A Multi-centrality Heuristic for the Bandwidth Reduction Problem

João Maues[1], Israel Mendonça[3], Glauco Amorim[2], Sanderson L. Gonzaga de Oliveira[4], Ana Isabel Pereira[5,6], Diego Brandão[2(✉)], and Pedro Henrique González[1]

[1] System Engineering and Computer Science Program, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil
{jvmaues,pegonzalez}@cos.ufrj.br

[2] Department of Computer Science, Federal Center for Technological Education of Rio de Janeiro, Rio de Janeiro, Brazil
{glauco.amorim,diego.brandao}@cefet-rj.br

[3] School of Advanced Science, Kumamoto University, Kumamoto, Japan
israel@cs.kumamoto-u.ac.jp

[4] Department of Science and Technology, Federal University of São Paulo, São Paulo, Brazil
sanderson.oliveira@unifesp.br

[5] Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, 5300-253 Bragança, Portugal
apereira@ipb.pt

[6] Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, 5300-253 Bragança, Portugal

Abstract. The Bandwidth Minimization Problem for Sparse Matrices is a well-known NP-Hard problem critical in numerous significant scientific applications. The Cuthill-McKee algorithm, a heuristic based on degree centrality for minimizing bandwidth, is a common solution approach. One can integrate other centrality measures into the Cuthill-McKee method or similar algorithms. This work explores the impact of utilizing these diverse centrality measures on the performance of the Cuthill-McKee heuristic. It introduces a novel multi-centrality constructive algorithm designed as an alternative for practical applications emphasizing efficient execution for large linear systems. The results demonstrate clear advantages of considering multiple centrality measures over solely degree centrality. This approach notably enhances the heuristic's effectiveness, offering significant improvements in solving complex bandwidth minimization problems.

Keywords: Bandwidth Reduction Problem · Graph Centrality · Cuthill-Mckee method

1 Introduction

Solving large-scale sparse linear systems, represented by $Ax = b$, where A is an $n \times n$ sparse matrix, x is the unknown vector of dimension n, and b is a known

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
O. Gervasi et al. (Eds.): ICCSA 2024, LNCS 14813, pp. 3–15, 2024.
https://doi.org/10.1007/978-3-031-64605-8_1

vector of dimension n, plays a crucial role in various scientific and engineering fields. Many simulations in science and engineering require solving extensive sparse linear systems, often representing the highest computational cost. For instance, as discussed by Kaveh [16], in structural mechanics problems, linear systems of real-world problems account for 30% to 50% of computational costs, reaching up to 80% in comparison to nonlinear systems in structural optimization problems.

The challenge of the Bandwidth Minimization Problem (BMP) traces back to the 1950s when structural engineers initially examined steel frameworks using computers [3]. By concentrating all non-zero elements within a slim band around the main diagonal and employing a reordering matrix, efficiency gains in operations like inversion and determinants can be achieved, resulting in time savings [3]. Bandwidth reduction heuristics aim to place non-zero coefficients of a sparse matrix as close as possible to the main diagonal. Figure 1 illustrates an example application of a bandwidth reduction heuristic to a matrix.

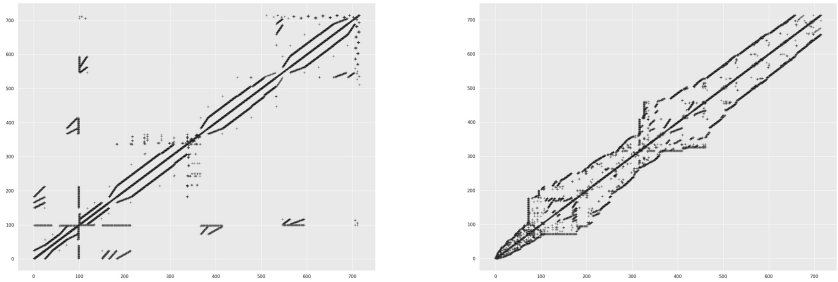


Fig. 1. Example of how a sparse matrix instance behaves before and after bandwidth reduction using the Cuthill-McKee algorithm for such reduction.

