# An evaluation of several heuristics for bandwidth and profile reductions to reduce the computational cost of the preconditioned Conjugate Gradient Method

**Sanderson L. Gonzaga de Oliveira, Júnior Assis Barreto Bernardes and Guilherme Oliveira Chagas**
Universidade Federal de Lavras
sanderson@dcc.ufla.br, jrassis@posgrad.ufla.br, guilherme.chagas@computacao.ufla.br

**ABSTRACT**

Hundreds of heuristics have been proposed to solve the problems of bandwidth and profile reductions since the 1960s. Specifically, we found 131 heuristics applied to these problems. Among them, 15 were selected in a way that no other simulation or comparison showed that these algorithms could be superseded by any other algorithm in the articles analyzed, in terms of bandwidth or profile reduction and also considering the computational costs of the heuristics. Therefore, these 15 heuristics were selected as the potential best low-cost methods to solve these problems. These 15 heuristics for bandwidth and profile reductions are compared here when solving linear systems using the Jacobi-preconditioned conjugate gradient method.

**KEYWORDS. Bandwidth reduction, profile reduction, combinatorial optimization, heuristics, reordering algorithms, sparse matrices, renumbering, graph labeling, Conjugate Gradient Method.**

## Introduction

In several science and engineering applications, a fundamental task is the resolution of large sparse linear systems $Ax = b$, in which $A$ is a sparse matrix. In general, this is the part of the simulation that requires the highest computational cost. A large amount of memory and a high processing cost are required to store and to solve these large-scale linear systems. For the low-cost solution of large and sparse linear systems and to reduce the memory space required, an appropriate vertex reordering is desirable to assure that the corresponding coefficient matrix $A$ will have narrow bandwidth and small profile. Thus, heuristics for bandwidth and profile reductions are used to achieve low execution times and memory requirements for solving large sparse linear systems [Gonzaga de Oliveira e Chagas, 2015]. Specifically, profile reduction is used to reduce storage costs of applications that employ the skyline data structure [Felippa, 1975] to represent large-scale matrices. Profile reductions have also been used in connection with other science and engineering applications, such as archeology [Kendall, 1969], biology [Karp, 1993], and information retrieval [Botafogo, 1993].

Let $A = [a_{ij}]$ be a symmetric matrix $n \times n$. The bandwidth of line $i$ is $\beta_i(A) = i - min(j : (1 \leq j < i)\ a_{ij} \neq 0)$. Bandwidth of $A$ is defined as $\beta(A) = max((1 \leq i \leq n)\ \beta_i(A)) = max((1 \leq i \leq n)\ (1 \leq j < i)\ (i - j) \mid a_{ij} \neq 0)$. The profile of $A$ is defined as $profile(A) = \sum_{i=1}^{n} \beta_i(A)$. The bandwidth and profile minimization problems are hard [Papadimitriou, 1976; Lin e Yuan, 1994] and, since the mid-1960s, several heuristics have been proposed to solve the bandwidth and profile reduction problems.

When a matrix is symmetric and strictly diagonally dominant, a prominent algorithm for solving large sparse linear systems is the Conjugate Gradient Method [Hestenes e Stiefel, 1952; Lanczos, 1952]. One can achieve a computational cost reduction of this method by applying a local ordering of the vertices [Duff e Meurant, 1989] of the corresponding graph of $A$ in order to improve cache hit rates [Gonzaga de Oliveira e Chagas, 2015]. It should be noted that it is also important to have an ordering which does not lead to a large increase of the number of iterations of this method when a preconditioner is applied. This local ordering can be reached by applying a heuristic for bandwidth reduction [Burgess e Giles, 1997; Das et al., 1992].

The main objective of this work is to evaluate 15 potential state-of-the-art low-cost heuristics for bandwidth and profile reductions with the intention of reducing the computational cost of the Jacobi-preconditioned conjugate gradient method (JPCGM). These 15 heuristics were selected from systematic reviews.

This paper is structured as follows. Section 2 describes the systematic reviews accomplished to identify the potential best low-cost heuristics for bandwidth and profile reductions. Section 3 shows how the tests were conducted in this study. Section 4 presents the results. Finally, section 5 addresses the conclusions.

## Systematic reviews

In systematic reviews [Chagas e Gonzaga de Oliveira, 2015; Gonzaga de Oliveira e Chagas, 2015; Bernardes e Gonzaga de Oliveira, 2015], 73 and 74 heuristics for bandwidth and profile reductions, respectively, were identified that had been published in the period of time spanning the 1960s to the present. From the heuristics identified in the systematic reviews, 17 heuristics were applied to both bandwidth and profile reductions. Thus, 130 heuristics for bandwidth and profile reductions were identified. These systematic reviews describe eight heuristics in each case as the most promising heuristics for bandwidth (Burgess e Lai [1986], WBRA [Esposito et al., 1998], FNCHC [Lim et al., 2003, 2004, 2007], GGPS [Wang et al., 2009], VNS-band [Mladenovic et al., 2010], hGPHH [Koohestani e Poli, 2011], CSS-band [Kaveh e Sharafi, 2012]) or profile reduction (Snay [1976], RCM-GL-FL [Fenves e Law, 1983], Sloan [1989], MPG [Medeiros et al., 1993], NSloan [Kumfert e Pothen, 1997], Sloan-MGPS [Reid e Scott, 1999], Hu e Scott [2001]). In addition, the reverse Cuthill-McKee method with pseudo-peripheral vertex given by the George-Liu algorithm

(RCM-GL) [George e Liu, 1981] was selected in both systematic reviews of heuristics for bandwidth and profile reductions. It should be noted that these 15 heuristics were selected because no other simulation or comparison showed that these 15 heuristics could be outperformed by any other heuristics in the articles analyzed, in terms of bandwidth or profile reduction, when the computation costs of the given heuristic were also taken into account. In particular, several heuristics for bandwidth or profile reduction have been designed through metaheuristics, but a few of them were selected as the most promising heuristics to solve these problems. A reason for this was the high execution times showed by the metaheuristic-based heuristics (see [Chagas e Gonzaga de Oliveira, 2015; Bernardes e Gonzaga de Oliveira, 2015]).

After we have published these systematic reviews, Torres-Jimenez et al. [2015] proposed a heuristic for bandwidth reduction based on a dual-representation simulated annealing (DRSA-band). The DRSA-band obtained results slightly better than results of the VNS-band heuristic [Mladenovic et al., 2010]. However, the DRSA-band demonstrated higher computational costs than the VNS-band in the results presented by Torres-Jimenez et al. [2015]. Thus, the DRSA-band heuristic was not considered as potentially the best low-cost heuristic with significant bandwidth reduction because its computational cost is higher than the computational cost of the VNS-band heuristic; apart from significantly reducing the bandwidth, a heuristic must also present low computing times when compared to other heuristics.

