Pesquisa Operacional (2023) 43: e268255 p.1-22
doi: 10.1590/0101-7438.2023.043.00268255

Printed version ISSN 0101-7438 / Online version ISSN 1678-5142
www.scielo.br/pope
ARTICLES

AN EVALUATION OF HEURISTIC METHODS FOR THE BANDWIDTH REDUCTION OF LARGE-SCALE GRAPHS

S. L. Gonzaga de Oliveira

Recebido em 28 de setembro de 2022 / Aceito em 10 de agosto de 2023

ABSTRACT. This paper studies the bandwidth reduction problem for large-scale sparse matrices in serial computations. A heuristic for bandwidth reduction reorders the rows and columns of a given sparse matrix, placing entries with a non-null value as close to the main diagonal as possible. Recently, a paper proposed the FNCHC+ heuristic. The heuristic method is a variant of the Fast Node Centroid Hill-Climbing algorithm. The FNCHC+ heuristic presented better results than the other existing heuristics in the literature when applied to reduce the bandwidth of large-scale graphs (of the underline matrices) with sizes up to 18.6 million vertices and up to 57.2 million edges. The present paper provides new experiments with even larger graphs. Specifically, the present study performs experiments with test problems containing up to 24 million vertices and 130 million edges. The results confirm that the FNCHC+ algorithm is the state-of-the-art metaheuristic algorithm for reducing the bandwidth of large-scale matrices.

Keywords: bandwidth reduction, reordering algorithms, sparse matrices.

1 INTRODUCTION

A graph $G(V,E)$ consists of two sets V and E. Set V contains vertices representing objects or entities. Set E comprises edges describing the pairwise association between vertices. Practitioners have designed practical solutions to complicated problems in several areas, modeling and analyzing them by employing graph structures (Barik et al., 2020).

Paging policies and modern hierarchical memory architecture favor algorithms that deal with the locality of reference. Efficient applications employ cache coherence. Specifically, a sequence of recent memory references should be clustered locally in the memory address space (Gonzaga de Oliveira et al., 2018b). However, graph processing presents an unsatisfactory cache locality. The reason is the erratic memory access (Balaji & Lucia, 2018). Cache coherence is unsatisfactory in applications with inadequate spatial and temporal locality in memory access. Hence, practitioners have performed efforts to accelerate applications modeled by graph structures. One of these efforts is to reduce the graph bandwidth. Thus, one can relate the bandwidth reduction problem to

Unifesp, São Paulo, SP, Brazil – E-mail: sanderson.oliveira@unifesp.br – https://orcid.org/0000-0003-4863-542X

a broad scope of significant practical applications in various application domains in engineering, science, industry, and technology (Gonzaga de Oliveira & Silva, 2021).

Heuristics for bandwidth reduction put non-null coefficients of a sparse matrix as close to the main diagonal as possible. Let $A = [a_{ij}]$ be an $n \times n$ symmetric matrix corresponding to an undirected graph $G = (V, E)$, where V and E are sets of n vertices and edges, respectively. The bandwidth of row i is $\beta_i(A) = i - \min_{1 \leq j \leq i} [j \mid a_{ij} \neq 0]$. The overall bandwidth of a matrix A is defined as $\beta(A) = \max_{1 \leq i \leq n} [\beta_i]$. This means that $a_{ij} = 0$ whenever $|i - j| > \beta$. Likewise, the bandwidth of G for a vertex labeling $S = \{s(v_1), s(v_2), \cdots, s\left(v_{|V|}\right)\}$ is defined as $\beta_S(G) = \max_{\{v,u\} \in E} [|s(v) - s(u)|]$, where $s(u)$ and $s(v)$ are labels of vertices u and v, respectively. Practitioners use identifiers in the interval $[1, |V|]$ to identify the vertices. Thus, S is a bijective 1-1 mapping from V to the set $\{1, 2, \ldots, |V|\}$). The bandwidth minimization problem is a notable $\mathcal{NP}$-hard problem (Papadimitriou, 1976). Thus, a common algorithm for the problem is a heuristic method with the implication that it attempts to find an ordering that delivers a small bandwidth in an acceptable amount of time, even for large-scale instances (Gonzaga de Oliveira & Carvalho, 2022).

Reordering strategies, which relabel the graph's vertices to yield better non-null patterns in the subjacent matrix, have long been employed in the sparse numerical linear algebra field (Marichal et al., 2021). Specifically, applications use heuristics for bandwidth reduction to favor spatial locality and low-cost executions in various scientific and engineering domains. A frequent application is to reduce the running times of linear system solvers originating from finite element or volume discretizations, chemical kinetics, circuit design, industrial electromagnetics, large-scale power transmission systems, numerical analysis, numerical geophysics, etc. (Esposito et al., 1998; Piñana et al., 2004; Lim et al., 2006; Koohestani, 2022). Researchers use a low-cost heuristic method in these situations. Hence, practical algorithms use graph theory concepts (Gonzaga de Oliveira & Silva, 2019, 2020).

The present computational experiment, however, concentrates on areas that do not require a strict restriction on the runtime of the heuristic method for bandwidth reduction. Examples of these domains are in saving large hypertext media (Berry et al., 1996), small-world networks (Higham, 2003), network survivability (Lim et al., 2006), VLSI design (Lim et al., 2006), visual analysis of data sets using visual similarity matrices (Mueller et al., 2007), graph entropy rate minimization (Bolanos et al., 2012), mesh layout optimization (Chen, 2017), symbolic model checking (Amparore et al., 2017), and seriation problem (Concas et al., 2019). Also, the efficient representation of sparse matrices is crucial in several applications (Lim et al., 2006). In particular, they are fundamental in big data applications. It is decisive in this field to employ a matrix storage format with low requirements. Furthermore, applications must avoid unproductive computations and need to provide the data locality (Marichal et al., 2021). For example, applications in communication networks, data organization, computation flow, and computational devices employ a graph representation of the matrix (Koohestani, 2020). A format with low memory requirements allows large-scale instances of the problem in context. Moreover, theory interest in the subject stimulates the design of heuristics that can reduce the bandwidth of large-scale matrices to a substantial proportion at an acceptable runtime. Therefore, it is possible to know how far the result delivered by a low-cost algorithm is from the best result already provided in a specific instance (Gonzaga de Oliveira & Carvalho, 2022).