The Cuthill-McKee method [4] stands out as the most classical heuristic for bandwidth reduction. Another highly effective algorithm to solve this problem and considered by many researchers to be the state of the art of PMB is DRSA [21]. However, current metaheuristics used to optimize this problem face a computational cost challenge during execution.

Many heuristics and metaheuristics have been developed to achieve higher bandwidth reductions [2]. Recent research has not found in the literature in-depth studies evaluating the impact of combinations of various centrality measures. This article studies different centralities on bandwidth reduction in sparsely populated matrices commonly used in the literature. It presents a new multi-centrality constructive algorithm designed as an alternative for practical applications with efficient execution for large linear systems. Until the present work, we found no in-depth studies in the literature evaluating the impact of using combinations of different centrality measures to guide the choice of vertex labeling in the BMP. Therefore, this work aims to carry out a comprehensive

study on the impact of different centralities on bandwidth reduction in sparse matrices commonly used in the literature. Furthermore, it proposes a multi-centrality heuristic designed as an alternative for practical applications emphasizing efficient execution to help solve large linear systems.

We organized the remaining structure of this article as follows. Section 2 presents a detailed description of BMP and a formulation. Related works are presented in Sect. 3, followed by the description of the proposed method and its components in Sect. 4. Section 5 describes computational experiments and their results. Finally, Sect. 6 presents the conclusions and future works.

2 Problem Description and Mathematical Formulation

We mathematically formulate the problem using as a base the formulation that can be found in [6]. Let a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, let $n = |V|$. A labeling f of G assigns integers $1, 2, ..., n$ to the vertices of G. The bandwidth of a graph G with labeling $f : V \rightarrow \mathbb{N}$ is defined as:

$$B_w(f) = \max_{\{u,v\} \in E} |f(u) - f(v)| \tag{1}$$

The bandwidth reduction problem aims to find a labeling f that minimizes $B_w(f)$. Suppose $A = [a_{ij}]$ represents the incidence matrix of a graph V. The bandwidth reduction problem aims to identify a permutation of rows and columns that maintains all non-zero elements of A within a band as close as possible to the main diagonal. This understanding of the problem emerges in practical applications aimed at solving non-singular systems of linear algebraic equations in the form $Ax = b$ [1].

3 Relevant Related Work

The literature review uses Scopus, a bibliographic database specializing in academic journal articles, to explore the BMP with the following search string: "bandwidth reduction of sparse matrix". The problem studied in this article has two relevant factors as its objective: the highest reduction in bandwidth in a favorable execution time. Focusing on these characteristics, Table 1 aims to highlight the most significant studies in the literature that address BMP, identifying the main methods used as proposed solutions to this challenge.

Table 1. Most promissing heuristic methods for bandwidth reduction.

Heuristic	Year	Bandwidth reduction	Cost
Cuthill-McKee [4]	1969	low	low
RCM [7]	1971	low	low
FNCHC [18]	2007	reasonable	high
VNS-band [19]	2010	high	high
KP-band [17]	2011	low	low
DRSA [21]	2015	high	high
RBFS-GL [9]	2018	low	low
ACHH [11,12,14]	2020	high	low
FNCHC+ [8,10]	2022	high	reasonable

A study focused on low computational cost heuristics conducted in [13] concluded that the Variable Neighborhood Search metaheuristic in [19] demonstrated better effectiveness in bandwidth reduction for instances from the SuiteSparse matrix collection [5], surpassing low-cost heuristics. FNCHC stood out as the second-best in this set but achieved the best results in 12 instances in a smaller set. Although the VNS-band achieved superior results, it incurred a considerably higher computational cost.

Finally, the latest relevant work related to the BMP was carried out in [14], presenting an ant colony hyperheuristic approach as a new and promising method for optimizing bandwidth in large-scale matrices. The study compared its effectiveness with low-cost heuristics. Results based on a wide range of sparse matrix instances indicate that the proposed approach has the potential to be more effective and efficient than existing low-cost heuristics in optimizing bandwidth for large-scale matrices.

Another relevant point to note in Table 1 is the case of the state of the art, the DRSA [21], but its execution times are higher than VNS-band [8,10,14]. Despite having a high bandwidth reduction rate, in practice, it does not make sense to use it, as the resolution of linear systems without such methods is usually faster.

