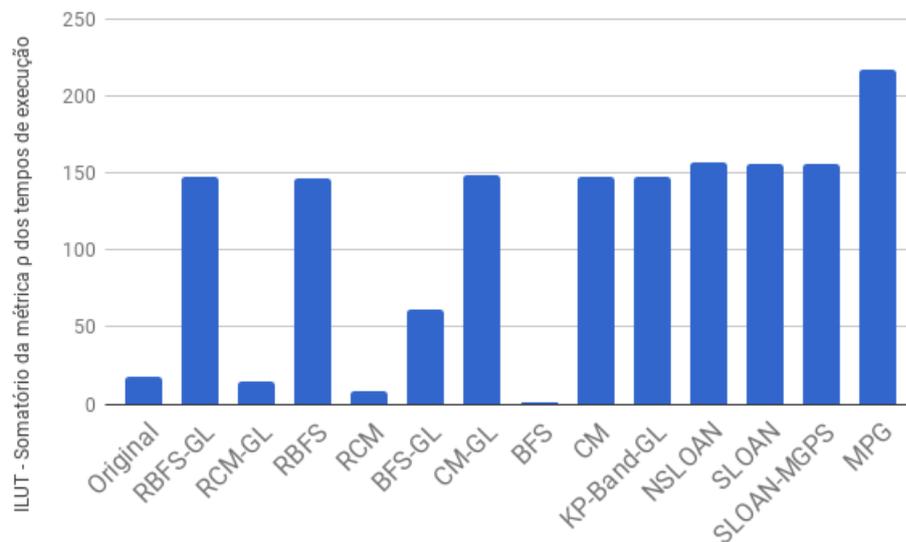
Os resultados das simulações com o método GMRES pré-condicionado nas instâncias da área *Circuit Simulation problem* encontram-se na Tabela 8.8. Essa área contempla o maior número de instâncias utilizadas nos testes, 12 das 20 instâncias. Além disso, muitas das maiores instâncias selecionadas pertencem a essa área. Talvez por isso, muitos dos pré-condicionadores testados apresentaram instabilidades e falharam em contribuir com a convergência do método GMRES em algumas das instâncias. Em três instâncias dessa área, nenhum pré-condicionador foi capaz de auxiliar na convergência do método GMRES no tempo estabelecido. São elas *Hamrle3*, *FullChip* e *pre2*.

O método GMRES pré-condicionado com ILU(k) convergiu em 5 das 12 instâncias dessa área. Em nenhum desses casos, o *speedup* obtido por esse pré-condicionador foi o melhor alcançado nessas instâncias. Nesse conjunto de instâncias, os pré-condicionadores em bloco VBILUT e VBILU(k) contribuíram com a convergência do método GMRES em apenas 1

instância, *ASIC\_680ks*. Entretanto, o pior desempenho entre os pré-condicionadores nas instâncias dessa área foi o do ILUC. Apenas na instância *rajat31*, com as ordenações produzidas pelas heurísticas CM-GL e CM houve convergência do GMRES com esse pré-condicionador. Esses resultados sugerem que a formulação Crout do ILUT não é a mais adequada para instâncias dessa área de atuação.

Os pré-condicionadores que forneceram os melhores resultados nesse conjunto de instâncias foram ILUT e ARMS. O método GMRES pré-condicionado com ILUT convergiu com, pelo menos, uma heurística de redução de largura de banda ou de *profile* em 6 das 12 instâncias dessa área. A soma dos tempos de execução nessas instâncias com o pré-condicionador ILUT é mostrado na Figura 8.5. Em todas essas instâncias em que houve convergência do método GMRES com o ILUT, o maior *speedup* entre os pré-condicionadores foi justamente o obtido por esse pré-condicionador. Em 3 dessas instâncias, a melhor ordenação foi a ordenação original da matriz. As heurísticas BFS e RBFS foram responsáveis por, respectivamente, 2 e 1 melhor resultado quando usadas em conjunto com o ILUT.

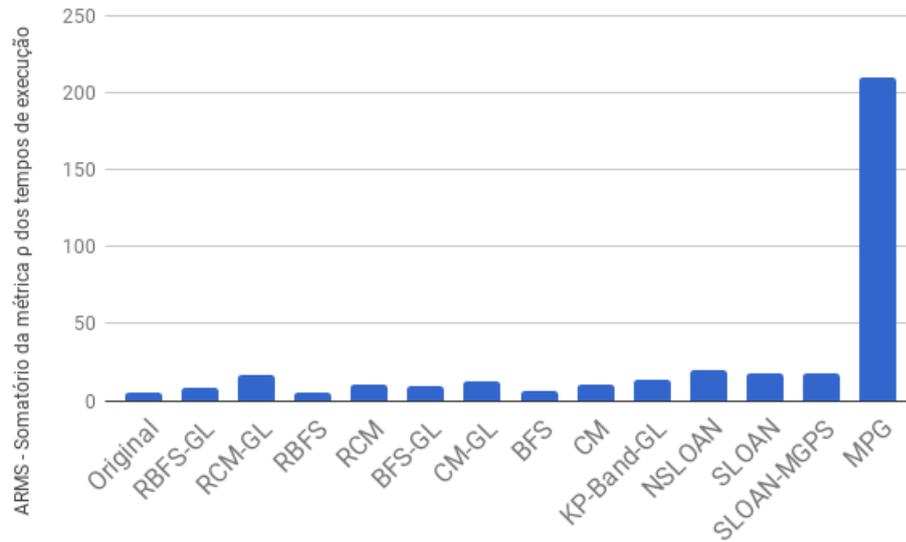
Figura 8.5 – Somatório da métrica  $\rho$  dos tempos de execução do método GMRES pré-condicionado nas instâncias da área *Circuit Simulation problem* em que houve convergência com o pré-condicionador ILUT



O método GMRES pré-condicionado com ARMS convergiu com, pelo menos, uma heurística de redução de largura de banda ou de *profile* em 9 das 12 instâncias dessa área. Uma análise a métrica  $\rho$  para essas instâncias indica que a melhor numeração para uso com o pré-condicionador ARMS foi, no geral, a numeração original da matriz. Entre as heurísticas, os melhores resultados foram conseguidos com as numerações produzidas por BFS e RBFS. A

soma dos tempos de execução nessas instâncias com o pré-condicionador ARMS é mostrado na Figura 8.6. É importante também observar que em 3 instâncias, *rajat24*, *rajat29* e *rajat30*, o pré-condicionador ARMS foi o único com o qual o método GMRES convergiu. Pode-se dizer que o ARMS foi um dos pré-condicionadores mais confiáveis nesse conjunto de instâncias.

Figura 8.6 – Soma dos tempos de execução do método GMRES pré-condicionado nas instâncias da área *Circuit Simulation problem* em que houve convergência com o pré-condicionador ARMS



Inicialmente, as instâncias *rajat21*, *rajat29*, *Hamrle3*, *FullChip* e *pre2* não apresentaram convergência com nenhum dos seis pré-condicionadores testados. Essas instâncias foram submetidas aos testes com o quarto conjunto de parâmetros indicado na Tabela 8.2. Apenas as instâncias *rajat21* e *rajat29* convergiram com o novo conjunto de parâmetros. Nos dois casos, o pré-condicionador responsável pela convergência foi o ARMS.

### 8.3 Discussões sobre as simulações para resolução de SELs com o método GMRES pré-condicionado

Os pré-condicionadores baseados em fatoração incompleta são sensíveis à ordem das incógnitas e das equações no SEL. Neste capítulo, as heurísticas RBFS-GL, RCM-GL, RBFS, RCM, BFS-GL, CM-GL, BFS, CM, NSloan, Sloan, Sloan-MGPS e MPG foram aplicadas em 20 instâncias assimétricas de grande porte, no intuito que as ordenações por elas produzidas contribuíssem para a redução do custo computacional do método GMRES pré-condicionado por ILUT, ILU(k), ILUC, ARMS, VBILUT e VBILU(k).

O número de condição da matriz é crucial para a convergência do método GMRES pré-condicionado e, como apontam Duff e Meurant (1989), "o número de condição de um problema fica pior à medida que o número de incógnitas aumenta". Dessa forma, a influência da heurística de ordenação utilizada tende a ser maior conforme as instâncias aumentam. Ainda, como explica Benzi (2002), os pré-condicionadores baseados em fatoração incompleta frequentemente falham em matrizes esparsas que não apresentam boa simetria estrutural ou numérica, dominância diagonal, ou são altamente indefinidas.

Nos testes conduzidos neste capítulo, 6 das 20 instâncias testadas (*webbase-1M*, *rajat21*, *rajat29*, *Hamrle3*, *FullChip*, *pre2*), não apresentaram convergência com nenhuma combinação de heurística de redução de largura de banda ou de *profile* e método pré-condicionador em conjunto com o GMRES. Após novas tentativas, as instâncias *rajat21* e *rajat29* convergiram com o uso do pré-condicionador ARMS.

Nas 14 instâncias remanescentes, o método GMRES pré-condicionado obteve convergência dentro dos parâmetros estabelecidos com pelo menos um dos métodos pré-condicionadores testados. Com o objetivo de procurar estabelecer quais combinações de heurísticas de ordenação e métodos pré-condicionadores exibem o melhor comportamento de convergência com o método GMRES, buscou-se compilar os resultados destes testes na Tabela 8.3.

Tabela 8.3 – Melhores combinações de heurísticas de ordenação e métodos pré-condicionadores, no geral, nos testes realizados com 20 instâncias assimétricas

| Área de aplicação            | Pré-condicionador + heurística |
|------------------------------|--------------------------------|
| Electromagnetics             | ILUT + RBFS                    |
| Optimization                 | ILUT + RBFS                    |
| Directed weighted graph      | ILUC + original                |
| Computational fluid dynamics | ILU(k) + original              |
| Circuit simulation           | ILUT + BFS                     |
|                              | ARMS + ([R]BFS ou original)    |

Por meio da Tabela 8.3 pode-se verificar que, com exceção dos pré-condicionadores em bloco, VBILUT e VBILU(k), cada pré-condicionador testado se sobressaiu em, pelo menos, uma área de atuação. Além disso, nota-se que na maioria das áreas de atuação das matrizes testadas, os melhores tempos de convergência foram observados quando não se utilizou nenhuma heurística de redução de largura de banda ou de *profile*. Apenas heurísticas baseadas em variações da busca em largura, RBFS e BFS, foram capazes de contribuir com a redução do

custo total de resolução de SELs com o método GMRES pré-condicionado em alguma área de atuação nos testes realizados. A seguir, o desempenho dos pré-condicionadores que obtiveram bons resultados nos testes é analisado.

### 8.3.1 ILUT

O pré-condicionador ILUT foi o responsável pelos menores tempos de processamento nas instâncias das áreas *Electromagnetics problem* e *Optimization problem*. Nos testes com instâncias do tipo *Circuit Simulation* o pré-condicionador ILUT obteve melhores resultados em quatro delas quando a ordenação original da matriz foi usada. Ressaltamos que instâncias dessa área foram bastante desafiadoras para o método GMRES com os pré-condicionadores testados. Tome, como exemplo, a instância *memchip*. O método GMRES pré-condicionado com ILUT e ordenação produzida por RBFS convergiu em 8,94 segundos. Já a ordenação produzida por BFS (isto é, sem inversão da ordenação gerada) acarretou em um erro durante a geração do pré-condicionador ILUT. Logo, essa instância foi altamente sensível às mudanças na ordem dos coeficientes, o que caracteriza uma matriz mal-condicionada.

Como afirma Saad (2003), nos casos em que o ILUT falha em um problema, pode-se obter melhores resultados diminuindo-se o valor  $\tau$  e aumentando-se o valor  $p$ . Isso permitirá mais elementos na fatoração e aumentará as chances de sucesso do pré-condicionador. Contudo, no trabalho de Chow e Saad (1997), mostra-se que nem sempre essa estratégia funciona. Para matrizes não estruturadas, podem surgir pivôs com valores muito pequenos na fatoração, ou mesmo com valor zero, o que torna o pré-condicionamento instável e suscetível a falhas. Além disso, aumentar demais o *thresholding* a fatoração pode destruir a precisão da fatoração. Para certos casos em que o ILUT falha, Chow e Saad (1997) sugerem tentar versões do ILUT com pivôs, como a versão ILUTP (*thresholding-based ILU with pivoting*).

### 8.3.2 ILU(k)

O pré-condicionador ILU(k) foi o melhor pré-condicionador para o método GMRES nos testes com instâncias do tipo *Computational Fluid Dynamics*. Em todas as três instâncias dessa área, a combinação do ILU(k) com as ordenações originais da matriz obteve resultados notáveis.

Em instâncias das áreas *Electromagnetics*, *Optimization* e *Circuit Simulation*, o pré-condicionador ILU(k) teve um desempenho ruim. Em geral, as fatorações como o ILU(k) são

pouco confiáveis para sistemas assimétricos genéricos (KAPORIN, 2010). Mesmo em tentativas posteriores com ILU(2), ILU(4) e ILU(12) que permitem mais *fill-in*, o pré-condicionador não contribuiu com a convergência do método GMRES.

### 8.3.3 ILUC

O pré-condicionador ILUC foi o melhor pré-condicionador para o método GMRES nos testes com instâncias do tipo *Directed Weighted Graph*. Os melhores resultados do método GMRES pré-condicionado com ILUC nessa área foram com a ordenação original da matriz.

A proposta da formulação Crout do ILU é diminuir o custo da busca por pivôs durante a fatoração, em comparação à fatoração ILU ordinária. Nos testes realizados, o ILUC teve melhor desempenho, em geral que o ILUT e o ILU(k) nas instâncias do tipo *Directed Weighted Graph*, e melhor desempenho que o ILUT nas instâncias do tipo *Computational Fluid Dynamics*.

### 8.3.4 ARMS

Nos testes deste capítulo, o pré-condicionador ARMS contribuiu para o melhor desempenho do método GMRES, em geral, na instância da área *Electromagnetics*. Ainda, o método GMRES pré-condicionado com ARMS obteve bons resultados em instâncias da área *Circuit Simulation*, conseguindo convergência no maior número de instâncias.

O pré-condicionador ARMS conseguiu convergência com, pelo menos, uma heurística de ordenação em 14 das 20 heurísticas testadas. Esse foi o melhor resultado entre todos os pré-condicionadores testados. O pré-condicionador ILUT levou o GMRES à convergir em 13 das 20 instâncias. No geral, as causas de não convergência do método GMRES pré-condicionado com ARMS, com as diferentes heurísticas de ordenação, foram o limite de tempo estabelecido. Nesse aspecto, o pré-condicionador ARMS se mostrou mais estável que o ILUT, e mais robusto para uso geral em instâncias assimétricas em simulações em que o tempo de execução pode ser mais longo.

### 8.3.5 Considerações finais

Os testes realizados neste capítulo consideraram seis pré-condicionadores e treze heurísticas de redução de largura de banda ou de *profile*, além da ordenação original, em 20 instâncias assimétricas de grandes dimensões, oriundas de cinco áreas de atuação diferentes. Devido à di-

versidade das instâncias utilizadas, era esperado que métodos baseados no subespaço de Krylov, como o GMRES, falhassem em algumas delas.

Um resultado notável observado nos testes conduzidos foi que, em geral, os menores tempos de processamento do método GMRES pré-condicionado foram obtidos com a ordenação original das instâncias. Este resultado vai de encontro aos resultados obtidos por outros trabalhos que estudaram os efeitos da reordenação das matrizes na convergência do método GMRES, em especial os de Benzi, Szyld e Duin (1999) e Benzi, Haws e Tuma (2000). Há diversos fatores que podem explicar esses resultados. Primeiramente, os testes realizados neste trabalho incluíram outros pré-condicionadores (ILUC, ARMS, VBILUT e VBILU(k)), além do ILUT e ILU(k) utilizados em Benzi, Szyld e Duin (1999), Benzi, Haws e Tuma (2000). A escolha dos parâmetros para as execuções do método GMRES com cada pré-condicionador também foi conduzida de forma diferente. Como visto em Chow e Saad (1997), entre muitos outros, diferenças sutis nos valores dos parâmetros dos pré-condicionadores baseados em fatoração incompleta são determinantes na convergência do GMRES. Por fim, as instâncias utilizadas neste trabalho possuem dimensões muito maiores que as utilizadas por Benzi, Szyld e Duin (1999), Benzi, Haws e Tuma (2000). Instâncias com número maior de coeficientes tendem a ter maiores números condição e, em consequência, serem mais sensíveis às ordenações empregadas (DUFF; MEURANT, 1989).

Conclui-se também que as ordenações produzidas por variações da busca em largura (BFS) obtiveram melhor desempenho, no geral, que as produzidas pela heurística Cuthill-McKee (CM). Ainda, as versões dessas duas heurísticas iniciadas por vértices pseudo-periféricos obtidos pelo algoritmo de George e Liu (GL) tornaram maiores os tempos totais para convergência do GMRES, no geral, que as versões sem esse recurso. Foi verificado também que, em geral, as heurísticas para redução de *profile* de Sloan, NSloan, Sloan-MGPS e MPG não são recomendadas para reduzir o tempo total do método GMRES em instâncias assimétricas de grande porte.

Tabela 8.4 – Tempos de execução (em segundos) do método GMRES pré-condicionado em instâncias assimétricas do tipo “*Electromagnetics problem*”.

|           | Pré-condicionador | Original            | RBFS-GL             | RCM-GL              | RBFS                | RCM                 | BFS-GL              | CM-GL               | BFS                 | CM                  | KP-Band-GL          | NSloan              | Sloan               | Sloan-MGPS          | MPG                 |
|-----------|-------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| tmt_unsym | ILUT              | 90.76 <sup>/2</sup> | 79.65 <sup>/2</sup> | 87.15 <sup>/2</sup> | 77.50 <sup>/2</sup> | 85.22 <sup>/2</sup> | 82.67 <sup>/2</sup> | 84.94 <sup>/2</sup> | 80.85 <sup>/2</sup> | 82.39 <sup>/2</sup> | 90.26 <sup>/2</sup> | 90.25 <sup>/2</sup> | 84.39 <sup>/2</sup> | 84.33 <sup>/2</sup> | 82.79 <sup>/2</sup> |
|           | ILUC              | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   |
|           | ILUK              | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   |
|           | ARMS              | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   |
|           | VBILUT            | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   |
|           | VBILUK            | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   | †                   |

Tabela 8.5 – Tempos de execução (em segundos) do método GMRES pré-condicionado em instâncias assimétricas do tipo “*Optimization problem*”.

|            | Pré-cond. | Original             | RBFS-GL              | RCM-GL                | RBFS                 | RCM                 | BFS-GL              | CM-GL  | BFS                 | CM                   | KP-Band-GL | NSloan              | Sloan               | Sloan-MGPS          | MPG                  |
|------------|-----------|----------------------|----------------------|-----------------------|----------------------|---------------------|---------------------|--------|---------------------|----------------------|------------|---------------------|---------------------|---------------------|----------------------|
| largebasis | ILUT      | *                    | 3.97                 | *                     | 3.85                 | *                   | *                   | 8.47   | *                   | 7.36                 | 8.76       | *                   | *                   | *                   | *                    |
|            | ILUC      | cond                 | †                    | †                     | †                    | †                   | †                   | †      | †                   | †                    | cond       | †                   | †                   | †                   | †                    |
|            | ILU(k)    | *                    | *                    | *                     | *                    | *                   | *                   | *      | *                   | *                    | cond       | *                   | *                   | *                   | *                    |
|            | ARMS      | 136.90 <sup>/3</sup> | 371.67 <sup>/3</sup> | 1115.10 <sup>/3</sup> | 386.04 <sup>/3</sup> | 65.04 <sup>/3</sup> | 62.70 <sup>/3</sup> | 746.62 | 61.48 <sup>/3</sup> | 389.70 <sup>/3</sup> | 699.66     | 72.82 <sup>/3</sup> | 76.11 <sup>/3</sup> | 12.12 <sup>/3</sup> | 107.94 <sup>/3</sup> |
|            | VBILUT    | *                    | *                    | 7.32                  | *                    | 9.73                | 7.26                | *      | 6.15                | *                    | 7.47       | *                   | *                   | *                   | *                    |
|            | VBILU(k)  | *                    | *                    | *                     | *                    | cond                | cond                | *      | cond                | *                    | cond       | cond                | cond                | cond                | cond                 |

Tabela 8.6 – Tempos de execução (em segundos) do método GMRES pré-condicionado em instâncias assimétricas do tipo “Directed Weighted Graph”.

|            | Pré-condicionador | Original | RBFS-GL | RCM-GL | RBFS  | RCM   | BFS-GL | CM-GL | BFS   | CM    | KP-Band-GL | NSloan | Sloan  | Sloan-MGPS | MPG    |
|------------|-------------------|----------|---------|--------|-------|-------|--------|-------|-------|-------|------------|--------|--------|------------|--------|
| cage13     | ILUT              | 8.26     | 16.01   | 21.64  | 13.54 | 18.34 | 16.96  | 20.66 | 13.59 | 18.40 | 23.24      | 24.47  | 24.66  | 24.61      | 225.62 |
|            | ILUC              | 5.37     | 13.26   | 17.27  | 9.25  | 14.22 | 12.41  | 16.95 | 9.41  | 13.98 | 18.05      | 20.21  | 19.23  | 19.28      | 219.90 |
|            | ILU(k)            | 11.41    | 18.26   | 24.02  | 15.45 | 21.09 | 19.22  | 22.95 | 16.36 | 20.28 | 26.38      | 26.79  | 25.06  | 25.03      | 258.25 |
|            | ARMS              | 729.74   | †       | †      | †     | †     | †      | †     | †     | †     | †          | †      | †      | †          | †      |
|            | VBILUT            | 15.38    | 24.50   | 31.78  | 21.91 | 28.38 | 27.65  | 28.79 | 23.72 | 26.39 | 35.08      | 36.32  | 31.03  | 31.45      | 231.06 |
|            | VBILU(k)          | 25.18    | 30.63   | 36.90  | 28.09 | 33.85 | 32.82  | 36.10 | 29.37 | 32.81 | 39.37      | 40.76  | 37.56  | 36.96      | 234.58 |
| cage14     | ILUT              | 31.47    | 65.23   | 84.06  | 50.87 | 67.27 | 63.74  | 81.09 | 51.82 | 68.23 | 86.63      | 97.18  | 103.17 | 97.32      | †      |
|            | ILUC              | 17.48    | 45.98   | 69.78  | 33.13 | 51.08 | 46.25  | 62.90 | 33.20 | 50.53 | 68.17      | 73.59  | 79.45  | 86.74      | †      |
|            | ILU(k)            | 49.81    | 81.66   | 102.63 | 65.71 | 86.35 | 81.71  | 99.40 | 69.15 | 82.87 | 109.11     | 106.16 | 111.32 | 98.65      | †      |
|            | ARMS              | †        | †       | †      | †     | †     | †      | †     | †     | †     | †          | †      | †      | †          | †      |
|            | VBILUT            | †        | †       | †      | †     | †     | †      | †     | †     | †     | †          | †      | †      | †          | †      |
|            | VBILU(k)          | †        | †       | †      | †     | †     | †      | †     | †     | †     | †          | †      | †      | †          | †      |
| webbase-1M | ILUT              | *        | *       | *      | *     | *     | *      | *     | *     | *     | *          | *      | *      | *          | *      |
|            | ILUC              | cond     | cond    | cond   | cond  | cond  | cond   | cond  | cond  | cond  | cond       | cond   | cond   | cond       | cond   |
|            | ILU(k)            | *        | *       | *      | *     | *     | *      | *     | *     | *     | *          | *      | *      | *          | *      |
|            | ARMS              | *        | *       | *      | ‡     | *     | *      | ‡     | ‡     | ‡     | ‡          | ‡      | *      | *          | *      |
|            | VBILUT            | *        | *       | *      | *     | *     | *      | *     | *     | *     | *          | *      | *      | *          | *      |
|            | VBILU(k)          | *        | *       | *      | *     | *     | *      | *     | *     | *     | *          | *      | *      | *          | *      |

Tabela 8.7 – Tempos de execução (em segundos) do método GMRES pré-condicionado em instâncias assimétricas do tipo “*Computational Fluid Dynamics problem*”.

|           | Pré-condicionador | Original | RBFS-GL | RCM-GL | RBFS   | RCM    | BFS-GL | CM-GL  | BFS    | CM     | KP-Band-GL | NSloan | Sloan  | Sloan-MGPS | MPG    |
|-----------|-------------------|----------|---------|--------|--------|--------|--------|--------|--------|--------|------------|--------|--------|------------|--------|
| atmosmodd | ILUT              | 38.47    | 30.63   | 37.47  | 26.27  | 33.42  | 29.51  | 37.76  | 26.19  | 33.42  | 37.97      | 42.08  | 42.43  | 42.07      | 41.33  |
|           | ILUC              | 31.89    | 26.30   | 33.75  | 23.03  | 30.34  | 26.42  | 33.69  | 23.06  | 30.15  | 34.65      | 35.38  | 35.28  | 35.73      | 37.52  |
|           | ILU(k)            | 18.39    | 24.28   | 32.76  | 21.84  | 28.99  | 25.58  | 31.80  | 21.68  | 28.98  | 33.98      | 33.63  | 32.31  | 33.46      | 34.58  |
|           | ARMS              | 33.03    | 34.08   | 44.37  | 35.68  | 44.48  | 41.02  | 43.65  | 37.87  | 42.87  | 48.89      | 45.09  | 45.14  | 44.44      | 45.85  |
|           | VBILUT            | †        | †       | †      | †      | †      | †      | †      | †      | †      | †          | †      | †      | †          | †      |
|           | VBILU(k)          | †        | †       | †      | †      | †      | †      | †      | †      | †      | †          | †      | †      | †          | †      |
| atmosmodj | ILUT              | 39.82    | 30.40   | 36.74  | 25.75  | 33.09  | 30.25  | 38.16  | 25.77  | 32.93  | 37.69      | 39.50  | 41.19  | 41.10      | 40.84  |
|           | ILUC              | 31.41    | 28.42   | 33.85  | 23.33  | 30.35  | 26.90  | 33.66  | 23.08  | 30.17  | 34.54      | 37.79  | 37.73  | 38.09      | 39.60  |
|           | ILU(k)            | 18.27    | 25.68   | 30.91  | 21.80  | 29.02  | 23.59  | 30.73  | 21.67  | 28.96  | 33.53      | 32.01  | 31.71  | 32.58      | 35.15  |
|           | ARMS              | 33.35    | 37.40   | 47.53  | 35.09  | 44.19  | 32.40  | 42.79  | 37.41  | 42.34  | 55.03      | 52.26  | 44.23  | 50.07      | 54.51  |
|           | VBILUT            | †        | †       | †      | †      | †      | †      | †      | †      | †      | †          | †      | †      | †          | †      |
|           | VBILU(k)          | †        | †       | †      | †      | †      | †      | †      | †      | †      | †          | †      | †      | †          | †      |
| atmosmodi | ILUT              | 27.48    | 31.33   | 39.21  | 27.39  | 36.19  | 30.28  | 40.35  | 27.64  | 35.58  | 40.81      | 42.17  | 44.26  | 44.35      | 41.44  |
|           | ILUC              | 21.11    | 27.44   | 37.96  | 24.73  | 33.10  | 27.94  | 35.96  | 24.56  | 33.73  | 38.82      | 36.51  | 38.03  | 37.78      | 39.05  |
|           | ILU(k)            | 10.21    | 19.92   | 27.79  | 13.92  | 22.55  | 19.28  | 28.09  | 14.13  | 22.24  | 28.63      | 31.31  | 31.12  | 30.04      | 29.84  |
|           | ARMS              | 30.21    | 173.91  | 301.85 | 178.12 | 288.45 | 178.16 | 294.34 | 174.02 | 303.06 | 186.77     | 237.24 | 234.38 | 239.30     | 250.36 |
|           | VBILUT            | †        | †       | †      | †      | †      | †      | †      | †      | †      | †          | †      | †      | †          | †      |
|           | VBILU(k)          | †        | †       | †      | †      | †      | †      | †      | †      | †      | †          | †      | †      | †          | †      |