Recently, Gonzaga de Oliveira & Carvalho (2022) proposed a new metaheuristic algorithm for bandwidth reduction. The authors referred to the new algorithm as the FNCHC+ heuristic. Furthermore, the method yielded better bandwidth results than the existing heuristics in the literature when using large-scale matrices. The present paper presents new experiments with the FNCHC+ heuristic applied to even larger matrices. Gonzaga de Oliveira & Carvalho (2022) showed that before the proposal of the FNCHC+ heuristic, the GPS algorithm (Gibbs et al., 1976) was the previous state-of-the-art-heuristic for the bandwidth reduction of large-scale matrices when the running times were not a crucial aspect in the analysis. Thus, this paper compares the results of the FNCHC+ heuristic with the results of the GPS algorithm when applied to 32 large-scale matrices taken from the SuiteSparse matrix collection (Davis & Hu, 2011). SuiteSparse is a large (and actively expanding) set of sparse matrices originating from real-world applications. Moreover, SuiteSparse is broadly used by the numerical linear algebra community for the design and performance evaluation of sparse matrix algorithms, permitting solid and replicable experimentation since performance results with artificially-generated matrices may be imprecise. Furthermore, the collection contains an expansive range of application fields, including 2D or 3D geometric domains (computational fluid dynamics, electromagnetics, structural engineering, thermodynamics, etc.) (Davis & Hu, 2011).

The present study deepens a previous effort (Gonzaga de Oliveira & Carvalho, 2022). Specifically, this work shows new experiments to compare the results of the FNCHC+ heuristic with the GPS algorithm when applied to 32 matrices with sizes of up to 24 million rows and columns and 130 million non-null coefficients. To the author's knowledge, no publication showed results from a metaheuristic algorithm for bandwidth reduction applied to graphs with such large dimensions.

The remainder of this paper is structured as follows. In Section 2, this study describes the related work in this field. In Section 3, this paper presents the FNCHC+ heuristic. The article describes how the study conducted the experiments in Section 4. The study shows and analyzes the experimental results in Section 5. Finally, the manuscript presents the main conclusions in Section 6.

2 RELATED WORK

The bandwidth minimization problem originates from the 1950s when structural engineers researched sparse linear system solvers (Livesley, 1960; Chinn et al., 1982; Gonzaga de Oliveira & Chagas, 2015). Researchers have proposed several heuristic methods for bandwidth reduction since the 1960s (Gonzaga de Oliveira & Chagas, 2015). This paper considers only the main recent advances in this section.

Chagas & Gonzaga de Oliveira (2015); Gonzaga de Oliveira & Chagas (2015); Gonzaga de Oliveira et al. (2018c,b); Gonzaga de Oliveira & Silva (2019, 2020); Gonzaga de Oliveira &

Carvalho (2022) reviewed the literature. These publications are reported below and indicate the most relevant heuristics for bandwidth reduction published so far.

One can divide the heuristics for bandwidth reduction into two main categories. The first category, reported in Section 2.1, comprises low-cost heuristics based on graph theory concepts, and the second category, presented in Section 2.2, encompasses metaheuristic algorithms.

2.1 Graph-theory algorithms

Graph-theory heuristic methods for bandwidth reduction are low-cost heuristics primarily employed to accelerate linear system solvers. Gonzaga de Oliveira & Chagas (2015) examined the results of several heuristic methods for bandwidth reduction based on graph theory concepts. Shortly, the most used graph-theory heuristic methods in this field are the Reverse Cuthill-McKee (RCM) (George & Liu, 1981) and GPS (Gibbs et al., 1976) methods.

Koohestani and Poli (Koohestani & Poli, 2011) employed genetic programming (GP) metaheuristic to devise the KP-band heuristic. The approach evolved the RCM method (George & Liu, 1981) and generated a new heuristic for bandwidth reduction.

Gonzaga de Oliveira et al. (2018c) reported that a reasonable bandwidth reduction accelerates linear system solvers. Thus, the authors showed that a reverse breadth-first search procedure with the starting vertex given by the George-Liu algorithm (George & Liu, 1979) (RBFS-GL for short) is a relevant heuristic to label graph's vertices of subjacent matrices.

Gonzaga de Oliveira & Silva (2019, 2020) introduced an ant colony hyperheuristic (ACHH) strategy for bandwidth reduction. Specifically, the objective was to accelerate a linear system solver (Gonzaga de Oliveira & Silva, 2019). The hyperheuristic evolved the RCM, KP-band, RBFS-GL, and a proposed heuristic named the RLK heuristic. The hyperheuristic approach receives a small set of small-sized matrices arising from an application field. As a result, the ACHH system generated a heuristic for the application domain. Thus, the strategy created several expert-level set reordering heuristics. The approach delivered better results than previous state-of-the-art low-cost heuristics for bandwidth reduction (RCM, KP-band, RBFS-GL). Gonzaga de Oliveira & Silva (2021) incorporated a local search into the approach and yielded even better results when executed with symmetric instances. Thus, when considering the running times as a relevant feature of the evaluation, the ACHH system (Gonzaga de Oliveira & Silva, 2019, 2020, 2021) is the current state-of-the-art strategy regarding low-cost heuristics for bandwidth reduction.

2.2 Metaheuristic algorithms

As previously mentioned, the present study concentrates on applications that do not demand a hard limit on the execution time of the heuristic for bandwidth reduction. These applications can use metaheuristic algorithms for the problem. The reason is that, in general, metaheuristic algorithms yield better results than low-cost graph-theory algorithms for bandwidth reduction at longer running times.

The development of metaheuristic algorithms for bandwidth reduction began in the 1990s (Chagas & Gonzaga de Oliveira, 2015). Researchers have designed heuristics with the most different metaheuristics, including GRASP-PR (Piñana et al., 2004), genetic algorithm (Lim et al., 2006; Czibula et al., 2013; Pop et al., 2014), simulated annealing (Rodriguez-Tello et al., 2008; Torres-Jimenez et al., 2015), ant colony optimization (Kaveh & Sharafi, 2009; Pintea et al., 2010, 2012; Czibula et al., 2013; Guan et al., 2019; Gonzaga de Oliveira & Silva, 2019, 2020), variable neighborhood search (Mladenovic et al., 2010), genetic programming (Koohestani & Poli, 2011), charged system search algorithm (Kaveh & Sharafi, 2012), colliding bodies optimization (Kaveh & Bijari, 2015), brain storm optimization (Mafteiu-Scai et al., 2017, 2019), biased random-key genetic algorithm (Silva et al., 2020), and iterated local search (Gonzaga de Oliveira & Carvalho, 2022).

While traditional metaheuristic algorithms (e.g., Mladenovic et al. (2010); Torres-Jimenez et al. (2015)) for the problem showed remarkable bandwidth results when applied to very small-sized matrices (i.e., matrices with sizes of up to approximately 1,000) (Gonzaga de Oliveira et al., 2016, 2017), these metaheuristic algorithms are very slow (Gonzaga de Oliveira et al., 2018b; Gonzaga de Oliveira & Silva, 2020; Gonzaga de Oliveira & Carvalho, 2022). A traditional metaheuristic algorithm significantly reduces the bandwidth of very small-sized matrices at a high cost, whereas the other algorithms do not perform well. However, a conventional metaheuristic algorithm takes several orders of magnitude more running time than low-cost graph-theory heuristics for bandwidth reduction (Gonzaga de Oliveira et al., 2016, 2017, 2018b,c; Gonzaga de Oliveira & Silva, 2020; Gonzaga de Oliveira & Carvalho, 2022).