4 Multi-centrality Heuristic Algorithm

This section presents the Multi-centrality Heuristic method (MCH), which combines a multi-start constructive heuristic with a multi-centrality approach in choosing vertex labeling for the bandwidth reduction problem. Section 4.1 introduces the Multi-centrality Constructive Algorithm. The second subsection explains how we selected the centralities to contemplate the heuristic proposal.

4.1 Multi-centrality Constructive Algorithm

The proposed Constructive Heuristic aims to generate a high-quality solution computationally efficiently through iterative restarts to explore various solutions

effectively in the quest for an optimal solution. In Algorithm 1.1, we construct a vertex sequence q using depth-first search to obtain a high-quality solution and label them as follows: 1, 2, ..., n. Following the labeling order according to the centrality, from the lowest to the highest at the current neighborhood level. The idea is based on an observation that adjacent vertices should have close labels in a good solution (with a small maximum bandwidth), as explained in the constructive approach proposed in [19]. In this specific algorithm, we add extra relevance by ordering vertices according to their centrality. Additionally, our procedure should allow us to generate different solutions in different restarts. To achieve this, we introduced randomness to select the first vertex.

Algorithm 1.1: Multi-centrality Constructive Algorithm

```
   Data: Graph A, ArrayCentrality
 1 centrality ← randomChoice(ArrayCentrality);
 2 for i ← 1 to |V| do
 3 |   mark[i] ← false;
 4 end
 5 k ← RandomInt(1, |V|);
 6 mark[k] ← true;
 7 q[1] ← k; ql ← 1; f[k] ← 1; l ← 1;
 8 while l < |V| do
 9 |   r ← 0;
10 |   for il ← 1 to ql do
11 |   |   i ← q[il];
12 |   |   for jl ← 1 to |A[i]| do
13 |   |   |   j ← a[i, jl];
14 |   |   |   if not mark[j] then
15 |   |   |   |   r ← r + 1; s[r] ← j;
16 |   |   |   end
17 |   |   end
18 |   end
19 |   q ← sorted(q, lambda(x) : centrality[x]);
20 |   j* ← 0;
21 |   for jl ← 1 to ql do
22 |   |   j ← q[j*]; l ← l + 1; f[j] ← l; j* ← j* + 1;
23 |   end
24 |   q ← s; ql ← r;
25 end
26 return f
```

Vertices for breadth-first search are placed in levels as follows: the first level contains only vertex k (randomly selected), and label one is assigned to it ($f(k) \leftarrow 1$); vertices adjacent to k are in the second level ordered in decreasing order, with labels 2, 3,... The third level contains vertices not included in previous levels but connected to vertices in the second level, and so on, with the labeling order always following the centrality values of each vertex.

Suppose there are n_i vertices in level i. The method randomly chooses a centrality to order the vertices q. The labeling process then takes place by the sequence of ordered vertices.

In addition, Algorithm 1.2 was implemented with a multi-start control feature to execute Algorithm 1.1. It takes as input a graph, the maximum number of iterations max_{iter}, and a list of centrality dictionaries, where each dictionary is constructed having a vertex linked to a centrality value at the vertex in question. Then, it runs the multi-start algorithm max_{iter} times, utilizing the calculated list of centrality as a parameter. The algorithm's objective is to perform multiple initializations (multi-start) of the main algorithm, possibly varying the centrality measure used, to explore different initial solutions and improve the chances of finding a high-quality solution.

Algorithm 1.2: Multi-start Centrality Heuristic

```
   Data: Graph A, int max_iter, List ArrayCentrality
1  f_sol ← inf;
2  for k ← 1 to max_iter do
3  |  f ← ConstructiveCentralities(A, ArrayCentrality)
4  |  if f < f_sol then
5  |  |  f_sol ← f
6  |  end
7  end
8  return f_sol
```

This algorithm combines the flexibility of multiple random calls for the initial choice of the first vertex to be labeled by the algorithm with the diversification provided by selecting different centrality measures. The idea is that diversified centralities can guide the construction of the solution in various ways, exploring several aspects of the graph's structure.