Despite not having been selected in the systematic reviews, the GPS heuristic [Gibbs et al., 1976] was implemented and its results were compared to the other heuristics in this computational experiment because it is one of the most classic low-cost heuristics tested in the field for both bandwidth and profile reductions. On the other hand, the RCM-GL-FL heuristic [Fenves e Law, 1983], despite being selected in the systematic review of heuristics for profile reduction, was not implemented in this work because it is simply a specific application of the RCM-GL method [George e Liu, 1981] for finite element discretizations. Therefore, 15 heuristics were implemented and tested in this work. Additionally, the exchange methods (EM) [Reid e Scott, 2002] (i.e., local search algorithms) were tested in conjunction with several of these 15 heuristics.

**Description of the tests, implementation of the heuristics, testing, and calibration**

Regarding the simulations with the 15 heuristics for bandwidth and profile reductions that were selected in systematic reviews, a 64-bit executable program of the VNS-band heuristic (which was kindly provided by one of the heuristic's authors) was used. This executable only runs with instances up to 500,000 vertices.

The FNCHC-heuristic source code was also kindly provided by one of the heuristic's authors. With this, the source code was converted and implemented in this present work in the C++ programming language. We asked all the 13 other heuristics' authors for the sources and/or executables of their algorithms. However, some authors answered that they no longer had the source code or executable, some authors did not respond, and others explained that the programs could not be provided. Then, the 13 other heuristics were also implemented in the C++ programming language so that the computational costs of the heuristics could be compared accordingly. Specifically, the g++ version 4.8.2 compiler was used. In addition, the *GNU Multiple Precision Floating-point Computations with Correct-Rounding* (MPFR) library with 512-bit precisions were employed to make it possible to achieve high precision in the computations.

It should be observed that it was not our objective that the results of the C++ programming language versions of the heuristics supersede all the results of the original implementations. Our goal was to implement reasonably efficient implementations of the heuristics tested in order to make it possible an appropriate comparison of the results of the 15 heuristics. On the other hand, we tested and calibrated the C++ programming language of the heuristics performed in order to compare our implementations with the codes used by the original proposers of the heuristics to ensure the codes we implemented were comparable to the algorithms that were originally proposed. We compared the results of the C++ programming language versions of the heuristics with the results presented in the original publications. It should be noted that these heuristics are simple to implement:

- the RCM-GL method [George e Liu, 1981],
- the heuristic of Snay [1976],
- the algorithm of Sloan [1989],
- the Medeiros-Pimenta-Goldenberg (MPG) heuristic [Medeiros et al., 1993],
- the Normalized Sloan (NSloan) heuristic [Kumfert e Pothen, 1997],
- Sloan's algorithm with pseudo-peripheral vertex given by the modified GPS algorithm (Sloan-MGPS) [Reid e Scott, 1999],
- the heuristic based on genetic programming hyper-heuristic (hGPHH) [Koohestani e Poli, 2011].

In particular, the RCM-GL method [George e Liu, 1981] is based on breadth-first search procedure, the heuristic of Snay [1976] and the hGPHH heuristic [Koohestani e Poli, 2011] are based on RCM method, and the MPG, NSloan, and Sloan-MGPS are based on Sloan's algorithm. Likewise other heuristics tested here, the hGPHH heuristic [Koohestani e Poli, 2011] is highly dependent on the starting vertex. Since Koohestani e Poli [2011] provided no explanation about the pseudo-peripheral vertex finder applied, the algorithm of George e Liu [1979] for computing a pseudo-peripheral vertex was employed here; consequently, this method was termed hGPHH-GL, i.e. it is an RCM-based heuristic designed through a genetic programming hyper-heuristic. Thus, in both algorithms, the starting vertex is given here by the George-Liu algorithm. In short, the C++ programming language implementations of the RCM-GL and hGPHH-GL heuristics obtained similar results to the results presented by Koohestani e Poli [2011] in relation to bandwidth reductions.

We closely observed the instructions provided by Lewis [1982] in a Fortran programming language source code in order to implement the GPS algorithm [Gibbs et al., 1976] in the C++ programming language. The C++ programming language GPS algorithm achieved better bandwidth results than the results presented by Lewis [1982].

The algorithm of Sloan [1989], NSloan [Kumfert e Pothen, 1997], Sloan-MGPS [Reid e Scott, 1999], Hu e Scott [2001], and Charged System Search for bandwidth reduction (CSS-band) [Kaveh e Sharafi, 2012] heuristics have important parameters that may affect the results. Exploratory investigations were conducted in order to determine the parameters, but in general, the better parameters are those suggested by the heuristics' authors: $w_1 = 1$ and $w_2 = 2$ (associated to global and local criteria corresponding to the distance of each vertex from the target end vertex and the degree of each vertex, respectively) within the Sloan [1989], Sloan-MGPS [Reid e Scott, 1999], and Hu-Scott [Hu e Scott, 2001] heuristics; and $w_1 = 2$ and $w_2 = 1$ within the NSloan heuristic [Kumfert e Pothen, 1997]. On the other hand, the other heuristics do not have parameters that influence the results.

We meticulously observed the guidances given by Sloan [1989] and also the recommendations reported at http://www.hsl.rl.ac.uk/archive/specs/mc40.pdf [STFC Rutherford Appleton Laboratory, 1963-2016] in a Fortran programming language source code in order to implement this heuristic in the C++ programming language. Moreover, we studied the Fortran programming language source code of the Sloan-MGPS available at http://www.hsl.rl.ac.uk/catalogue/mc60.html [STFC Rutherford Appleton Laboratory, 1963-2016] in order to implement this heuristic in the C++ programming language. The C++ programming language versions of the algorithm of Sloan [1989] and the Sloan-MGPS heuristic [Reid e Scott, 1999] yielded better profile results than the results of these heuristics presented by Reid e Scott [1999].

The C++ programming language version of the NSloan heuristic achieved comparable results to the results of the original version [Kumfert e Pothen, 1997] in relation to profile reductions. The C++ programming language version of the MPG heuristic achieved better profile results (in the 30 instances provided by Kumfert e Pothen [1997]) than the results of the original version [Kumfert e Pothen, 1997].