Tabela 8.8 – Tempos de execução (em segundos) do método GMRES pré-condicionado em instâncias assimétricas do tipo “*Circuit Simulation problem*”.

|            | Pré-cond. | Original             | RBFS-GL             | RCM-GL              | RBFS                | RCM                 | BFS-GL               | CM-GL               | BFS                  | CM                  | KP-Band-GL           | NSloan | Sloan | Sloan-MGPS | MPG                 |
|------------|-----------|----------------------|---------------------|---------------------|---------------------|---------------------|----------------------|---------------------|----------------------|---------------------|----------------------|--------|-------|------------|---------------------|
| rajat21    | ILUT      | *                    | *                   | *                   | *                   | *                   | *                    | *                   | *                    | *                   | *                    | *      | *     | *          | *                   |
|            | ILUC      | ‡                    | ‡                   | cond                | ‡                   | cond                | cond                 | ‡                   | cond                 | ‡                   | cond                 | cond   | cond  | cond       | ‡                   |
|            | ILU(k)    | ‡                    | *                   | ‡                   | *                   | ‡                   | ‡                    | *                   | ‡                    | *                   | ‡                    | *      | *     | *          | *                   |
|            | ARMS      | 8.99 <sup>/4</sup>   | 14.57 <sup>/4</sup> | 13.86 <sup>/4</sup> | 25.22 <sup>/4</sup> | 12.86 <sup>/4</sup> | 11.97 <sup>/4</sup>  | 20.86 <sup>/4</sup> | 10.98 <sup>/4</sup>  | 27.08 <sup>/4</sup> | *                    | *      | *     | *          | ‡                   |
|            | VBILUT    | *                    | *                   | *                   | *                   | *                   | *                    | *                   | *                    | *                   | *                    | *      | *     | *          | *                   |
|            | VBILU(k)  | ‡                    | *                   | ‡                   | *                   | ‡                   | ‡                    | *                   | ‡                    | *                   | ‡                    | *      | *     | *          | *                   |
| rajat29    | ILUT      | *                    | *                   | *                   | *                   | *                   | *                    | *                   | †                    | *                   | *                    | ‡      | ‡     | ‡          | ‡                   |
|            | ILUC      | ‡                    | ‡                   | cond                | ‡                   | cond                | cond                 | ‡                   | ‡                    | ‡                   | cond                 | ‡      | ‡     | ‡          | ‡                   |
|            | ILU(k)    | ‡                    | *                   | ‡                   | *                   | ‡                   | ‡                    | ‡                   | ‡                    | *                   | ‡                    | ‡      | ‡     | ‡          | ‡                   |
|            | ARMS      | 133.53 <sup>/4</sup> | 56.28 <sup>/4</sup> | 30.78 <sup>/4</sup> | 54.32 <sup>/4</sup> | 29.06 <sup>/4</sup> | 246.75 <sup>/4</sup> | 66.86 <sup>/4</sup> | 157.03 <sup>/4</sup> | 65.22 <sup>/4</sup> | *                    | ‡      | ‡     | ‡          | ‡                   |
|            | VBILUT    | ‡                    | ‡                   | ‡                   | ‡                   | ‡                   | ‡                    | ‡                   | ‡                    | ‡                   | ‡                    | ‡      | ‡     | ‡          | ‡                   |
|            | VBILU(k)  | ‡                    | ‡                   | ‡                   | ‡                   | ‡                   | ‡                    | ‡                   | ‡                    | ‡                   | ‡                    | ‡      | ‡     | ‡          | ‡                   |
| rajat30    | ILUT      | *                    | *                   | *                   | *                   | *                   | *                    | *                   | †                    | *                   | *                    | †      | †     | †          | †                   |
|            | ILUC      | †                    | †                   | cond                | †                   | cond                | cond                 | †                   | cond                 | †                   | cond                 | †      | †     | †          | †                   |
|            | ILU(k)    | †                    | *                   | †                   | *                   | †                   | †                    | †                   | †                    | †                   | †                    | †      | †     | †          | †                   |
|            | ARMS      | 228.11               | 85.30               | 93.41               | 81.24               | 91.34               | 350.80               | 90.11               | 319.75               | 87.87               | 359.91 <sup>/3</sup> | †      | †     | †          | †                   |
|            | VBILUT    | †                    | †                   | †                   | †                   | †                   | †                    | †                   | †                    | †                   | †                    | †      | †     | †          | †                   |
|            | VBILU(k)  | †                    | †                   | †                   | †                   | †                   | †                    | †                   | †                    | †                   | †                    | †      | †     | †          | †                   |
| ASIC_680ks | ILUT      | 0.35                 | 2.44                | 3.43                | 1.37                | 2.77                | 2.99                 | 3.41                | 2.38                 | 2.13                | 3.89                 | 7.90   | 8.03  | 8.02       | 53.10               |
|            | ILUC      | †                    | †                   | †                   | †                   | †                   | †                    | †                   | †                    | †                   | †                    | †      | †     | †          | †                   |
|            | ILU(k)    | 0.97                 | 3.13                | 7.97                | 2.00                | 7.20                | 13.83                | 4.16                | 13.72                | 2.76                | 16.12                | 7.80   | 7.24  | 7.23       | 53.84               |
|            | ARMS      | 14.66                | 3.58                | 11.67               | 2.48                | 3.26 <sup>/2</sup>  | 3.68                 | 4.32                | 2.79                 | 3.27                | 4.76                 | 11.49  | 12.01 | 12.01      | 54.78 <sup>/2</sup> |
|            | VBILUT    | 6.12                 | 8.36                | 9.90                | 9.41                | 8.81                | 8.90                 | 10.36               | 8.01                 | 8.49                | 8.89                 | 12.30  | 14.59 | 13.75      | 62.79               |
|            | VBILU(k)  | 2.81                 | 5.05                | 18.23               | 3.89                | 18.22               | 40.90                | 6.46                | 43.28                | 4.69                | 50.47                | 9.10   | 9.08  | 9.07       | 56.00               |

Continua na próxima página

|           | Pré-cond. | Original | RBFS-GL | RCM-GL | RBFS   | RCM    | BFS-GL | CM-GL  | BFS    | CM     | KP-Band-GL | NSloan | Sloan | Sloan-MGPS | MPG    |
|-----------|-----------|----------|---------|--------|--------|--------|--------|--------|--------|--------|------------|--------|-------|------------|--------|
| ASIC_680k | ILUT      | *        | *       | *      | *      | 6.90   | *      | *      | 4.20   | *      | *          | †      | †     | †          | †      |
|           | ILUC      | †        | *       | †      | *      | †      | †      | †      | †      | †      | †          | †      | †     | †          | †      |
|           | ILU(k)    | †        | *       | †      | *      | †      | †      | *      | †      | *      | †          | †      | †     | †          | †      |
|           | ARMS      | 141.58   | 182.19  | 187.09 | 180.77 | 185.49 | 132.68 | 158.25 | 131.24 | 156.93 | 187.08     | †      | †     | †          | †      |
|           | VBILUT    | †        | †       | †      | †      | †      | †      | †      | †      | †      | †          | †      | †     | †          | †      |
|           | VBILU(k)  | †        | †       | †      | †      | †      | †      | †      | †      | †      | †          | †      | †     | †          | †      |
| Hamrle3   | ILUT      | *        | *       | *      | *      | *      | *      | *      | *      | *      | *          | *      | *     | *          | ‡      |
|           | ILUC      | cond     | cond    | cond   | cond   | cond   | cond   | cond   | cond   | cond   | cond       | cond   | cond  | cond       | ‡      |
|           | ILU(k)    | *        | *       | *      | *      | *      | *      | *      | *      | *      | *          | *      | *     | *          | ‡      |
|           | ARMS      | *        | *       | *      | *      | *      | *      | *      | *      | *      | *          | *      | *     | *          | ‡      |
|           | VBILUT    | ‡        | ‡       | ‡      | ‡      | ‡      | ‡      | ‡      | ‡      | ‡      | ‡          | ‡      | ‡     | ‡          | ‡      |
|           | VBILU(k)  | ‡        | ‡       | ‡      | ‡      | ‡      | ‡      | ‡      | ‡      | ‡      | ‡          | ‡      | ‡     | ‡          | ‡      |
| memchip   | ILUT      | *        | 18.03   | *      | 8.94   | *      | *      | 28.06  | *      | 21.14  | *          | †      | *     | *          | *      |
|           | ILUC      | †        | †       | †      | †      | †      | †      | †      | †      | †      | †          | †      | †     | †          | †      |
|           | ILU(k)    | *        | 29.88   | *      | 21.24  | *      | *      | 40.48  | *      | 33.33  | *          | *      | *     | *          | *      |
|           | ARMS      | 21.37    | 105.62  | 118.23 | 101.37 | 109.41 | 135.48 | 107.90 | 125.46 | 100.23 | 106.44     | 143.19 | 90.07 | 93.12      | 378.31 |
|           | VBILUT    | †        | †       | †      | †      | †      | †      | †      | †      | †      | †          | †      | †     | †          | †      |
|           | VBILU(k)  | †        | †       | †      | †      | †      | †      | †      | †      | †      | †          | †      | †     | †          | †      |
| FullChip  | ILUT      | *        | *       | *      | *      | *      | *      | *      | ‡      | *      | *          | ‡      | ‡     | ‡          | ‡      |
|           | ILUC      | ‡        | cond    | cond   | ‡      | cond   | cond   | ‡      | ‡      | ‡      | cond       | ‡      | ‡     | ‡          | ‡      |
|           | ILU(k)    | ‡        | ‡       | ‡      | ‡      | ‡      | ‡      | ‡      | ‡      | ‡      | ‡          | ‡      | ‡     | ‡          | ‡      |
|           | ARMS      | ‡        | ‡       | ‡      | ‡      | ‡      | ‡      | ‡      | ‡      | ‡      | ‡          | ‡      | ‡     | ‡          | ‡      |
|           | VBILUT    | ‡        | ‡       | ‡      | ‡      | ‡      | ‡      | ‡      | ‡      | ‡      | ‡          | ‡      | ‡     | ‡          | ‡      |
|           | VBILU(k)  | ‡        | ‡       | ‡      | ‡      | ‡      | ‡      | ‡      | ‡      | ‡      | ‡          | ‡      | ‡     | ‡          | ‡      |

|              | Pré-cond. | Original              | RBFS-GL              | RCM-GL                | RBFS                  | RCM                  | BFS-GL                | CM-GL                 | BFS                  | CM                    | KP-Band-GL           | NSloan | Sloan                | Sloan-MGPS           | MPG     |
|--------------|-----------|-----------------------|----------------------|-----------------------|-----------------------|----------------------|-----------------------|-----------------------|----------------------|-----------------------|----------------------|--------|----------------------|----------------------|---------|
| Freescale1   | ILUT      | 303.01                | 49.67                | 50.63 <sup>/2</sup>   | 33.10                 | 36.37 <sup>/2</sup>  | 28.50 <sup>/2</sup>   | 68.71                 | 16.63 <sup>/2</sup>  | 55.12                 | 52.53 <sup>/2</sup>  | 198.44 | 180.80               | 179.37               | †       |
|              | ILUC      | †                     | cond                 | cond                  | †                     | cond                 | cond                  | †                     | cond                 | †                     | cond                 | †      | †                    | †                    | †       |
|              | ILU(k)    | 447.74 <sup>/3</sup>  | 360.02 <sup>/3</sup> | †                     | 948.00 <sup>/3</sup>  | †                    | †                     | 380.02 <sup>/3</sup>  | †                    | 905.50 <sup>/3</sup>  | †                    | †      | 577.65 <sup>/3</sup> | 582.50 <sup>/3</sup> | †       |
|              | ARMS      | 1039.31 <sup>/2</sup> | 998.72 <sup>/2</sup> | 1073.64 <sup>/2</sup> | 1035.55 <sup>/2</sup> | 941.52 <sup>/2</sup> | 1002.48 <sup>/2</sup> | 1029.89 <sup>/2</sup> | 998.89 <sup>/2</sup> | 1052.66 <sup>/2</sup> | 992.52 <sup>/2</sup> | †      | †                    | †                    | †       |
|              | VBILUT    | †                     | †                    | †                     | †                     | †                    | †                     | †                     | †                    | †                     | †                    | †      | †                    | †                    | †       |
|              | VBILU(k)  | †                     | †                    | †                     | †                     | †                    | †                     | †                     | †                    | †                     | †                    | †      | †                    | †                    | †       |
| circuit5M_dc | ILUT      | 2.80                  | 24.67                | *                     | 10.31                 | *                    | 22.75                 | *                     | 10.69                | *                     | 43.58                | 38.06  | 42.06                | 37.00                | †       |
|              | ILUC      | †                     | †                    | †                     | †                     | cond                 | †                     | cond                  | †                    | cond                  | †                    | †      | †                    | †                    | †       |
|              | ILU(k)    | 6.813                 | 26.334               | *                     | 11.769                | *                    | 29.999                | *                     | 15.610               | *                     | 51.113               | 47.015 | 45.822               | 40.668               | †       |
|              | ARMS      | 15.66                 | 30.24                | 51.12                 | 16.05                 | 35.79                | 29.61                 | 46.27                 | 13.78                | 35.33                 | 50.50                | 50.17  | 54.88                | 49.50                | †       |
|              | VBILUT    | †                     | †                    | †                     | †                     | †                    | †                     | †                     | †                    | †                     | †                    | †      | †                    | †                    | †       |
|              | VBILU(k)  | †                     | †                    | †                     | †                     | †                    | †                     | †                     | †                    | †                     | †                    | †      | †                    | †                    | †       |
| rajat31      | ILUT      | 4.10                  | *                    | 53.34 <sup>/2</sup>   | *                     | 34.01                | 250.25 <sup>/3</sup>  | *                     | 9.97                 | *                     | *                    | *      | *                    | *                    | *       |
|              | ILUC      | †                     | †                    | †                     | †                     | †                    | cond                  | 55.71                 | cond                 | 60.31                 | cond                 | †      | †                    | †                    | †       |
|              | ILU(k)    | 16.47                 | *                    | 122.84 <sup>/2</sup>  | *                     | 43.74                | 29.44                 | *                     | 21.46                | *                     | 57.03                | *      | *                    | *                    | *       |
|              | ARMS      | 8.21                  | 27.52                | 51.71                 | 15.80                 | 41.87                | 26.84                 | 51.28                 | 15.42                | 41.81                 | 49.44                | 36.19  | 35.85                | 41.43                | 1041.39 |
|              | VBILUT    | †                     | †                    | †                     | †                     | †                    | †                     | †                     | †                    | †                     | †                    | †      | †                    | †                    | †       |
|              | VBILU(k)  | †                     | †                    | †                     | †                     | †                    | †                     | †                     | †                    | †                     | †                    | †      | †                    | †                    | †       |
| pre2         | ILUT      | *                     | *                    | *                     | *                     | *                    | *                     | *                     | *                    | *                     | *                    | *      | *                    | *                    | *       |
|              | ILUC      | cond                  | cond                 | cond                  | cond                  | cond                 | cond                  | cond                  | cond                 | cond                  | cond                 | cond   | cond                 | cond                 | cond    |
|              | ILU(k)    | ‡                     | *                    | ‡                     | *                     | ‡                    | ‡                     | *                     | ‡                    | *                     | ‡                    | *      | *                    | *                    | *       |
|              | ARMS      | ‡                     | ‡                    | ‡                     | ‡                     | ‡                    | ‡                     | ‡                     | ‡                    | ‡                     | ‡                    | ‡      | ‡                    | ‡                    | ‡       |
|              | VBILUT    | *                     | *                    | *                     | *                     | *                    | *                     | *                     | *                    | *                     | *                    | *      | *                    | *                    | *       |
|              | VBILU(k)  | *                     | *                    | *                     | *                     | *                    | *                     | *                     | *                    | *                     | *                    | *      | *                    | *                    | *       |

## 9. CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho, buscou-se avaliar o desempenho de métodos de redução de largura de banda ou de *profile* na redução do custo de execução do método GMRES pré-condicionado. Os métodos de interesse neste estudo foram aqueles considerados no estado da arte para redução de largura de banda ou de *profile* por Bernardes e Oliveira (2015), Chagas e Oliveira (2015), Oliveira, Bernardes e Chagas (2016). Além deles, foram propostas heurísticas para redução de largura de banda e de *profile* de matrizes baseadas na meta-heurística *Iterated Local Search*.

A aplicação da meta-heurística *Iterated Local Search* nos problemas em questão culminou com o desenvolvimento de duas heurísticas: *ILS-Band*, com foco em redução de largura de banda e *ILS-Prof*, com foco em redução de *profile*. As heurísticas propostas foram comparadas à heurísticas que podem ser consideradas como as potenciais heurísticas de baixo custo computacional no estado da arte para reduções de largura de banda e de *profile*. Nos testes realizados com instâncias simétricas, a heurística proposta ILS-Band obteve o segundo melhor desempenho geral em redução de largura de banda de acordo com a métrica  $\rho$ . A heurística ILS-Prof foi a quinta melhor em redução de *profile* nesse conjunto de instâncias. Nos testes realizados com instâncias assimétricas, a heurística proposta ILS-Band obteve a terceira melhor posição em redução de largura de banda de acordo com a métrica  $\rho$ . Quanto à redução de *profile* nesse conjunto de instâncias, a heurística ILS-Prof foi também a quinta melhor.

As heurísticas ILS-Band e ILS-Prof propostas apresentaram tempos de execução menores dos que os obtidos pelos outros métodos heurísticos testados que incluem alguma forma de busca local, como FNC-HC e VNS-Band. Contudo, esses tempos foram ainda muito superiores aos de outros métodos heurísticos testados. Esse resultado fez com que as heurísticas propostas ILS-Band e ILS-Prof não fossem consideradas promissoras para a tarefa de redução do custo computacional do método GMRES pré-condicionado.

Foram conduzidos experimentos com heurísticas de baixo custo selecionadas para se avaliar os efeitos das reordenações produzidas por elas na convergência do método GMRES utilizando pré-condicionadores baseados em fatoração incompleta (ILUT, ILU(k), ILUC, VBI-LUT e VBILU(k)) e em *multigrid* (ARMS). Os experimentos com o método GMRES pré-condicionado foram realizados em 20 instâncias assimétricas de grande porte oriundas de cinco áreas de aplicação. Matrizes desse tipo podem sofrer com altos números de condição e podem possuir características que tirem a estabilidade dos pré-condicionadores aplicados. Ainda

assim, apenas 4 das instâncias não apresentaram convergência com nenhuma combinação de heurística de ordenação e método pré-condicionador.

Nos testes conduzidos, verificou-se que a escolha da melhor combinação de heurística de numeração de vértices e pré-condicionador é altamente dependente da área de aplicação, e mesmo das características das instâncias dentro de uma mesma área. Essa observação corrobora os resultados obtidos por Chow e Saad (1997), Benzi, Szyld e Duin (1999). Não obstante, umas das contribuições deste trabalho foi indicar potenciais combinações de pré-condicionadores e heurísticas de ordenação que, no geral, promovem reduções do custo do método GMRES em áreas específicas.

Ainda, os experimentos conduzidos neste trabalho mostraram que o tempo de execução do método GMRES pré-condicionado em conjunto com as ordenações produzidas pelas heurísticas selecionadas foi pior, no geral, do que com a ordenação original da matriz, nas instâncias selecionadas. Nas áreas em que houve diminuição no tempo de execução do método GMRES, as heurísticas que levaram a melhores resultados foram, no geral, as mais simples. Em especial, ordenações produzidas por variações da busca em largura (BFS) produziram bons resultados nas instâncias testadas. Estes últimos resultados estão em consonância com os observados no trabalho de Oliveira, Bernardes e Chagas (2017), em que foi empregado o método dos gradientes conjugados pré-condicionado pela fatoração incompleta de Cholesky.

Apesar das heurísticas propostas terem apresentado os menores tempos, em geral, entre métodos meta-heurísticos testados, espera-se, em trabalhos futuros, melhorar ainda mais o desempenho das heurísticas ILS-Band e ILS-Prof quanto ao tempo de execução. Técnicas que adicionem alguma forma de memória à heurística, como a da busca tabu, podem contribuir para que as heurísticas não fiquem presas a ótimos locais potenciais e produzam respostas melhores e mais rápidas.

Experimentos futuros envolvendo o método GMRES podem se beneficiar de implementações paralelas desse resolutor. Tais implementações podem ser focadas apenas em GPUs, em multi-CPU ou em esquemas híbridos GPU-CPU. Experimentos que apontem qual a melhor plataforma para implementações paralelas do método GMRES pré-condicionado devem ser realizados. Nos casos de implementações em GPUs, os melhores *kernels* para operações de produtos vetoriais envolvendo matrizes esparsas (SpMV) devem ser identificados. Versões paralelas de pré-condicionadores baseados em fatoração incompleta LU, bem como de heurísticas

para redução de largura de banda ou de *profile* também devem ser considerados em trabalhos futuros.

Por fim, certas áreas de aplicação consideradas nos testes com o método GMRES pré-condicionado possuíam poucas instâncias que atendiam aos critérios definidos neste trabalho. Espera-se realizar testes futuros com o método GMRES pré-condicionado em um número maior de instâncias de diferentes áreas de aplicação.

## REFERÊNCIAS

- APPLEGATE, D.; COOK, W.; ROHE, A. Chained Lin-Kernighan for large traveling salesman problems. **INFORMS Journal on Computing**, INFORMS, v. 15, n. 1, p. 82–92, 2003.
- ARNOLDI, W. E. The principle of minimized iterations in the solution of the matrix eigenvalue problem. **Quarterly of applied mathematics**, v. 9, n. 1, p. 17–29, 1951.
- BENZI, M. Preconditioning techniques for large linear systems: A survey. **Journal of Computational Physics**, Elsevier, v. 182, n. 2, p. 418–477, 2002.
- BENZI, M.; HAWS, J. C.; TUMA, M. Preconditioning highly indefinite and nonsymmetric matrices. **SIAM Journal on Scientific Computing**, SIAM, v. 22, n. 4, p. 1333–1353, 2000.
- BENZI, M.; SZYLD, D. B.; DUIN, A. V. Orderings for incomplete factorization preconditioning of nonsymmetric problems. **SIAM Journal on Scientific Computing**, SIAM, v. 20, n. 5, p. 1652–1670, 1999.
- BERNARDES, J. A. B.; OLIVEIRA, S. L. Gonzaga de. A systematic review of heuristics for profile reduction of symmetric matrices. **Procedia Computer Science**, Elsevier, v. 51, p. 221–230, 2015.
- BESTEN, M. D.; STÜTZLE, T.; DORIGO, M. Design of iterated local search algorithms. In: SPRINGER. **Workshops on Applications of Evolutionary Computation**. Berlin, Heidelberg: Springer, 2001. p. 441–451.
- BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. **ACM computing surveys (CSUR)**, ACM, v. 35, n. 3, p. 268–308, 2003.
- BRIGGS, W. L.; MCCORMICK, S. F.; HENSON, V. E. **A multigrid tutorial**. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2000. v. 72.
- BROZOLO, G. R. D.; ROBERT, Y. Parallel conjugate gradient-like algorithms for solving sparse nonsymmetric linear systems on a vector multiprocessor. **Parallel Computing**, Elsevier, v. 11, n. 2, p. 223–239, 1989.
- BURGESS, D.; GILES, M. Renumbering unstructured grids to improve the performance of codes on hierarchical memory machines. **Advances in Engineering Software**, Elsevier, v. 28, n. 3, p. 189–201, 1997.
- CAMATA, J.; ROSSA, A.; VALLI, A.; CATABRIGA, L.; CAREY, G.; COUTINHO, A. Reordering and incomplete preconditioning in serial and parallel adaptive mesh refinement and coarsening flow solutions. **International Journal for Numerical Methods in Fluids**, Wiley Online Library, v. 69, n. 4, p. 802–823, 2012.
- CAMPOS, V.; PIÑANA, E.; MARTÍ, R. Adaptive memory programming for matrix bandwidth minimization. **Annals of Operations Research**, Springer, v. 183, n. 1, p. 7–23, 2011.
- CATABRIGA, L.; CAMATA, J. J.; VALLI, A. M.; COUTINHO, A.; CAREY, G. F. Reordering effects on preconditioned Krylov methods in AMR solutions of flow and transport. In: **XVII Iberian Latin American Congress on Computational Methods in Engineering**. Belém, Pará, Brazil: Brazilian Assoc. for Comp. Mechanics & Latin American Assoc. of Comp. Methods in Engineering, 2006. v. 1, p. 1–14.

- CHAGAS, G. O.; OLIVEIRA, S. L. Gonzaga de. Metaheuristic-based heuristics for symmetric-matrix bandwidth reduction: a systematic review. **Procedia Computer Science**, Elsevier, v. 51, p. 211–220, 2015.
- CHOW, E.; SAAD, Y. Experimental study of ILU preconditioners for indefinite matrices. **Journal of Computational and Applied Mathematics**, Elsevier, v. 86, n. 2, p. 387–414, 1997.
- CONGRAM, R. K.; POTTS, C. N.; VELDE, S. L. van de. An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem. **INFORMS Journal on Computing**, INFORMS, v. 14, n. 1, p. 52–67, 2002.
- CORSO, G. M. D.; MANZINI, G. Finding exact solutions to the bandwidth minimization problem. **Computing**, Springer, v. 62, n. 3, p. 189–203, 1999.
- COUTURIER, R.; DOMAS, S. Sparse systems solving on GPUs with GMRES. **The journal of Supercomputing**, Springer, v. 59, n. 3, p. 1504–1516, 2012.
- CUNHA, R. D. D.; HOPKINS, T. A parallel implementation of the restarted GMRES iterative algorithm for nonsymmetric systems of linear equations. **Advances in Computational Mathematics**, Springer, v. 2, n. 3, p. 261–277, 1994.
- CUTHILL, E.; MCKEE, J. Reducing the bandwidth of sparse symmetric matrices. In: **ACM. Proceedings of the 1969 24th national conference**. New York, NY, USA, 1969. p. 157–172.
- CZIBULA, G.; CRIȘAN, G.-C.; PINTEA, C.-M.; CZIBULA, I.-G. Soft computing approaches on the bandwidth problem. **Informatica**, Vilnius University Institute of Mathematics and Informatics, v. 24, n. 2, p. 169–180, 2013.
- DAS, R.; MAVRIPLIS, D.; SALTZ, J.; GUPTA, S.; PONNYSAMY, R. Design and implementation of a parallel unstructured Euler solver using software primitives. **AIAA journal**, v. 32, n. 3, p. 489–496, 1994.
- DAVIS, T. A.; HU, Y. The University of Florida sparse matrix collection. **ACM Transactions on Mathematical Software (TOMS)**, ACM, v. 38, n. 1, p. 1, 2011.
- DEVRIES, B.; IANNELLI, J.; TREFFTZ, C.; O’HEARN, K. A.; WOLFFE, G. Parallel implementations of FGMRES for solving large, sparse non-symmetric linear systems. **Procedia Computer Science**, Elsevier, v. 18, p. 491–500, 2013.
- DUFF, I. S.; GRIMES, R. G.; LEWIS, J. G. Users’ guide for the Harwell-Boeing sparse matrix collection (Release I). Citeseer, 1992.
- DUFF, I. S.; KOSTER, J. On algorithms for permuting large entries to the diagonal of a sparse matrix. **SIAM Journal on Matrix Analysis and Applications**, SIAM, v. 22, n. 4, p. 973–996, 2001.
- DUFF, I. S.; MEURANT, G. A. The effect of ordering on preconditioned conjugate gradients. **BIT Numerical Mathematics**, Springer, v. 29, n. 4, p. 635–657, 1989.
- DUFF, I. S.; REID, J. K.; SCOTT, J. A. The use of profile reduction algorithms with a frontal code. **International Journal for Numerical Methods in Engineering**, Wiley Online Library, v. 28, n. 11, p. 2555–2568, 1989.