The combination with the constructive algorithm proposed in [19] aims to find good-quality solutions while maintaining an acceptable runtime for practical use in sparse linear systems resolutions.

4.2 Selecting Centralities

Before conducting the experiments, it was essential to identify the centralities most pertinent to bandwidth reduction. Thus, the methodology adjusted the Cuthill-McKee algorithm, detailed in [4], by modifying the criteria for selecting the labeling order based on various centralities from existing literature. It aimed to assess the impact of each centrality on the bandwidth reduction.

We implemented each selected centrality with two labeling orders: one from the lowest to the highest degree (Smallest) and another from the highest to the lowest degree (Largest). Based on the highest reduction, Table 2 outlines the implementation choices for each centrality.

Table 2. Bandwidth Reductions by Centrality.

Node	Centrality	Reduction(%)
A	Smallest Closeness	52.90
B	Smallest Degree	51.96
C	Smallest Katz	46.84
D	Smallest Harmonic	50.58
E	Largest Eigenvector	50.09
F	Smallest Betweenness	51.03

Table 3. Heatmap of the correlation between centralities.

Correlations	A	B	C	D	E	F
A	1					
B	0.97	1				
C	0.89	0.88	1			
D	0.93	0.91	0.94	1		
E	0.92	0.91	0.94	1	1	
F	0.96	0.97	0.86	0.89	0.89	1

We analyzed the relationship between bandwidth reductions among the algorithms. The correlation was calculated for each pair of results obtained from the implemented algorithms, as shown in Table 3. Typically, a correlation exceeding 0.7 is considered high, but in this study, we focused on correlations higher than or equal to 0.8.

To investigate these correlations, we represent a graph, as illustrated in Fig. 2. In this graph, each vertex V represents an implemented centrality, and an edge E is established between two vertices if the correlation between their corresponding values exceeds the desired correlation threshold, as shown in Table 4. After formulating the problem for the specified correlations, we approach the minimum dominating set problem as described in [22] and solve a binary integer programming problem, where the chosen vertices are the solution to the problem in question.

The highest reduction potential emerged from a correlation of 0.9, involving vertices representing the highest eigenvector centrality and the lowest degree centrality. Consequently, opting for other combinations was deemed unnecessary to avoid redundancy and ensure the implementation of a high-quality solution to the problem.

5 Computational Experiments

We divided this section into three distinct subsections. The introductory subsection outlines the details of the implementation and tuning parameters. The last

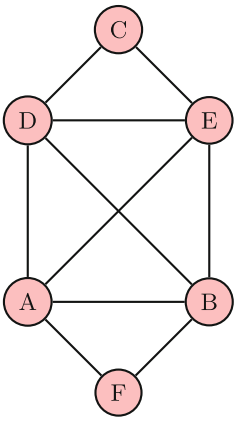


Fig. 2. Graph G with nodes V

Table 4. Correlations edge

Edge	Correlation
AB	0.97
AD	0.93
AE	0.92
AF	0.96
BD	0.91
BE	0.91
BF	0.97
CD	0.94
CE	0.94
DE	1.00

two sections comprehensively analyze the principal considerations of the algorithm proposed in this work, establishing insightful comparisons with existing literature.

5.1 Implementation Details and Parameters Tuning

We implemented the proposed method in Python 3.9.12 and executed it ten times for each instance on an Intel Core 11th generation processor, i7, with a frequency of 2.80 GHz and 16.0 GB of RAM. The matrices used in this work for comparison were the same as those used by Mladenovid et al. [19]. We used a set of 43 instances from the SuiteSparse Matrix Collection [5] to assess the effectiveness of each implemented algorithm in bandwidth reduction.

Before experimenting, it was essential to establish a maximum iteration parameter first. This parameter could not be too low as it would test a few restart solutions. Furthermore, it could not be too high as it would increase the computational cost of the method. After evaluating a few possibilities, we established $max_{iter} = 100$.