The DRSA algorithm (Torres-Jimenez et al., 2015) may be seen as the current state-of-the-art metaheuristic algorithm for bandwidth reduction when applied to very small matrices with sizes of up to approximately 1,000 rows and columns. The method surpassed the VNS-band heuristic (Mladenovic et al., 2010) regarding the quality solution. The VNS-band heuristic was the previous state-of-the-art metaheuristic algorithm for the bandwidth reduction of very small matrices. Furthermore, a simulated annealing heuristic (Rodriguez-Tello et al., 2008) was the state of the practice metaheuristic algorithm for bandwidth reduction before VNS-band.

Lim et al. (Lim et al., 2007) proposed the Node Centroid with Hill Climbing (NCHC) strategy based on a previous effort (Lim et al., 2006). Furthermore, Lim et al. (Lim et al., 2007) performed a breakthrough scientific paper in metaheuristic algorithms for the bandwidth reduction of large-scale matrices. The authors proposed the FNCHC heuristic. Furthermore, the method represents a low-cost implementation of the node-centroid-based strategy to solve the bandwidth reduction problem. Moreover, the heuristic presented lower running times than traditional metaheuristic algorithms for bandwidth reduction (Gonzaga de Oliveira et al., 2018b). To the best of the author's knowledge, FNCHC was the first metaheuristic algorithm able to handle medium-size instances, i.e., graphs with sizes ranging from 100,000 to 1,000,000 (Gonzaga de Oliveira & Carvalho, 2022) with reasonable running times.

A previous investigation (Gonzaga de Oliveira & Carvalho, 2022) proposed a heuristic method based on the iterated local search metaheuristic. The authors referred to the new algorithm as the ILS-band heuristic. ILS-band dominated FNCHC when applied to large-size matrices, i.e., matrices higher than one million.

Gonzaga de Oliveira & Carvalho (2022) also proposed a variant of the FNCHC heuristic. The authors referred to the new method as the FNCHC+ heuristic. The FNCHC+ heuristic dominated the FNCHC (Lim et al., 2007), VNS-band (Mladenovic et al., 2010) (and, hence, DRSA), and ILS-band heuristics concerning the solution quality and running times when applied to large-scale matrices (Gonzaga de Oliveira & Carvalho, 2022). The FNCHC+ heuristic also surpassed the resulting heuristics from the ACHH approach (Gonzaga de Oliveira & Silva, 2020) regarding the quality solution of large-scale matrices. Additionally, the FNCHC+ heuristic outperformed GPS (Gibbs et al., 1976) concerning the solution quality when applied to matrices with sizes up to 18,571,154 rows and columns and 57,156,537 non-null coefficients (Gonzaga de Oliveira & Carvalho, 2022). However, the GPS algorithm provided, in general, better running times in the experiments. Thus, the present study shows experiments to verify whether the FNCHC+ heuristic yields better bandwidth results than the GPS algorithm with even larger matrices. Furthermore, this paper also studies the running times of the heuristic methods.

Without considering the running times, Gonzaga de Oliveira et al. (2018b); Gonzaga de Oliveira & Silva (2020) showed that the GPS algorithm was the most competitive method to provide remarkably satisfactory bandwidth results when applied to large-scale matrices. Thus, as previously mentioned, this paper compares the results of the FNCHC+ heuristic with the results of the GPS algorithm when applied to 32 large-scale matrices with sizes of up to 24 million rows and columns and 130 million non-null coefficients. To the best of the author's knowledge, no publication presented results from a metaheuristic algorithm for bandwidth reduction applied to graphs with such large dimensions.

3 THE FNCHC+ HEURISTIC

The FNCHC+ heuristic (Gonzaga de Oliveira & Carvalho, 2022) derived from the FNCHC heuristic (Lim et al., 2007). Furthermore, FNCHC is a fast implementation of the NCHC heuristic. Therefore, the FNCHC+ heuristic (Gonzaga de Oliveira & Carvalho, 2022) is a variant of the Node Centroid Hill Climbing (NCHC) heuristic (Lim et al., 2006).

The NCHC approach iterates i_{max} times. A breadth-first search procedure starting from a random vertex provides the initial labeling in each iteration. Then, NCHC performs vertex label adjustments for global searches followed by a local search Hill-Climbing procedure. The approach adjusts vertex labels as a central (named centroid) position regarding its adjacent vertices. Afterward, the method creates a new ordering applying the Hill-Climbing subroutine to reach a local optimum. NCHC performs vertex label adjustments and the local search Hill-Climbing procedure NC_{max} times.

As previously mentioned, Lim et al. (Lim et al., 2007) developed a low-cost version of the NCHC algorithm. The authors referred to the heuristic as the Fast Node Centroid Hill Climbing (FNCHC) heuristic. FNCHC automatically adjusts parameters according to the matrix size.

Analogous to the original algorithms, the FNCHC+ heuristic employs a labeling adjustment approach and a Hill-Climbing procedure to reorder a given matrix. Algorithm 1 shows simplified pseudocode of the FNCHC+ heuristic. Algorithm 1 creates initial labeling by carrying out the BFS procedure with a random starting vertex in line 3 of Algorithm 1. The heuristic modifies the original ordering by adjusting the labeling concerning a given vertex and its neighbors, called the node centroid, in line 5. The heuristic determines a value for each vertex of the graph representation of the matrix. The node centroid subroutine provides the values. Subsequently, the heuristic labels the vertices according to these values in ascending order.

Algorithm 1: The FNCHC+ heuristic (Gonzaga de Oliveira & Carvalho, 2022).