The heuristic of Hu e Scott [2001] is more complicated to be implemented than the other heuristics implemented here. We implemented this heuristic in the C++ programming language, but our results are not comparable to the results shown by Hu e Scott [2001]. Our C++ programming language of this heuristic is not a reasonably efficient implementation of this heuristic. In spite of that, in exploratory investigations, we realized that this is a high-cost heuristic, mostly for computing matrix multiplications, so that we did not make a greater programming effort to obtain a better implementation of this heuristic because it cannot outperform the other heuristics tested here when applied to reduce the computational cost of an iterative linear system solver. As described, a reordering algorithm must significantly reduce the bandwidth or profile at low cost: it cannot be expensive when compared to other algorithms. It should be noted that Hu e Scott [2001] showed no computational cost of their proposed heuristic. It was selected as one of the potential best heuristic for profile reduction because it showed promising profile results. However, when examining it to be implemented, it was realized that the heuristic performs matrix multiplications at each iteration.

As far as we know, results of the heuristic of Snay [1976], Burgess e Lai [1986], Wonder Bandwidth Reduction Algorithm (WBRA) [Esposito et al., 1998], Generic GPS (GGPS) [Wang et al., 2009], and CSS-band [Kaveh e Sharafi, 2012] heuristics have only been presented in their original papers and, unfortunately, we did not find the instances where these five heuristics were tested. In spite of that, since an efficient implementation comes at a cost in programming effort, we precisely followed the explanations of the algorithms provided by their authors in order to obtain reasonably efficient implementations of these heuristics.

The workstations used in the execution of the simulations contained an Intel® Core™ I3-550 (4MB Cache, CPU 3.20GHz $\times$ 4, 16GB of main memory DDR3 1333MHz) (Intel; Santa Clara, CA, United States). The Ubuntu 14.04 LTS 64-bit operating system with Linux kernel-version 3.13.0-39-generic was used.

Three sequential runs, with both a reordering algorithm and with the JPCGM, were performed in each instance. More executions were not necessary as the standard deviation and coefficient of variation observed were small within these three sequential runs. Moreover, it should be observed that we followed the recommendations given by Johnson [2002] for this experimental analysis of 15 low-cost heuristics for bandwidth and profile reductions aiming at reducing the computational cost of the JPCGM.

**Results and analysis**

This section presents and analyzes the results obtained in simulations using the JPCGM executed after reordering algorithms. Specifically, this section shows the results of the resolutions of linear systems (containing symmetric matrices) arising from the discretization of the heat conduction equation by finite volumes with meshes generated by Voronoi diagrams (and Delaunay triangulation) [Gonzaga de Oliveira et al., 2015].

Tables 1 and 2 show the dimension $n$ of the respective coefficient matrix of the linear system (or the number of vertices of the graph associated to the coefficient matrix), the sparsity (%) of these instances, the name of the reordering algorithms applied, the bandwidth and profile results, the average (in relation to the three sequential runs) results of the heuristics in relation to the computational cost, in seconds (s), and the memory requirements, in mebibytes (MiB). In addition, these tables provide the number of iterations and the total computational cost, in seconds, of the JPCGM. Furthermore, in spite of the small number of executions for each heuristic in each instance, Tables 1 and 2 show the standard deviation and coefficient of variation, referring to the total computational cost of JPCGM. In the last column of these two tables, reduction by using a heuristic for bandwidth and profile reductions is shown in relation to the JPCGM runtimes without reordering the graph vertices, i.e. computational costs without reordering are shown in the first line in a corresponding test with each instance in these tables. The best speedup of the JPCGM is highlighted in the last columns in Tables 1 and 2. Additionally, numbers in bold face are the best results.

The Hu e Scott [2001] and adjacent exchange methods [Reid e Scott, 2002], in conjunction with the other heuristics, were dominated by several heuristics when applied to the linear system composed of 4846 unknowns (when considering the speedup of the JPCGM). Consequently, these methods were not applied to larger linear systems.

The hGPHH-GL heuristic obtained the best results in reducing the computational cost of the JPCGM when applied to the linear systems with up to 50,592 unknowns and to the linear system composed of 232,052 unknonws. The RCM-GL method [George e Liu, 1981] obtained the best results in reducing the computational cost of the JPCGM when applied to the linear system composed of 108,683 unknowns. The MPG [Medeiros et al., 1993] obtained the best results in reducing the computational cost of the JPCGM when applied to the linear systems composed of 492,853, and 965,545 unknowns.

Figures 1 and 2 (without the results of the heuristics that obtained the worst results in this set of instances), built from a wide variety of references that were part of this work, show the speedup of the JPCGM when applied to the linear systems described (line plots were used for clarity). In particular, Figure 1 shows that the GPS [Gibbs et al., 1976], Burgess-Lai [Burgess e Lai, 1986], Hu-Scott [Hu e Scott, 2001], GGPS [Wang et al., 2009], CSS-band [Kaveh e Sharafi, 2012] heuristics and several heuristics applied in conjunction with the adjacent exchange methods [Reid e Scott, 2002] were dominated by the other heuristics evaluated. Figure 2 shows that the MPG [Medeiros et al., 1993], hGPHH-GL [Koohestani e Poli, 2011], and RCM-GL [George e Liu, 1981] heuristics obtained in general the best results in the instances tested.

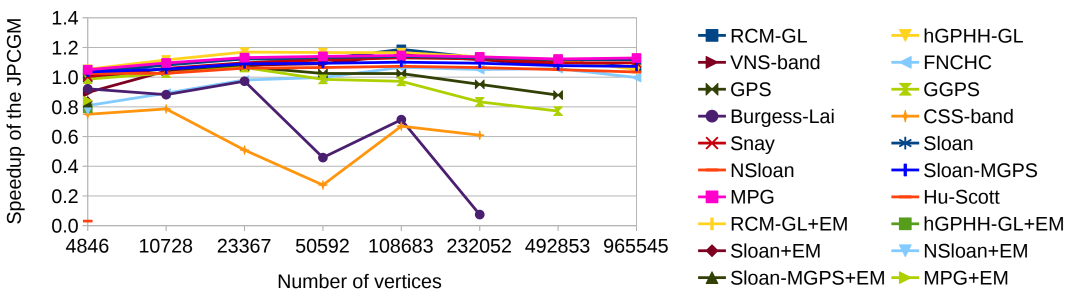



Figure 1: Speedup of the JPCGM obtained using several heuristics for bandwidth and profile reductions (tests shown in Tables 1 and 2) to matrices that belong to linear systems that originate from the discretization of the heat conduction equation by finite volumes.