- DUTTO, L. C. The effect of ordering on preconditioned GMRES algorithm, for solving the compressible navier-stokes equations. **International Journal for Numerical Methods in Engineering**, Wiley Online Library, v. 36, n. 3, p. 457–497, 1993.
- D'AZEVEDO, E. F.; FORSYTH, P. A.; TANG, W.-P. Ordering methods for preconditioned conjugate gradient methods applied to unstructured grid problems. **SIAM Journal on Matrix Analysis and Applications**, SIAM, v. 13, n. 3, p. 944–961, 1992.
- ERHEL, J. A parallel GMRES version for general sparse matrices. **Electronic Transactions on Numerical Analysis**, v. 3, n. 12, p. 160–176, 1995.
- FELIPPA, C. A. Solution of linear equations with skyline-stored symmetric matrix. **Computers & Structures**, Elsevier, v. 5, n. 1, p. 13–29, 1975.
- FERRONATO, M. Preconditioning for sparse linear systems at the dawn of the 21st century: History, current developments, and future perspectives. **ISRN Applied Mathematics**, Hindawi Publishing Corporation, v. 2012, 2012.
- GEORGE, A.; LIU, J. W. An implementation of a pseudoperipheral node finder. **ACM Transactions on Mathematical Software (TOMS)**, ACM, v. 5, n. 3, p. 284–295, 1979.
- GEORGE, J. A. **Computer implementation of the finite element method**. Stanford, CA, USA, 1971.
- GHYSELS, P.; LI, X. S.; GORMAN, C.; ROUET, F.-H. A robust parallel preconditioner for indefinite systems using hierarchical matrices and randomized sampling. In: **IEEE. Parallel and Distributed Processing Symposium (IPDPS), 2017 IEEE International**. Orlando, FL, USA, 2017. p. 897–906.
- GIBBS, N. E. Algorithm 509: A hybrid profile reduction algorithm. **ACM Transactions on Mathematical Software (TOMS)**, ACM, v. 2, n. 4, p. 378–387, 1976.
- GIBBS, N. E.; POOLE JR, W. G.; STOCKMEYER, P. K. An algorithm for reducing the bandwidth and profile of a sparse matrix. **SIAM Journal on Numerical Analysis**, SIAM, v. 13, n. 2, p. 236–250, 1976.
- GUIDETTI, K.; CATABRIGA, L.; BOERES, M. C.; RANGEL, M. C. A study of the influence of sparse matrices reordering algorithms on Krylov-type preconditioned iterative methods. In: **XXXI Iberian Latin American Congress on Computational Methods in Engineering**. Buenos Aires, Argentina: Asociación Argentina de Mecánica Computacional, 2010. XXIX, p. 2323–2343.
- HASHIMOTO, H.; YAGIURA, M.; IBARAKI, T. An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. **Discrete Optimization**, Elsevier, v. 5, n. 2, p. 434–456, 2008.
- HE, K.; TAN, S. X.-D.; ZHAO, H.; LIU, X.-X.; WANG, H.; SHI, G. Parallel GMRES solver for fast analysis of large linear dynamic systems on GPU platforms. **INTEGRATION, the VLSI journal**, Elsevier, v. 52, p. 10–22, 2016.
- HESTENES, M. R.; STIEFEL, E. Methods of conjugate gradients for solving linear systems. **Journal of Research of the National Bureau of Standards**, v. 49, n. 6, p. 409–436, 1952.

HYSOM, D.; POTHEM, A. A scalable parallel algorithm for incomplete factor preconditioning. **SIAM Journal on Scientific Computing**, SIAM, v. 22, n. 6, p. 2194–2215, 2001.

IBARAKI, T.; IMAHORI, S.; NONOBE, K.; SOBUE, K.; UNO, T.; YAGIURA, M. An iterated local search algorithm for the vehicle routing problem with convex time penalty functions. **Discrete Applied Mathematics**, Elsevier, v. 156, n. 11, p. 2050–2069, 2008.

ISAZADEH, A.; IZADKHAH, H.; MOKARRAM, A. A learning based evolutionary approach for minimization of matrix bandwidth problem. **Applied Mathematics and Information Sciences**, v. 6, n. 1, p. 51–57, 2012.

KAPORIN, I. Scaling, preconditioning, and superlinear convergence in GMRES-type iterations. In: **Matrix Methods: Theory, Algorithms and Applications: Dedicated to the Memory of Gene Golub**. NJ, USA: World Scientific, 2010. p. 273–295.

KARYPIS, G.; KUMAR, V. Parallel threshold-based ILU factorization. In: IEEE. **Supercomputing, ACM/IEEE 1997 Conference**. New York, NY, USA, 1997. p. 28–28.

KAVEH, A.; SHARAFI, P. A simple ant algorithm for profile optimization of sparse matrices. **Asian Journal of Civil Engineering (Building and Housing)**, v. 9, n. 1, p. 35–46, 2007.

KAVEH, A.; SHARAFI, P. Optimal priority functions for profile reduction using ant colony optimization. **Finite Elements in Analysis and Design**, Elsevier, v. 44, n. 3, p. 131–138, 2008.

KAVEH, A.; SHARAFI, P. Ordering for bandwidth and profile minimization problems via Charged System Search algorithm. **Iranian Journal of Science and Technology Transaction of Civil Engineering**, v. 36, n. C1, p. 39–52, 2012.

KHODJA, L. Z.; COUTURIER, R.; GIERSCH, A.; BAHI, J. M. Parallel sparse linear solver with GMRES method using minimization techniques of communications for GPU clusters. **The journal of Supercomputing**, Springer, v. 69, n. 1, p. 200–224, 2014.

KING, I. P. An automatic reordering scheme for simultaneous equations derived from network systems. **International Journal for Numerical Methods in Engineering**, Wiley Online Library, v. 2, n. 4, p. 523–533, 1970.

KOOHESTANI, B.; POLI, R. A genetic programming approach to the matrix bandwidth-minimization problem. In: **Parallel Problem Solving from Nature, PPSN XI**. Berlin, Heidelberg: Springer, 2010. p. 482–491.

KOOHESTANI, B.; POLI, R. A hyper-heuristic approach to evolving algorithms for bandwidth reduction based on genetic programming. In: **Research and Development in Intelligent Systems XXVIII**. London: Springer, 2011. p. 93–106.

KOOHESTANI, B.; POLI, R. A genetic programming approach for evolving highly-competitive general algorithms for envelope reduction in sparse matrices. In: **Parallel Problem Solving from Nature-PPSN XII**. Berlin, Heidelberg: Springer, 2012. v. 7492, p. 287–296.

KOOHESTANI, B.; POLI, R. On the application of genetic programming to the envelope reduction problem. In: IEEE. **4th Computer Science and Electronic Engineering Conference (CEECE)**. Colchester, UK, 2012. p. 53–58.

- KOOHESTANI, B.; POLI, R. Evolving an improved algorithm for envelope reduction using a hyper-heuristic approach. **IEEE Transactions on Evolutionary Computation**, IEEE, v. 18, n. 4, p. 543–558, 2014.
- KUMFERT, G.; POTHEN, A. Two improved algorithms for envelope and wavefront reduction. **BIT Numerical Mathematics**, Springer, v. 37, n. 3, p. 559–590, 1997.
- LANCZOS, C. Solution of systems of linear equations by minimized iterations. **Journal of Research of the National Bureau of Standards**, Citeseer, v. 49, n. 1, p. 33–53, 1952.
- LEWIS, R. R. Simulated annealing for profile and fill reduction of sparse matrices. **International journal for numerical methods in engineering**, Wiley Online Library, v. 37, n. 6, p. 905–925, 1994.
- LI, N.; SAAD, Y.; CHOW, E. Crout versions of ILU for general sparse matrices. **SIAM Journal on Scientific Computing**, SIAM, v. 25, n. 2, p. 716–728, 2003.
- LI, R.; SAAD, Y. GPU-accelerated preconditioned iterative linear solvers. **The Journal of Supercomputing**, Springer, v. 63, n. 2, p. 443–466, 2013.
- LI, Z.; SAAD, Y.; SOSONKINA, M. pARMS: a parallel version of the algebraic recursive multilevel solver. **Numerical linear algebra with applications**, Wiley Online Library, v. 10, n. 5-6, p. 485–509, 2003.
- LIM, A.; BRIAN, R.; XIAO, F. Integrated genetic algorithm with hill climbing for bandwidth minimization problem. In: SPRINGER. **Genetic and Evolutionary Computation Conference**. Berlin, 2003. p. 1594–1595.
- LIM, A.; LIN, J.; RODRIGUES, B.; XIAO, F. Ant colony optimization with hill climbing for the bandwidth minimization problem. **Applied Soft Computing**, Elsevier, v. 6, n. 2, p. 180–188, 2006.
- LIM, A.; LIN, J.; XIAO, F. Particle Swarm Optimization and Hill Climbing for the bandwidth minimization problem. **Applied Intelligence**, Springer, v. 26, n. 3, p. 175–182, 2007.
- LIM, A.; RODRIGUES, B.; XIAO, F. A new node centroid algorithm for bandwidth minimization. In: MORGAN KAUFMANN PUBLISHERS INC. **Proceedings of the 18th international joint conference on Artificial intelligence**. Acapulco, Mexico, 2003. p. 1544–1545.
- LIM, A.; RODRIGUES, B.; XIAO, F. A centroid-based approach to solve the bandwidth minimization problem. In: IEEE. **System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on**. Big Island, HI, 2004. p. 6–pp.
- LIM, A.; RODRIGUES, B.; XIAO, F. Heuristics for matrix bandwidth reduction. **European Journal of Operational Research**, Elsevier, v. 174, n. 1, p. 69–91, 2006.
- LIM, A.; RODRIGUES, B.; XIAO, F. A fast algorithm for bandwidth minimization. **International Journal on Artificial Intelligence Tools**, World Scientific, v. 16, n. 03, p. 537–544, 2007.
- LIN, Y.; YUAN, J. Profile minimization problem for matrices and graphs. **Acta Mathematicae Applicatae Sinica (English Series)**, Springer, v. 10, n. 1, p. 107–112, 1994.

- LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search. In: **Handbook of metaheuristics**. Boston, MA: Springer, 2003. p. 320–353.
- LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search: Framework and applications. In: **Handbook of metaheuristics**. Boston, MA: Springer, 2010. p. 363–397.
- LUGON, B.; CATABRIGA, L. Algoritmos de reordenamento de matrizes esparsas aplicados a preconditionadores ILU(p). **XLV Simpósio Brasileiro de Pesquisa Operacional**, p. 2343–2354, 2013.
- LUGON, B.; CATABRIGA, L. Algoritmos de reordenamento de matrizes esparsas aplicados a preconditionadores ILU(p). **XLV Simpósio Brasileiro de Pesquisa Operacional**, p. 2343–2354, 2013.
- MAMAGHANI, A. S.; MEYBODI, M. R. A learning automaton based approach to solve the graph bandwidth minimization problem. In: IEEE. **Application of Information and Communication Technologies (AICT), 2011 5th International Conference on**. Baku, Azerbaijan, 2011. p. 1–5.
- MARTÍ, R.; CAMPOS, V.; PIÑANA, E. A branch and bound algorithm for the matrix bandwidth minimization. **European Journal of Operational Research**, Elsevier, v. 186, n. 2, p. 513–528, 2008.
- MARTÍ, R.; LAGUNA, M.; GLOVER, F.; CAMPOS, V. Reducing the bandwidth of a sparse matrix with tabu search. **European Journal of Operational Research**, Elsevier, v. 135, n. 2, p. 450–459, 2001.
- MARTIN, O. C.; OTTO, S. W. Partitioning of unstructured meshes for load balancing. **Concurrency and Computation: Practice and Experience**, Wiley Online Library, v. 7, n. 4, p. 303–314, 1995.
- MEDEIROS, S.; PIMENTA, P.; GOLDENBERG, P. An algorithm for profile and wavefront reduction of sparse matrices with a symmetric structure. **Engineering computations**, MCB UP Ltd, v. 10, n. 3, p. 257–266, 1993.
- MLADENOVIC, N.; UROSEVIC, D.; PÉREZ-BRITO, D.; GARCÍA-GONZÁLEZ, C. G. Variable neighbourhood search for bandwidth reduction. **European Journal of Operational Research**, Elsevier, v. 200, n. 1, p. 14–27, 2010.
- OLIVEIRA, S. L. G. de; BERNARDES, J. A.; CHAGAS, G. O. An evaluation of low-cost heuristics for matrix bandwidth and profile reductions. **Computational and Applied Mathematics**, Springer, p. 1–60, 2016.
- OLIVEIRA, S. L. G. de; BERNARDES, J. A. B.; CHAGAS, G. O. An evaluation of reordering algorithms to reduce the computational cost of the incomplete Cholesky-conjugate gradient method. **Computational and Applied Mathematics**, Springer, p. 1–40, 2017.
- OLIVEIRA, S. L. Gonzaga de. **Introdução à geração de malhas triangulares**. São Carlos: SBMAC, 2015.
- OLIVEIRA, S. L. Gonzaga de; CHAGAS, G. O. **Introdução a heurísticas para redução de largura de banda de matrizes**. São Carlos: SBMAC, 2014.

- OSEI-KUFFUOR, D.; SAAD, Y. Preconditioning Helmholtz linear systems. **Applied numerical mathematics**, Elsevier, v. 60, n. 4, p. 420–431, 2010.
- PAPADIMITRIOU, C. H. The NP-completeness of the bandwidth minimization problem. **Computing**, Springer, v. 16, n. 3, p. 263–270, 1976.
- PAQUETE, L.; STÜTZLE, T. An experimental investigation of iterated local search for coloring graphs. In: **Applications of Evolutionary Computing**. Berlin, Heidelberg: Springer, 2002. p. 122–131.
- PENNA, P. H. V.; SUBRAMANIAN, A.; OCHI, L. S. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. **Journal of Heuristics**, Springer, v. 19, n. 2, p. 201–232, 2013.
- PESSANHA, J. E.; PORTUGAL, C.; SAAVEDRA, O. Investigação crítica do desempenho do GMRES pré-condicionado via fatoração incompleta LU em estudos de fluxo de carga. **Sba Controle & Automação**, SciELO Brasil, v. 20, n. 4, p. 564–572, 2009.
- PIÑANA, E.; PLANA, I.; CAMPOS, V.; MARTÍ, R. GRASP and path relinking for the matrix bandwidth minimization. **European Journal of Operational Research**, Elsevier, v. 153, n. 1, p. 200–210, 2004.
- PINTEA, C.-M.; CRIȘAN, G.-C.; CHIRA, C. A hybrid ACO approach to the matrix bandwidth minimization problem. In: ROMAY, M. G.; CORCHADO, E.; SEBASTIAN, M. T. G. (Ed.). **Hybrid Artificial Intelligence Systems**. Berlin, Heidelberg: Springer, 2010. v. 6076, p. 405–412.
- PINTEA, C.-M.; CRIȘAN, G. C.; CHIRA, C. Hybrid ant models with a transition policy for solving a complex problem. **Logic Journal of IGPL**, Oxford Univ Press, v. 20, n. 3, p. 560–569, 2012.
- POP, P.; MATEI, O.; COMES, C.-A. Reducing the bandwidth of a sparse matrix with a genetic algorithm. **Optimization**, Taylor & Francis, v. 63, n. 12, p. 1851–1876, 2014.
- PYZARA, A.; BYLINA, B.; BYLINA, J. The influence of a matrix condition number on iterative methods' convergence. In: IEEE. **Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on**. Szczecin, Poland, 2011. p. 459–464.
- REID, J.; SCOTT, J. Ordering symmetric sparse matrices for small profile and wavefront. **International Journal for Numerical Methods in Engineering**, v. 45, n. 12, p. 1737–1755, 1999.
- RODRIGUEZ-TELLO, E.; HAO, J.-K.; TORRES-JIMENEZ, J. An improved simulated annealing algorithm for bandwidth minimization. **European Journal of Operational Research**, Elsevier, v. 185, n. 3, p. 1319–1335, 2008.
- RUGE, J. W.; STÜBEN, K. Algebraic multigrid. In: **Multigrid methods**. Boston, MA: SIAM, 1987. p. 73–130.
- SAAD, Y. Ilut: A dual threshold incomplete LU factorization. **Numerical linear algebra with applications**, Wiley Online Library, v. 1, n. 4, p. 387–402, 1994.

- SAAD, Y. Finding exact and approximate block structures for ILU preconditioning. **SIAM Journal on Scientific Computing**, SIAM, v. 24, n. 4, p. 1107–1123, 2002.
- SAAD, Y. **Iterative methods for sparse linear systems**. 2nd. ed. Philadelphia, PA: SIAM, 2003. ISBN 0898715342.
- SAAD, Y.; SCHULTZ, M. H. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. **SIAM Journal on scientific and statistical computing**, SIAM, v. 7, n. 3, p. 856–869, 1986.
- SAAD, Y.; SOULAIMANI, A.; TOUIHRI, R. Variations on algebraic recursive multilevel solvers (ARMS) for the solution of cfd problems. **Applied numerical mathematics**, Elsevier, v. 51, n. 2-3, p. 305–327, 2004.
- SAAD, Y.; SUCHOMEL, B. ARMS: An algebraic recursive multilevel solver for general sparse linear systems. **Numerical linear algebra with applications**, Wiley Online Library, v. 9, n. 5, p. 359–378, 2002.
- SAAD, Y. e. a. **ITSOL v.2.0: ITERATIVE SOLVERS package**. 2017. Disponível em: <<http://www-users.cs.umn.edu/~saad/software/ITSOL/>>.
- SAKURABA, C. S.; RONCONI, D. P.; BIRGIN, E. G.; YAGIURA, M. Metaheuristics for large-scale instances of the linear ordering problem. **Expert Systems with Applications**, Elsevier, v. 42, n. 9, p. 4432–4442, 2015.
- SÁNCHEZ-ORO, J.; LAGUNA, M.; DUARTE, A.; MARTÍ, R. Scatter search for the profile minimization problem. **Networks**, Wiley Online Library, v. 65, n. 1, p. 10–21, 2015.
- SCHIAVINOTTO, T.; STÜTZLE, T. The linear ordering problem: Instances, search space analysis and algorithms. **Journal of Mathematical Modelling and Algorithms**, Springer, v. 3, n. 4, p. 367–402, 2004.
- SLOAN, S. An algorithm for profile and wavefront reduction of sparse matrices. **International Journal for Numerical Methods in Engineering**, Wiley Online Library, v. 23, n. 2, p. 239–251, 1986.
- SOULÄIMANI, A.; SALAH, N. B.; SAAD, Y. Enhanced GMRES acceleration techniques for some CFD problems. **International Journal of Computational Fluid Dynamics**, Taylor & Francis Group, v. 16, n. 1, p. 1–20, 2002.
- STÜTZLE, T. Iterated local search for the quadratic assignment problem. **European Journal of Operational Research**, Elsevier, v. 174, n. 3, p. 1519–1539, 2006.
- STÜTZLE, T.; HOOS, H. H. Analysing the run-time behaviour of iterated local search for the travelling salesman problem. In: **Essays and Surveys in Metaheuristics**. Boston, MA: Springer, 2002. p. 589–611.
- TORRES-JIMENEZ, J.; IZQUIERDO-MARQUEZ, I.; GARCIA-ROBLEDO, A.; GONZALEZ-GOMEZ, A.; BERNAL, J.; KACKER, R. N. A dual representation simulated annealing algorithm for the bandwidth minimization problem on graphs. **Information Sciences**, Elsevier, v. 303, p. 33–49, 2015.

- VALDEZ, G. C.; MEDINA, S. S. B. Iterated local search for the linear ordering problem. **International Journal of Combinatorial Optimization Problems and Informatics**, International Journal of Combinatorial Optimization Problems & Informatics, v. 3, n. 1, p. 12, 2012.
- VANSTEENWEGEN, P.; SOUFFRIAU, W.; BERGHE, G. V.; OUDHEUSDEN, D. V. Iterated local search for the team orienteering problem with time windows. **Computers & Operations Research**, Elsevier, v. 36, n. 12, p. 3281–3290, 2009.
- VILLANUEVA, J. D. T.; HUACUJA, H. J. F.; RANGEL, R. P.; VALADEZ, J. M. C.; SOBERANES, H. J. P.; BARBOSA, J. J. G. Iterated local search algorithm for the linear ordering problem with cumulative costs (LOPCC). In: **Soft Computing for Intelligent Control and Mobile Robotics**. Berlin, Heidelberg: Springer, 2010. v. 318, p. 395–404.
- VORST, H. A. Van der. Bi-cgstab: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. **SIAM Journal on scientific and Statistical Computing**, SIAM, v. 13, n. 2, p. 631–644, 1992.
- ZHANG, J. A sparse approximate inverse preconditioner for parallel preconditioning of general sparse matrices. **Applied Mathematics and Computation**, Elsevier, v. 130, n. 1, p. 63–85, 2002.

**APÊNDICE A – Resultados das simulações para redução de largura de banda e de *profile***

Nesse apêndice, apresentam-se os resultados das simulações para redução de largura de banda e de *profile* com as heurísticas ILS-Band e ILS-Prof propostas e outras dez heurísticas da literatura consideradas no estado da arte para a resolução desses problemas. As simulações estão divididas entre as executadas sobre 124 instâncias simétricas e as executadas sobre 48 instâncias assimétricas da *SuiteSparse Matrix Collection*.

Tabela A1 – Resultados de redução de largura de banda das heurísticas testadas em 124 instâncias simétricas da *SuiteSparse Matrix Collection*

| Instância       | n     | Original | RBFS-GL  | KP-band-GL | RCM-GL     | GPS         | MPG      | NSloan   | Sloan    | Sloan-MGPS | ILS-Prof   | ILS-Band    | VNS-Band (60s) | FNC-HC      |
|-----------------|-------|----------|----------|------------|------------|-------------|----------|----------|----------|------------|------------|-------------|----------------|-------------|
| utm5940         | 5940  | 497      | 4761     | 3869       | 3659       | 381         | 1763     | 2076     | 1662     | 1626       | 3613       | 497         | 1552           | <b>278</b>  |
| G67             | 10000 | 9900     | 200      | <b>199</b> | <b>199</b> | <b>199</b>  | 350      | 343      | 321      | 341        | 200        | 200         | 2497           | <b>199</b>  |
| crack           | 10240 | 10195    | 191      | 184        | 186        | <b>177</b>  | 641      | 2057     | 524      | 524        | 191        | 188         | 582            | <b>172</b>  |
| sit100          | 10262 | 7381     | 398      | 399        | <b>395</b> | 396         | 1159     | 3806     | 1464     | 1266       | 396        | 398         | 2615           | 414         |
| shuttle_eddy    | 10429 | 9597     | 196      | 210        | 196        | 176         | 814      | 770      | 830      | 824        | 201        | 169         | 586            | <b>166</b>  |
| vsp_p0291...    | 10498 | 10386    | 6642     | 6920       | 6022       | 6722        | 10318    | 10069    | 9682     | 10079      | 6748       | 6377        | 3655           | <b>3092</b> |
| bundle1         | 10581 | 10461    | 6631     | 6278       | 6029       | 3340        | 8207     | 7970     | 7931     | 8014       | 5509       | 3018        | 4166           | <b>2532</b> |
| wing_nodal      | 10937 | 10884    | 1442     | 1258       | 1060       | 1060        | 5417     | 6219     | 3696     | 6057       | 879        | <b>801</b>  | 2681           | 893         |
| CurlCurl_0      | 11083 | 1748     | 708      | 849        | 643        | 548         | 3965     | 4308     | 3727     | 4901       | 532        | 671         | 3305           | <b>400</b>  |
| fe_4elt2        | 11143 | 10838    | 204      | 205        | 200        | 181         | 4388     | 2959     | 862      | 758        | 199        | 217         | 3597           | <b>164</b>  |
| linverse        | 11999 | <b>4</b> | 6        | 7          | <b>4</b>   | <b>4</b>    | <b>4</b> | 7        | 6        | 7          | <b>4</b>   | <b>4</b>    | <b>4</b>       | 8           |
| ncvxqp1         | 12111 | 11107    | 2798     | 2776       | 2767       | 2743        | 10295    | 10437    | 9586     | 10162      | 3320       | 2679        | 3674           | <b>2145</b> |
| vibrobox        | 12328 | 12162    | 4840     | 4822       | 4640       | 4594        | 11717    | 11093    | 11157    | 11384      | 3762       | 2834        | 3643           | <b>2643</b> |
| stokes64        | 12546 | 8450     | 370      | 389        | 384        | 346         | 824      | 1547     | 593      | 593        | 519        | 392         | 1156           | <b>319</b>  |
| tuma2           | 12992 | 9018     | 333      | 338        | 328        | 332         | 1573     | 4634     | 1582     | 1511       | 546        | 323         | 385            | <b>294</b>  |
| cbuckle         | 13681 | 13577    | 778      | 953        | 943        | 1001        | 2951     | 3113     | 3308     | 3100       | 13577      | 757         | 2648           | <b>754</b>  |
| cyl6            | 13681 | 13577    | 719      | <b>713</b> | <b>713</b> | <b>713</b>  | 2631     | 2576     | 2610     | 2574       | 13577      | 719         | 2487           | 719         |
| crystm02        | 13965 | 459      | 275      | 266        | 272        | <b>158</b>  | 490      | 606      | 573      | 573        | 242        | 277         | 3955           | 261         |
| bcstk29         | 13992 | 1157     | 892      | 990        | 905        | 989         | 3543     | 3714     | 6041     | 7199       | 881        | 766         | 5470           | <b>694</b>  |
| Pres_Poisson    | 14822 | 12583    | 362      | 335        | 379        | 334         | 495      | 312      | 570      | 601        | 300        | 295         | 2941           | <b>292</b>  |
| opt1            | 15449 | 1632     | 1568     | 1509       | 1539       | 1348        | 5825     | 6543     | 6052     | 7734       | 1632       | 1479        | 4609           | <b>1263</b> |
| hangGlider_4    | 15561 | 13832    | 13829    | 14214      | 13829      | 13806       | 15552    | 15547    | 15547    | 15559      | 14813      | 12734       | 7114           | <b>6915</b> |
| Dubcoval        | 16129 | 16052    | 511      | 502        | 500        | 534         | 790      | 1831     | 909      | 909        | 512        | 512         | 3472           | <b>416</b>  |
| olafu           | 16146 | 593      | 564      | 556        | 569        | 533         | 8650     | 5581     | 4914     | 5915       | 663        | 565         | 689            | <b>531</b>  |
| net50           | 16320 | 16246    | 6558     | 6728       | 7071       | 5032        | 14238    | 14359    | 14273    | 14359      | 11544      | 5417        | 5124           | <b>3603</b> |
| fe_sphere       | 16386 | 439      | 247      | 257        | 256        | 258         | 6362     | 3618     | 568      | 402        | 256        | 256         | 3991           | <b>237</b>  |
| pds10           | 16558 | 15451    | 3767     | 3223       | 3916       | 3902        | 16409    | 14641    | 14671    | 14647      | 15451      | 2492        | 5341           | <b>2323</b> |
| ex3sta1         | 16782 | 12921    | 4541     | 4876       | 5078       | 4129        | 16400    | 16422    | 16415    | 16599      | 5136       | 3891        | 4751           | <b>3475</b> |
| ramage02        | 16830 | 3388     | 2938     | 3114       | 3046       | 2513        | 3265     | 3822     | 3362     | 3181       | 2943       | <b>2289</b> | 4942           | 2308        |
| cti             | 16840 | 12942    | 614      | 662        | <b>592</b> | 613         | 4900     | 9440     | 2188     | 2511       | 601        | 613         | 4487           | 589         |
| gyro            | 17361 | 5144     | 1776     | 1739       | 1739       | 1703        | 10595    | 12534    | 10156    | 12872      | 1603       | 1187        | 1307           | <b>940</b>  |
| lowThrust_7     | 17378 | 15449    | 14374    | 10439      | 11130      | 11129       | 17193    | 17323    | 17323    | 17323      | 11551      | 8096        | 5667           | <b>5455</b> |
| L-9             | 17983 | 2378     | 228      | 245        | 219        | 276         | 4334     | 5182     | 378      | 738        | 211        | 277         | 487            | <b>192</b>  |
| nd6k            | 18000 | 16766    | 4776     | 4091       | 4320       | <b>4037</b> | 11818    | 10920    | 10900    | 10727      | 4150       | 4181        | 5904           | 4078        |
| crplat2         | 18010 | 17757    | 521      | 595        | 476        | <b>437</b>  | 7388     | 7465     | 7871     | 6886       | 519        | 495         | 3302           | 444         |
| tandem_vtx      | 18454 | 18411    | 1673     | 1671       | 1665       | 1912        | 5899     | 4320     | 4524     | 4427       | 1205       | <b>1084</b> | 5843           | 1391        |
| ford1           | 18728 | 18704    | 296      | 323        | 309        | 293         | 2672     | 9722     | 2277     | 2403       | 282        | <b>264</b>  | 3458           | <b>265</b>  |
| fxm4_6          | 18892 | 18815    | 9501     | 9008       | 8925       | 8925        | 18524    | 17049    | 17637    | 17163      | 18814      | <b>4141</b> | 5945           | 5536        |
| whitaker3_dual  | 19190 | 19178    | 103      | 101        | 105        | 73          | 403      | 2363     | 305      | 304        | 201        | 103         | 5188           | 94          |
| pattern1        | 19242 | 19231    | 14011    | 14062      | 11106      | 8564        | 18358    | 16487    | 16591    | 16430      | 15617      | 8603        | 9609           | <b>8490</b> |
| bodyy6          | 19366 | 18605    | 333      | 329        | 348        | 295         | 4436     | 4822     | 1437     | 1128       | 350        | 266         | 2360           | <b>259</b>  |
| raefsky4        | 19779 | 11786    | 1157     | 1023       | 1038       | 1052        | 6891     | 7447     | 6538     | 7356       | 1159       | <b>980</b>  | 2146           | 983         |
| Si5H12          | 19896 | 7105     | 4021     | 4020       | 4154       | <b>3836</b> | 9269     | 10872    | 11016    | 10085      | 4051       | 4096        | 4731           | 4069        |
| LF10000         | 19998 | 3        | <b>3</b> | <b>3</b>   | <b>3</b>   | <b>3</b>    | <b>3</b> | <b>3</b> | <b>3</b> | <b>3</b>   | <b>3</b>   | <b>3</b>    | <b>3</b>       | 5           |
| qpband          | 20000 | 15000    | <b>2</b> | <b>2</b>   | <b>2</b>   | <b>2</b>    | <b>2</b> | <b>2</b> | <b>2</b> | <b>2</b>   | <b>2</b>   | <b>2</b>    | 4003           | <b>2</b>    |
| Trefethen_20000 | 20000 | 16384    | 7351     | 7453       | 7285       | 7545        | 19308    | 18070    | 19104    | 17871      | 7366       | 7366        | 8054           | <b>7142</b> |
| tsyl201         | 20685 | 20384    | 1149     | 1124       | 1012       | 1001        | 2357     | 2562     | 2562     | 2365       | <b>827</b> | <b>827</b>  | 3255           | 910         |
| tubel           | 21498 | 2759     | 2707     | 2742       | 2743       | 1144        | 14545    | 14109    | 14336    | 13901      | 2679       | 1959        | 1199           | <b>1141</b> |