Input: a graph $G = (V, E)$ representation of the input matrix; a labeling S; maximum number of iterations i_{max}; maximum number of iterations NC_{max} for the $NodeCentroid$ subroutine; an integer κ;

Output: a new labeling S^*;

1 $S^* \leftarrow S$;
2 **for** ($i \leftarrow 1$; $i \leq i_{max}$; $i \leftarrow i+1$) **do**
3     $S \leftarrow Breadth\text{-}firstSearchProcedure(S)$;
4   **for** ($j \leftarrow 1$; $j \leq NC_{max}$; $j \leftarrow j+1$) **do**
        `// the NodeCentroid procedure labels the vertices`
5         $S' \leftarrow NodeCentroid(S)$;
6   **if** ($\beta_{S'}(G) - \beta_{S^*}(G) < \kappa$) **then** $S' \leftarrow HillClimbing(S')$;
7   **if** ($\beta_{S'}(G) < \beta_{S^*}(G)$) **then** $S^* \leftarrow S'$;
8 **return** S^*;

The local search Hill-Climbing subroutine is the bottleneck for the speed of the heuristic method. Thus, the FNCHC+ heuristic invokes the Hill-Climbing procedure only when a new solution S' has better (or similar) bandwidth than the best solution S^* found so far (see line 6). The heuristic employs the Hill-Climbing procedure if $\beta_{S'}(G) - \beta_{S^*}(G) < \kappa$ for $\kappa \in \mathbb{Z}$. Specifically, κ is a parameter in the algorithm. Thus, the heuristic method invokes the Hill-Climbing procedure only with promising new solutions because the Hill-Climbing procedure is a high-cost procedure. Additionally, the maximum number of iterations i_{max} (see line 2) and the maximum number of iterations NC_{max} (see line 4) for the $NodeCentroid$ subroutine are parameters in the FNCHC+ heuristic. Using these parameters appropriately makes it possible to apply the heuristic method to large-scale matrices yielding satisfactory results (Gonzaga de Oliveira & Carvalho, 2022).

4 DESCRIPTION OF THE TESTS

In addition to the FNCHC+ heuristic, this paper evaluates the results of the GPS algorithm (Gibbs et al., 1976). This study used the C++ programming language (employing the g++ version 5.4.0 compiler) to code the heuristics for bandwidth reduction analyzed.

The machine used in the execution of the tests with 32 matrices taken from the SuiteSparse matrix collection (Davis & Hu, 2011) featured an Intel® Xeon® Gold Scalable Processor (formerly Skylake) 5120 containing 192 GiB of main memory DIMM DDR4 2400 MT/s [Intel; Santa Clara, CA, United States]. This machine used the Ubuntu 19.10 operating system with Linux kernel version 5.3.0-64-generic. The SuiteSparse matrix collection contains matrices arising from a diversity of areas, including undirected graphs, undirected graphs with communities, undirected random graphs, undirected multigraph, directed graphs, electromagnetics problems, structural problems, semiconductor process problems, 2D/3D problem, optimization problem, and circuit simulation problem. In particular, this study took the matrices from the SuiteSparse matrix collection in order of size of the number of edges starting from the highest graphs used by Gonzaga de Oliveira & Carvalho (2022).

This study performed three serial runs for each graph and calculated the geometric means. The heuristics run on each component of the graph.

For each algorithm H applied to each set of matrices, Gonzaga de Oliveira et al. (2018b) proposed a metric to evaluate the quality of the bandwidth results yielded by the algorithms considered. For a group composed of N matrices, the publication computed $\rho = \sum_{i=1}^{N} \frac{\beta_H(i)-\beta_{min}(i)}{\beta_{min}(i)}$, where $\beta_H(i)$ is the bandwidth yielded when using an algorithm H applied to the matrix i, $\beta_{min}(i)$ is the lowest bandwidth provided in the matrix i (with the use of the heuristics evaluated and the original bandwidth). According to this metric, a method that yields consistent bandwidth results performs better than a heuristic that delivers the best bandwidth when applied to many instances but yields extremely unsatisfactory results in a few other matrices. Furthermore, when applied to a given matrix, the best algorithm commonly returns a small β even when another heuristic delivers the best bandwidth result.

The ratio $\frac{\beta_H(i)-\beta_{min}(i)}{\beta_{min}(i)}$ in metric ρ is not bounded. Specifically, it can be arbitrarily large if a heuristic performs unsatisfactorily on a given instance i. Although this was the objective, a recent publication proposed another metric. For each algorithm H in each dataset, Gonzaga de Oliveira et al. (2018a) computed $\upsilon = \sum_{i=1}^{N} \frac{\beta_H(i)-\beta_{min}(i)}{\beta_{max}(i)}$, where $\beta_{max}(i)$ is the largest bandwidth yielded in the matrix i (with the use of the heuristics evaluated and the original bandwidth). The ratio in metric υ is bounded. Specifically, it ensures $0 \leq \frac{\beta_H(i)-\beta_{min}(i)}{\beta_{max}(i)} < 1$. The metric υ does not excessively penalize an algorithm that delivers large bandwidth results.

This study also employs $\mathtt{t} = \sum_{i=1}^{N} \frac{\mathrm{t(s)}_H(i)-\mathrm{t(s)}_{min}(i)}{\mathrm{t(s)}_{min}(i)}$ to evaluate the running times of the heuristics (Gonzaga de Oliveira et al., 2018b; Gonzaga de Oliveira & Carvalho, 2022), where t represents time and s represents seconds. The nonsymmetric matrix A_u is added to its transpose matrix A_u^{T} $\left(A_u + A_u^{\mathrm{T}}\right)$. Subsequently, the implementations calculate the bandwidths of the resulting symmetric matrix [Gonzaga de Oliveira et al. (2018b) shows this strategy in detail].

5 RESULTS AND ANALYSIS

This section evaluates the bandwidth results yielded by the FNCHC+ heuristic against the GPS and ILS-band heuristics. Subsection 5.1 summarizes the results delivered by the FNCHC+, GPS, and ILS-band heuristics in a previous effort (Gonzaga de Oliveira & Carvalho, 2022). Subsection 5.2 presents new results from the FNCHC+ and GPS heuristics. Subsection 5.3 discusses the calibration of the FNCHC+'s parameters and the running times of the heuristic method.

5.1 Previous results yielded by the most promising alternative algorithms

This subsection describes previous results provided by five alternative algorithms. Subsections 5.1.1, 5.1.2, 5.1.3, 5.1.4, and 5.1.5 describes previous results produced by the ACHH, DRSA, VNS-band, FNCHC, and ILS-band heuristics, respectively. In tables in this subsection, β_0 denotes the original bandwidth of the matrix.

5.1.1 ACHH (Gonzaga de Oliveira & Silva, 2019, 2020, 2021)

As previously mentioned, the ACHH system (Gonzaga de Oliveira & Silva, 2019, 2020, 2021) is the current state-of-the-art strategy regarding low-cost heuristics for bandwidth reduction when considering the running times as a pertinent characteristic of the appraisal. Table 1 summarizes the results presented by a previous effort (Gonzaga de Oliveira & Carvalho, 2022) comparing the GPS, FNCHC+, ILS-band, and ACHH heuristics when applied to 10 matrices with sizes ranging from 70,656 to 1,585,478. The authors ran the heuristic methods with a timeout of 1,200 s.

Table 1 – Summary of the results given by GPS, FNCHC+, ILS-band, and ACHH approaches applied to reduce the bandwidth of 10 matrices contained in the SuiteSparse matrix collection (Gonzaga de Oliveira & Carvalho, 2022).