5.2 MCH1 × MCH2 × MCH3

After choosing the parameters and centralities of the algorithm, we implemented three versions of MCH to understand which centralities would perform best, as presented and discussed in this subsection. Using the proposed methodology of constructive multi-start by centralities presented in this study, the initial versions developed were as follows. The constructive method Version 1 (MCH1) uses degree centrality in its implementation. In contrast, the constructive method Version 2 (MCH2) employs the centrality of eigenvectors as the basis for your vertex labeling decision. The Version 3 (MCH3) construction method implements a random call approach with both centralities.

Thus, we executed each version 10 times for each instance; initially, Version 3 showed apparent superiority over the other versions, as shown in Table 3. Then, to evaluate the statistical significance of the differences in the values of the best solutions between the three implemented versions, a two-to-two hypothesis test was performed between the best bandwidth reduction values for each version. Given that the data did not fit a normal distribution and the methodology paired the measurements from the experiments, the non-parametric Wilcoxon Signed Rank Test was used [15,20]. This test allows the rejection of the null hypothesis at a significance level denoted by θ, with a probability of $(1 - \theta) \times 100\%$. This study considered a significance level of $\theta = 0.05$. The formulated hypotheses are as follows:

- Null Hypothesis (H0): The median difference is zero.
- Alternative Hypothesis (H1): The median difference is not zero.

The null hypothesis requires no differences in the values of the best solutions between each implemented version and the absence thereof. We expect to observe positive and negative differences if the null hypothesis is valid. However, the methodology rejected the null hypothesis because the p-value is 0.005, lower than θ. It indicates more positive differences than negative ones, corroborating the conclusion that Version 3 outperforms the other versions.

5.3 Multi-centrality Heuristic × Methods Published in the Literature

After conducting the initial experiments and the execution time was favorable, we sought to measure algorithms in the literature that had a time of less than 0.5 sec to compare the results obtained by the proposed methodology with low-cost heuristics. In addition to these initially implemented algorithms, the constructive component of the VNS-band metaheuristic proposed in [19] was implemented for comparison purposes.

The results presented in Table 5 highlight the execution times associated with the heuristics developed in this work and other important ones in the literature. Although Cuthill-McKee presents the lowest average execution time (0.03 s), suggesting effectiveness in only one specific instance, it obtained the worst overall result in bandwidth reductions, as shown in Table 5.

Table 5. Comparison of times between heuristics in seconds.

Heuristic	Average time(s)
Cuthill-McKee	0.03
VNS Constructive	0.16
MCH	0.29
GPS	31.80
FNCHC	146.20
VNS-band	157.33

The VNS constructive component, with a bandwidth value of 11, still presents relevant performance and an average execution time of 0.16, indicating efficiency in finding solutions in a favorable time in performance and demonstrating better consistency in instances with a high value of N (nodes), as shown in Table 5.

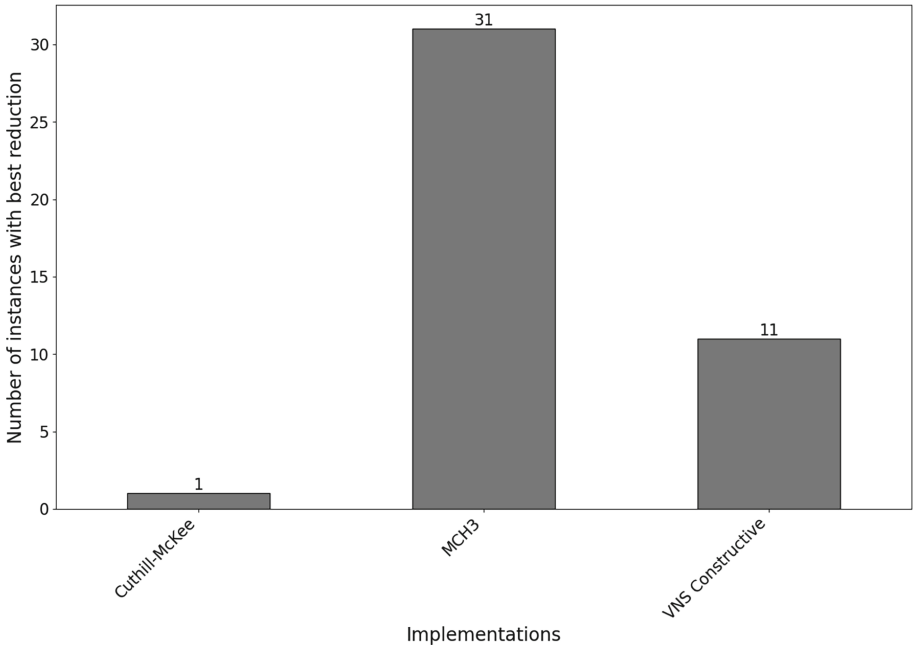


Fig. 3. Number of instances with best reductions by Heuristic.