Continua na próxima página

| Instância    | n     | Original | RBFS-GL     | KP-band-GL | RCM-GL   | GPS        | MPG      | NSloan | Sloan    | Sloan-MGPS | ILS-Prof    | ILS-Band    | VNS-Band (60s) | FNC-HC       |
|--------------|-------|----------|-------------|------------|----------|------------|----------|--------|----------|------------|-------------|-------------|----------------|--------------|
| biplane-9    | 21701 | 4165     | 439         | 517        | 443      | 422        | 4640     | 8309   | 2353     | 2248       | 3065        | 430         | 526            | <b>375</b>   |
| trdheim      | 22098 | 533      | 783         | 732        | 804      | 719        | 3930     | 4102   | 4084     | 4068       | 869         | <b>533</b>  | 842            | 638          |
| pli          | 22695 | 1357     | 1977        | 1861       | 1961     | 1725       | 11553    | 11104  | 10204    | 14565      | <b>1357</b> | <b>1357</b> | 1862           | 1730         |
| tuma1        | 22967 | 16032    | 477         | 508        | 489      | 474        | 2338     | 16580  | 2227     | 2210       | 868         | 493         | 3928           | <b>444</b>   |
| msc23052     | 23052 | 23046    | 1310        | 1314       | 1394     | 1462       | 17500    | 15659  | 15527    | 18025      | 1466        | 1416        | 4016           | <b>1095</b>  |
| ca-CondMat   | 23133 | 23019    | 9440        | 9301       | 9496     | 10431      | 21211    | 20247  | 20473    | 20111      | 9457        | 9704        | 10627          | <b>5301</b>  |
| stufe-10     | 24010 | 7466     | 700         | 673        | 948      | <b>516</b> | 10112    | 10430  | 1695     | 3847       | 7185        | 617         | 5854           | <b>607</b>   |
| de2010       | 24115 | 23905    | 657         | 582        | 583      | 797        | 11326    | 3573   | 2342     | 2342       | 917         | 641         | 2298           | <b>473</b>   |
| rajat09      | 24482 | 24473    | 522         | 523        | 525      | 526        | 1006     | 1025   | 979      | 975        | 522         | 527         | 6745           | <b>272</b>   |
| crystk03     | 24696 | 572      | 1033        | 1015       | 1034     | <b>557</b> | 1061     | 1062   | 1062     | 1062       | 791         | 572         | 3312           | 906          |
| pcrystk03    | 24696 | 572      | 1033        | 1020       | 1020     | <b>557</b> | 1060     | 1062   | 1062     | 1062       | 791         | 572         | 3312           | 914          |
| dtoc         | 24993 | 19994    | <b>4</b>    | <b>4</b>   | <b>4</b> | <b>4</b>   | <b>4</b> | 5      | <b>4</b> | <b>4</b>   | <b>4</b>    | <b>4</b>    | 6              | 6            |
| hi2010       | 25016 | 8836     | 1081        | 1029       | 1093     | 925        | 7795     | 3092   | 3440     | 2419       | 1760        | 1191        | 1329           | <b>679</b>   |
| ri2010       | 25181 | 22696    | 936         | 992        | 790      | 692        | 9213     | 4024   | 2648     | 2843       | 2172        | 756         | 6265           | <b>581</b>   |
| bsstk37      | 25503 | 19193    | 1534        | 1493       | 1427     | 1456       | 20352    | 16920  | 16453    | 19569      | 19193       | 1403        | 2308           | <b>1378</b>  |
| smt          | 25710 | 19228    | 2649        | 2501       | 2530     | 2227       | 14654    | 15441  | 16176    | 15255      | 2512        | 2491        | 5295           | <b>1977</b>  |
| brainpc2     | 27607 | 20707    | 20690       | 20690      | 20689    | 13799      | 27589    | 26752  | 26025    | 25783      | 23024       | 19168       | <b>10042</b>   | 20689        |
| bratu3d      | 27792 | 26687    | 944         | 962        | 943      | 988        | 2287     | 2282   | 1626     | 1598       | <b>941</b>  | 952         | 6448           | 955          |
| bloweya      | 30004 | 20004    | 19993       | 19994      | 19994    | 19989      | 29992    | 29993  | 29993    | 29993      | 12729       | <b>9618</b> | 9636           | 10878        |
| rajat10      | 30202 | 30193    | 584         | 582        | 581      | 586        | 1132     | 1230   | 1054     | 1122       | 588         | 583         | 10018          | <b>318</b>   |
| big_dual     | 30269 | 29706    | 360         | 369        | 369      | 359        | 3170     | 6693   | 1573     | 2004       | 29706       | 373         | 6222           | <b>336</b>   |
| wathen100    | 30401 | 304      | 533         | 566        | 551      | 492        | 1579     | 3658   | 1519     | 1290       | 490         | <b>304</b>  | 732            | 502          |
| helm3d01     | 32226 | 32070    | 2619        | 2675       | 2607     | 2855       | 7338     | 6957   | 7528     | 7082       | 2558        | <b>2155</b> | 7578           | 2279         |
| lp11         | 32460 | 29706    | 14643       | 13495      | 14226    | 16414      | 30877    | 29439  | 29490    | 29469      | 11504       | <b>7538</b> | 10336          | 9673         |
| vt2010       | 32580 | 31554    | 786         | 763        | 717      | 668        | 14045    | 3932   | 3020     | 3834       | 9431        | 742         | 2129           | <b>616</b>   |
| delaunay_n15 | 32768 | 32506    | 899         | 910        | 873      | <b>737</b> | 5291     | 4341   | 3172     | 3460       | 2838        | 2802        | 6698           | 793          |
| se           | 32768 | 16383    | 4525        | 4525       | 4525     | 4525       | 15513    | 21236  | 11343    | 11343      | 974         | <b>904</b>  | 8560           | 4012         |
| SiO          | 33401 | 10976    | 6575        | 6467       | 6456     | 6044       | 13834    | 13127  | 13112    | 14945      | <b>4921</b> | 6116        | 7811           | 6596         |
| ship_001     | 34920 | 32993    | <b>1055</b> | 1095       | 1090     | 1073       | 2732     | 2161   | 2285     | 2169       | 1055        | 1055        | 2867           | <b>987</b>   |
| aug3dcqp     | 35543 | 29215    | 994         | 994        | 994      | 985        | 2619     | 2383   | 2286     | 2335       | 2072        | 836         | 7812           | <b>714</b>   |
| pdb1HYS      | 36417 | 34064    | 3129        | 2936       | 2709     | 2489       | 18083    | 16829  | 17073    | 17275      | 2384        | 2810        | 9997           | <b>2332</b>  |
| shock-9      | 36476 | 2538     | 340         | 307        | 319      | 293        | 4460     | 6509   | 1264     | 1090       | 2538        | 337         | 1742           | <b>279</b>   |
| pwt          | 36519 | 35455    | 366         | 360        | 367      | 340        | 2848     | 8585   | 2459     | 1698       | 392         | <b>336</b>  | 2345           | 340          |
| email-Enron  | 36692 | 34414    | 20865       | 21881      | 21857    | 18431      | 32315    | 33130  | 33280    | 33280      | 29547       | 22718       | 17768          | <b>9789</b>  |
| pkustk05     | 37164 | 37013    | 1765        | 1716       | 1705     | 1661       | 24325    | 28684  | 11403    | 18281      | 1487        | 1469        | 9391           | <b>1248</b>  |
| finance256   | 37376 | 37348    | 2023        | 2076       | 2053     | 2313       | 8277     | 5257   | 8889     | 8711       | 1889        | <b>1309</b> | 2025           | 1968         |
| c-57         | 37833 | 37774    | 17864       | 17904      | 17033    | 18533      | 37677    | 36906  | 36813    | 37810      | 20725       | 13353       | 10471          | <b>9974</b>  |
| minsurfo     | 40806 | 202      | 204         | 203        | 203      | <b>202</b> | 399      | 358    | 302      | 302        | 204         | <b>202</b>  | 7113           | 204          |
| OPF_10000    | 43887 | 42825    | 2576        | 2557       | 2476     | 2440       | 17700    | 11841  | 10237    | 15734      | 2499        | 2007        | 1696           | <b>1470</b>  |
| vanbody      | 47072 | 3526     | 2212        | 2170       | 2211     | 2437       | 30647    | 25546  | 24262    | 26544      | 2269        | 2212        | 2699           | <b>1955</b>  |
| gridgena     | 48962 | 405      | 372         | 443        | 537      | 344        | 5874     | 10358  | 6777     | 7325       | 566         | 405         | 11826          | <b>336</b>   |
| stokes128    | 49666 | 33282    | 775         | 776        | 768      | 690        | 1497     | 5912   | 1106     | 1110       | 1096        | 771         | 1198           | <b>660</b>   |
| cvxbqp1      | 50000 | 33333    | 1670        | 1665       | 1685     | 1690       | 8380     | 29265  | 7830     | 8063       | 1578        | <b>1400</b> | 6859           | 1534         |
| sparsine     | 50000 | 45454    | 25538       | 25689      | 27707    | 30730      | 39915    | 41667  | 40137    | 44307      | 26366       | 19232       | <b>17134</b>   | 18281        |
| copter2      | 55476 | 55279    | 2671        | 2698       | 2744     | 2705       | 14788    | 10280  | 11039    | 10990      | 2550        | 2294        | 5637           | <b>2047</b>  |
| dixmaan1     | 60000 | 40000    | 7           | 7          | 7        | 8          | 14       | 11514  | 11       | 12         | 12          | 8           | 19994          | <b>6</b>     |
| blockqp1     | 60012 | 40012    | 40010       | 60005      | 60005    | 30008      | 60008    | 60008  | 60008    | 60008      | 56929       | 32648       | 29648          | <b>25007</b> |
| Dubcova2     | 65025 | 64820    | 1024        | 1010       | 1012     | 1072       | 1725     | 3247   | 1751     | 1709       | 1023        | 1024        | 15848          | <b>834</b>   |
| qa8fk        | 66127 | 1048     | 2035        | 1972       | 1991     | 1421       | 13421    | 23946  | 6970     | 6996       | 1384        | <b>1048</b> | 6056           | 1888         |
| gas_sensor   | 66917 | 2901     | 1979        | 1954       | 1962     | 1842       | 4853     | 10225  | 5527     | 5265       | <b>1582</b> | 1958        | 8733           | 1740         |

Continua na próxima página

| Instância             | n      | Original | RBFS-GL    | KP-band-GL | RCM-GL     | GPS         | MPG      | NSloan    | Sloan    | Sloan-MGPS | ILS-Prof     | ILS-Band     | VNS-Band<br>(60s) | FNC-HC       |
|-----------------------|--------|----------|------------|------------|------------|-------------|----------|-----------|----------|------------|--------------|--------------|-------------------|--------------|
| H2O                   | 67024  | 14976    | 7436       | 7096       | 7252       | 7511        | 28406    | 15636     | 15976    | 14771      | 9232         | <b>6883</b>  | 16037             | 7309         |
| cfdl                  | 70656  | 6229     | 3359       | 3379       | 3465       | 3035        | 19685    | 24569     | 20956    | 19141      | <b>2467</b>  | 2779         | 11839             | 2799         |
| finan512              | 74752  | 74724    | 1323       | 1290       | 1331       | 1331        | 3879     | 3922      | 4109     | 3993       | 1446         | 1283         | 1279              | <b>1233</b>  |
| rail_79841            | 79841  | 79811    | 552        | 501        | 550        | 454         | 8704     | 27447     | 8430     | 3691       | 552          | 552          | 14672             | <b>421</b>   |
| cont-201              | 80595  | 69996    | 404        | 405        | <b>403</b> | <b>403</b>  | 602      | 27390     | 599      | 664        | 578          | <b>403</b>   | 15917             | 404          |
| apache1               | 80800  | 808      | 792        | 794        | 792        | 792         | 1278     | 1485      | 1240     | 1261       | 792          | <b>791</b>   | 16647             | 809          |
| shallow_water1        | 81920  | 40959    | 345        | 345        | 335        | <b>322</b>  | 908      | 2044      | 783      | 886        | <b>322</b>   | 353          | 23242             | 328          |
| shallow_water2        | 81920  | 40959    | 347        | 334        | 324        | <b>322</b>  | 839      | 6752      | 841      | 759        | <b>322</b>   | <b>322</b>   | 23242             | 335          |
| thermal1              | 82654  | 80916    | 218        | 231        | 231        | 215         | 1268     | 1314      | 648      | 713        | 464          | <b>210</b>   | 4773              | 230          |
| consph                | 83334  | 44024    | 6087       | 5400       | 5956       | <b>4144</b> | 12315    | 13667     | 13024    | 14688      | 24224        | 5441         | 16859             | 4533         |
| c-72                  | 84064  | 84063    | 33969      | 33990      | 33964      | 54091       | 83164    | 83335     | 83335    | 83335      | 41011        | <b>21125</b> | 23703             | 28399        |
| ncvxqp7               | 87500  | 79996    | 18360      | 17841      | 18302      | 17767       | 79663    | 77023     | 73556    | 74583      | 16183        | <b>12248</b> | 27101             | 15059        |
| olesnik0              | 88263  | 64453    | 1255       | 1241       | 1274       | 1583        | 4257     | 29659     | 4375     | 4287       | 1223         | 1205         | 17261             | <b>1131</b>  |
| denormal              | 89400  | 596      | 598        | 601        | 597        | 597         | 893      | 1173      | 930      | 927        | 600          | <b>596</b>   | 28966             | 614          |
| s3dkq4m2              | 90449  | 614      | 1217       | 1207       | 1206       | 614         | 4553     | 7193      | 4563     | 4697       | <b>608</b>   | <b>608</b>   | 8113              | 1002         |
| s4dkt3m2              | 90449  | 614      | 609        | <b>608</b> | <b>608</b> | <b>608</b>  | 1138     | 1781      | 1109     | 1123       | 843          | 614          | 5852              | 677          |
| boyd1                 | 93279  | 93269    | 89512      | 89508      | 89508      | 46637       | 93276    | 93276     | 93276    | 93276      | 93275        | 82026        | 46640             | <b>46636</b> |
| tandem_dual           | 94069  | 70502    | 2562       | 2638       | 2386       | 2280        | 13583    | 16779     | 8372     | 8084       | <b>1889</b>  | 2283         | 26476             | 2227         |
| pkustk13              | 94893  | 90800    | 5035       | 4968       | 5122       | 5207        | 27889    | 33149     | 28852    | 29951      | 5368         | 5002         | 25067             | <b>4463</b>  |
| Si34H36               | 97569  | 18908    | 15834      | 15715      | 15826      | 17017       | 39080    | 41309     | 40843    | 41350      | <b>10867</b> | 12329        | 30703             | 16063        |
| 2cubes_sphere         | 101492 | 100407   | 4751       | 4732       | 4772       | 6855        | 11160    | 13807     | 11645    | 11023      | 5313         | 4543         | 32116             | <b>4509</b>  |
| thermomech_TC         | 102158 | 102138   | <b>261</b> | 275        | 271        | 270         | 2801     | 2881      | 897      | 898        | 294          | 277          | 7121              | 282          |
| thermomech_TK         | 102158 | 102138   | 261        | 269        | 258        | 270         | 2765     | 2891      | 898      | 900        | 310          | 264          | 7121              | 266          |
| filter3D              | 106437 | 8276     | 3675       | 3369       | 3586       | 3690        | 8873     | 9792      | 9744     | 9062       | 3564         | <b>2475</b>  | 5448              | 2482         |
| 598a                  | 110971 | 110811   | 7750       | 8160       | 8292       | 9142        | 29494    | 23215     | 21975    | 23236      | 4629         | <b>4336</b>  | 28479             | 7195         |
| Ge87H76               | 112985 | 21340    | 18398      | 18417      | 18397      | 17988       | 69682    | 72334     | 70915    | 68311      | <b>12673</b> | 14887        | 35118             | 18529        |
| Núm. melhores         |        |          | 4          | 6          | 10         | 21          | 4        | 2         | 3        | 3          | 16           | 34           | 4                 | 66           |
| $\Sigma \rho_{\beta}$ |        |          | 68.9797    | 65.8537    | 63.4706    | 39.4073     | 725.6986 | 2960.7800 | 474.2705 | 495.9671   | 257.2746     | 25.8714      | 6308.8192         | 17.3802      |

Tabela A2 – Resultados de redução de *profile* das heurísticas testadas em 124 instâncias simétricas da *SuiteSparse Matrix Collection*

| Instância       | n     | Original     | RBFS-GL      | KP-band-GL   | RCM-GL       | GPS          | MPG             | NSloan          | Sloan          | Sloan-MGPS      | ILS-Prof        | ILS-Band        | VNS-Band (60s) | FNC-HC    |
|-----------------|-------|--------------|--------------|--------------|--------------|--------------|-----------------|-----------------|----------------|-----------------|-----------------|-----------------|----------------|-----------|
| utm5940         | 5940  | 1827101      | 2777355      | 2524863      | 2420531      | 2257927      | <b>1337963</b>  | 1961690         | 1848141        | 1859518         | 2295900         | 3618808         | 8042772        | 2191696   |
| G67             | 10000 | 1970197      | 1340182      | 1340113      | 1339539      | 1342507      | <b>1337963</b>  | 1340859         | 1340682        | 1340901         | 1340629         | 1340217         | 3636308        | 1340155   |
| crack           | 10240 | 43760373     | 955652       | 1254362      | 939730       | 968807       | <b>743316</b>   | 751327          | 763669         | 763669          | 955652          | 950752          | 1870122        | 1288747   |
| sit100          | 10262 | 18393811     | 2165854      | 2863739      | 2287710      | 2299615      | 1546879         | <b>1426845</b>  | 1768363        | 1653998         | 2136300         | 2456495         | 11701230       | 2899577   |
| shuttle_eddy    | 10429 | 11656811     | 709140       | 821739       | 709132       | 708216       | <b>571277</b>   | 590143          | 615362         | 602377          | 709368          | 689615          | 1535150        | 818476    |
| vsp_p0291...    | 10498 | 23156785     | 20902534     | 40454285     | 19436006     | 39442296     | 9705108         | 8936198         | <b>8896922</b> | 8924193         | 15089756        | 20048375        | 22450862       | 19742849  |
| bundle1         | 10581 | 54480684     | 6674071      | 26075552     | 3378140      | 9345694      | 1458364         | 1463849         | 1505171        | <b>1435063</b>  | 2099836         | 9109017         | 18976399       | 13465346  |
| wing_nodal      | 10937 | 38007800     | 8270663      | 8733842      | 6641127      | 7708187      | 4554919         | <b>4408160</b>  | 5750870        | 4417277         | 5035546         | 5129318         | 17841331       | 7498325   |
| CurlCurl_0      | 11083 | 8739886      | 3725931      | 4857019      | 3439146      | 3014835      | <b>2352119</b>  | 2353965         | 2462524        | 3115227         | 2843015         | 3748322         | 12622069       | 3279450   |
| fe_4elt2        | 11143 | 24510012     | 1250914      | 1338920      | 1241109      | 1195667      | 1029087         | 1091763         | 1066667        | <b>1022376</b>  | 1117705         | 1272008         | 4723646        | 1321826   |
| linverse        | 11999 | <b>41989</b> | 49184        | 65973        | <b>41989</b> | <b>41989</b> | <b>41989</b>    | <b>41989</b>    | <b>41989</b>   | <b>41989</b>    | <b>41989</b>    | <b>41989</b>    | <b>41989</b>   | 76320     |
| ncvxqp1         | 12111 | 54092365     | 18241757     | 22650932     | 18084291     | 22599196     | <b>8766615</b>  | 10220168        | 8318470        | 9318470         | 15976557        | 16678656        | 24135426       | 19205017  |
| vibrobox        | 12328 | 45720334     | 31441783     | 39589512     | 30710064     | 37205034     | <b>7953845</b>  | 8543384         | 9045913        | 9212234         | 18266679        | 16799800        | 30128113       | 23651379  |
| stokes64        | 12546 | 35031296     | 3046793      | 3230996      | 3164508      | 2750556      | <b>1901430</b>  | 2391023         | 1950875        | 1952400         | 2939682         | 3228200         | 6833264        | 2848067   |
| tuma2           | 12992 | 36191690     | 2364400      | 2404371      | 2303930      | 2291139      | 1735925         | 1829202         | 1825779        | 1825779         | 2249977         | 2306500         | 2964050        | 2513199   |
| cbuckle         | 13681 | 3203982      | 6075770      | 6340530      | 6147635      | 6126820      | 6855167         | 6978967         | 6976049        | 6894964         | <b>3203979</b>  | 6123019         | 22204321       | 6200229   |
| cy16            | 13681 | 3205584      | 6143551      | 6150898      | 6107868      | 6095232      | 7967456         | 7671941         | 7673688        | 7673748         | <b>3205584</b>  | 6174423         | 18410920       | 6209610   |
| crystm02        | 13965 | 6134814      | 2152188      | 2245014      | 2139233      | 1924542      | 1950245         | <b>1889094</b>  | 1929387        | 1929387         | 1949770         | 2152219         | 2739660        | 2300782   |
| bcstk29         | 13992 | 7441166      | 6998455      | 8946474      | 7129315      | 7699086      | <b>3249121</b>  | 4785106         | 5586636        | 5955595         | 5162327         | 23513985        | 7649063        | 7649063   |
| Pres_Poisson    | 14822 | 9789525      | 3081623      | 3078179      | 3114030      | 2916499      | 2832494         | 2858324         | 2831709        | <b>2831252</b>  | 2889410         | 2965473         | 32119189       | 3112863   |
| opt1            | 15449 | 8263744      | 9039003      | 13210502     | 9064654      | 10317995     | 6935343         | 7067090         | <b>6893576</b> | 7300450         | 8263744         | 8734313         | 38890526       | 12116673  |
| hangGlider_4    | 15561 | 101628355    | 74735493     | 118432051    | 74151361     | 98263263     | 6649811         | 1769837         | 1769837        | <b>1767333</b>  | 6566244         | 41606654        | 61870725       | 60315661  |
| Dubcoval        | 16129 | 120327369    | 2730009      | 5374546      | 2722691      | 3486708      | 2599412         | <b>2303298</b>  | 2640901        | 2640901         | 2726367         | 2728189         | 14948511       | 4917621   |
| olafu           | 16146 | 4951980      | 5055509      | 5184744      | 4997669      | 5068842      | 4683716         | 4521433         | <b>4229465</b> | 4667417         | 4836307         | 5025848         | 6692905        | 5448340   |
| net50           | 16320 | 104280210    | 37922795     | 56551957     | 47948321     | 37897524     | <b>26648414</b> | 28079908        | 27835774       | 28015365        | 40218330        | 34804872        | 48003402       | 36719920  |
| fe_sphere       | 16386 | 2883775      | 2940545      | 2916820      | 2910971      | 2810512      | 3157683         | 2956851         | 2809411        | 3041623         | <b>2808260</b>  | <b>2808260</b>  | 11535547       | 3121316   |
| pds10           | 16558 | 18035808     | 31909849     | 35994525     | 31732482     | 31521682     | 12908541        | <b>12733380</b> | 13284665       | 13282254        | 18035806        | 22424672        | 46804896       | 27950223  |
| ex3sta1         | 16782 | 105040221    | 34783052     | 55049283     | 44737081     | 44660460     | 67505878        | 68352183        | 69242050       | 68908256        | 34298447        | <b>33713621</b> | 52892755       | 38734237  |
| ramage02        | 16830 | 24469981     | 23966577     | 35188485     | 25303839     | 25327737     | <b>21614387</b> | 21619813        | 21694105       | 21739060        | 23850754        | 24376104        | 54623349       | 29394507  |
| cti             | 16840 | 15472870     | 6847560      | 7251931      | 7127826      | 7233765      | 6180171         | <b>6081988</b>  | 6941454        | 6633498         | 6762818         | 7218527         | 29967706       | 7467946   |
| gyro            | 17361 | 14326901     | 6650738      | 13970511     | 6465443      | 6623306      | 6052982         | 7606740         | <b>4992024</b> | 6985374         | 6543893         | 6227013         | 12675891       | 10220288  |
| lowThrust_7     | 17378 | 88555135     | 80285553     | 110999006    | 72707788     | 72688629     | 1311456         | <b>455611</b>   | <b>455611</b>  | <b>455611</b>   | 38211337        | 43042052        | 60123813       | 52371375  |
| L-9             | 17983 | 2668337      | 2959777      | 3178073      | 2801555      | 3469588      | 2208909         | <b>1703416</b>  | 2418927        | 2484588         | 2195135         | 3574946         | 3391273        | 2627210   |
| nd6k            | 18000 | 119066489    | 53998686     | 54258235     | 51917267     | 53719783     | 46317130        | 46745846        | 46730769       | <b>45686618</b> | 49518794        | 51878908        | 78716939       | 57749634  |
| crplat2         | 18010 | 22125142     | 6063836      | 6471714      | 5811374      | 5823822      | 6340245         | 5761990         | 5647276        | 6140062         | <b>5402210</b>  | 5674853         | 35234938       | 6231120   |
| tandem_vtx      | 18454 | 81008429     | 15299895     | 19027027     | 15461056     | 14416158     | <b>5240441</b>  | 6376337         | 6360822        | 6144489         | 8998901         | 8904615         | 48357187       | 19138307  |
| ford1           | 18728 | 35197733     | 3088734      | 3543874      | 3141965      | 3034119      | <b>2186467</b>  | 2787461         | 2414599        | 2393700         | 2404475         | 2662814         | 7723399        | 3417199   |
| fxm4_6          | 18892 | 10272265     | 73905152     | 114011881    | 72498893     | 72498726     | 16411453        | 34473257        | 33809242       | 33988397        | <b>10272015</b> | 22794118        | 62985459       | 64773828  |
| whitaker3_dual  | 19190 | 7155543      | 1330046      | 1348612      | 1329069      | 1241010      | <b>1200436</b>  | 1291981         | 1304860        | 1299239         | 1333444         | 1340204         | 5500747        | 1325163   |
| pattern1        | 19242 | 142682527    | 97259173     | 147356630    | 107201971    | 99077712     | 113207201       | 116022955       | 116730238      | 115795259       | <b>90930708</b> | 102226122       | 112275719      | 103231414 |
| bodyy6          | 19366 | 14954304     | 4139397      | 4110894      | 4242899      | 4008541      | 3150424         | <b>3062640</b>  | 3656826        | 3953376         | 3396630         | 3661092         | 8983610        | 3814323   |
| raefsky4        | 19779 | 19611188     | 12290053     | 13173417     | 12461912     | 12351011     | 8185383         | 8113474         | 8489384        | <b>8086695</b>  | 10709300        | 11302017        | 27109930       | 13091137  |
| Si5H12          | 19896 | 90257212     | 57393158     | 57612868     | 57665157     | 56185768     | <b>53406707</b> | 54088996        | 54150395       | 54455066        | 56755713        | 55258520        | 68958420       | 60944440  |
| LF10000         | 19998 | 49990        | <b>49990</b> | <b>49990</b> | <b>49990</b> | <b>49990</b> | <b>49990</b>    | <b>49990</b>    | <b>49990</b>   | <b>49990</b>    | <b>49990</b>    | <b>49990</b>    | <b>49990</b>   | 50481     |
| qpbnd           | 20000 | 75000000     | <b>15000</b> | 20000        | <b>15000</b> | 20000        | <b>15000</b>    | <b>15000</b>    | <b>15000</b>   | <b>15000</b>    | <b>15000</b>    | <b>15000</b>    | 16004          | 19999     |
| Trefethen_20000 | 20000 | 148723029    | 100331174    | 101901890    | 100048992    | 101499853    | 99479275        | 98805966        | 98583152       | <b>98537508</b> | 99608072        | 100734648       | 115088989      | 98759812  |
| tsyl201         | 20685 | 54454233     | 11498723     | 15842797     | 11137897     | 11719074     | <b>10270513</b> | 10343035        | 10343035       | 10356438        | 10525746        | 10525746        | 35405766       | 15672202  |
| tubel           | 21498 | 21781370     | 21342996     | 22004640     | 21771810     | 16946287     | <b>16360887</b> | 23159697        | 23059363       | 23017087        | 18867827        | 18851302        | 18029971       | 18935711  |