Figure 3 shows the execution times, in seconds, of the six least expensive heuristics for bandwidth and profile reductions tested with matrices originating from the discretization of the heat conduction equation by finite volumes. In particular, Figure 3a shows that the Sloan-MGPS heuristic [Reid e Scott, 1999] obtained largest execution times than the other five heuristics in the instances tested. Figure 3b shows that the NSloan heuristic [Kumfert e Pothen, 1997] was the least expensive heuristic evaluated.

## Conclusions

The results of 15 heuristics for bandwidth and profile reductions when reducing the computational cost of solving linear systems using the Jacobi-preconditioned conjugate gradient method (JPCGM) have been presented. These 15 heuristics were selected in systematic reviews and may be

Table 1: Resolution of linear systems (up to 50,592 unknowns and derived from the discretization of the heat conduction equation by finite volumes) using the JPCGM and vertices labeled by reordering algorithms.

| $n$ | Heuristic | $\beta$ | *profile* | Heuristic | | JPCGM | | $\sigma$ | $C_v$ | Speedup |
|---|---|---|---|---|---|---|---|---|---|---|
| (%) | | | | t(s) | m.(MiB) | iter. | t(s) | | (%) | |
| 4846 (0.14) | Without reordering | 4769 | 9116750 | - | - | 319 | 9.97 | 0.04 | 0.46 | - |
| | RCM-GL | 161 | 468110 | 0.005 | 0 | 319 | 9.55 | 0.19 | 2.01 | 1.043 |
| | hGPHH-GL | 161 | 518637 | 0.005 | 0 | 319 | **9.46** | 0.05 | 0.52 | **1.053** |
| | VNS-band | 154 | 499797 | 1.054 | 44.51 | 319 | 10.06 | 0.15 | 1.50 | 0.897 |
| | FNCHC | **121** | 437962 | 2.287 | 2.22 | 319 | 10.04 | 0.08 | 0.86 | 0.809 |
| | GPS | 140 | 406081 | 0.166 | 0.32 | 319 | 9.83 | 0.19 | 1.90 | 0.997 |
| | GGPS | 130 | 422281 | 0.273 | 0.56 | 319 | 9.83 | 0.22 | 2.31 | 0.987 |
| | Burgess-Lai | 233 | 417653 | 0.545 | 0 | 319 | 10.27 | 0.05 | 0.48 | 0.922 |
| | CSS-band | 4765 | 8596483 | 2.360 | 22.16 | 319 | 10.93 | 0.27 | 2.51 | 0.750 |
| | Snay | 1047 | 352116 | 0.284 | 0 | 319 | 9.65 | 0.01 | 0.14 | 1.004 |
| | Sloan | 774 | 297591 | 0.006 | 0 | 319 | 9.66 | 0.01 | 0.07 | 1.031 |
| | NSloan | 608 | 472650 | **0.003** | 0 | 319 | 9.72 | 0.09 | 0.89 | 1.025 |
| | Sloan-MGPS | 658 | 288047 | 0.023 | 0 | 319 | 9.61 | 0.02 | 0.18 | 1.035 |
| | MPG | 1124 | 317356 | 0.006 | 0 | 319 | 9.49 | 0.01 | 0.11 | 1.050 |
| | Hu-Scott | 3501 | 640372 | 317.169 | 100.46 | 319 | 9.73 | 0.02 | 0.22 | 0.030 |
| | RCM-GL+EM | 476 | 358017 | 2.709 | 0 | 319 | 9.72 | 0.03 | 0.28 | 0.802 |
| | HGPHH+EM | 377 | 391204 | 2.965 | 0 | 319 | 9.64 | 0.02 | 0.16 | 0.791 |
| | Snay-GL+EM | 1088 | 384289 | 2.464 | 0.09 | 319 | 9.56 | 0.01 | 0.09 | 0.829 |
| | Sloan+EM | 919 | 268663 | 2.350 | 0 | 319 | 9.54 | 0.04 | 0.39 | 0.839 |
| | NSloan+EM | 601 | 348643 | 3.056 | 0.09 | 319 | 9.73 | 0.01 | 0.05 | 0.780 |
| | Sloan-MGPS+EM | 635 | **256907** | 2.390 | 0 | 319 | 9.64 | 0.05 | 0.55 | 0.829 |
| | MPG+EM | 1077 | 307725 | 2.374 | 0 | 319 | 9.47 | 0.01 | 0.09 | 0.842 |
| 10728 (0.06) | Without reordering | 10626 | 45314579 | - | - | 462 | 37.35 | 0.36 | 0.96 | - |
| | RCM-GL | 270 | 1579179 | 0.02 | 0 | 462 | 34.19 | 0.06 | 0.17 | 1.092 |
| | hGPHH-GL | 275 | 1768638 | 0.02 | 0 | 462 | **33.44** | 0.04 | 0.12 | **1.116** |