Metric	β_0	GPS	FNCHC+	ILS-band	ACHH
Number of best results	1	**4**	3	3	2
ρ	3151.0	**0.8**	1.1	1.9	2.1
υ	8.26	0.08	**0.06**	0.19	0.19
t	—	817	13,989	5,704	0

Table 1 shows that, in general, GPS, FNCHC+, and ILS-band provide overall better bandwidth results than the resulting heuristic from the ACHH strategy at longer running times, recalling that the present study concentrates on heuristic methods that yield high bandwidth reductions and execute at reasonable execution costs. Thus, this computational experiment will not run the resulting heuristics from the ACHH strategy.

5.1.2 DRSA (Torres-Jimenez et al., 2015)

As previously mentioned, the DRSA heuristic (Torres-Jimenez et al., 2015) may be seen as the current state-of-the-art metaheuristic algorithm for bandwidth reduction when applied to very small matrices with sizes of up to approximately 1,000 rows and columns. The method dominated VNS-band (Mladenovic et al., 2010) concerning the quality solution at longer running times. VNS-band was the previous state-of-the-art metaheuristic algorithm for the bandwidth reduction of very small matrices. However, the DRSA heuristic was at least five times slower than the VNS-band heuristic to yield the best bandwidth solution (Torres-Jimenez et al., 2015). The DRSA heuristic returned impressive results when applied to matrices with sizes up to approximately 1,000, whereas other heuristics [GRASP-PR (Piñana et al., 2004), SA (Rodriguez-Tello et al., 2008), VNS-band (Mladenovic et al., 2010)] do not perform well. However, Gonzaga de Oliveira & Silva (2020) revealed that three low-cost heuristic methods [KP-band (Koohestani & Poli, 2011), RCM (George & Liu, 1981), and RBFS-GL (Gonzaga de Oliveira et al., 2018c)] yielded better bandwidth results than the DRSA heuristic did when applied to matrices with sizes of approximately 5,000. The appraisal (Gonzaga de Oliveira & Silva, 2020) considered the running times as a crucial aspect of the analysis. Thus, the high execution costs make the metaheuristic-based algorithm unfeasible for sparse large-scale matrix factorization. One can also add other problems related to matrix factorization. This situation occurs when the context involves large-scale instances of the problem. Consequently, a traditional metaheuristic algorithm for bandwidth reduction is not practical for large-scale matrix processing (i.e., matrices with more than one million lines and columns) (Gonzaga de Oliveira & Carvalho, 2022).

5.1.3 VNS-band (Torres-Jimenez et al., 2015)

A previous effort (Gonzaga de Oliveira & Carvalho, 2022) showed that the VNS-band heuristic (Mladenovic et al., 2010) predominantly delivers high-quality solutions when applied to very small-sized matrices (Mladenovic et al., 2010) (i.e., matrices with sizes of approximately 1,000). However, Gonzaga de Oliveira & Carvalho (2022) showed that this metaheuristic-based algorithm did not produce competing results when applied to small-sized matrices (ranging from 5,940 to 112,985). The authors executed the VNS-band heuristic with a timeout several times longer than the FNCHC+, GPS, ILS-band, and FNCHC heuristics. The publication showed that the four heuristics delivered a higher number of best results than the VNS-band heuristic at lower execution costs. Since Gonzaga de Oliveira & Carvalho (2022) showed that FNCHC+, GPS, ILS-band, and FNCHC dominated VNS-band concerning solution quality and running times when executed with small-sized matrices, the present study will not apply the VNS-band heuristic to large-scale matrices.

5.1.4 FNCHC (Lim et al., 2007)

The FNCHC heuristic (Lim et al., 2007) delivered overall better results than the ILS-band heuristic when applied to 172 small-sized matrices (with sizes ranging from 5,940 to 112,985) at longer running times (Gonzaga de Oliveira & Carvalho, 2022). Gonzaga de Oliveira & Carvalho (2022) also applied the FNCHC heuristic to 25 large-scale matrices, i.e., matrices with more than one million lines and columns. The authors aborted executions higher than 1,200 s. Some executions of the FNCHC heuristic took more than two hours. It is not practical to wait hours for the solution of a heuristic method. Furthermore, Gonzaga de Oliveira & Carvalho (2022) aborted 23 out of 25 executions with the FNCHC heuristic. Thus, the publication showed that the FNCHC heuristic (Lim et al., 2007) is impractical for large-scale matrices. On the other hand, the authors aborted only 4 out of 76 executions of the ILS-band heuristic when applied to large-scale matrices. Furthermore, the authors showed that FNCHC+, GPS, and ILS-band dominated FNCHC when applied to large-scale matrices.

5.1.5 ILS-band (Gonzaga de Oliveira & Carvalho, 2022)

Gonzaga de Oliveira & Carvalho (2022) showed that the ILS-band heuristic dominated previous state-of-the-art metaheuristic algorithms when applied to large-scale matrices. Then, this section summarizes the results yielded by the ILS-band, GPS, and FNCHC+ heuristics provided by Gonzaga de Oliveira & Carvalho (2022).

Large-scale symmetric matrices. Table 2 reproduces the results presented by Gonzaga de Oliveira & Carvalho (2022) comparing the FNCHC+, ILS-band, and GPS heuristics when applied to 45 symmetric matrices with sizes ranging from 1,000,000 to 18,571,154. The authors aborted four executions with each ILS-band and GPS because the runs took more than 1,200 s. Moreover, the study aborted another execution with the GPS algorithm. The method could not run because of high memory usage (more than 15 GB). Thus, Gonzaga de Oliveira & Carvalho (2022) calculated the metrics ρ and υ associated with the FNCHC+, ILS-band, and GPS heuristics with 45, 41, and 40 matrices, respectively, recalling that the metrics are smaller using fewer instances. Even so, Table 2 shows that the FNCHC+ heuristic delivered better results than the ILS-band and GPS heuristics at longer running times. Furthermore, metric ρ in Table 2 reveals that the ILS-band heuristic provided slightly better results than the GPS algorithm at longer running times.

Table 2 – Summary of the results delivered by the FNCHC+, ILS-band, and GPS heuristics applied to reduce the bandwidth of 45 large-scale symmetric instances (Gonzaga de Oliveira & Carvalho, 2022).

Metric	β_0	FNCHC+		ILS-band		GPS	
		β	t	β	t	β	t
Number of best results	3	**29**	5	9	8	8	**32**
ρ υ and t	41,852.99 35.6	**1.39** **0.3**	915	4.49 0.4	560	4.53 0.8	**84**

Gonzaga de Oliveira & Carvalho (2022) aborted executions with ILS-band when applied to the three highest large-scale graphs (containing more than 16 million vertices and more than 38 million edges) employed in the testbed because the heuristic took more than 1,200 s. Thus, the publication did not execute ILS-band with larger graphs. On the contrary, the FNCHC+ heuristic delivered satisfactory results in runs with these matrices.