Continua na próxima página

| Instância    | n     | Original   | RBFS-GL      | KP-band-GL | RCM-GL     | GPS            | MPG              | NSloan          | Sloan           | Sloan-MGPS      | ILS-Prof        | ILS-Band        | VNS-Band (60s) | FNC-HC    |
|--------------|-------|------------|--------------|------------|------------|----------------|------------------|-----------------|-----------------|-----------------|-----------------|-----------------|----------------|-----------|
| biplane-9    | 21701 | 3872319    | 4783609      | 5476632    | 4759152    | 4983313        | 2919945          | <b>1999469</b>  | 3543844         | 3846360         | 3807876         | 4760046         | 6079471        | 4865598   |
| trdheim      | 22098 | 5584881    | 3976955      | 7735967    | 4016541    | 3897309        | 2945649          | 2942978         | 2941833         | <b>2938977</b>  | 4160685         | 5584881         | 4400997        | 8341053   |
| pli          | 22695 | 19878760   | 23908272     | 24324036   | 23885641   | 23484231       | 41931736         | 44974025        | 45175787        | 52774094        | <b>19878758</b> | 19878760        | 25447453       | 23493999  |
| tuma1        | 22967 | 112240761  | 6159577      | 6370514    | 6180524    | 6159578        | <b>4768040</b>   | 7602840         | 5294783         | 5368698         | 6094555         | 6184758         | 13605200       | 6657837   |
| msc23052     | 23052 | 183660755  | 16003552     | 19858732   | 17238679   | 17122364       | <b>9407920</b>   | 10514056        | 9978563         | 11860876        | 16315402        | 16024261        | 48606620       | 19784417  |
| ca-CondMat   | 23133 | 179600115  | 48373060     | 127854302  | 48822904   | 88285070       | <b>25497315</b>  | 28855022        | 28882880        | 28718047        | 49107211        | 49906370        | 87126140       | 72516034  |
| stufe-10     | 24010 | 2637362    | 10526959     | 10854350   | 14652989   | 8758432        | 5895549          | 2739975         | 7721893         | 7046416         | <b>2631860</b>  | 10271750        | 33237516       | 10322089  |
| de2010       | 24115 | 68667881   | 4523656      | 7567918    | 4337924    | 6060462        | 2358230          | 2132531         | <b>2085691</b>  | <b>2085691</b>  | 3401026         | 4484974         | 8963151        | 7836052   |
| rajat09      | 24482 | 7791481    | 3619930      | 6405911    | 3644645    | 6419371        | 3664122          | <b>3603810</b>  | 3747098         | 3738385         | 3607597         | 3685440         | 9702327        | 6369958   |
| crystk03     | 24696 | 13649391   | 14362178     | 14752315   | 14388876   | 12891339       | <b>12809116</b>  | 12903278        | 12891198        | 12907305        | 157129973       | 13619473        | 13649391       | 27039861  |
| pcrystk03    | 24696 | 13649391   | 14362178     | 14807654   | 14244620   | 12891339       | <b>12776072</b>  | 12909318        | 12919266        | 12887172        | 13619746        | 13649391        | 27039861       | 14670351  |
| dtoc         | 24993 | 137432506  | <b>54976</b> | 74968      | 56976      | <b>54976</b>   | 59974            | <b>54976</b>    | <b>54976</b>    | <b>54976</b>    | 57477           | 59979           | 118548         | 124714    |
| hi2010       | 25016 | 57803984   | 6398259      | 11568615   | 6550539    | 5527330        | 1293294          | 1240080         | 1300780         | <b>1141727</b>  | 3423870         | 6786704         | 8010800        | 10052823  |
| ri2010       | 25181 | 73734773   | 6786547      | 13109805   | 6419840    | 7203852        | 2553410          | 3056657         | <b>2316444</b>  | 2514432         | 5350533         | 6379264         | 30301346       | 10430454  |
| bsstk37      | 25503 | 11302540   | 17900032     | 20422180   | 17144927   | 17488662       | <b>9722386</b>   | 10120387        | 9854744         | 10510696        | 11302540        | 15234185        | 30346232       | 20534855  |
| smt          | 25710 | 82049527   | 29915676     | 43230688   | 28910885   | 31445665       | 27966295         | 29147803        | 29627283        | 29394795        | 29871004        | <b>27823678</b> | 76828072       | 40256624  |
| brainpc2     | 27607 | 238146600  | 190369939    | 333113441  | 190388419  | 285577234      | <b>31264433</b>  | 82082330        | 117793830       | 157129995       | 94363671        | 188068371       | 155704223      | 333074338 |
| bratu3d      | 27792 | 176677007  | 16100457     | 17973854   | 16015093   | 16162765       | 15987303         | <b>15916711</b> | 15957140        | 15973227        | 16059096        | 16222501        | 57342370       | 18053720  |
| bloweya      | 30004 | 150115006  | 149997194    | 349874305  | 150035948  | 349655162      | 129992           | 108107          | 107424          | <b>106667</b>   | 71408299        | 75306057        | 166351454      | 145169324 |
| rajat10      | 30202 | 10371751   | 4987880      | 8700887    | 5121106    | 8824360        | 5058820          | 5049477         | <b>4960638</b>  | 5151033         | 5194599         | 4968287         | 15624076       | 8073976   |
| big_dual     | 30269 | 3636049    | 6333958      | 6581617    | 6353296    | 6253788        | 4738598          | <b>2702502</b>  | 5292962         | 4983758         | 3636048         | 6527011         | 13287621       | 6997386   |
| wathen100    | 30401 | 7645600    | 7024341      | 10940848   | 7192353    | 8548323        | 6780110          | <b>6106997</b>  | 6963250         | 6702309         | 7286537         | 7645600         | 8673715        | 10743429  |
| helm3d01     | 32226 | 488963313  | 34160090     | 54339700   | 34225395   | 42562876       | <b>24521168</b>  | 25481465        | 25698588        | 25785823        | 34289959        | 34111094        | 118117480      | 55022728  |
| lpl1         | 32460 | 105223934  | 125809199    | 245179557  | 136695362  | 185090406      | 18439546         | <b>14411689</b> | 17117497        | 14992145        | 81026551        | 84702893        | 192707846      | 195793744 |
| vt2010       | 32580 | 35599542   | 7272213      | 13579216   | 7202736    | 7858979        | 4057742          | 3531528         | <b>3478061</b>  | 3641450         | 5965974         | 7221386         | 10337513       | 13681535  |
| delaunay_n15 | 32768 | 349148752  | 15156878     | 18199527   | 14813339   | 13853980       | <b>8982222</b>   | 10295092        | 9123906         | 10270861        | 53467697        | 53424443        | 41367765       | 17362366  |
| se           | 32768 | 212504577  | 90232937     | 99692211   | 90232937   | 99693910       | 47030583         | 48763450        | 55481204        | 55481204        | <b>13084452</b> | 15017836        | 153101342      | 91155727  |
| SiO          | 33401 | 181551040  | 148324534    | 153244825  | 147361631  | 138949035      | <b>116353517</b> | 117294957       | 117494733       | 118159032       | 117155902       | 137174013       | 181208712      | 158083934 |
| ship_001     | 34920 | 258753262  | 16645276     | 24886938   | 16621642   | 20817547       | <b>15442109</b>  | 15487140        | 15577637        | 15493173        | 16645276        | 16645276        | 30454486       | 24865139  |
| aug3dcqp     | 35543 | 162465715  | 15086610     | 17026900   | 15078887   | 14828318       | 9018883          | 7883975         | <b>7882483</b>  | 7892406         | 14882774        | 15110840        | 35251821       | 17387313  |
| pdb1HYS      | 36417 | 85893224   | 54145909     | 60786479   | 50584140   | 56995785       | <b>31177486</b>  | 32502588        | 32445562        | 32473178        | 41537082        | 45915064        | 216845390      | 60043305  |
| shock-9      | 36476 | 6323171    | 7154536      | 7197412    | 7087634    | 6717138        | <b>3966198</b>   | 5709448         | 6362797         | 6295715         | 6323171         | 7132852         | 12287903       | 7411063   |
| pwt          | 36519 | 88246186   | 5545473      | 5829032    | 5589537    | 5640951        | 4905492          | <b>4815117</b>  | 4886819         | 5080596         | 5268420         | 5497046         | 11190182       | 5868383   |
| email-Enron  | 36692 | 456214647  | 100982343    | 426907301  | 101646059  | 234048649      | <b>24179120</b>  | 39959018        | 40459899        | 40459891        | 74657359        | 109584186       | 207528740      | 185826463 |
| pkustk05     | 37164 | 58406934   | 29965328     | 36878257   | 30075121   | 29886042       | 65469461         | 37940689        | <b>28140697</b> | 30776790        | 28463658        | 29504244        | 167392014      | 33095409  |
| finance256   | 37376 | 241399968  | 33050067     | 48151913   | 33643397   | 39514304       | 6702596          | 6397794         | 6367595         | <b>6339123</b>  | 17684817        | 14590040        | 52283288       | 48065478  |
| c-57         | 37833 | 515506328  | 257539158    | 420827222  | 232102639  | 270515381      | <b>18518265</b>  | 29199059        | 27614004        | 27845199        | 195786898       | 203417512       | 1886688380     | 211402464 |
| minsurfo     | 40806 | 8201805    | 5547297      | 5524057    | 5540296    | <b>5515105</b> | 5515105          | 5515256         | <b>5515105</b>  | <b>5515105</b>  | 5555344         | 6858455         | 8978203        | 6425937   |
| OPF_10000    | 43887 | 354965540  | 27365060     | 50016317   | 26724331   | 31704268       | 2951267          | 2489700         | 2305710         | <b>2256773</b>  | 15752205        | 13851926        | 24930457       | 42238137  |
| vanbody      | 47072 | 47896815   | 51621819     | 65949604   | 50289775   | 53418981       | <b>33726915</b>  | 37946583        | 36886525        | 36826959        | 36475199        | 51621819        | 63581173       | 66311968  |
| gridgena     | 48962 | 15499962   | 11910912     | 12843539   | 12602940   | 10951258       | <b>10948496</b>  | 12904999        | 11103051        | 11341348        | 11853147        | 12293843        | 67687201       | 12675930  |
| stokes128    | 49666 | 548545024  | 25474108     | 25686651   | 25240252   | 21731222       | <b>15418196</b>  | 24931898        | 15652574        | 15655001        | 24182424        | 25323813        | 25955418       | 22637782  |
| cvxbqp1      | 50000 | 819815962  | 43383098     | 52801368   | 43413273   | 53398841       | 36530379         | <b>26088467</b> | 34070451        | 33956062        | 37098633        | 37315872        | 47544537       | 51383072  |
| sparsine     | 50000 | 1137001002 | 633468627    | 760335126  | 642203430  | 731917744      | 443528350        | 463569659       | 453138982       | 487569082       | 542249831       | 567374228       | 6222070794     | 647366549 |
| copter2      | 55476 | 1084047722 | 75100445     | 92125642   | 76164385   | 74416281       | 47130636         | 50520263        | <b>46801114</b> | 48713374        | 59129239        | 58258871        | 151710188      | 87904797  |
| dixmaan1     | 60000 | 1200019999 | 370398       | 372122     | 380231     | 389964         | 359963           | <b>348458</b>   | 359959          | 359959          | 360309          | 384965          | 266780933      | 359969    |
| blockqp1     | 60012 | 600310011  | 1200600049   | 1800414059 | 1080466058 | 850485039      | <b>719986</b>    | 360438020       | 120626025       | 600250014       | 199643944       | 488137114       | 664360583      | 625453767 |
| Dubcova2     | 65025 | 1954927827 | 22087999     | 43480789   | 22075139   | 28621659       | 21509938         | 21819791        | 2197673         | <b>20843842</b> | 22090665        | 22091337        | 208686061      | 38352486  |
| qa8fk        | 66127 | 64862523   | 77288049     | 80536411   | 76335676   | 66852070       | 73392688         | 76250029        | 72010410        | 71736088        | <b>64031108</b> | 64862523        | 134937310      | 81148885  |
| gas_sensor   | 66917 | 69391231   | 80882583     | 82830931   | 80345254   | 73871884       | 69820999         | <b>64266712</b> | 69930149        | 69597992        | 67003478        | 80706514        | 225817042      | 78183097  |

Continua na próxima página

| Instância             | n      | Original   | RBFS-GL    | KP-band-GL      | RCM-GL          | GPS             | MPG              | NSloan           | Sloan           | Sloan-MGPS       | ILS-Prof         | ILS-Band   | VNS-Band (60s) | FNC-HC     |
|-----------------------|--------|------------|------------|-----------------|-----------------|-----------------|------------------|------------------|-----------------|------------------|------------------|------------|----------------|------------|
| H2O                   | 67024  | 726502155  | 354821308  | 357395061       | 353951785       | 372197160       | 339604846        | 323307033        | 325540803       | <b>314033571</b> | 370110471        | 350870845  | 710661807      | 369991375  |
| cfdl                  | 70656  | 103460652  | 125811085  | 132732812       | 131735999       | 105832955       | 58618821         | 73716728         | 61703713        | <b>56708430</b>  | 76303742         | 93949032   | 385734946      | 119735689  |
| finan512              | 74752  | 961176736  | 56337680   | 81042583        | 57385952        | 83402704        | 11818422         | <b>11192912</b>  | 11874206        | 11918518         | 21168294         | 52849855   | 84711343       | 83331859   |
| rail_79841            | 79841  | 551148483  | 16742832   | 19083148        | 16738720        | 14720371        | 13976217         | <b>9306620</b>   | 13828506        | 15029539         | 16742832         | 16742832   | 42504348       | 17498396   |
| cont-201              | 80595  | 1634949108 | 21503884   | 21659078        | 21389843        | 21397724        | 21393582         | 21386988         | <b>21334863</b> | 21336249         | 21370588         | 21393684   | 559298398      | 23570820   |
| apache1               | 80800  | 64705043   | 43398207   | 43486781        | 43573695        | 43551322        | 43314099         | 43381780         | 43347110        | 43346836         | 43580816         | 43577760   | 155874779      | 49146926   |
| shallow_water1        | 81920  | 35117727   | 22867384   | 22887719        | 22369881        | 21920594        | 22419016         | <b>21897736</b>  | 23687824        | 22685200         | 21902078         | 23095403   | 77658932       | 22176916   |
| shallow_water2        | 81920  | 35117727   | 22879469   | 22406173        | 21977576        | 21920594        | <b>21857395</b>  | 22241893         | 22597269        | 21885650         | 21918268         | 21918331   | 77658932       | 22425894   |
| thermal1              | 82654  | 175625317  | 12023625   | 12976354        | 12143186        | 12199245        | 10302343         | <b>10283016</b>  | 10577399        | 10737259         | 11805699         | 11943346   | 20092531       | 12841911   |
| consph                | 83334  | 245646007  | 297747273  | 293147332       | 293427167       | 241426337       | 191659193        | <b>189954349</b> | 192998906       | 214249681        | 236516801        | 286028826  | 669250510      | 252816707  |
| c-72                  | 84064  | 2255525662 | 1143992285 | 1843783696      | 1133335836      | 1549828571      | <b>100920670</b> | 112412268        | 111268803       | 112412268        | 784329682        | 703616837  | 1083658227     | 1354043890 |
| ncvxqp7               | 87500  | 2983313879 | 846714413  | 1064929086      | 847645802       | 1069372595      | <b>241450601</b> | 277141702        | 282172854       | 271782257        | 516245686        | 532122276  | 1335480147     | 950167150  |
| olesnik0              | 88263  | 1235233247 | 37560237   | 64217339        | 37508882        | 47955579        | <b>25431328</b>  | 25909625         | 25541709        | 25995854         | 35690261         | 36562841   | 280536455      | 64046204   |
| denormal              | 89400  | 53016878   | 35798796   | 35833959        | 35727844        | <b>35727370</b> | 35729783         | 35728421         | 35758436        | 35762065         | 35776941         | 53016878   | 1148161534     | 39589218   |
| s3dkq4m2              | 90449  | 54981444   | 67377576   | 67034249        | 66387617        | 54981444        | 56243536         | 53233151         | 56047345        | 56046942         | <b>42759294</b>  | 42759387   | 192855810      | 61092456   |
| s4dkt3m2              | 90449  | 54981444   | 42794753   | <b>42759294</b> | <b>42759294</b> | <b>42759294</b> | 42764966         | 42770013         | 42762890        | 42762777         | 54206152         | 54981444   | 110674149      | 44200140   |
| boyd1                 | 93279  | 1508966    | 1953655021 | 4013758116      | 920382333       | 1446006835      | <b>691870</b>    | 8214189          | 8214189         | 8545217          | 971620           | 1370824317 | 1148140770     | 1308753335 |
| tandem_dual           | 94069  | 487466755  | 129713225  | 142783578       | 128071557       | 119020989       | 80193344         | <b>65878258</b>  | 92247784        | 90967464         | 76105831         | 103249749  | 276139384      | 158340461  |
| pkustk13              | 94893  | 234310721  | 152841330  | 326903588       | 149854271       | 227776463       | <b>133169679</b> | 148014371        | 145700853       | 149082222        | 158526011        | 152973744  | 1151436298     | 315598816  |
| Si34H36               | 97569  | 1121290978 | 1016296049 | 1078787168      | 1024850608      | 989135758       | 768882514        | 795209187        | 793775648       | 794242316        | <b>756362088</b> | 759877008  | 1857924813     | 1108421633 |
| 2cubes_sphere         | 101492 | 483241271  | 273087405  | 352369397       | 271086176       | 276620528       | <b>169299238</b> | 193039264        | 188689009       | 201280732        | 243544022        | 237771123  | 1486492983     | 325061948  |
| thermomech_TC         | 102158 | 2667823445 | 17806392   | 18835939        | 17872907        | 18153270        | <b>15429699</b>  | 15529114         | 15985756        | 16043867         | 17014959         | 17908476   | 32633880       | 19838401   |
| thermomech_TK         | 102158 | 2667823445 | 17806392   | 18806540        | 17766378        | 18153270        | <b>15371737</b>  | 15480751         | 16020622        | 16160089         | 17015033         | 17823406   | 32633880       | 19351294   |
| filter3D              | 106437 | 260719523  | 92262938   | 191806705       | 94108423        | 133330442       | <b>85938851</b>  | 95934490         | 95449701        | 94371728         | 91493938         | 112694002  | 222196706      | 203318106  |
| 598a                  | 110971 | 3432278586 | 403268891  | 540921032       | 416146211       | 462726798       | 149675436        | 146513068        | 192660114       | 189109404        | <b>146220564</b> | 151162670  | 1420462485     | 533090046  |
| Ge87H76               | 112985 | 1438496399 | 1349544985 | 1446829334      | 1345431237      | 1423650695      | 1202067278       | 1246564128       | 1242964682      | 1243953205       | <b>983866414</b> | 1008572614 | 2343232906     | 1474559091 |
| Núm. melhores         |        |            | 3          | 2               | 4               | 6               | 48               | 33               | 18              | 22               | 17               | 6          | 2              | 0          |
| $\Sigma \rho_{\beta}$ |        |            | 6244.3311  | 12141.8189      | 4565.4111       | 6926.4435       | 23.4274          | 531.2092         | 205.0242        | 874.4056         | 1098.4834        | 3576.9875  | 5528.3829      | 4456.8798  |