| | VNS-band | 552 | 1746660 | 1.14 | 135.62 | 462 | 34.73 | 0.41 | 1.14 | 1.041 |
| | FNCHC | 208 | 1637305 | 6.12 | 2.49 | 462 | 35.74 | 0.98 | 2.74 | 0.892 |
| | GPS | **207** | 1358676 | 0.63 | 1.54 | 462 | 34.69 | 0.23 | 0.66 | 1.057 |
| | GGPS | 226 | 1477595 | 1.00 | 1.38 | 462 | 35.35 | 0.63 | 1.79 | 1.028 |
| | Burgess-Lai | 398 | 1365197 | 6.38 | 0 | 462 | 35.96 | 0.12 | 0.32 | 0.882 |
| | CSS-band | 10625 | 42631422 | 4.71 | 79.92 | 462 | 42.81 | 0.78 | 1.82 | 0.786 |
| | Snay | 1289 | 1064597 | 0.87 | 0.26 | 462 | 34.39 | 0.46 | 1.33 | 1.059 |
| | Sloan | 1391 | **1012101** | 0.02 | 0 | 462 | 34.51 | 0.07 | 0.20 | 1.082 |
| | NSloan | 890 | 1511606 | **0.01** | 0 | 462 | 36.41 | 0.05 | 0.13 | 1.026 |
| | Sloan-MGPS | 1249 | 1023132 | 0.09 | 0 | 462 | 35.34 | 0.12 | 0.33 | 1.054 |
| | MPG | 2282 | 1097773 | 0.02 | 0 | 462 | 34.06 | 0.13 | 0.37 | 1.096 |
| 23367 (0.03) | Without reordering | 23167 | 216212086 | - | - | 671 | 124.39 | 0.53 | 0.42 | - |
| | RCM-GL | 313 | 4664523 | 0.07 | 0 | 671 | 110.69 | 2.01 | 0.82 | 1.123 |
| | hGPHH-GL | 314 | 5129407 | 0.08 | 0 | 671 | **106.32** | 0.26 | 0.25 | **1.169** |
| | VNS-band | 1564 | 8889127 | 1.54 | 371.12 | 671 | 113.53 | 1.74 | 1.53 | 1.081 |
| | FNCHC | 333 | 5519183 | 15.64 | 2.96 | 671 | 111.14 | 0.64 | 0.57 | 0.981 |
| | GPS | **293** | 4221479 | 3.57 | 3.34 | 671 | 112.56 | 3.52 | 3.12 | 1.071 |
| | GGPS | 314 | 5007391 | 5.30 | 3.51 | 671 | 111.47 | 2.72 | 2.45 | 1.065 |
| | Burgess-Lai | 465 | 4296542 | 11.20 | 0 | 671 | 116.66 | 0.29 | 0.25 | 0.973 |
| | CSS-band | 23183 | 202741325 | 94.78 | 549.09 | 671 | 149.49 | 1.59 | 1.05 | 0.509 |
| | Snay | 1824 | 3427644 | 2.96 | 0.69 | 671 | 110.43 | 1.94 | 1.76 | 1.097 |
| | Sloan | 2565 | **3419063** | 0.07 | 0 | 671 | 110.67 | 0.06 | 0.05 | 1.123 |
| | NSloan | 2057 | 5252921 | **0.05** | 0 | 671 | 117.18 | 0.05 | 0.05 | 1.061 |
| | Sloan-MGPS | 2650 | 3614398 | 0.33 | 0 | 671 | 113.77 | 0.12 | 0.13 | 1.090 |
| | MPG | 3988 | 3775704 | 0.06 | 0 | 671 | 109.85 | 0.26 | 0.24 | 1.132 |
| 50592 (0.01) | Without reordering | 50461 | 1020411959 | - | - | 970 | 391.19 | 0.13 | 0.03 | - |
| | RCM-GL | 647 | 17502999 | 0.26 | 0 | 970 | 348.44 | 10.71 | 3.07 | 1.122 |
| | hGPHH-GL | 640 | 19173913 | 0.26 | 0 | 970 | **335.17** | 0.48 | 0.14 | **1.166** |
| | VNS-band | 7377 | 47268027 | 3.03 | 967.09 | 970 | 355.55 | 8.48 | 2.38 | 1.091 |
| | FNCHC | 479 | 17779307 | 43.05 | 5.55 | 970 | 349.03 | 2.05 | 0.59 | 0.998 |
| | GPS | **466** | 14149442 | 24.65 | 5.51 | 970 | 357.03 | 10.56 | 2.96 | 1.025 |
| | GGPS | 499 | 15922368 | 55.64 | 7.72 | 970 | 341.25 | 2.13 | 0.63 | 0.986 |
| | Burgess-Lai | 961 | 14447154 | 491.74 | 0 | 970 | 361.89 | 2.08 | 0.58 | 0.458 |
| | CSS-band | 50287 | 951213645 | 954.92 | 2701 | 970 | 474.22 | 10.54 | 2.22 | 0.274 |
| | Snay | 2941 | **10776787** | 10.80 | 1.29 | 970 | 339.83 | 0.37 | 0.11 | 1.116 |
| | Sloan | 6025 | 11975091 | 0.23 | 0 | 970 | 345.52 | 0.31 | 0.09 | 1.131 |
| | NSloan | 4314 | 16059399 | **0.09** | 0 | 970 | 367.24 | 0.25 | 0.07 | 1.065 |
| | Sloan-MGPS | 3914 | 11784471 | 1.12 | 0 | 970 | 356.52 | 0.41 | 0.11 | 1.094 |
| | MPG | 9026 | 12000823 | 0.15 | 0.09 | 970 | 342.95 | 0.39 | 0.11 | 1.140 |