Large-scale nonsymmetric matrices. Table 3 reproduces the results presented by Gonzaga de Oliveira & Carvalho (2022) comparing the FNCHC+, ILS-band, and GPS heuristics when applied to 31 nonsymmetric matrices with sizes ranging from 1,000,005 to 9,845,725. The authors aborted eight executions with GPS because the runs took more than 1,200 s. Thus, Gonzaga de Oliveira & Carvalho (2022) calculated the metrics ρ and υ associated with the GPS heuristics with only 23 matrices. As previously mentioned, the metrics are smaller using fewer instances. Even so, the table shows that the FNCHC+ heuristic delivered better results than the GPS algorithm at higher execution costs. Furthermore, the table reveals that the FNCHC+ heuristic surpassed the ILS-band heuristic concerning solution quality and running times. Moreover, metrics ρ and υ in Table 3 shows that the GPS algorithm delivered better results than the ILS-band heuristic, recalling that the study used 23 matrices to compute the metrics associated with the GPS algorithm and employed 31 instances to calculate the metrics related to the ILS-band heuristic.

Table 3 – Summary of the results delivered by the FNCHC+, GPS, and ILS-band heuristics applied to reduce the bandwidth of 31 large-scale nonsymmetric matrices contained in the SuiteSparse matrix collection (Gonzaga de Oliveira & Carvalho, 2022).

Metric	β_0	FNCHC+		GPS		ILS-band	
		β	t	β	t	β	t
Number of best results	0	**21**	10	4	**11**	6	10
ρ and t	1440	**1.5**	242	4.6	**64**	54.8	331
υ	20	0.7		**0.6**		8.4	

Gonzaga de Oliveira & Carvalho (2022) showed that ILS-band presented difficulties in delivering results in less than 1,200 seconds when applied to the highest nonsymmetric matrix (wb-edu) employed in the testbed, composed of almost 10 million rows and columns and more than 57.1 million non-null coefficients. Thus, the authors did not execute ILS-band with higher instances. In contrast, FNCHC+ produced a satisfactory bandwidth result when applied to this test problem.

Discussion: FNCHC+ $\times$ ILS-band. Table 2 and 3 show that the FNCHC+ heuristic surpassed the ILS-band heuristic when applied to 76 large-scale matrices. Specifically, Table 3 reveals that FNCHC strongly dominated ILS-band concerning solution quality and running times when applied to large-scale nonsymmetric matrices. Although metric t in Table 2 shows that the FNCHC+ heuristic took longer running times than the ILS-band heuristic when executed with large-scale symmetric matrices, Gonzaga de Oliveira & Carvalho (2022) aborted four executions with ILS-band when applied to some of these instances because the runs took more than 1,200 s. Thus, the publication also showed that FNCHC+ dominated ILS-band regarding solution quality and execution costs when executed with large-scale symmetric matrices. Consequently, the present study will not apply the ILS-band heuristic to larger matrices.

Discussion: FNCHC+ $\times$ GPS. Table 2 and 3 show that the FNCHC+ heuristic delivered better results than the GPS algorithm at higher execution costs when applied to 76 large-scale matrices. Metric υ in Table 3 reveals that the GPS algorithm ($\upsilon = 0.6$) delivered slightly better results than the FNCHC+ heuristic ($\upsilon = 0.7$). However, the study computed the metric with results provided by the GPS heuristic when applied to 23 out of 31 large-scale nonsymmetric matrices. The reason was that the heuristic method executed longer than 1,200 seconds when applied to eight large-scale nonsymmetric matrices. Thus, the present computational experiment compares the results of the FNCHC+ heuristic with the results of the GPS algorithm when applied to even larger matrices. Therefore, Subsection 5.2 shows new experiments to evaluate the results of the FNCHC+ and GPS heuristics when applied to matrices with sizes up to 24 million rows and columns and 130 million non-null coefficients.

5.2 New results yielded by the FNCHC+ and GPS heuristics applied to matrices with sizes up to 24 millions

This section evaluates the bandwidth results returned by the FNCHC+ heuristic against the GPS algorithm. Table 4 shows the results delivered by the FNCHC+ and GPS heuristics when applied to 32 large-scale matrices. The table highlights in boldface the best results.

The GPS algorithm executed longer than 1,500 s when applied to the matrices nlpkkt120, ML_Geer, and Flan_1565. As previously mentioned, waiting a long time for a heuristic for bandwidth reduction is not acceptable. Thus, the study aborted the executions. Gonzaga de Oliveira & Carvalho (2022) noted that the problem of the GPS algorithm was in the step that finds the first of two pseudo-peripheral vertices. In these test cases, the study circumvented the problem using the George-Liu algorithm (George & Liu, 1979) in this step instead of employing the original algorithm. In particular, the study did not execute a GPS-GL heuristic to all matrices because the original GPS algorithm yields better bandwidth results at longer running times than this variant. However, even the GPS-GL algorithm ran longer than 1,500 s when executed with the matrices circuit5M, channel-500x100x100-b050, kron_g500-logn20, and Bump_2911. Oppositely, the FNCHC+ heuristic yielded high-quality solutions at a low cost in executions with these test problems. Metrics ρ and υ in Table 4 show that the FNCHC+ heuristic returned predominantly better bandwidth results than the GPS algorithm. The study used only 28 results to calculate the metrics because the computational experiment aborted four executions with the GPS algorithm.

This study employed the Wilcoxon matched-pairs signed-rank test to evaluate the bandwidth reduction delivered by the FNCHC+ and GPS algorithms. The null hypothesis for the test was that the methods delivered identical bandwidth results, i.e., the median difference was zero. The alternative hypothesis was that the median difference was positive. Thus, the study used

Table 4 – Bandwidths delivered by the FNCHC+ and GPS heuristics executed to reduce the bandwidth of 32 large-scale instances [t represents time, s represents seconds, † indicates that the study aborted runs longer than 1,500 seconds].