Tabela A3 – Resultados de tempo de processamento, em segundos, das heurísticas testadas em 124 instâncias simétricas da *SuiteSparse Matrix Collection*

| Instância       | n     | RBFS-GL       | KP-band-GL | RCM-GL        | GPS      | MPG     | NSloan        | Sloan         | Sloan-MGPS | ILS-Prof | ILS-Band | VNS-Band (60s) | FNC-HC   |
|-----------------|-------|---------------|------------|---------------|----------|---------|---------------|---------------|------------|----------|----------|----------------|----------|
| utm5940         | 5940  | <b>0.0204</b> | 0.0246     | 0.0251        | 0.0499   | 0.0259  | 0.0208        | 0.0222        | 0.0216     | 0.7839   | 1.5019   | 60.1545        | 4.9966   |
| G67             | 10000 | 0.0197        | 0.0215     | 0.0210        | 0.0359   | 0.0113  | 0.0080        | <b>0.0078</b> | 0.0079     | 0.6978   | 0.7745   | 60.1399        | 4.8525   |
| crack           | 10240 | 0.0232        | 0.0265     | 0.0257        | 0.0391   | 0.0169  | <b>0.0140</b> | 0.0153        | 0.0148     | 0.8639   | 2.6714   | 60.1150        | 4.0353   |
| sit100          | 10262 | 0.0215        | 0.0245     | 0.0240        | 0.0374   | 0.0296  | <b>0.0114</b> | 0.0118        | 0.0119     | 1.0094   | 3.2048   | 60.1198        | 5.5160   |
| shuttle_eddy    | 10429 | 0.0240        | 0.0283     | 0.0281        | 0.1109   | 0.0196  | 0.0161        | <b>0.0137</b> | 0.0148     | 1.0666   | 3.5796   | 60.1173        | 5.3044   |
| vsp_p0291...    | 10498 | 0.0309        | 0.0378     | 0.0373        | 0.0834   | 0.0632  | 0.0318        | <b>0.0268</b> | 0.0308     | 1.9389   | 2.7782   | 60.1417        | 26.3748  |
| bundle1         | 10581 | <b>0.2025</b> | 0.2779     | 0.2697        | 144.0595 | 1.6442  | 0.4078        | 0.3586        | 0.3311     | 5.8809   | 15.7771  | 60.5051        | 10.2382  |
| wing_nodal      | 10937 | <b>0.0404</b> | 0.0516     | 0.0511        | 0.0940   | 0.0584  | 0.0444        | 0.0414        | 0.0408     | 2.6645   | 10.1698  | 60.1578        | 10.8779  |
| CurlCurl_0      | 11083 | 0.0255        | 0.0320     | 0.0316        | 0.3851   | 0.0594  | 0.0232        | 0.0219        | 0.0203     | 1.4567   | 5.8753   | 60.1670        | 8.2456   |
| fe_4elt2        | 11143 | 0.0237        | 0.0271     | 0.0265        | 0.0408   | 0.0174  | <b>0.0126</b> | 0.0163        | 0.0130     | 1.8651   | 2.7301   | 60.1168        | 4.9260   |
| linverse        | 11999 | 0.0239        | 0.0284     | 0.0283        | 0.0890   | 0.0141  | 0.0131        | <b>0.0126</b> | 0.0136     | 1.4001   | 1.2622   | 60.1915        | 1.5777   |
| ncvxqp1         | 12111 | 0.0269        | 0.0309     | 0.0297        | 0.0665   | 0.0420  | 0.0230        | <b>0.0202</b> | 0.0235     | 1.3764   | 3.7865   | 60.1723        | 13.3586  |
| vibrobox        | 12328 | <b>0.0784</b> | 0.1093     | 0.1116        | 0.1664   | 0.1064  | 0.0992        | 0.1101        | 0.0932     | 4.4822   | 18.6532  | 60.3561        | 22.9753  |
| stokes64        | 12546 | 0.0284        | 0.0351     | 0.0377        | 0.3520   | 0.0326  | 0.0232        | <b>0.0226</b> | 0.0237     | 1.4995   | 3.5147   | 60.2148        | 9.0789   |
| tuma2           | 12992 | 0.0217        | 0.0246     | 0.0239        | 0.0589   | 0.0198  | 0.0118        | <b>0.0113</b> | 0.0117     | 0.9852   | 1.5514   | 60.1505        | 4.7006   |
| cbuckle         | 13681 | <b>0.0903</b> | 0.1394     | 0.1402        | 3.6594   | 0.1992  | 0.1511        | 0.1510        | 0.1258     | 4.9434   | 16.2318  | 60.5684        | 21.4500  |
| cy16            | 13681 | 0.0973        | 0.1258     | 0.1296        | 2.7209   | 0.2843  | 0.0878        | <b>0.0863</b> | 0.0890     | 5.3659   | 20.6593  | 60.4706        | 22.1377  |
| crystm02        | 13965 | <b>0.0483</b> | 0.0651     | 0.0689        | 0.8670   | 0.0649  | 0.0589        | 0.0594        | 0.0608     | 4.9506   | 7.7914   | 60.3732        | 11.3017  |
| bcstk29         | 13992 | <b>0.0801</b> | 0.1266     | 0.1276        | 0.2244   | 0.1417  | 0.0993        | 0.0922        | 0.0843     | 5.2622   | 23.7488  | 60.3213        | 20.1360  |
| Pres_Poisson    | 14822 | <b>0.1130</b> | 0.1478     | 0.1370        | 3.0178   | 0.1897  | 0.1733        | 0.1716        | 0.1722     | 5.7590   | 14.7006  | 60.5736        | 21.1334  |
| opt1            | 15449 | 0.3629        | 0.5102     | 0.5144        | 6.4193   | 2.1747  | 0.3774        | 0.4040        | 0.3214     | 13.4644  | 51.6958  | 60.7475        | 33.7349  |
| hangGlider_4    | 15561 | <b>0.0320</b> | 0.0427     | 0.0418        | 4.4338   | 0.0484  | 0.0413        | 0.0383        | 0.0338     | 2.2984   | 2.5103   | 60.2804        | 28.6092  |
| Dubcoval        | 16129 | <b>0.0515</b> | 0.0693     | 0.0719        | 1.6463   | 0.0783  | 0.0608        | 0.0589        | 0.0614     | 2.5227   | 8.0211   | 60.3854        | 12.2898  |
| olafu           | 16146 | <b>0.1338</b> | 0.2082     | 0.2182        | 1.2029   | 0.2552  | 0.1461        | 0.1457        | 0.1459     | 7.0413   | 28.7863  | 60.7624        | 23.8253  |
| net50           | 16320 | <b>0.2970</b> | 0.3746     | 0.3015        | 79.1596  | 6.2310  | 0.3914        | 0.3884        | 0.3619     | 11.5350  | 15.6027  | 60.6561        | 58.0320  |
| fe_sphere       | 16386 | 0.0337        | 0.0372     | 0.0352        | 0.0352   | 0.0323  | 0.0240        | 0.0216        | 0.0192     | 1.3988   | 3.8117   | 60.2233        | 9.5381   |
| pds10           | 16558 | 0.0360        | 0.0445     | 0.0441        | 1.2859   | 0.0726  | 0.0466        | 0.0480        | 0.0328     | 2.1266   | 6.0075   | 60.2450        | 17.4180  |
| ex3stal         | 16782 | <b>0.1606</b> | 0.2199     | 0.2088        | 7.1821   | 0.3546  | 0.2361        | 0.2216        | 0.2168     | 9.2716   | 30.7721  | 60.4407        | 29.0748  |
| ramage02        | 16830 | <b>0.8117</b> | 1.0687     | 1.0210        | 43.4957  | 2.8562  | 0.8490        | 0.8441        | 0.8907     | 22.0227  | 92.5474  | 61.1035        | 53.4161  |
| cti             | 16840 | 0.0295        | 0.0355     | 0.0345        | 0.0918   | 0.0330  | 0.0231        | 0.0229        | 0.0228     | 27.7990  | 4.4075   | 60.2235        | 15.9811  |
| gyro            | 17361 | <b>0.1218</b> | 0.1989     | 0.2172        | 0.4724   | 0.4531  | 0.2082        | 0.2368        | 0.1823     | 7.2235   | 32.1519  | 60.8166        | 17.7507  |
| lowThrust_7     | 17378 | <b>0.0475</b> | 0.0666     | 0.0626        | 4.9505   | 0.0596  | 0.0499        | 0.0492        | 0.0487     | 3.3902   | 5.2264   | 60.3601        | 63.7133  |
| L-9             | 17983 | 0.0279        | 0.0323     | 0.0329        | 0.0633   | 0.0199  | <b>0.0189</b> | 0.0205        | 0.0197     | 9.2674   | 3.0527   | 60.2190        | 7.7141   |
| nd6k            | 18000 | <b>1.3233</b> | 2.3682     | 2.9556        | 34.9241  | 8.6857  | 10.7771       | 7.4482        | 4.0541     | 57.5573  | 145.8650 | 64.9234        | 97.8758  |
| crplat2         | 18010 | <b>0.1040</b> | 0.1527     | 0.1774        | 0.8724   | 0.2972  | 0.1502        | 0.1703        | 0.1776     | 7.6706   | 25.2494  | 60.5039        | 23.7441  |
| tandem_vtx      | 18454 | <b>0.0589</b> | 0.0788     | 0.0773        | 0.3434   | 0.1106  | 0.0871        | 0.0927        | 0.0870     | 3.6611   | 16.3735  | 60.3077        | 19.5866  |
| ford1           | 18728 | 0.0286        | 0.0342     | 0.0326        | 0.0675   | 0.0290  | 0.0231        | <b>0.0216</b> | 0.0225     | 3.0076   | 5.1736   | 60.2584        | 8.0863   |
| fxm4_6          | 18892 | <b>0.0824</b> | 0.1204     | 0.1283        | 1.8988   | 0.1938  | 0.1154        | 0.1066        | 0.1137     | 4.2881   | 15.7968  | 60.4148        | 76.7772  |
| whitaker3_dual  | 19190 | 0.0236        | 0.0310     | 0.0299        | 0.0621   | 0.0364  | 0.0174        | 0.0173        | 0.0171     | 1.2423   | 1.5541   | 60.2603        | 6.3052   |
| pattern1        | 19242 | 3.3709        | 4.8699     | <b>3.2106</b> | 16.1594  | 98.7080 | 5.0544        | 4.8634        | 5.1600     | 65.2552  | 154.2615 | 63.9100        | 127.6115 |
| bodyy6          | 19366 | 0.0317        | 0.0400     | 0.0387        | 0.0811   | 0.0324  | 0.0264        | <b>0.0263</b> | 0.0267     | 11.2739  | 5.4588   | 60.3149        | 10.9231  |
| raefsky4        | 19779 | <b>0.1946</b> | 0.2773     | 0.2402        | 1.8665   | 0.4289  | 0.3338        | 0.3591        | 0.3211     | 9.2601   | 43.4627  | 61.0371        | 37.9574  |
| Si5H12          | 19896 | <b>0.1556</b> | 0.2251     | 0.2040        | 6.5413   | 1.1769  | 0.4213        | 0.4680        | 0.2945     | 69.0908  | 27.1988  | 60.7486        | 38.9175  |
| LF10000         | 19998 | 0.0288        | 0.0349     | 0.0340        | 0.1050   | 0.0199  | <b>0.0189</b> | 0.0202        | 0.0194     | 3.1907   | 1.6224   | 60.3386        | 1.5600   |
| qpband          | 20000 | <b>0.0180</b> | 0.0209     | 0.0198        | 0.0841   | 0.0458  | 0.0449        | 0.0446        | 0.0496     | 2.0122   | 1.3500   | 60.2724        | 1.3726   |
| Trefethen_20000 | 20000 | <b>0.1068</b> | 0.1602     | 0.1485        | 2.5115   | 0.4724  | 0.1557        | 0.1385        | 0.1467     | 5.2290   | 8.9607   | 60.5694        | 44.7556  |
| tsyl201         | 20685 | 0.5614        | 0.7265     | <b>0.5560</b> | 4.4624   | 4.6914  | 0.7718        | 0.7708        | 0.7885     | 17.4836  | 78.8392  | 61.0897        | 43.6910  |
| tubel           | 21498 | <b>0.1045</b> | 0.1505     | 0.1500        | 0.3690   | 0.2617  | 0.1267        | 0.1251        | 0.1262     | 6.7676   | 33.0990  | 60.5626        | 30.2771  |

Continua na próxima página

| Instância    | n     | RBFS-GL       | KP-band-GL | RCM-GL | GPS       | MPG      | NSloan        | Sloan         | Sloan-MGPS | ILS-Prof | ILS-Band | VNS-Band (60s) | FNC-HC   |
|--------------|-------|---------------|------------|--------|-----------|----------|---------------|---------------|------------|----------|----------|----------------|----------|
| biplane-9    | 21701 | 0.0290        | 0.0405     | 0.0376 | 0.0796    | 0.0336   | 0.0256        | <b>0.0242</b> | 0.0269     | 1.5173   | 3.4416   | 60.3310        | 11.3483  |
| trdheim      | 22098 | <b>0.2339</b> | 0.3423     | 0.3315 | 3.1910    | 3.8866   | 0.3083        | 0.3087        | 0.3144     | 13.5374  | 66.5513  | 60.8582        | 26.9829  |
| pli          | 22695 | <b>0.1818</b> | 0.3180     | 0.3194 | 9.9220    | 0.6400   | 0.5557        | 0.5722        | 0.4287     | 10.4836  | 43.3930  | 60.7299        | 45.0692  |
| tuma1        | 22967 | 0.0303        | 0.0379     | 0.0368 | 0.1227    | 0.0444   | <b>0.0256</b> | 0.0277        | 0.0279     | 1.8838   | 2.8507   | 60.3892        | 8.9951   |
| msc23052     | 23052 | <b>0.2154</b> | 0.2896     | 0.2501 | 0.8388    | 0.4470   | 0.2767        | 0.3002        | 0.3049     | 9.9474   | 48.1750  | 61.1486        | 35.7301  |
| ca-CondMat   | 23133 | <b>0.0649</b> | 0.0830     | 0.0877 | 0.2095    | 0.2568   | 0.0925        | 0.0886        | 0.0910     | 2.9473   | 3.9745   | 60.4694        | 54.8247  |
| stufe-10     | 24010 | <b>0.0345</b> | 0.0457     | 0.0451 | 0.1208    | 0.0413   | 0.0391        | 0.0388        | 0.0371     | 1.6851   | 3.4943   | 60.3520        | 17.6323  |
| de2010       | 24115 | 0.0411        | 0.0521     | 0.0506 | 0.1091    | 0.0511   | 0.0427        | <b>0.0403</b> | 0.0424     | 2.0852   | 5.6077   | 60.3992        | 11.3101  |
| rajat09      | 24482 | 0.0308        | 0.0384     | 0.0373 | 0.1381    | 0.0703   | 0.0293        | <b>0.0275</b> | 0.0301     | 1.8814   | 3.6942   | 60.3827        | 8.4213   |
| crystk03     | 24696 | <b>0.2565</b> | 0.3427     | 0.3048 | 32.6441   | 0.4665   | 0.3634        | 0.3674        | 0.3664     | 14.1767  | 40.3517  | 61.5437        | 48.9546  |
| pcrystk03    | 24696 | <b>0.2683</b> | 0.3432     | 0.3192 | 33.1230   | 0.4605   | 0.3656        | 0.3585        | 0.3862     | 15.3787  | 40.3506  | 60.8525        | 49.1221  |
| dtoc         | 24993 | 0.0326        | 0.0382     | 0.0361 | 0.3440    | 0.0262   | <b>0.0233</b> | 0.0244        | 0.0256     | 2.0859   | 2.6214   | 60.3861        | 2.1429   |
| hi2010       | 25016 | <b>0.0362</b> | 0.0490     | 0.0494 | 0.1264    | 0.0483   | 0.0432        | 0.0443        | 0.0449     | 2.0851   | 5.0269   | 60.4236        | 11.0959  |
| ri2010       | 25181 | <b>0.0434</b> | 0.0571     | 0.0544 | 0.1394    | 0.0559   | 0.0479        | 0.0476        | 0.0551     | 2.3296   | 5.2779   | 60.4443        | 12.2165  |
| bsstk37      | 25503 | <b>0.1236</b> | 0.1962     | 0.2135 | 0.9037    | 0.4295   | 0.2665        | 0.2497        | 0.2193     | 9.4015   | 40.4283  | 61.0413        | 39.8729  |
| smt          | 25710 | <b>0.8219</b> | 1.1465     | 0.9142 | 27.7625   | 4.9263   | 1.6324        | 1.8237        | 1.2966     | 26.6585  | 112.2545 | 62.9113        | 62.6995  |
| brainpc2     | 27607 | 0.0429        | 0.0578     | 0.0538 | 0.2723    | 856.9546 | 0.0453        | <b>0.0386</b> | 0.0457     | 3.3042   | 7.7997   | 60.5857        | 9.6201   |
| bratu3d      | 27792 | <b>0.0531</b> | 0.0667     | 0.0643 | 0.3259    | 0.0893   | 0.0633        | 0.0623        | 0.0615     | 2.7052   | 9.2282   | 60.5413        | 9.8317   |
| bloweya      | 30004 | 0.0419        | 0.0564     | 0.0542 | 97.6164   | 0.0481   | 0.0416        | <b>0.0411</b> | 0.0419     | 3.4862   | 5.5847   | 60.6061        | 89.8033  |
| rajat10      | 30202 | 0.0379        | 0.0483     | 0.0477 | 0.1924    | 0.1032   | 0.0388        | <b>0.0366</b> | 0.0468     | 2.4836   | 5.8620   | 60.5628        | 11.0173  |
| big_dual     | 30269 | 0.0416        | 0.0515     | 0.0493 | 0.1648    | 0.0734   | 0.0438        | <b>0.0407</b> | 0.0420     | 2.0836   | 4.3170   | 60.5714        | 13.8554  |
| wathen100    | 30401 | <b>0.0733</b> | 0.1007     | 0.0993 | 7.3604    | 0.1040   | 0.0902        | 0.0878        | 0.0866     | 6.4390   | 14.8792  | 60.7747        | 28.1051  |
| helm3d01     | 32226 | <b>0.1217</b> | 0.1759     | 0.1604 | 0.3205    | 0.3705   | 0.1821        | 0.1701        | 0.1704     | 5.8511   | 36.2670  | 60.9876        | 30.9421  |
| lpl1         | 32460 | <b>0.0830</b> | 0.1136     | 0.1261 | 0.2427    | 0.1715   | 0.1501        | 0.1110        | 0.1227     | 5.9319   | 12.6146  | 60.7163        | 13.3117  |
| vt2010       | 32580 | <b>0.0540</b> | 0.0725     | 0.0702 | 0.1883    | 0.0773   | 0.0671        | 0.0721        | 0.0601     | 2.9360   | 6.6131   | 60.6691        | 15.8875  |
| delaunay_n15 | 32768 | <b>0.0540</b> | 0.0775     | 0.0752 | 0.1797    | 0.1037   | 0.0647        | 0.0814        | 0.0620     | 43.6645  | 23.3199  | 60.6520        | 23.0136  |
| se           | 32768 | <b>0.0419</b> | 0.0577     | 0.0549 | 0.1125    | 0.2449   | 0.0506        | 0.0480        | 0.0490     | 3.7467   | 11.8282  | 60.6660        | 45.8377  |
| SiO          | 33401 | <b>0.3136</b> | 0.4551     | 0.4397 | 0.9166    | 1.2108   | 0.4917        | 0.5431        | 0.4419     | 29.7288  | 38.6839  | 61.4902        | 65.5825  |
| ship_001     | 34920 | <b>1.0018</b> | 1.3611     | 1.3287 | 5.0949    | 6.7953   | 1.2393        | 1.2244        | 1.2296     | 31.9726  | 146.0020 | 63.6388        | 64.9832  |
| aug3dcqp     | 35543 | 0.0501        | 0.0642     | 0.0564 | 0.3588    | 0.1188   | 0.0462        | 0.0448        | 0.0447     | 2.6529   | 18.9485  | 60.7389        | 25.2291  |
| pdb1HYS      | 36417 | <b>0.9284</b> | 1.2559     | 1.2951 | 3.0772    | 3.7259   | 2.2441        | 2.4242        | 1.4377     | 32.4557  | 141.3415 | 63.3907        | 89.0965  |
| shock-9      | 36476 | <b>0.0528</b> | 0.0679     | 0.0658 | 0.2022    | 0.0670   | 0.0610        | 0.0581        | 0.0585     | 2.5476   | 6.3578   | 60.7078        | 17.9756  |
| pwt          | 36519 | 0.0706        | 0.0898     | 0.0903 | 0.9168    | 0.0849   | 0.0765        | 0.0709        | 0.0680     | 11.6202  | 11.1266  | 60.7383        | 14.1882  |
| email-Enron  | 36692 | <b>0.1170</b> | 0.1666     | 0.1560 | 0.5380    | 0.4028   | 0.2310        | 0.2302        | 0.2323     | 6.1678   | 8.5593   | 60.8394        | 101.2275 |
| pkustk05     | 37164 | 0.3905        | 0.4991     | 0.4978 | 2.2216    | 0.9133   | 0.3708        | 0.3596        | 0.3575     | 16.7664  | 76.4437  | 61.4446        | 56.5105  |
| finance256   | 37376 | <b>0.0665</b> | 0.0973     | 0.0911 | 0.2706    | 0.1199   | 0.0786        | 0.0775        | 0.0783     | 5.3062   | 25.0580  | 60.9027        | 29.5993  |
| c-57         | 37833 | <b>0.0920</b> | 0.1331     | 0.1170 | 255.4630  | 0.3769   | 0.2595        | 0.2765        | 0.1345     | 6.8902   | 15.2841  | 60.9984        | 125.9508 |
| minsurfo     | 40806 | <b>0.0605</b> | 0.0775     | 0.0776 | 0.3545    | 0.0649   | 0.0630        | 0.0611        | 0.0613     | 3.3241   | 6.3620   | 60.9069        | 13.7585  |
| OPF_10000    | 43887 | <b>0.1009</b> | 0.1375     | 0.1367 | 0.4389    | 0.1481   | 0.1322        | 0.1312        | 0.1370     | 6.4165   | 27.9251  | 61.2803        | 33.4187  |
| vanbody      | 47072 | <b>0.3092</b> | 0.4446     | 0.4686 | 1.5150    | 0.8467   | 0.4866        | 0.4722        | 0.4967     | 18.8992  | 96.2528  | 62.5369        | 80.7322  |
| gridgena     | 48962 | <b>0.0875</b> | 0.1246     | 0.1339 | 0.5532    | 0.2008   | 0.1096        | 0.1103        | 0.1056     | 6.3921   | 18.5109  | 61.4579        | 38.0681  |
| stokes128    | 49666 | <b>0.1083</b> | 0.1499     | 0.1458 | 4.4829    | 0.2441   | 0.1308        | 0.1330        | 0.1339     | 7.3889   | 18.5449  | 61.5425        | 48.7858  |
| cvxbqp1      | 50000 | <b>0.0979</b> | 0.1528     | 0.1420 | 0.5943    | 0.1923   | 0.1228        | 0.1234        | 0.1199     | 21.2503  | 35.8104  | 61.4902        | 49.0385  |
| sparsine     | 50000 | <b>0.6191</b> | 0.8463     | 0.7917 | 2461.0000 | 5.1335   | 0.7230        | 0.7260        | 0.7845     | 26.4900  | 44.3949  | 62.7557        | 230.3445 |
| copter2      | 55476 | <b>0.1863</b> | 0.2611     | 0.2458 | 0.5287    | 0.3433   | 0.2268        | 0.2247        | 0.2282     | 28.2856  | 75.3418  | 61.7989        | 65.9936  |
| dixmaan1     | 60000 | <b>0.0694</b> | 0.0992     | 0.0897 | 0.8820    | 0.0804   | 0.0729        | 0.0733        | 0.0739     | 9.6271   | 5.1639   | 61.9648        | 6.9738   |
| blockqp1     | 60012 | <b>0.1155</b> | 0.1778     | 0.1765 | 342.6390  | 291.3796 | 0.1612        | 0.1603        | 0.1540     | 9.1836   | 17.0559  | 62.0238        | 153.3410 |
| Dubcova2     | 65025 | <b>0.2320</b> | 0.3149     | 0.3076 | 15.9510   | 0.4142   | 0.3320        | 0.3270        | 0.3296     | 11.3746  | 39.7724  | 62.8072        | 76.5056  |
| qa8fk        | 66127 | <b>0.2250</b> | 0.3224     | 0.3524 | 54.7788   | 0.4573   | 0.4002        | 0.3989        | 0.4040     | 22.2827  | 76.1954  | 62.9155        | 97.0394  |
| gas_sensor   | 66917 | <b>0.2962</b> | 0.3714     | 0.3820 | 65.8371   | 0.4497   | 0.4139        | 0.4041        | 0.4432     | 14.0564  | 45.8698  | 63.0819        | 100.9180 |

Continua na próxima página

| Instância             | n      | RBFS-GL       | KP-band-GL | RCM-GL  | GPS        | MPG        | NSloan  | Sloan   | Sloan-MGPS | ILS-Prof   | ILS-Band   | VNS-Band (60s) | FNC-HC     |
|-----------------------|--------|---------------|------------|---------|------------|------------|---------|---------|------------|------------|------------|----------------|------------|
| H2O                   | 67024  | <b>0.5588</b> | 0.7833     | 0.7529  | 1.6975     | 8.6456     | 0.8017  | 0.9733  | 0.8506     | 165.1657   | 124.9082   | 63.6743        | 118.1910   |
| cfdl                  | 70656  | <b>0.3252</b> | 0.4327     | 0.3965  | 20.8932    | 0.5953     | 0.4918  | 0.4700  | 0.4859     | 27.9988    | 120.4260   | 63.2592        | 109.1950   |
| finan512              | 74752  | <b>0.1280</b> | 0.1827     | 0.1695  | 0.6424     | 0.2492     | 0.1751  | 0.1732  | 0.1743     | 14.2363    | 57.7832    | 62.9115        | 47.2335    |
| rail_79841            | 79841  | <b>0.1196</b> | 0.1701     | 0.1646  | 0.8112     | 0.2367     | 0.1610  | 0.1571  | 0.1521     | 8.6248     | 27.9550    | 63.1493        | 49.3325    |
| cont-201              | 80595  | <b>0.1251</b> | 0.1708     | 0.1530  | 1.1750     | 0.1827     | 0.1655  | 0.1642  | 0.1617     | 9.9304     | 27.8199    | 63.1833        | 69.8154    |
| apache1               | 80800  | <b>0.1484</b> | 0.1983     | 0.1831  | 1.1539     | 0.2100     | 0.1986  | 0.1918  | 0.1917     | 12.1842    | 25.0811    | 63.1678        | 89.5916    |
| shallow_water1        | 81920  | <b>0.1006</b> | 0.1387     | 0.1488  | 1.1055     | 0.3936     | 0.1659  | 0.1126  | 0.1296     | 24.0436    | 7.3355     | 63.0814        | 42.9358    |
| shallow_water2        | 81920  | <b>0.0979</b> | 0.1548     | 0.1371  | 1.0543     | 0.4103     | 0.1541  | 0.1531  | 0.1583     | 24.4512    | 7.5020     | 63.1049        | 43.2950    |
| thermal1              | 82654  | <b>0.1259</b> | 0.2001     | 0.1801  | 0.7096     | 0.3186     | 0.3065  | 0.3058  | 0.3157     | 10.5340    | 17.6245    | 63.2304        | 36.3578    |
| consph                | 83334  | <b>0.8017</b> | 1.2344     | 1.5265  | 720.1520   | 5.3969     | 4.0529  | 3.6473  | 3.1438     | 46.6827    | 128.3630   | 66.6537        | 84.0069    |
| c-72                  | 84064  | <b>0.1803</b> | 0.2703     | 0.2896  | 3.2181     | 0.5087     | 0.3021  | 0.2958  | 0.2966     | 14.7890    | 42.1797    | 64.3364        | 1041.6500  |
| ncvxqp7               | 87500  | <b>0.2190</b> | 0.3228     | 0.2973  | 3.0339     | 1.6935     | 0.3501  | 0.3845  | 0.3282     | 283.0120   | 73.7512    | 64.0954        | 222.0185   |
| olesnik0              | 88263  | <b>0.1657</b> | 0.2347     | 0.2256  | 0.7571     | 0.8585     | 0.2370  | 0.2336  | 0.2503     | 11.5624    | 34.8054    | 63.7518        | 77.4491    |
| denormal              | 89400  | <b>0.1835</b> | 0.2549     | 0.2602  | 1.8859     | 0.3048     | 0.2685  | 0.2696  | 0.2649     | 56.6576    | 21.7314    | 64.0649        | 99.2586    |
| s3dkq4m2              | 90449  | <b>0.5469</b> | 0.8267     | 0.7293  | 29.3711    | 1.4366     | 0.8039  | 0.7974  | 0.8022     | 44.0125    | 227.5900   | 66.4396        | 155.9845   |
| s4dkt3m2              | 90449  | <b>0.3807</b> | 0.5604     | 0.6144  | 2.4286     | 0.9398     | 0.5163  | 0.5131  | 0.5237     | 79.2760    | 156.5415   | 64.4956        | 141.6645   |
| boyd1                 | 93279  | <b>0.3950</b> | 0.7913     | 0.7332  | 17.4941    | 186.1944   | 0.6163  | 0.6034  | 0.6188     | 18.1355    | 60.1750    | 64.1620        | 294.4345   |
| tandem_dual           | 94069  | <b>0.1212</b> | 0.1742     | 0.1708  | 0.5005     | 0.3316     | 0.1697  | 0.1596  | 0.1736     | 93.9624    | 42.6671    | 64.0222        | 91.1535    |
| pkustk13              | 94893  | <b>1.3051</b> | 1.7670     | 1.4726  | 43.7136    | 7.3081     | 2.1965  | 2.5529  | 2.1678     | 51.7379    | 258.0230   | 65.9505        | 169.6185   |
| Si34H36               | 97569  | <b>1.3057</b> | 1.9341     | 1.6334  | 3.9644     | 9.4581     | 2.0055  | 2.2277  | 2.1295     | 221.6830   | 319.2550   | 67.9498        | 194.9035   |
| 2cubes_sphere         | 101492 | <b>0.3721</b> | 0.5511     | 0.5206  | 1.8497     | 4.6887     | 0.5461  | 0.5019  | 0.6149     | 33.7637    | 107.1672   | 65.4856        | 134.2855   |
| thermomech_TC         | 102158 | <b>0.2130</b> | 0.3258     | 0.3015  | 1.2098     | 0.3508     | 0.2920  | 0.3132  | 0.3124     | 101.8651   | 38.9185    | 65.0597        | 69.2077    |
| thermomech_TK         | 102158 | <b>0.2139</b> | 0.3238     | 0.3026  | 1.1824     | 0.3509     | 0.3006  | 0.3052  | 0.2828     | 98.6794    | 36.3973    | 65.0707        | 64.2321    |
| filter3D              | 106437 | <b>0.5214</b> | 0.7573     | 0.7638  | 1.9218     | 2.9743     | 1.0871  | 1.0147  | 0.9270     | 28.1781    | 179.6825   | 66.8755        | 108.5985   |
| 598a                  | 110971 | <b>0.5702</b> | 0.8399     | 0.8145  | 1.9919     | 1.3909     | 0.9120  | 0.9403  | 0.8818     | 69.9385    | 239.2945   | 66.6182        | 191.6230   |
| Ge87H76               | 112985 | <b>2.0776</b> | 3.0696     | 2.5715  | 8.7368     | 15.8565    | 3.5208  | 3.6725  | 3.1778     | 176.3295   | 383.1670   | 71.2212        | 249.2590   |
| Núm. melhores         |        | 89            | 0          | 2       | 0          | 0          | 7       | 17      | 9          | 0          | 0          | 0              | 0          |
| $\Sigma \rho_{\beta}$ |        | 12.1685       | 60.2935    | 55.1058 | 16230.9755 | 25465.2125 | 57.6520 | 54.3151 | 44.1020    | 15458.2754 | 24114.7163 | 144707.0067    | 52696.7912 |