Table 2: Resolution of linear systems (ranging from 108,683 to 965,545 unknowns and derived from the discretization of the heat conduction equation by finite volumes) using the JPCGM and vertices labeled by heuristics for bandwidth and profile reductions.

| $n$ (%) | Heuristic | $\beta$ | *profile* | Heuristic | | JPCGM | | $\sigma$ | $C_v$ (%) | Speedup |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | t(s) | m.(MiB) | iter. | t(s) | | | |
| 108683 (0.006) | Without reordering | 108216 | 4725435534 | - | - | 1398 | 1238.76 | 29.02 | 2.34 | - |
| | RCM-GL | 868 | 54290089 | 0.25 | 0 | 1398 | **1043.15** | 2.57 | 0.25 | **1.187** |
| | hGPHH-GL | 868 | 59697870 | 0.26 | 0 | 1399 | 1062.85 | 5.69 | 0.53 | 1.165 |
| | VNS-band | 16676 | 157588720 | 8.64 | 2293.76 | 1398 | 1067.87 | 2.34 | 0.22 | 1.151 |
| | FNCHC | 738 | 55924986 | 116.36 | 11.63 | 1398 | 1045.24 | 4.70 | 0.45 | 1.066 |
| | GPS | **642** | 48744729 | 139.22 | 12.62 | 1398 | 1070.03 | 5.69 | 0.53 | 1.024 |
| | GGPS | 736 | 54051275 | 225.84 | 23.33 | 1398 | 1048.35 | 0.79 | 0.08 | 0.972 |
| | Burgess-Lai | 1262 | 45836037 | 617.56 | 0 | 1398 | 1117.72 | 0.68 | 0.06 | 0.714 |
| | CSS-band | 108390 | 99051906 | 369.55 | 586.52 | 1400 | 1479.53 | 11.81 | 0.80 | 0.670 |
| | Snay | 4281 | **33664485** | 29.12 | 2.98 | 1398 | 1066.89 | 1.28 | 0.12 | 1.130 |
| | Sloan | 10272 | 36886113 | 0.75 | 2.32 | 1398 | 1086.46 | 1.19 | 0.11 | 1.139 |
| | NSloan | 7826 | 52122701 | **0.19** | 2.32 | 1398 | 1154.12 | 0.73 | 0.06 | 1.073 |
| | Sloan-MGPS | 12656 | 38154730 | 4.18 | 2.32 | 1398 | 1121.04 | 0.35 | 0.03 | 1.101 |
| | MPG | 15172 | 38512718 | 0.34 | 2.32 | 1399 | 1080.37 | 0.57 | 0.05 | 1.146 |
| 232052 (0.003) | Without reordering | 231672 | 21652820640 | - | - | 2034 | 3814.37 | 0.33 | 0.01 | - |
| | RCM-GL | 1471 | 183551861 | 0.58 | 0 | 2039 | 3370.62 | 5.59 | 0.17 | 1.131 |
| | hGPHH-GL | 1471 | 200679371 | 0.61 | 0 | **2030** | 3352.42 | 20.39 | 0.61 | **1.138** |
| | VNS-band | 17036 | 313594920 | 33.49 | 5048.32 | **2030** | 3379.31 | 33.09 | 0.98 | 1.118 |
| | FNCHC | 1145 | 182638143 | 287.81 | 22.42 | 2031 | 3340.45 | 18.27 | 0.55 | 1.051 |
| | GPS | **1104** | 148697458 | 585.48 | 25.42 | 2032 | 3422.30 | 2.65 | 0.08 | 0.952 |
| | GGPS | 1243 | 170494162 | 1236.95 | 43.79 | 2031 | 3337.75 | 9.26 | 0.28 | 0.834 |
| | Burgess-Lai | 2048 | 153659093 | 47604.01 | 0 | 2038 | 3589.31 | 1.11 | 0.03 | 0.075 |
| | CSS-band | 231606 | 2857086021 | 1551.79 | 841.35 | 2039 | 4709.74 | 6.28 | 0.13 | 0.609 |
| | Snay | 6212 | **102404593** | 94.28 | 6.19 | **2030** | **3301.87** | 4.88 | 0.15 | 1.123 |
| | Sloan | 14640 | 119489765 | 2.46 | 4.13 | 2032 | 3372.41 | 5.53 | 0.16 | 1.130 |
| | NSloan | 23577 | 174602136 | **0.44** | 4.13 | 2032 | 3582.52 | 6.59 | 0.18 | 1.065 |
| | Sloan-MGPS | 26056 | 124108449 | 14.87 | 4.21 | 2032 | 3471.88 | 3.54 | 0.14 | 1.094 |
| | MPG | 22919 | 118186600 | 0.88 | 4.13 | 2031 | 3354.02 | 8.77 | 0.26 | 1.137 |
| 492853 (0.002) | Without reordering | 492100 | 97893937993 | - | - | 2927 | 11619.2 | 11.49 | 0.10 | - |
| | RCM-GL | 1805 | 510308269 | 1.48 | 0 | **2925** | 10440.46 | 12.96 | 0.22 | 1.113 |
| | hGPHH-GL | 1814 | 557734280 | 1.57 | 0 | 2932 | 10395.07 | 16.07 | 0.15 | 1.118 |
| | VNS-band | 41845 | 1083304907 | 137.67 | 10844.16 | 2936 | 10580.39 | 11.12 | 0.11 | 1.084 |
| | FNCHC | 1660 | 583349111 | 749.03 | 74.19 | 2932 | **10233.68** | 14.51 | 0.14 | 1.058 |
| | GPS | **1454** | 460254575 | 2700.88 | 49.85 | 2932 | 10516.23 | 25.43 | 0.34 | 0.879 |
| | GGPS | 1734 | 571294474 | 4759.91 | 107.57 | 2932 | 10297.02 | 16.08 | 0.16 | 0.772 |
| | Snay | 10273 | **329516129** | 323.58 | 12.97 | 2935 | 10270.53 | 26.6 | 0.26 | 1.097 |
| | Sloan | 44208 | 373429562 | 8.42 | 9.99 | 2934 | 10378.44 | 16.51 | 0.16 | 1.119 |
| | NSloan | 28222 | 510609978 | **1.19** | 9.93 | 2932 | 11054.62 | 41.74 | 0.38 | 1.051 |
| | Sloan-MGPS | 43022 | 391323009 | 54.07 | 9.99 | 2928 | 10727.11 | 13.94 | 0.13 | 1.078 |
| | MPG | 55654 | 387601624 | 2.64 | 9.96 | 2932 | 10348.92 | 2.15 | 0.02 | **1.122** |
| 965545 (0.001) | Without reordering | 964827 | 377848438952 | - | - | 4358 | 34061.44 | 11.67 | 0.03 | - |
| | RCM-GL | 2201 | 1332260636 | 2.21 | 0 | 4346 | 31978.36 | 37.83 | 0.12 | 1.065 |
| | hGPHH-GL | 2315 | 1495703784 | 2.37 | 0 | 4370 | 32392.77 | 15.64 | 0.05 | 1.051 |
| | FNCHC | **2189** | 1562324154 | 1636.39 | 139.75 | 4365 | 32511.67 | 30.43 | 0.09 | 0.997 |
| | Snay | 13292 | **950057407** | 1049.95 | 26.09 | **4341** | **30016.38** | 205.23 | 0.68 | 1.096 |
| | Sloan | 42459 | 1007346790 | 22.79 | 16.98 | 4365 | 30453.62 | 41.81 | 0.14 | 1.118 |
| | NSloan | 56571 | 1379371308 | **1.90** | 17.07 | 4362 | 32913.96 | 59.64 | 0.18 | 1.035 |
| | Sloan-MGPS | 60821 | 1007161537 | 166.78 | 17.07 | 4361 | 31589.23 | 77.27 | 0.24 | 1.073 |
| | MPG | 77152 | 982945106 | 6.02 | 17.24 | 4360 | 30154.23 | 60.63 | 0.22 | **1.129** |

seen as the potential state-of-the-art low-cost heuristics for bandwidth and profile reductions [Chagas e Gonzaga de Oliveira, 2015; Bernardes e Gonzaga de Oliveira, 2015; Gonzaga de Oliveira e Chagas, 2015]. Additionally, the adjacent exchange methods [Reid e Scott, 2002] were also evaluated in conjunction with several heuristics. The results of the implementations of these heuristics for bandwidth and profile reductions reported here show their expected value accordingly to the existing publications in the area. In particular, the results obtained here showed that the adjacent exchange methods, in spite of reducing the profile of the instances, are not powerful enough for reducing computational costs of the JPCGM.