Matrix	n	$\|E\|$	GPS	t(s)	FNCHC+	t(s)	κ,i_{max},NC_{max}
kron_g500-logn20	1,048,576	89,239,674	†	†	**706513**	25	1,1,1
dielFilterV3real	1,102,824	89,306,020	30314	135	**23441**	51	10,10,10
hollywood-2009	1,139,905	113,891,327	814891	778	**761533**	50	1,1,1
Serena	1,391,349	65,923,050	119984	43	**77391**	52	10,10,10
Geo_1438	1,437,960	64,594,650	30623	1082	**24356**	39	10,10,10
Long_Coup_dt0	1,470,152	88,559,144	**32735**	104	35477	629	40,40,40
Long_Coup_dt6	1,470,152	88,559,144	**32735**	104	35477	629	40,40,40
Hook_1498	1,498,023	62,415,468	29033	59	**26659**	42	10,10,10
ML_Geer	1,504,002	110,879,972	**4667**	105	4841	801	30,30,30
Flan_1565	1,564,794	118,970,838	**17328**	205	19196	458	30,30,30
vas_stokes_2M	2,146,677	65,129,037	118855	370	**109327**	593	30,30,30
Cube_Coup_dt0	2,164,760	129,370,904	31526	955	**30331**	503	30,30,30
Cube_Coup_dt6	2,164,760	129,370,904	31526	1147	**30331**	505	30,30,30
Bump_2911	2,911,419	130,641,318	†	†	**70379**	5	0,1,1
nlpkkt120	3,542,400	100,388,192	86876	951	**54658**	65	10,10,10
com-LiveJournal	3,997,962	69,362,378	2186870	161	**2154624**	74	1,1,1
rgg_n_2_22_s0	4,194,304	60,718,396	6549	118	**5637**	96	10,10,10
channel-500x100x100-b050	4,802,000	85,362,744	†	†	**23879**	74	10,10,10
soc-LiveJournal1	4,847,571	68,993,773	3066903	770	**2336099**	755	20,20,20
cage15	5,154,859	99,199,551	420177	526	**417647**	1057	30,30,30
ljournal-2008	5,363,260	79,023,142	2714967	850	**2504915**	705	20,20,20
circuit5M	5,558,326	59,524,291	†	†	**4224322**	8	1,1,1
rgg_n_2_23_s0	8,388,608	127,002,786	9605	443	**7861**	1134	30,30,30
hugetrace-00010	12,057,441	36,164,358	3954	495	**3837**	1381	20,15,100
road_central	14,081,816	33,866,826	**5526**	253	6082	1079	10,30,40
hugetrace-00020	16,002,413	47,997,626	6238	552	**5086**	577	10,10,40
delaunay_n24	16,777,216	100,663,202	**21307**	484	22725	743	20,20,20
hugebubbles-00000	18,318,143	54,940,162	4404	757	**4391**	1468	5,20,5
hugebubbles-00010	19,458,087	58,359,528	**4161**	843	5180	240	10,10,10
hugebubbles-00020	21,198,119	63,580,358	**3984**	871	4582	913	20,20,20
GAP-road	23,947,347	57,708,624	6902	687	**6731**	856	20,20,20
road_usa	23,947,347	57,708,624	6902	691	**6731**	848	20,20,20
Number of best results			8	16	**24**	16	—
ρ			—	3.13	—	**0.87**	—
υ			—	2.38	—	**0.77**	—
t			—	602	—	**28**	—

a two-tailed test because the study did not specify a direction. Additionally, the study needed to determine if there was any difference between the groups compared (i.e., the possibility of positive or negative differences). In this case, $R_+ = 436$ and $R_- = 92$. The test statistic is, therefore, $T = 92 \leq T_{crit(\alpha=0.001,32)} = 94$. The study presented statistically significant evidence at $\alpha = 0.001$ (i.e., the significance level of 0.001 is related to the 99.9% confidence level) to show that the median difference is positive. This result means that the FNCHC+ heuristic surpassed the GPS algorithm. At a significance level of 0.01, p-value is 0.00128.

Figure 1 shows the number of best results and metrics $\mathtt{t}$, ρ, and υ computed with the bandwidth results returned by the two heuristics. The study used 1,500 s as the running time of the GPS algorithm when applied to the matrices circuit5M, channel-500x100x100-b050, kron_g500-logn20, and Bump_2911 to calculate metric $\mathtt{t}$. Thus, Table 4 and Figure 1 show that the FNCHC+ heuristic surpassed the GPS algorithm regarding the output solution quality and running times. Moreover, each algorithm returned 16 times the best running time. However, metric $\mathtt{t}$ in Table 4 and Figure 1 reveals that the FNCHC+ heuristic was faster than the GPS algorithm.

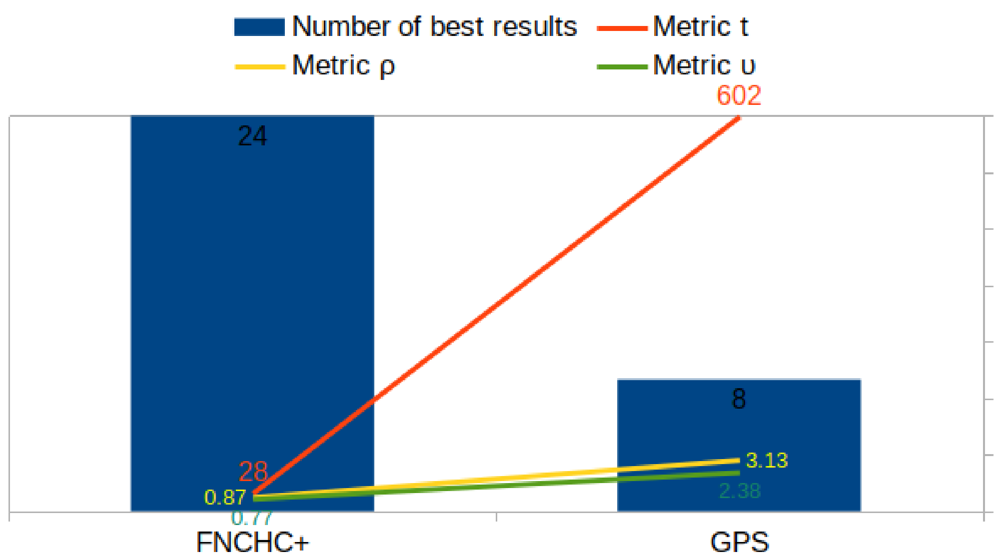

Figure 1 – Results yielded by the FNCHC+ and GPS heuristics executed to reduce the bandwidth of 32 large-scale matrices. Metric $\mathtt{t}$ reveals that the FNCHC+ heuristic executed predominantly at shorter running times than the GPS algorithm.

5.3 Parameter calibration and running times

An advantage of the FNCHC+ over the GPS algorithm is that, through the parameters, one can control the solution quality and running times. This study performed some tests to calibrate the parameters of the FNCHC+ heuristic for each test problem. However, the study did not run the FNCHC+ heuristic with even better parameters when this heuristic delivered the best results.

Therefore, the heuristic method can likely yield even better results at shorter running times if applied to those matrices with more calibrated parameters.