The VNS-band, FNCHC, and GPS heuristics have higher execution times, with VNS-band and FNCHC standing out with 157.33 and 146.20 s, respectively. Despite having a high power of bandwidth reduction, these approaches may be less efficient when the resolution time of a linear system is shorter than the execution time of such heuristics.

Versions 1, 2, and 3 showed a low variation in the execution times, with values of 0.41, 0.42, and 0.29, respectively. From a practical point of view, they proved to be heuristics that behave more consistently and are competitive in terms of efficiency.

In summary, analysis of execution times reveals performance variability between heuristics. While Cuthill-McKee is good in a single case, other heuristics, such as Version 3 and Constructive VNS, demonstrate broader consistency across different scenarios. Heuristics with higher times, such as VNS-band, FNCHC, and GPS, may be less suitable when the resolution time of a linear system without reduction is already shorter than the execution time of these heuristics.

6 Conclusions and Future Works

The research presented in this work focuses on developing and evaluating a new multi-centrality constructive algorithm for bandwidth minimization in sparse matrices. By introducing the use of centrality measures in the algorithm, the study aims to increase the efficiency and effectiveness of bandwidth reduction techniques for large linear systems through computational experiments and comparisons with traditional heuristics such as Cuthill-McKee.

Our computational experiments reveal that strategies implemented within a relatively simple construction procedure can outperform known algorithms used in practice for bandwidth reduction, such as Cuthill-McKee. In particular, the reset procedure based on random centrality choices confirms the benefit of encouraging diversification of this study. Thus, the additional diversification results in satisfactory quality solutions. One can use the approach in practice for the bandwidth reduction problem, combined with the fact that constructive heuristics consume considerably less computer time.

Based on the results presented in this work, a possible direction of study is the implementation of hybrid versions of the VNS-band. An additional approach would be to implement a machine learning classification model in the instance preprocessing, allowing for analysis of the behavior of reductions with different centrality strategies per instance class. This comprehensive approach not only enriches the understanding of the performance of different centralities but also offers valuable insights by exploring the nuances of the approach proposed in this work.

Acknowledgement. The authors would like to thank the following Brazilian Agencies: CAPES and the National Council for Scientific and Technological Development - CNPq. DB thanks Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) for the financial support through the grants E-26/210.798/2024. Additionally, the authors are grateful to the Foundation for Science and Technology (FCT, Portugal) for financial support through national funds FCT/MCTES (PIDDAC) to CeDRI, UIDB/05757/2020 (DOI: 10.54499/UIDB/05757/2020) and UIDP/05757/2020 (DOI: 10.54499/UIDB/05757/2020) and SusTEC, LA/P/0007/2020 (DOI: 10.54499/LA/P/0007/2020).

References

1. Marti, R., Laguna, M., Glover, F., Campos, V.: Reducing the bandwidth of a sparse matrix with tabu search. Eur. J. Oper. Res. **135**(2), 450–459 (2001). https://doi.org/10.1016/S0377-2217(00)00325-8. https://www.sciencedirect.com/science/article/pii/S0377221700003258. Financial Modelling
2. Chagas, G.O., Gonzaga de Oliveira, S.L.: Metaheuristic-based heuristics for symmetric-matrix bandwidth reduction: a systematic review. Procedia Comput. Sci. **51**, 211–220 (2015). https://doi.org/10.1016/j.procs.2015.05.229. https://www.sciencedirect.com/science/article/pii/S1877050915010376. International Conference on Computational Science, ICCS 2015