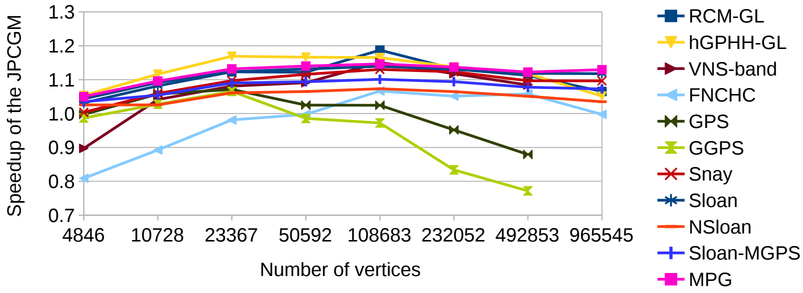
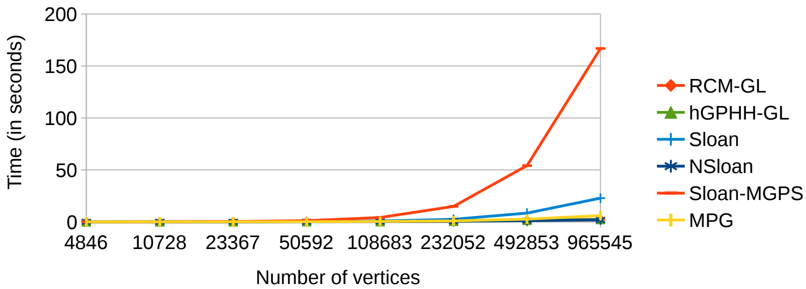
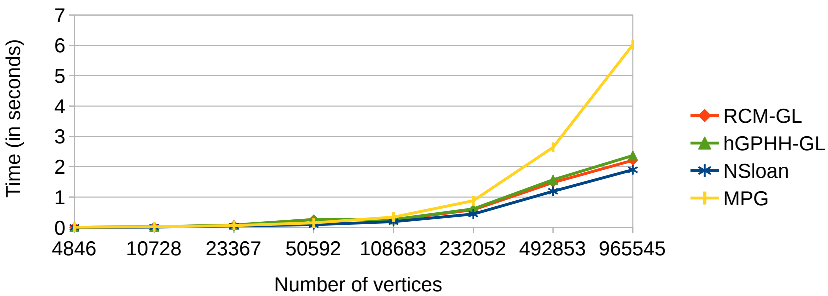



Figure 2: Speedup of the JPCGM obtained using 11 heuristics for bandwidth and profile reductions (tests shown in Tables 1 and 2) to matrices that belong to linear systems that originate from the discretization of the heat conduction equation by finite volumes.




(a)




(b)

Figure 3: Execution times, in seconds, of [(a) 6; (b) 4] reordering algorithms.

In experiments using linear systems from the discretization of the heat conduction equations by finite volumes composed of instances comprised of almost 1,000,000 unknowns, the MPG heuristic [Medeiros et al., 1993] performed best in the two largest instances tested when applied to reduce the computational cost of the JPCGM. Moreover, the hGPHH-GL [Koohestani e Poli, 2011] (based on the RCM method) and RCM-GL [George e Liu, 1981] methods were the easiest heuristic to implement. These two low-cost algorithms obtained reasonable results in the eight instances tested. In particular, the hGPHH-GL [Koohestani e Poli, 2011] and RCM-GL [George e Liu, 1981] methods achieved the best results when applied to the six smallest instances tested.

On the other hand, one can attest that there is no a unique and optimal method that supersedes all the other heuristics in all instances: the choice of a heuristic for bandwidth or profile reduction is highly dependent on the structure of the instance. Thus, the computational experiment described in this paper does not assure dominance of an algorithm over the others with respect to the reordering of graph vertices. In spite of this, evidence from the experiments described reveals that a high-cost heuristic that significantly reduces the bandwidth or profile of the matrix may not be better than a low-cost method that reduces reasonably the bandwidth or profile of the matrix, such as is the case of the MPG [Medeiros et al., 1993], Sloan [1989], NSloan [Kumfert e Pothen, 1997], Sloan-MGPS [Reid e Scott, 1999], RCM-GL [George e Liu, 1981], and hGPHH-GL [Koohestani e Poli, 2011] heuristics, especially when linear systems are solved using the JPCGM.

In future works, we plan to implement and analyze the following preconditioners: Algebraic Multigrid, ILUT, Successive Over-Relaxation (SOR), Symmetric SOR, and Gauss-Seidel. In detail, we expect to investigate the efficiency of reordering algorithms in conjunction with the computation of incomplete or approximate factorization based preconditioners as well approximate inverse preconditioners. These preconditioners shall be employed as preconditioners of the Conjugate Gradient Method and the Generalized Minimal Residual (GMRES) method [Saad e Schultz, 1986] in order to analyze their computational performance in conjunction with heuristics for bandwidth or profile reduction. Parallel strategies of the algorithms mentioned are also intended to be studied.

A 512-bit extended precision was employed in this work. Certainly, this reduces rounding errors, but it also increases the processing time by a large factor and it may not be performed when solving many real-world applications. We plan to examine what occurs in double-precision arithmetic in future studies.

**Acknowledgments**

This work was undertaken with the support of the FAPEMIG - Fundação de Amparo à Pesquisa do Estado de Minas Gerais (Minas Gerais Research Support Foundation), Minas Gerais, Brazil. We would like to thank Prof. Dr. Dragan Urosevic, from the Mathematical Institute SANU, for sending us the VNS-band executable programs. We would like to thank Prof. Dr. Fei Xiao, from Beepi, for sending us the source code of the FNCHC heuristic. In addition, we would like to thank the reviewers for their valuable comments and suggestions.

**References**

Bernardes, J. A. B. e Gonzaga de Oliveira, S. L. (2015). A systematic review of heuristics for profile reduction of symmetric matrices. *Procedia Computer Science (International Conference on Computational Science, ICCS 2015)*, 51:221–230.

Botafogo, R. A. (1993). Cluster analysis for hypertext systems. In Korfhage, R., Rasmussen, E., e Willett, P., editors, *Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, p. 116–125, Pittsburgh, PA. ACM.

Burgess, D. A. e Giles, M. (1997). Renumbering unstructured grids to improve the performance of codes on hierarchial memory machines. *Advances in Engineering Software*, 28(3):189–201.

Burgess, I. W. e Lai, P. K. F. (1986). A new node renumbering algorithm for bandwidth reduction. *International Journal for Numerical Methods in Engineering*, 23:1693–1704.

Chagas, G. O. e Gonzaga de Oliveira, S. L. (2015). Metaheuristic-based heuristics for symmetric-matrix bandwidth reduction: A systematic review. *Procedia Computer Science (International Conference on Computational Science, ICCS 2015)*, 51:211–220.

Das, R., Mavriplis, D. J., Saltz, J. H., Gupta, S. K., e Ponnusamy, R. (1992). The design and implementation of a parallel unstructured euler solver using software primitives. Technical Report AD-A249 437, Institute for Computer Applications in Science and Engineering - NASA, Virgina, USA.

Duff, I. S. e Meurant, G. A. (1989). The effect of ordering on preconditioned conjugate gradients. *BIT Numerical Mathematics*, 29(4):635–657.