Tabela A4 – Resultados de redução de largura de banda das heurísticas testadas em 48 instâncias assimétricas da *SuiteSparse Matrix Collection*

| Instância           | n      | Original   | RBFS-GL     | KP-band-GL | RCM-GL      | GPS         | MPG    | NSloan | Sloan | Sloan-MGPS | ILS-Prof    | ILS-Band     | VNS-Band<br>(50s) | FNC-HC       |
|---------------------|--------|------------|-------------|------------|-------------|-------------|--------|--------|-------|------------|-------------|--------------|-------------------|--------------|
| ted_A               | 10605  | 4291       | 222         | 196        | 210         | 174         | 259    | 248    | 242   | 243        | 197         | <b>137</b>   | 250               | 148          |
| lhr10               | 10672  | 8419       | 10380       | 10276      | 10388       | 4797        | 10252  | 10110  | 9977  | 10010      | 8421        | 8418         | 3502              | <b>2897</b>  |
| igbt3               | 10938  | 10905      | 358         | 323        | 339         | 368         | 1635   | 2281   | 1755  | 1755       | 635         | <b>302</b>   | 1697              | 307          |
| k3plates            | 11107  | 265        | 286         | 282        | 280         | 334         | 1530   | 1402   | 1590  | 1485       | 447         | 265          | 622               | <b>260</b>   |
| bips98_1450         | 11305  | 8914       | 10510       | 10768      | 10216       | 1745        | 10153  | 10110  | 10177 | 10177      | 11000       | 7795         | 3101              | <b>1714</b>  |
| cage10              | 11397  | 5348       | 2504        | 2474       | 2589        | 2578        | 9194   | 7577   | 7795  | 7467       | 2677        | 2499         | 3680              | <b>2015</b>  |
| inlet               | 11730  | <b>142</b> | 497         | 296        | 548         | 149         | 268    | 209    | 259   | 265        | <b>142</b>  | <b>142</b>   | 327               | 148          |
| pesa                | 11738  | 11327      | 351         | 363        | 352         | 350         | 5901   | 5927   | 6776  | 6380       | 277         | <b>265</b>   | 1232              | 267          |
| big                 | 13209  | 12916      | 419         | 346        | 403         | 300         | 3270   | 2182   | 2438  | 1025       | <b>296</b>  | 328          | 3105              | <b>296</b>   |
| mimo46x46_system    | 13250  | 10782      | 12966       | 12336      | 12292       | <b>628</b>  | 7920   | 6114   | 4316  | 4316       | 12969       | 9476         | 737               | <b>663</b>   |
| xingo_afonso_itaipu | 13250  | 10782      | 12195       | 13017      | 12903       | <b>628</b>  | 7920   | 6941   | 4316  | 4316       | 12969       | 9513         | 737               | 663          |
| airfoi1_2d          | 14214  | 3002       | 771         | 707        | 1027        | 440         | 2248   | 3115   | 1878  | 2195       | 3002        | 723          | 4601              | <b>434</b>   |
| barth5              | 15606  | 15080      | 13434       | 14178      | 14557       | 374         | 4043   | 3827   | 1185  | 2145       | 10088       | 10569        | 4933              | <b>339</b>   |
| powersim            | 15838  | 11304      | 15727       | 15740      | 15729       | 1059        | 10112  | 7548   | 7129  | 7936       | 11363       | 2813         | 784               | <b>661</b>   |
| memplus             | 17758  | 17751      | 9877        | 10330      | 10090       | 10757       | 16770  | 17611  | 17611 | 17611      | 8875        | 5873         | <b>4806</b>       | 5071         |
| FEM_3D_thermal1     | 17880  | 13787      | 655         | 692        | 692         | 565         | 1837   | 702    | 1854  | 1796       | 568         | 562          | 1059              | <b>486</b>   |
| chipcool1           | 20082  | 1395       | 1255        | 1254       | 1211        | 1201        | 3448   | 4068   | 3628  | 3628       | <b>916</b>  | 1090         | 5032              | 1036         |
| cz20468             | 20468  | 514        | 67          | 67         | 69          | 42          | 130    | 1488   | 140   | 124        | 66          | 68           | 103               | <b>28</b>    |
| bayer04             | 20545  | 20544      | 20379       | 20454      | 20446       | 7852        | 19534  | 17005  | 17120 | 16742      | 18988       | 18904        | 6574              | <b>4475</b>  |
| xingo3012           | 20944  | 16122      | 19946       | 19695      | 19634       | 1491        | 18144  | 14177  | 14301 | 14301      | 17031       | 14845        | 1423              | <b>1137</b>  |
| std1_Jac2           | 21982  | 18189      | 21864       | 21963      | 21946       | 13058       | 20548  | 20737  | 20885 | 20739      | 18189       | 18189        | 7891              | <b>6944</b>  |
| af23560             | 23560  | <b>304</b> | 345         | 527        | 420         | 305         | 1274   | 599    | 546   | 547        | 476         | <b>304</b>   | 2391              | 308          |
| hvdc1               | 24842  | 24762      | 8206        | 8158       | 8535        | 2107        | 23468  | 13321  | 13321 | 13321      | 5169        | 4227         | 3096              | <b>1780</b>  |
| epb2                | 25228  | 2035       | 4794        | 4805       | 4780        | 1044        | 11067  | 7046   | 7046  | 7433       | 2035        | 941          | 6493              | <b>712</b>   |
| wang3               | 26064  | 900        | <b>675</b>  | 678        | <b>675</b>  | <b>675</b>  | 1456   | 1297   | 1284  | 1300       | <b>675</b>  | <b>675</b>   | 7455              | 693          |
| 2D_27628_bjtcai     | 27628  | 27304      | 450         | 506        | 453         | 431         | 3603   | 4455   | 3278  | 4044       | 747         | 449          | 4605              | <b>417</b>   |
| mixtank_new         | 29957  | 28080      | 4061        | 3880       | 3919        | 3780        | 20182  | 21315  | 20458 | 21042      | <b>2504</b> | 3137         | 8581              | 3090         |
| poli4               | 33833  | 33784      | 33449       | 33497      | 33408       | 8865        | 33447  | 32034  | 32034 | 32034      | 33449       | 33292        | 10984             | <b>6296</b>  |
| Zhao1               | 33861  | 32907      | 983         | 964        | 983         | <b>490</b>  | 1694   | 12428  | 1377  | 1267       | 496         | 499          | 6504              | 504          |
| onetone1            | 36057  | 8505       | 27408       | 29110      | 29240       | 8127        | 15866  | 12053  | 11510 | 11739      | 8505        | 8505         | 11719             | <b>6493</b>  |
| chem_master1        | 40401  | <b>201</b> | <b>201</b>  | 202        | <b>201</b>  | <b>201</b>  | 400    | 301    | 301   | 301        | <b>201</b>  | <b>201</b>   | 6243              | 204          |
| xenon1              | 48600  | 10001      | 2023        | 1982       | 2042        | 2404        | 9967   | 10894  | 11560 | 9725       | <b>1838</b> | 1845         | 13938             | 1935         |
| ec132               | 51993  | 51840      | 22102       | 40926      | 40885       | 1824        | 10513  | 8220   | 5585  | 5498       | 16828       | 16223        | 14089             | <b>1805</b>  |
| bcircuit            | 68902  | 67086      | 1509        | 1242       | <b>1218</b> | 1534        | 5577   | 6988   | 5250  | 4999       | 1402        | 1285         | 1880              | 1337         |
| shyy161             | 76480  | <b>160</b> | 62941       | 52577      | 26094       | 239         | 8440   | 6444   | 473   | 505        | <b>160</b>  | <b>160</b>   | 15600             | 238          |
| circuit_4           | 80209  | 80193      | 73389       | 73393      | 73384       | 31644       | 67003  | 66970  | 66970 | 66970      | 55553       | 73089        | <b>22035</b>      | 36567        |
| epb3                | 84617  | <b>284</b> | 567         | 569        | 565         | 286         | 567    | 1594   | 771   | 756        | <b>284</b>  | <b>284</b>   | 10718             | 289          |
| poisson3Db          | 85623  | 85563      | 9552        | 9996       | 9758        | 10607       | 19269  | 22260  | 22219 | 21678      | 8230        | <b>7631</b>  | 26576             | 8142         |
| rajat28             | 87190  | 87090      | 55208       | 56998      | 53806       | 29029       | 86285  | 78984  | 77561 | 77561      | 87090       | 39968        | <b>28178</b>      | 31820        |
| LeGresley_87936     | 87936  | 81497      | 85746       | 76487      | 76323       | 3372        | 71989  | 38141  | 15333 | 15333      | 84505       | 23700        | 17559             | <b>2665</b>  |
| rajat18             | 94294  | 94189      | 84605       | 84435      | 84586       | 32857       | 89042  | 76134  | 76134 | 76134      | 94189       | 51655        | <b>29974</b>      | 34315        |
| ASIC_100ks          | 99190  | 98439      | 21825       | 20436      | 21641       | 17160       | 92072  | 90056  | 86531 | 90120      | 98439       | <b>14400</b> | 31187             | 17540        |
| ASIC_100k           | 99340  | 99106      | 99178       | 99183      | 99183       | 98976       | 99334  | 99338  | 99338 | 99338      | 98955       | 74205        | <b>49626</b>      | <b>49626</b> |
| matrix_9            | 103430 | 103429     | 5672        | 6003       | 6180        | 4050        | 25695  | 29166  | 28933 | 33140      | 8328        | 4986         | 29388             | <b>2621</b>  |
| hcircuit            | 105676 | 105663     | 24274       | 23849      | 6422        | 5262        | 69561  | 27724  | 27829 | 26575      | 19402       | 47490        | 93310             | <b>4047</b>  |
| lung2               | 109460 | 107141     | 62801       | 62800      | 62800       | <b>6683</b> | 29949  | 29949  | 29949 | 29949      | 107141      | 15957        | 10331             | 11256        |
| rajat23             | 110355 | 110255     | 106875      | 107402     | 106758      | 18981       | 103844 | 75051  | 75051 | 80237      | 110255      | 104621       | 28541             | <b>14427</b> |
| Baumann             | 112211 | 1111       | <b>1086</b> | 1089       | <b>1086</b> | <b>1086</b> | 1939   | 1958   | 1963  | 1987       | <b>1086</b> | <b>1086</b>  | 22624             | 1117         |

|                     |  |          |          |          |         |          |          |          |         |          |          |          |        |    |
|---------------------|--|----------|----------|----------|---------|----------|----------|----------|---------|----------|----------|----------|--------|----|
| Núm. melhores       |  | 3        | 0        | 4        | 7       | 0        | 0        | 0        | 0       | 0        | 10       | 12       | 5      | 24 |
| $\sum \rho_{\beta}$ |  | 620.4706 | 563.9001 | 397.0526 | 13.4804 | 333.9680 | 343.3967 | 204.9323 | 55.4512 | 210.5791 | 132.8431 | 348.1568 | 3.2285 |    |

Tabela A5 – Resultados de redução de *profile* das heurísticas testadas em 48 instâncias assimétricas da *SuiteSparse Matrix Collection*

| Instância           | n      | Original   | RBFS-GL         | KP-band-GL | RCM-GL          | GPS              | MPG              | NSloan           | Sloan           | Sloan-MGPS      | ILS-Prof          | ILS-Band          | VNS-Band (50s) | FNC-HC     |
|---------------------|--------|------------|-----------------|------------|-----------------|------------------|------------------|------------------|-----------------|-----------------|-------------------|-------------------|----------------|------------|
| ted_A               | 10605  | 24944805   | 1528808         | 1517817    | 1355083         | 1435767          | 1306075          | 1403260          | 1398577         | 1383835         | 1189439           | <b>1130996</b>    | 1555911        | 1488245    |
| lhr10               | 10672  | 24671734   | 49868537        | 44679263   | 49250105        | 40201867         | <b>30861570</b>  | 31432851         | 33532595        | 35159484        | 42484086          | 42484088          | 39547603       | 32976002   |
| igbt3               | 10938  | 45856042   | 3319771         | 3147705    | 3179929         | 3148234          | 2107045          | <b>2035165</b>   | 2104198         | 2104198         | 2432910           | 3112235           | 11208316       | 3257278    |
| k3plates            | 11107  | 1746869    | 3692962         | 3639110    | 3646661         | 3633608          | 2829739          | <b>2715129</b>   | 2906148         | 2866159         | 3169463           | 3621254           | 5353882        | 3884810    |
| bips98_1450         | 11305  | 7771187    | 6647726         | 7121744    | 6647073         | 12610989         | 2748965          | <b>228033</b>    | 2261779         | 2261779         | 6262374           | 11685050          | 17147340       | 15607085   |
| cage10              | 11397  | 14938283   | 33050985        | 33127736   | 33956221        | 30489549         | 24597247         | 24806429         | 24781826        | <b>23775722</b> | 27087491          | 32467340          | 45735925       | 30799133   |
| inlet               | 11730  | 1004256    | 2088798         | 1518540    | 1985451         | 1557156          | 1734457          | 1562720          | 1747614         | 1801291         | <b>1191711</b>    | <b>1191711</b>    | 2332766        | 1983400    |
| pesa                | 11738  | 37892459   | 3213795         | 3229664    | 3234609         | 3205746          | 1810366          | <b>1676937</b>   | 2761140         | 2912575         | 1837729           | 1854034           | 5509557        | 2929516    |
| big                 | 13209  | 53960367   | 5996470         | 5303617    | 5845010         | 4948747          | <b>3022694</b>   | 3066284          | 3604116         | 4174837         | 3210776           | 4796084           | 14614038       | 5537866    |
| mimo46x46_system    | 13250  | 9318367    | 7371023         | 7292013    | 7322328         | 6140503          | 1228566          | 1150516          | <b>1095833</b>  | <b>1095833</b>  | 7590144           | 16363157          | 7227523        | 6857132    |
| xingo_afonso_itaipu | 13250  | 9318367    | 7541287         | 7327380    | 7369175         | 6140507          | 1228566          | <b>1080006</b>   | 1095833         | 1095833         | 7686005           | 16358063          | 7227523        | 7257614    |
| airfoil_2d          | 14214  | 2435598    | 10624103        | 10038696   | 13807930        | 7761926          | 6526546          | 6533599          | 7255947         | 8869502         | <b>4715379</b>    | 9537650           | 63771664       | 8623098    |
| barth5              | 15606  | 4058103    | 7710787         | 7064130    | 6159734         | 4548388          | 1976421          | <b>1891059</b>   | 3754938         | 3290718         | 6463092           | 7251077           | 13143202       | 5220360    |
| powersim            | 15838  | 26047268   | 34610156        | 36280422   | 34609254        | 9445591          | <b>1014022</b>   | 1371034          | 1156393         | 1139895         | 2211388           | 4858848           | 9350462        | 9231595    |
| memplur             | 17758  | 141187718  | 142426669       | 141221920  | 138424536       | 139469750        | <b>42061860</b>  | 70680741         | 70680741        | 70680741        | 64889840          | 56829375          | 77810519       | 156268014  |
| FEM_3D_thermal1     | 17880  | 11445306   | 9840487         | 9845111    | 9858410         | 9079106          | <b>8756714</b>   | 9024245          | 9017610         | 9017610         | 9164330           | 9164330           | 73764727       | 9591827    |
| chipcool1           | 20082  | 11902605   | 29963908        | 29876884   | 29414269        | 29261046         | 22864308         | 22735731         | 24219334        | 24219334        | <b>22193682</b>   | 22329111          | 98567746       | 30843448   |
| cz20468             | 20468  | 1196152    | 623337          | 619057     | 623927          | 591933           | 589652           | <b>578891</b>    | 608682          | 601296          | 622087            | 622609            | 675316         | 589564     |
| bayer04             | 20545  | 131403879  | 154407064       | 158522700  | 159086875       | 114773251        | <b>67957306</b>  | 76429544         | 70952877        | 78566851        | 157560556         | 1344138106        | 134414930      | 90967498   |
| xingo3012           | 20944  | 49244571   | 23016071        | 23735454   | 23692347        | 17855055         | 3778828          | 3651393          | <b>2838575</b>  | <b>2838575</b>  | 23750806          | 34480874          | 23277634       | 18518150   |
| std1_Jac2           | 21982  | 86101829   | 196574357       | 199204516  | 183768391       | 229471894        | 137571489        | 150079294        | 150218370       | 146702413       | <b>103277299</b>  | <b>103277299</b>  | 184651020      | 151943849  |
| af23560             | 23560  | 7094330    | 13558471        | 13253631   | 13228122        | 13172231         | 13175515         | <b>13157726</b>  | 13178555        | 13177229        | 13165583          | 14188640          | 66052094       | 13625365   |
| hvdcl               | 24842  | 68348635   | 31740367        | 31875404   | 32081783        | 42293356         | 8853881          | 5556872          | 5556872         | 5556872         | <b>5401928</b>    | 22013531          | 56337346       | 54881747   |
| epb2                | 25228  | 6803892    | 16610197        | 16625146   | 16548905        | 18838613         | 14137581         | 15061057         | 15061057        | 15406264        | <b>10189084</b>   | 14109198          | 125508489      | 19373552   |
| wang3               | 26064  | 22644785   | 25614052        | 25656153   | 25545046        | <b>25525904</b>  | 25607191         | 25587747         | 25624516        | 25653880        | 25643285          | 25539320          | 74469889       | 26696005   |
| 2D_27628_bjtcai     | 27628  | 7141587    | 13009922        | 13192763   | 13006315        | 13078837         | 11215293         | <b>10693847</b>  | 11235199        | 11145683        | 11838229          | 13009930          | 48385803       | 13763460   |
| mixtank_new         | 29957  | 397143539  | 114060711       | 113812401  | 111981632       | 99529791         | 115762197        | 124459346        | 122874657       | 127492867       | <b>83298641</b>   | 100108645         | 291363267      | 190903413  |
| poli4               | 33833  | 469894056  | 38886906        | 38425096   | 38409912        | 48803968         | <b>8329690</b>   | 12432053         | 12432053        | 12432053        | 38882784          | 149426249         | 39248655       | 40620878   |
| Zhao1               | 33861  | 365490595  | 26677513        | 26268424   | 26677227        | 26851046         | 26896455         | <b>26210270</b>  | 27113551        | 27357229        | 26546584          | 26655464          | 156491593      | 27932910   |
| onetone1            | 36057  | 36601159   | 221224462       | 231505295  | 238606370       | 229978401        | 92944014         | 105156418        | 105813122       | 106206248       | <b>45565543</b>   | <b>45565543</b>   | 391362869      | 242653061  |
| chem_master1        | 40401  | 8080400    | <b>10867400</b> | 10868781   | <b>10867400</b> | <b>10867400</b>  | 10867599         | 10867599         | 10867599        | 10867599        | <b>10867400</b>   | <b>10867400</b>   | 19542137       | 11111333   |
| xenon1              | 48600  | 77047005   | 127380864       | 118729096  | 126539009       | 114537877        | 104057144        | 103307532        | 106569032       | 109614924       | <b>99730318</b>   | 100850287         | 682038535      | 122560969  |
| ec132               | 51993  | 346202475  | 172549505       | 282993045  | 271229546       | 97867732         | <b>72521717</b>  | 73116157         | 80364666        | 80153075        | 136709643         | 145296696         | 456870403      | 98721170   |
| bcircuit            | 68902  | 488289014  | 109991667       | 100112592  | 98793138        | 82568942         | 14895200         | <b>11737585</b>  | 18766290        | 18991448        | 24502079          | 38374638          | 137145318      | 112231148  |
| shyy161             | 76480  | 12186076   | 30693729        | 27731994   | 21125003        | 22943037         | 20269935         | 20814949         | 23291036        | 23605144        | <b>20261590</b>   | 20276153          | 24400381       | 24493665   |
| circuit_4           | 80209  | 26212339   | 2219447077      | 2149859833 | 2220277339      | 1541984697       | <b>560700434</b> | 1334360951       | 1334360951      | 1334360951      | 1089469393        | 1574188126        | 1301950947     | 1783592001 |
| epb3                | 84617  | 23890785   | 40015343        | 40176085   | 39907530        | 40168868         | 39867166         | 35644108         | 41262018        | 41256981        | <b>12144048</b>   | <b>12144048</b>   | 377842643      | 42176582   |
| poisson3Db          | 85623  | 3419311554 | 860077724       | 870486522  | 875298904       | 759243839        | 615465483        | 755003391        | 763589678       | 759730677       | <b>487291877</b>  | 540307969         | 2406766149     | 831041679  |
| rajat28             | 87190  | 881414140  | 2146604657      | 1993518359 | 2159275316      | 2145841912       | 2297728734       | 1423222062       | 1419195604      | 1419195593      | 1177017682        | <b>1124468694</b> | 1750334082     | 2169921375 |
| LeGresley_87936     | 87936  | 1954349960 | 243503216       | 254344316  | 255293477       | 203418157        | 78215102         | 55675362         | <b>44588507</b> | 44672672        | 239256826         | 501587170         | 307748423      | 246291777  |
| rajat18             | 94294  | 943395788  | 1349339573      | 1414326152 | 1399018164      | 2094132431       | 2340492674       | 1690820378       | 1690820378      | 1690820376      | <b>1188606664</b> | 1591248983        | 1641541540     | 2015819035 |
| ASIC_100ks          | 99190  | 152743554  | 1693918521      | 1634166094 | 1650345324      | 1743037215       | 663690906        | 819317369        | 821806957       | 817512499       | <b>478531931</b>  | 844532660         | 3032996878     | 1421001348 |
| ASIC_100k           | 99340  | 2951888685 | 2416576554      | 3572966909 | 4038637644      | 1601840386       | 740413323        | 826623463        | 826623463       | 826623463       | <b>185713467</b>  | 3156005982        | 553107049      | 3788629841 |
| matrix_9            | 103430 | 396507913  | 495045266       | 570250880  | 538661162       | 412585651        | 405710521        | <b>388448362</b> | 405460241       | 398668319       | 420931439         | 477076916         | 3243281996     | 413543364  |
| hcircuit            | 105676 | 4367574045 | 479150501       | 459115714  | 428340760       | 333276878        | 137567939        | 159265331        | 148191062       | 142449943       | <b>137193365</b>  | 232831437         | 440788671      | 372202202  |
| lung2               | 109460 | 43425018   | 4025076344      | 4025082257 | 4025018634      | 685916146        | <b>2954327</b>   | 3813893          | 3609697         | 3609697         | 67863490          | 1922795188        | 1273519979     | 1341141513 |
| rajat23             | 110355 | 359789933  | 1818631424      | 1888240580 | 1687856660      | 1737821623       | <b>467214519</b> | 622237608        | 622237608       | 638117231       | 563264261         | 820141814         | 1314071584     | 1298658491 |
| Baumann             | 112211 | 123444210  | 166552864       | 166678382  | 166744924       | <b>165940324</b> | 166235680        | 166154593        | 166157481       | 166160875       | 166311904         | 166420474         | 707520725      | 178300223  |
| Núm. melhores       |        |            | 1               | 0          | 1               | 3                | 10               | 13               | 3               | 0               | 17                | 7                 | 0              | 0          |
| $\Sigma \rho\beta$  |        |            | 1485.7874       | 1492.5542  | 1494.0318       | 317.0497         | 14.0587          | 15.6093          | 18.5199         | 19.0931         | 61.2993           | 760.1803          | 638.9250       | 557.6532   |