3. Chinn, P.Z., Chvátalová, J., Dewdney, A.K., Gibbs, N.E.: The bandwidth problem for graphs and matrices—a survey. J. Graph Theory **6**(3), 223–254 (1982). https://doi.org/10.1002/jgt.3190060302. https://onlinelibrary.wiley.com/doi/abs/10.1002/jgt.3190060302
4. Cuthill, E., McKee, J.: Reducing the bandwidth of sparse symmetric matrices. In: Proceedings of the 1969 24th National Conference (1969). https://doi.org/10.1145/800195.805928
5. Davis, T.A., Hu, Y.: The University of Florida sparse matrix collection. ACM Trans. Math. Softw. **38**(1), 1–25 (2011)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences), 1st edn. W. H. Freeman (1979). http://www.amazon.com/Computers-Intractability-NP-Completeness-Mathematical-Sciences/dp/0716710455
7. George, J.A.: Computer implementation of the finite element method. Ph.D. thesis, Computer Science Department, Stanford University, CA, USA (1971)
8. Gonzaga de Oliveira, S.L.: An evaluation of heuristic methods for the bandwidth reduction of large-scale graphs. Revista Pesquisa Operacional **43**(e268255), 1–22 (2023)
9. Gonzaga de Oliveira, S.L., Bernardes, J.A.B., Chagas, G.O.: An evaluation of reordering algorithms to reduce the computational cost of the incomplete Cholesky-conjugate gradient method. Comput. Appl. Math. **37**, 2965–3004 (2018)
10. Gonzaga de Oliveira, S.L., Carvalho, C.: Metaheuristic algorithms for the bandwidth reduction of large-scale matrices. J. Comb. Optim. **43**, 727–784 (2022)
11. Gonzaga de Oliveira, S.L., Silva, L.M.: Evolving reordering algorithms using an ant colony hyperheuristic approach for accelerating the convergence of the ICCG method. Eng. Comput. **36**, 1857–1873 (2020)
12. Gonzaga de Oliveira, S.L., Silva, L.M.: Low-cost heuristics for matrix bandwidth reduction combined with a Hill-Climbing strategy. Rairo - Oper. Res. **55**(4), 2247–2264 (2021)
13. Gonzaga de Oliveira, S.L., Bernardes, J.A.B., Chagas, G.O.: An evaluation of low-cost heuristics for matrix bandwidth and profile reductions. Comput. Appl. Math. **37**(2), 1412–1471 (2018). https://doi.org/10.1007/s40314-016-0394-9
14. Gonzaga de Oliveira, S., Silva, L.: An ant colony hyperheuristic approach for matrix bandwidth reduction. Appl. Soft Comput. **94**, 106434 (2020). https://doi.org/10.1016/j.asoc.2020.106434. https://www.sciencedirect.com/science/article/pii/S1568494620303744
15. Hettmansperger, T.P., McKean, J.W.: Robust Nonparametric Statistical Methods. CRC Press (2010)
16. Kaveh, A.: Structural Mechanics: Graph and Matrix Methods. Research Studies Press (2004)
17. Koohestani, B., Poli, R.: A hyper-heuristic approach to evolving algorithms for bandwidth reduction based on genetic programming. In: Bramer, M., Petridis, M., Nolle, L. (eds.) Research and Development in Intelligent Systems XXVIII, pp. 93–106. Springer, London (2011). https://doi.org/10.1007/978-1-4471-2318-7_7
18. Lim, A., Rodrigues, B., Xiao, F.: Int. J. Artif. Intell. Tools **16**, 537–544 (2007)
19. Mladenovic, N., Urosevic, D., Pérez-Brito, D., García-González, C.: Variable neighbourhood search for bandwidth reduction. Eur. J. Oper. Res. **200**, 14–27 (2010). https://doi.org/10.1016/j.ejor.2008.12.015

20. Pratt, J.W.: Remarks on zeros and ties in the Wilcoxon signed rank procedures. J. Am. Stat. Assoc. **54**, 655–667 (1959). https://api.semanticscholar.org/CorpusID:120225998
21. 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. Inf. Sci. **303**, 33–49 (2015)
22. West, D.B.: Introduction to Graph Theory, 2 edn. Prentice Hall (2000)