Esposito, A., Catalano, M. S. F., Malucelli, F., e Tarricone, L. (1998). A new matrix bandwidth reduction algorithm. *Operations Research Letters*, 23:99–107.

Felippa, C. A. (1975). Solution of linear equations with skyline-stored symmetric matrix. *Computers and Structures*, 5(1):13–29.

Fenves, S. J. e Law, K. H. (1983). A two-step approach to finite element ordering. *International Journal for Numerical Methods in Engineering*, 19(6):891–911.

George, A. e Liu, J. W. (1981). *Computer solution of large sparse positive definite systems*. Prentice-Hall, Englewood Cliffs.

George, A. e Liu, J. W. H. (1979). An implementation of a pseudoperipheral node finder. *ACM Transactions on Mathematical Software*, 5(3):284–295.

Gibbs, N. E., Poole, W. G., e Stockmeyer, P. K. (1976). An algorithm for reducing the bandwidth and profile of a sparse matrix. *SIAM Journal on Numerical Analysis*, 13(2):236–250.

Gonzaga de Oliveira, S. L. e Chagas, G. O. (2015). A systematic review of heuristics for symmetric-matrix bandwidth reduction: methods not based on metaheuristics. In *The XLVII Brazilian Symposium of Operational Research*, Ipojuca-PE, Brazil. Sobrapo.

Gonzaga de Oliveira, S. L., Oliveira, F. S., e Chagas, G. O. (2015). A novel approach to the weighted laplacian formulation applied to 2-d delaunay triangulations. In *Computational Science and Its Applications – ICCSA 2015*, volume 9155 of *Lecture Notes in Computer Science*, p. 502–515. Springer International Publishing.

Hestenes, M. R. e Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(36):409–436.

Hu, Y. e Scott, J. A. (2001). A multilevel algorithm for wavefront reduction. *SIAM Journal on Scientific Computing*, 23(4):1352–1375.

Johnson, D. (2002). A Theoretician's Guide to the Experimental Analysis of Algorithms. In Goldwasser, M., Johnson, D. S., e McGeoch, C. C., editors, *Proceedings of the 5th and 6th DIMACS Implementation Challenges*, Providence. American Mathematical Society.

Karp, R. (1993). Mapping the genome: Some combinatorial problems arising in molecular biology. In Kosaraju, R., Johnson, D., e Aggarwal, A., editors, *STOC'93 Proceedings of the twenty-fifth Annual ACM Symposium on Theory of Computing*, p. 278–285, San Diego, CA. ACM.

Kaveh, A. e Sharafi, P. (2012). Ordering for bandwidth and profile minimization problems via charged system search algorithm. *IJST Transactions of Civil Engineering*, 36(2):39–52.

Kendall, D. G. (1969). Incidence matrices, interval graphs and seriation in archaeology. *Pacific Journal of Mathematics*, 28(3):565–570.

Koohestani, B. e Poli, R. (2011). A hyper-heuristic approach to evolving algorithms for bandwidth reduction based on genetic programming. In *Research and Development in Intelligent Systems XXVIII*, p. 93–106, London, UK. Springer London.

Kumfert, G. e Pothen, A. (1997). Two improved algorithms for envelope and wavefront reduction. Technical Report NAS1-19480, Institute for Computer Applications in Science and Engineering NASA Langley Research Center, Hampton, Virginia.

Lanczos, C. (1952). Solutions of systems of linear equations by minimized iterations. *Journal of research of the National Bureau of Standards*, 49(1):33–53.

Lewis, J. G. (1982). Implementation of the gibbs-poole-stockmeyer and gibbs-king algorithms. *ACM Trans. Math. Softw.*, 8(2):180–189.

Lim, A., Rodrigues, B., e Xiao, F. (2003). A new node centroid algorithm for bandwidth minimization. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, p. 1544–1545, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lim, A., Rodrigues, B., e Xiao, F. (2004). A centroid-based approach to solve the bandwidth minimization problem. In *Proceedings of the Hawaii International Conference on System Sciences*, volume 37, p. 1203–1208, Big Island, HI, United States. Institute of Electrical and Electronics Engineers Inc.

Lim, A., Rodrigues, B., e Xiao, F. (2007). A fast algorithm for bandwidth minimization. *International Journal on Artificial Intelligence Tools*, 3:537–544.

Lin, Y. X. e Yuan, J. J. (1994). Profile minimization problem for matrices and graphs. *Acta Mathematicae Applicatae Sinica*, 10(1):107–122.

Medeiros, S. R. P., Pimenta, P. M., e Goldenberg, P. (1993). Algorithm for profile and wavefront reduction of sparse matrices with a symmetric structure. *Engineering computations*, 10(3):257–266.

Mladenovic, N., Urosevic, D., Pérez-Brito, D., e García-González, C. G. (2010). Variable neighbourhood search for bandwidth reduction. *European Journal of Operational Research*, 200: 14–27.

Papadimitriou, C. H. (1976). The NP-completeness of bandwidth minimization problem. *Computing Journal*, 16:177–192.

Reid, J. K. e Scott, J. A. (1999). Ordering symmetric sparse matrices for small profile and wavefront. *International Journal for Numerical Methods in Engineering*, 45(12):1737–1755.

Reid, J. K. e Scott, J. A. (2002). Implementing Hager's exchange methods for matrix profile reduction. *ACM Transactions on Mathematical Software*, 28(4):377–391.

Saad, Y. e Schultz, M. H. (1986). Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7:856–869.

Sloan, S. W. (1989). A Fortran program for profile and wavefront reduction. *International Journal for Numerical Methods in Engineering*, 28(11):2651–2679.

Snay, R. A. (1976). Reducing the profile of sparse symmetric matrices. *Bulletin Geodésique*, 50 (4):341–352.

STFC Rutherford Appleton Laboratory (1963-2016). The Science and Technology Facilities Council. HSL. A collection of Fortran codes for large scale scientific computation. `http://www.hsl.rl.ac.uk`. Accessed: December, 2015.

Torres-Jimenez, J., Izquierdo-Marquez, I., Garcia-Robledo, A., Gonzalez-Gomez, A., Bernal, J., e Kacker, R. N. (2015). A dual representation simulated annealing algorithm for the bandwidth minimization problem on graphs. *Information Sciences*, 33:33–49.

Wang, Q., Guo, Y. C., e Shi, X. W. (2009). A generalized GPS algorithm for reducing the bandwidth and profile of a sparse matrix. *Progress in Electromagnetics Research*, 90:121–136.