Tabela A6 – Resultados de tempo de processamento, em segundos, das heurísticas testadas em 48 instâncias assimétricas da *Suite Sparse Matrix Collection*

| Instância           | n      | RBFS-GL       | KP-band-GL | RCM-GL | GPS      | MPG      | NSloan        | Sloan         | Sloan-MGPS     | ILS-Prof | ILS-Band | VNS-Band (50s) | FNC-HC   |
|---------------------|--------|---------------|------------|--------|----------|----------|---------------|---------------|----------------|----------|----------|----------------|----------|
| ted_A               | 10605  | <b>0.1145</b> | 0.1402     | 0.1365 | 1.3814   | 0.4601   | 0.1731        | 0.1686        | 0.1851         | 4.1907   | 16.4430  | 50.6025        | 11.769   |
| lhr10               | 10672  | <b>0.0477</b> | 0.0637     | 0.0645 | 1.0707   | 0.4433   | 0.3081        | 0.2584        | 0.2215         | 1.6058   | 3.2763   | 50.3581        | 34.80285 |
| igbt3               | 10938  | <b>0.0571</b> | 0.0750     | 0.0730 | 0.3988   | 0.0868   | 0.0718        | 0.0730        | 0.0702         | 6.0294   | 8.5020   | 50.3130        | 14.4351  |
| k3plates            | 11107  | <b>0.0560</b> | 0.0768     | 0.0814 | 0.2545   | 0.1308   | 0.1029        | 0.1023        | 0.0977         | 6.6989   | 9.5229   | 50.5093        | 16.6384  |
| bips98_1450         | 11305  | 0.02046364    | 0.0222     | 0.0215 | 0.0479   | 0.0213   | <b>0.0177</b> | 0.0179        | 0.0204         | 1.0887   | 2.0482   | 50.2394        | 4.322495 |
| cage10              | 11397  | <b>0.0390</b> | 0.0490     | 0.0484 | 0.1458   | 0.1055   | 0.0515        | 0.0520        | 0.0524         | 3.0639   | 4.3617   | 50.2592        | 21.42845 |
| inlet               | 11730  | <b>0.0521</b> | 0.0742     | 0.0735 | 0.3460   | 0.1117   | 0.1025        | 0.0976        | 0.1126         | 1.7357   | 6.3687   | 50.4823        | 11.5468  |
| pesa                | 11738  | 0.02537016    | 0.0288     | 0.0282 | 0.0603   | 0.0250   | 0.0222        | <b>0.0207</b> | 0.0215         | 3.0550   | 4.7261   | 50.1681        | 8.08772  |
| big                 | 13209  | <b>0.0251</b> | 0.0334     | 0.0319 | 0.0670   | 0.0298   | 0.0257        | 0.0285        | 0.0252         | 5.3358   | 3.6451   | 50.2023        | 9.368495 |
| mimo46x46_system    | 13250  | 0.02194796    | 0.0233     | 0.0219 | 0.0537   | 0.0211   | 0.0192        | <b>0.0190</b> | 0.0220         | 1.2824   | 2.5516   | 50.1932        | 4.945995 |
| xingo_afonso_itaipu | 13250  | 0.02153102    | 0.0238     | 0.0229 | 0.0498   | 0.0215   | 0.0193        | 0.0195        | <b>0.01856</b> | 1.0798   | 1.6007   | 50.1771        | 5.07843  |
| airfoil_2d          | 14214  | <b>0.0516</b> | 0.0733     | 0.0752 | 0.1942   | 0.1301   | 0.1115        | 0.1123        | 0.1387         | 1.3310   | 7.7960   | 50.5073        | 18.06465 |
| barth5              | 15606  | <b>0.0240</b> | 0.0278     | 0.0268 | 0.0940   | 0.0292   | 0.0261        | 0.0267        | 0.0313         | 1.0949   | 1.7613   | 50.1912        | 9.046615 |
| powersim            | 15838  | <b>0.0227</b> | 0.0273     | 0.0273 | 0.0586   | 0.0264   | 0.0228        | 0.0236        | 0.0232         | 1.6825   | 4.1641   | 50.2605        | 9.67221  |
| memplus             | 17758  | <b>0.0317</b> | 0.0431     | 0.0419 | 0.1084   | 0.0771   | 0.0657        | 0.0645        | 0.0621         | 1.8814   | 4.5265   | 50.3344        | 29.3187  |
| FEM_3D_thermal1     | 17880  | <b>0.0702</b> | 0.0963     | 0.0992 | 3.5811   | 0.1276   | 0.1112        | 0.1136        | 0.1112         | 5.1643   | 12.1416  | 50.6625        | 28.59335 |
| chipcool1           | 20082  | <b>0.0981</b> | 0.1271     | 0.1205 | 0.2518   | 0.1876   | 0.1462        | 0.1381        | 0.1386         | 3.8054   | 15.9848  | 50.6073        | 25.89095 |
| cz20468             | 20468  | <b>0.0390</b> | 0.0540     | 0.0550 | 0.2491   | 0.0908   | 0.0740        | 0.0842        | 0.0825         | 5.0032   | 11.2156  | 50.4674        | 6.915075 |
| bayer04             | 20545  | <b>0.0415</b> | 0.0579     | 0.0565 | 0.2196   | 0.5033   | 0.1046        | 0.1068        | 0.1014         | 2.3310   | 5.2068   | 50.4884        | 58.11975 |
| xingo3012           | 20944  | <b>0.0270</b> | 0.0325     | 0.0306 | 0.0894   | 0.0434   | 0.0405        | 0.0383        | 0.0365         | 1.8607   | 3.0839   | 50.3522        | 8.104725 |
| std1_Jac2           | 21982  | <b>0.4806</b> | 0.6491     | 0.5967 | 7.3145   | 14.9038  | 6.6621        | 6.6716        | 6.6628         | 5.0556   | 17.6001  | 52.0032        | 105.6455 |
| af23560             | 23560  | <b>0.0727</b> | 0.1019     | 0.1039 | 0.3580   | 0.2940   | 0.1407        | 0.1388        | 0.1410         | 19.8327  | 5.1623   | 50.8004        | 31.91645 |
| hvdcl               | 24842  | <b>0.0428</b> | 0.0577     | 0.0548 | 0.1330   | 0.0649   | 0.0581        | 0.0575        | 0.0558         | 5.7782   | 6.1291   | 50.5175        | 27.82375 |
| epb2                | 25228  | <b>0.0390</b> | 0.0525     | 0.0520 | 0.1850   | 0.1264   | 0.0764        | 0.0743        | 0.0754         | 2.0204   | 11.7196  | 50.5394        | 21.1954  |
| wang3               | 26064  | <b>0.0442</b> | 0.0580     | 0.0575 | 0.2041   | 0.0764   | 0.0664        | 0.0664        | 0.0642         | 8.8049   | 7.7975   | 50.5591        | 29.37565 |
| 2D_27628_bjtcai     | 27628  | <b>0.0725</b> | 0.1013     | 0.1014 | 0.3219   | 0.2165   | 0.1082        | 0.1162        | 0.1229         | 11.3172  | 9.5292   | 50.8493        | 35.2644  |
| mixtank_new         | 29957  | <b>0.9066</b> | 1.0866     | 1.0083 | 259.2303 | 3.2757   | 2.5975        | 2.4859        | 2.6221         | 61.2439  | 82.1067  | 52.8081        | 90.8827  |
| poli4               | 33833  | <b>0.0323</b> | 0.0353     | 0.0349 | 0.1469   | 0.0745   | 0.0561        | 0.0547        | 0.0523         | 1.9085   | 5.3783   | 50.6462        | 48.5469  |
| Zhao1               | 33861  | <b>0.0602</b> | 0.0769     | 0.0736 | 0.3757   | 0.4784   | 0.0786        | 0.0771        | 0.0755         | 11.4671  | 11.4468  | 50.8205        | 41.1204  |
| onetone1            | 36057  | <b>0.0900</b> | 0.1273     | 0.1225 | 0.9825   | 0.7124   | 0.3014        | 0.2979        | 0.2967         | 8.1091   | 5.7493   | 51.0595        | 73.709   |
| chem_master1        | 40401  | <b>0.0609</b> | 0.0824     | 0.0774 | 0.3860   | 0.0869   | 0.0844        | 0.0836        | 0.0929         | 5.1569   | 8.0364   | 51.0514        | 29.84665 |
| xenon1              | 48600  | <b>0.3336</b> | 0.4029     | 0.3978 | 3.6534   | 0.9017   | 0.5246        | 0.4947        | 0.5630         | 42.0395  | 45.6470  | 52.4375        | 38.90935 |
| ec132               | 51993  | <b>0.0984</b> | 0.1780     | 0.1554 | 1.9270   | 1.9135   | 1.6464        | 1.8788        | 1.5460         | 10.9750  | 14.0007  | 51.7268        | 30.5772  |
| bcircuit            | 68902  | <b>0.1162</b> | 0.1666     | 0.1554 | 0.4214   | 0.2029   | 0.1864        | 0.1806        | 0.1846         | 21.0736  | 41.6134  | 52.5335        | 50.54515 |
| shyy161             | 76480  | <b>0.0792</b> | 0.1144     | 0.1095 | 2.5546   | 0.1642   | 0.1647        | 0.1661        | 0.1640         | 5.5343   | 6.1399   | 52.7851        | 56.29615 |
| circuit_4           | 80209  | <b>0.1000</b> | 0.1345     | 0.1337 | 4.0781   | 3.4778   | 3.1946        | 3.1823        | 3.8445         | 7.4992   | 9.3681   | 53.1366        | 36.3078  |
| epb3                | 84617  | <b>0.1176</b> | 0.1589     | 0.1558 | 1.3029   | 1.1815   | 0.2052        | 0.2110        | 0.2094         | 3.2078   | 15.6129  | 53.5965        | 74.35685 |
| poisson3Db          | 85623  | <b>1.2856</b> | 1.6763     | 1.5391 | 6.7725   | 14.2176  | 3.3115        | 3.5702        | 3.6216         | 102.3131 | 176.3010 | 57.1718        | 176.1115 |
| rajat28             | 87190  | <b>0.1656</b> | 0.2531     | 0.2409 | 4.9201   | 5.3110   | 4.7890        | 5.0036        | 4.1128         | 10.5598  | 19.7834  | 53.9226        | 39.31645 |
| LeGresley_87936     | 87936  | <b>0.1823</b> | 0.2881     | 0.2329 | 0.6580   | 0.3581   | 0.3126        | 0.3055        | 0.3258         | 7.7922   | 20.1262  | 54.2827        | 42.1455  |
| rajat18             | 94294  | <b>0.1349</b> | 0.2006     | 0.1940 | 3.6578   | 3.9124   | 3.3132        | 3.3042        | 3.2821         | 9.5471   | 16.1836  | 54.3219        | 40.5265  |
| ASIC_100ks          | 99190  | <b>0.1703</b> | 0.2433     | 0.2327 | 0.7327   | 0.7312   | 0.3681        | 0.3644        | 0.3746         | 12.6499  | 64.9006  | 54.8141        | 46.0318  |
| ASIC_100k           | 99340  | <b>0.3418</b> | 0.5824     | 0.5237 | 232.0313 | 225.1798 | 225.3074      | 223.8386      | 234.8950       | 22.8801  | 29.6697  | 55.4503        | 186.8515 |
| matrix_9            | 103430 | <b>0.3697</b> | 0.5247     | 0.5080 | 97.4564  | 31.2975  | 0.6502        | 0.6589        | 0.7111         | 18.6788  | 46.5454  | 56.7357        | 179.6075 |
| hcircuit            | 105676 | <b>0.1388</b> | 0.2060     | 0.1991 | 1.2709   | 0.7970   | 0.7897        | 0.7811        | 0.8869         | 17.4605  | 21.0619  | 59.6858        | 76.3394  |
| lung2               | 109460 | <b>0.1160</b> | 0.1597     | 0.1612 | 0.5276   | 0.2245   | 0.2130        | 0.2092        | 0.2120         | 5.0194   | 30.0554  | 55.3130        | 160.0315 |
| rajat23             | 110355 | <b>0.1521</b> | 0.2215     | 0.2166 | 1.0070   | 0.7148   | 0.7018        | 0.6861        | 0.6891         | 11.1079  | 19.3731  | 55.4983        | 46.0894  |
| Baumann             | 112211 | <b>0.1762</b> | 0.2426     | 0.2411 | 2.1275   | 0.3556   | 0.3269        | 0.3283        | 0.3272         | 44.8872  | 42.1012  | 55.9188        | 139.341  |

|                  |        |         |         |           |          |          |          |          |           |           |            |            |   |
|------------------|--------|---------|---------|-----------|----------|----------|----------|----------|-----------|-----------|------------|------------|---|
| Núm. melhores    | 44     | 0       | 0       | 0         | 0        | 0        | 1        | 2        | 1         | 0         | 0          | 0          | 0 |
| $\sum \rho\beta$ | 0.6908 | 17.1953 | 15.2310 | 1618.9824 | 981.8740 | 810.1102 | 808.0996 | 839.7661 | 4330.0166 | 7140.2979 | 45652.1925 | 22667.2414 |   |

## APÊNDICE B – Parametrização do método GMRES

Antes de se iniciar as simulações com resolução de SELs com o método GMRES pré-condicionado é preciso que se definam diversos parâmetros dos pré-condicionadores e do método GMRES, em si. Nesta seção, mostra-se como foram feitos os testes para escolha do melhor conjunto de parâmetros para as simulações subsequentes.

É importante reforçar que a escolha dos parâmetros para o GMRES pré-condicionado é altamente dependente da instância que se deseja resolver. Por esse motivo, os valores definidos para os parâmetros nesta seção são apenas um ponto de partida comum para as simulações realizadas no Capítulo 8.

Para os pré-condicionadores baseados em fatoração incompleta ILUT, ILUC e VBILUT, os parâmetros a se considerar são  $\tau$ , relacionado com a tolerância para descarte de elementos com valor pequeno durante a fatoração e  $p$ , que define o número máximo de elementos por linha, conforme explicado na Seção 7.2.2.

Nos pré-condicionadores ILU(k) e VBILU(k), o parâmetro  $k$  controla o nível de fill-in da fatoração, como explicado na Seção 7.2.1. A utilização de valores maiores de  $k$  tendem a produzir melhores sistemas pré-condicionados, porém com custos computacionais de geração mais elevados.

Um dos parâmetros a se considerar com o pré-condicionador ARMS é o número de níveis da fatoração, como explicado na Seção 7.2.4. A implementação ARMS disponível na biblioteca ITSOL permite definir também o parâmetro  $tol_{dd}$ , relacionado com o grau de dominância da diagonal principal, e o parâmetro  $block\_size$ , que define o tamanho dos conjuntos (blocos) independentes gerados a cada nível.

### .1 Determinação dos parâmetros para o método GMRES pré-condicionado

A versão do método GMRES implementada na biblioteca ITSOL v2.0 (SAAD, 2017) é o GMRES com reinícios, ou GMRES(m), em que  $m$  é número de iterações internas até que o algoritmo seja reiniciado. Nessa versão do método GMRES, como explicado na Seção 7.1.2, a escolha de um valor adequado para o parâmetro  $m$  é fundamental para a convergência do método.

Cada um dos parâmetros mencionados, de cada pré-condicionador e do GMRES, possui um intervalo de valores aceitáveis que é definido de acordo com as características da instância. Quando possível, buscou-se referências na literatura para a escolha desse intervalo de valores.

De maneira a evitar um número exageradamente grande de combinações, o conjunto de valores definidos para cada parâmetro foi restrito a poucos elementos, mas que representassem intervalos de valores coerentes com as características das matrizes utilizadas nos testes com o método GMRES pré-condicionado. A Tabela B1 resume os valores testados para cada parâmetro. Para estes testes, considerou-se que as melhores combinações de parâmetros fossem as que levassem a um menor tempo total de processamento, no geral.

Tabela B1 – Parâmetros dos pré-condicionadores e respectivos intervalos de valores utilizados nos testes.

| Método             | Parâmetro         | Valores testados                               |
|--------------------|-------------------|------------------------------------------------|
| GMRES(m)           | m                 | 30, 50, 100, 250                               |
| ILUT, ILUC, VBILUT | $\tau$            | $10^{-1}, 10^{-3}, 10^{-6}, 10^{-9}$           |
|                    | $p$               | 15, 50, 100, grauMaximo                        |
| ILU(k), VBILU(k)   | k                 | 0, 1, 2, 3, 4                                  |
| ARMS               | <i>num_niveis</i> | 3, 5, 10                                       |
|                    | <i>tol_dd</i>     | 0.1, 0.4, 0.7                                  |
|                    | <i>block_size</i> | 30, 100, $\lfloor \text{grauMaximo}/2 \rfloor$ |

Para GMRES(m), pode-se encontrar na literatura trabalhos que utilizaram GMRES(20) (BENZI; HAWS; TUMA, 2000; DUFF; KOSTER, 2001), GMRES(30) (CAMATA et al., 2012), GMRES(50) (CHOW; SAAD, 1997), GMRES(60) (OSEI-KUFFUOR; SAAD, 2010), GMRES(100) ((LI; SAAD; SOSONKINA, 2003)), entre outros. Devido às dimensões das instâncias utilizadas nos testes conduzidos para redução do custo computacional do método GMRES pré-condicionado deste trabalho (Capítulo 8), optou-se por favorecer valores maiores para este parâmetro. Nos testes realizados para definição dos valores dos parâmetros, os valores de  $m$  escolhidos para GMRES(m) foram 30, 50, 100 e 250.

Exemplos de valores encontrados na literatura para parâmetro  $\tau$  usado na fatoração incompleta com *threshold* passam por  $\tau = 10^{-1}$  (BENZI; HAWS; TUMA, 2000; DUFF; KOSTER, 2001),  $\tau = 10^{-2}$  (BENZI; SZYLD; DUIN, 1999),  $\tau = 10^{-3}$  (BENZI; SZYLD; DUIN, 1999; SAAD; SUCHOMEL, 2002),  $\tau = 10^{-4}$  e  $\tau = 10^{-5}$  (BENZI; SZYLD; DUIN, 1999; LI; SAAD; SOSONKINA, 2003), para citar alguns. Para o parâmetro  $p$ , podem-se encontrar  $p = 5$  (BENZI; HAWS; TUMA, 2000),  $p = 10$  (BENZI; SZYLD; DUIN, 1999),  $p=15$  (SOULÄIMANI; SALAH; SAAD, 2002),  $p=30$  (CHOW; SAAD, 1997),  $p=50$  (SAAD; SUCHOMEL, 2002),  $p=60$  (LI; SAAD; SOSONKINA, 2003), entre outros. Em (BENZI; HAWS; TUMA, 2000, p.1345-1346), os autores reconhecem que os valores pequenos de  $\tau$  e  $p$  utilizados em seu

trabalho são incomuns na prática e que "geralmente, tolerâncias ( $\tau$ ) muito menores e valores de  $p$  muito maiores são usados, particularmente para problemas difíceis". Por este motivo, em virtude das dimensões das matrizes utilizadas nos testes deste trabalho com o método GMRES pré-condicionado, optou-se por incluir valores maiores de  $p$  e valores menores de  $\tau$  que os normalmente encontrados na literatura. Ainda, como também enfatiza (CHOW; SAAD, 1997), essa escolha implica em pré-condicionadores mais robustos e pode ser crucial para a convergência do método GMRES pré-condicionado em instâncias mais difíceis. Deste modo, nos testes para escolha de parâmetros para o método GMRES, os valores definidos para  $\tau$  foram  $10^{-1}$ ,  $10^{-3}$ ,  $10^{-6}$  e  $10^{-9}$ . Os valores de  $p$  escolhidos foram 15, 50, 100 e o valor do maior grau entre os vértices do grafo correspondente à matriz de entrada.

O parâmetro  $k$  dos pré-condicionadores baseados no ILU( $k$ ) assume um variado conjunto de valores na literatura. Por exemplo, nos trabalhos de (BENZI; SZYLD; DUIN, 1999; DUFF; KOSTER, 2001) utiliza-se ILU(0) e ILU(1), e em (LUGON; CATABRIGA, 2013b), são feitos testes com ILU(0), ILU(4), ILU(8), ILU(12), ILU(16), ILU(20). Neste trabalho, devido às dimensões das instâncias testadas, valores grandes de  $k$  poderiam demandar muito tempo de processamento e, mesmo, exaurir os recursos computacionais da máquina. Veja (LUGON; CATABRIGA, 2013b) para comparações do *fill-in* resultante de diferentes valores de  $k$ , por exemplo. Por isso utilizou-se valores pequenos de  $k$ , entre 0 e 4, nos testes para a escolha dos valores dos parâmetros do método GMRES.

Para o parâmetro *num\_niveis* do ARMS, pode-se encontrar valores de *num\_niveis*=4 (OSEI-KUFFUOR; SAAD, 2010) e *num\_niveis*=5 (SAAD; SUCHOMEL, 2002), por exemplo. Nos testes deste trabalho, foram usados os valores 3, 5 e 10 de maneira empírica. Não se encontrou referências sobre os parâmetros *tol\_dd* e *block\_size* do ARMS na literatura. Para o parâmetro *tol\_dd*, que admite valores entre 0 e 1, foram escolhidos 0.1, 0.4 e 0.7 e para o parâmetro *block\_size* os valores 30, 100 e o (piso) do maior grau entre os vértices do grafo dividido por 2.

## .2 Execuções e parâmetros escolhidos

Para cada pré-condicionador, cada possível combinação de parâmetros foi executada em um conjunto de 35 instâncias da *SuiteSparse Matrix Collection* com até 150 mil vértices. As instâncias foram selecionadas de cinco áreas diferentes: 2D/3D, *circuit simulation*, *computati-*

*onal fluid dynamics, optimization e structural*. A Tabela B2 mostra as instâncias selecionadas e as respectivas áreas de aplicação dos problemas que as originaram.

Após a seleção dos parâmetros, o método GMRES pré-condicionado foi utilizado nas instâncias selecionadas com as diferentes combinações de parâmetros dos pré-condicionadores. As iterações foram encerradas quando excederam  $|n|$ , em que  $n$  é o número de linhas da matriz, ou quando a norma do resíduo ficou menor que  $10^{-8}$ . Quando o vetor  $b$  não foi fornecido, utilizou-se um vetor de valores aleatórios entre -1 e 1.

Como mencionado, a escolha do parâmetro  $m$ , em GMRES( $m$ ), é fundamental para o comportamento de convergência do GMRES. Para os testes deste trabalho, optou-se por fixar um valor de  $m$  em todas as execuções do método GMRES. O valor de  $m$  escolhido foi aquele que obteve o menor somatório da métrica  $\rho$  quanto ao tempo de execução para execuções com todas os pré-condicionadores e suas possíveis combinações de parâmetros. Isto é, para cada valor de  $m$  listado na Tabela B1, cada pré-condicionador foi executado com todas as combinações de parâmetros possíveis (também listados na referida tabela), e calculou-se o valor de  $\rho$  da soma dos tempos de cada uma delas.

A compilação desses resultados é exibida na Tabela B3. Como pode ser observado, o menor valor de  $\sum \rho$  para GMRES( $m$ ) foi  $m = 50$ . Isto é, na média, as execuções com GMRES(50) com todas as outras combinações de parâmetros de pré-condicionadores obtiveram os menores tempos, em geral, segundo a métrica  $\rho$ . Então, esse foi o valor utilizado para GMRES( $m$ ) nos testes do Capítulo 8.

Após a definição de  $m = 50$  para as simulações, buscou-se identificar quais valores de parâmetros dos pré-condicionadores levavam o GMRES(50) a convergir com os menores tempos. Assim, calculou-se o valor de  $\sum \rho$  (equação 6.1) quanto à média das somas dos tempos nas execuções com GMRES(50), para cada valor possível dos parâmetros listados. As Tabelas B4 a B9 mostram os valores de  $\sum \rho$  para cada parâmetro.

Os melhores valores para  $\tau$  nos testes foram obtidos com  $\tau(10^{-3})$ , vide Tabela B4. Seguindo os valores mostrados na Tabela B5, o valor adotado para  $p$  foi  $p=50$ . O valor escolhido para o parâmetro ILU( $k$ ) foi  $k=2$ , conforme a Tabela B6. Os parâmetros escolhidos para o pré-condicionador ARMS foram  $num\_niveis=5$ ,  $tol\_dd=0.4$ , por um pequena margem, e  $block\_size=30$ , como pode ser visto nas Tabelas B7 a B9.

É natural que, devido a variedade de características presentes nas instâncias testadas (número de vértices, número de elementos não nulos e, principalmente, número de condição,

para citar algumas), a escolha de parâmetros não seja a ideal para todas elas. Contudo, acredita-se que os valores obtidos nesses testes sejam pontos de partida adequados para simulações com instâncias de maior porte, além de estarem em consonância com valores usados por outros trabalhos na literatura.

Tabela B2 – Áreas de aplicação de 35 instâncias da *SuiteSparse Matrix Collection* usadas nos testes. O número de vértices é indicado por  $n$  e o número de coeficientes não nulos, por  $nnz$ .

| Área de aplicação            | Instância      | $n$    | $nnz$   |
|------------------------------|----------------|--------|---------|
| 2D/3D                        | cant           | 62451  | 4007383 |
|                              | consph         | 83334  | 6010480 |
|                              | cop20k_A       | 121192 | 2624331 |
|                              | Dubcova2       | 65025  | 1030225 |
|                              | Dubcova3       | 146689 | 3636643 |
|                              | torso2         | 115967 | 1033473 |
| Circuit simulation           | ASIC_100ks     | 99190  | 578890  |
|                              | G2_circuit     | 150102 | 726674  |
|                              | bcircuit       | 68902  | 375558  |
| Computational fluid dynamics | cf1            | 70656  | 1825580 |
|                              | cf2            | 123440 | 3085406 |
|                              | lung2          | 109460 | 492564  |
|                              | shallow_water1 | 81920  | 327680  |
|                              | venkat01       | 62424  | 1717792 |
| Optimization                 | blockqp1       | 60012  | 640033  |
|                              | brainpc2       | 27607  | 179395  |
|                              | c-57           | 37833  | 403373  |
|                              | cont-201       | 80595  | 438795  |
|                              | cvxbqp1        | 50000  | 349968  |
|                              | dixmaanl       | 60000  | 299998  |
|                              | gridgena       | 48962  | 512084  |
|                              | minsurfo       | 40806  | 203622  |
|                              | ncvxqp1        | 12111  | 73963   |
|                              | qpband         | 20000  | 45000   |
| Structural                   | apache1        | 80800  | 542184  |
|                              | bcstk37        | 25503  | 1140977 |
|                              | bodyy6         | 19366  | 134208  |
|                              | cbuckle        | 13681  | 676515  |
|                              | msc23052       | 23052  | 1142686 |
|                              | olafu          | 16146  | 1015156 |
|                              | raefsky4       | 19779  | 1316789 |
|                              | ship_001       | 34920  | 3896496 |
|                              | smt            | 25710  | 3749582 |
|                              | sparsine       | 50000  | 1548988 |
|                              | vanbody        | 47072  | 2329056 |

Tabela B3 – Parametrização do método GMRES: valores de  $\sum \rho$  para o parâmetro  $m$  nas 35 instâncias selecionadas, com todas as combinações de parâmetros dos pré-condicionadores

|               | GMRES(30) | GMRES(50)    | GMRES(100) | GMRES(250) |
|---------------|-----------|--------------|------------|------------|
| $\sum \rho_t$ | 4.473     | <b>2.246</b> | 2.285      | 3.737      |

Tabela B4 – Parametrização do método GMRES( $m=50$ ): valores de  $\sum \rho$  para o parâmetro  $\tau$

|             | $\tau(10^{-1})$ | $\tau(10^{-3})$ | $\tau(10^{-6})$ | $\tau(10^{-9})$ |
|-------------|-----------------|-----------------|-----------------|-----------------|
| $\sum \rho$ | 19.80           | <b>12.30</b>    | 32.68           | 100.60          |

Tabela B5 – Parametrização do método GMRES( $m=50$ ): valores de  $\sum \rho$  para o parâmetro  $p$

|             | $p(15)$ | $p(50)$      | $p(100)$ | $p(\text{grauMaximo})$ |
|-------------|---------|--------------|----------|------------------------|
| $\sum \rho$ | 11.71   | <b>11.21</b> | 17.39    | 17.30                  |

Tabela B6 – Parametrização do método GMRES( $m=50$ ): valores de  $\sum \rho$  para o parâmetro  $k$

|             | ILU(0) | ILU(1)      | ILU(2) | ILU(3) | ILU(4) |
|-------------|--------|-------------|--------|--------|--------|
| $\sum \rho$ | 5.67   | <b>2.33</b> | 4.33   | 7.24   | 9.27   |

Tabela B7 – Parametrização do método GMRES( $m=50$ ): valores de  $\sum \rho$  para o parâmetro  $num\_niveis$

|             | $num\_niveis(3)$ | $num\_niveis(5)$ | $num\_niveis(10)$ |
|-------------|------------------|------------------|-------------------|
| $\sum \rho$ | <b>4.11</b>      | 7.50             | 30.08             |

Tabela B8 – Parametrização do método GMRES( $m=50$ ): valores de  $\sum \rho$  para o parâmetro  $tol\_dd$

|             | $tol\_dd(0.1)$ | $tol\_dd(0.4)$ | $tol\_dd(0.7)$ |
|-------------|----------------|----------------|----------------|
| $\sum \rho$ | <b>4.19</b>    | 21.55          | 47.95          |

Tabela B9 – Parametrização do método GMRES( $m=50$ ): valores de  $\sum \rho$  para o parâmetro  $block\_size$

|             | $block\_size(30)$ | $block\_size(100)$ | $block\_size(\text{grauMaximo}/2)$ |
|-------------|-------------------|--------------------|------------------------------------|
| $\sum \rho$ | 25.99             | 4.19               | <b>3.37</b>                        |