The running time of the FNCHC+ heuristic depends on the size of the input matrix and the parameters κ, i_{max}, and NC_{max}. The larger the parameter κ, the more calls are made to the high-cost local search *HillClimbing*. In particular, *HillClimbing* is the subroutine with the highest execution cost. The parameter i_{max} determines the number of iterations in Algorithm 1. Finally, the NC_{max} parameter defines the number of calls to the *NodeCentroid* subroutine. The study did not calibrate the parameters of the FNCHC+ heuristic for runs with matrices in which the GPS algorithm did not return results, as with the matrices kron_g500-logn20, Bump_2911, channel-500x100x100-b050, and circuit5M. The study set the three parameters to 1 in these test cases. With other matrices, it was possible to set the parameters of the FNCHC+ heuristic to 10 or 20. Hence, FNCHC+ obtained better bandwidth results than the solutions returned by the GPS algorithm. As previously mentioned, one can choose better parameters for the FNCHC+ heuristic to achieve even better results. Thus, the execution times of the FNCHC+ heuristic shown in Table 4 alternate between high and low running times because they mainly depend on the established parameters.

Figure 2 shows the running times of the FNCHC+ heuristic applied to eight matrices using similar parameters ($\kappa \in \{20,30\}$, $i_{max} \in \{15,30\}$, and $NC_{max} \in \{30,100\}$). The figure shows that, with similar parameters, the running time of the FNCHC+ heuristic is approximately linear in the number of vertices of the input graph, although more calls to the local search procedure increase the running time of the heuristic. Furthermore, the running times of the heuristic are much longer than the running times of a graph-theory algorithm, such as the resulting heuristics from the ACHH approach, because the FNCHC+ heuristic has much more code.

6 SUMMARY AND CONCLUSIONS

Gonzaga de Oliveira & Carvalho (2022) showed that the FNCHC+ heuristic delivered better bandwidth results than the GPS algorithm at higher execution costs when applied to matrices with sizes up to 18,571,154 rows and columns and 57,156,537 non-null coefficients. The present study provided further experiments with these two heuristics. The investigation performed extensive experiments on large-scale problem matrices with sizes up to 23,947,347 (GAP-road, road_usa) rows and columns and 130,641,318 (Bump_2911) non-null coefficients. The FNCHC+ heuristic surpassed the GPS algorithm concerning the quality solution and running times with matrices with such sizes. In particular, the FNCHC+ heuristic computed the matrix Bump_2911 in 5 seconds, whereas the GPS algorithm took more than 1,500 seconds in this matrix. Thus, the study aborted the execution. To the best of the author's knowledge, no other metaheuristic algorithm for bandwidth reduction that computes matrices of such a dimension in a acceptable time exists in the literature.

The FNCHC+ heuristic is fast because it employs a different strategy from traditional metaheuristic algorithms. Instead of calling the high-cost local-search *HillClimbing* algorithm at each

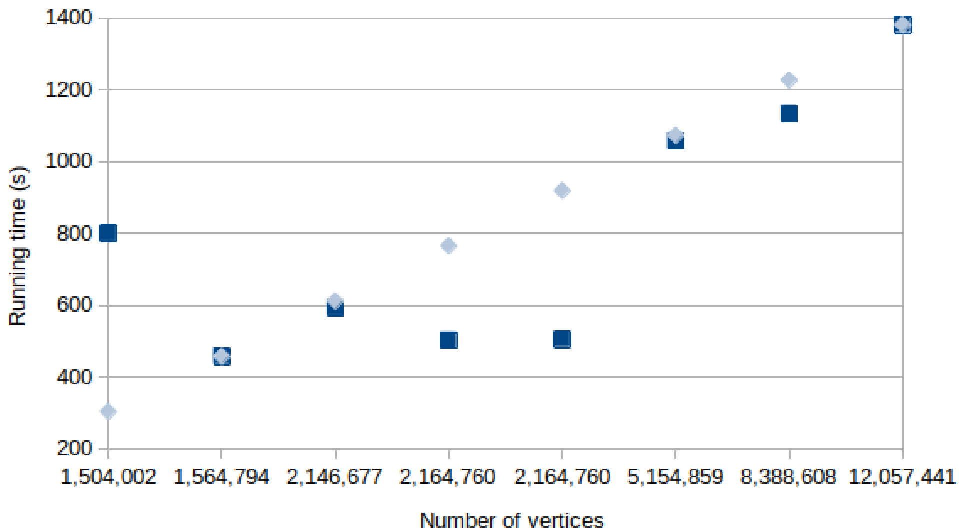

Figure 2 – Running times of the FNCHC+ heuristic applied to eight large-scale matrices.

iteration, as performed in conventional metaheuristic algorithms, the FNCHC+ heuristic invokes the Hill-Climbing subroutine only when the approach finds a promising solution. This strategy means that the FNCHC+ heuristic invokes the Hill-Climbing subroutine only when a new solution provides better (or similar) bandwidth than the best solution found so far. Furthermore, by controlling the number of calls to the local search procedure, the number of iterations, and the number of calls to the Node Centroid subroutine, the FNCHC+ heuristic yielded satisfactory results at short running times when applied to large-scale instances.

The results showed that the FNCHC+ heuristic provided, in general, better bandwidth results at shorter running times than the GPS algorithm when applied to 32 large-scale matrix testbeds. For this reason, one can conclude that the FNCHC+ heuristic provides a high rate of bandwidth reduction of large-scale matrices at a reasonable running time. Thus, one can consider that the FNCHC+ heuristic is currently the state-of-the-art metaheuristic algorithm for the bandwidth reduction of large-scale matrices.

Researchers solve very large-scale problems today with the use of parallel computations. Massive problems in the present day are in the order of billions of degrees of freedom [see Gonzaga de Oliveira & Carvalho (2022) and reference therein]. A future study will investigate the influence of orderings in parallel implementations using OpenMP and Message Passing Interface systems. Another future step in the study is a systematic review of parallel heuristics for bandwidth reductions.

References

AMPARORE EG, BECCUTI M & DONATELLI S. 2017. Gradient-based variable ordering of decision diagrams for systems with structural units. In: D'SOUZA D & KUMAR KN (Eds.), *ATVA 2017. Lecture Notes in Computer Science*, vol. 10482. pp. 184–200. Cham: Springer.

BALAJI V & LUCIA B. 2018. When is Graph Reordering an Optimization? Studying the Effect of Lightweight Graph Reordering Across Applications and Input Graphs. In: *IEEE International Symposium on Workload Characterization (IISWC)*. pp. 203–214. Raleigh, NC, USA: IEEE.

BARIK R, MINUTOLI M, HALAPPANAVAR M, TALLENT NR & KALYANARAMAN A. 2020. Vertex Reordering for Real-World Graphs and Applications: An Empirical Evaluation. In: *2020 IEEE International Symposium on Workload Characterization (IISWC)*. pp. 240–251. Beijing, China: IEEE.

BERRY MW, HENDRICKSON B & RAGHAVAN P. 1996. Sparse matrix reordering schemes for browsing hypertext. In: RENEGAR J, SHUB M & SMALE S (Eds.), *Lectures in Applied Mathematics*, vol. 32. pp. 99–123. T. lhe Mathematics of Numerical Analysis.

BOLANOS ME, AVIYENTE S & RADHA H. 2012. Graph entropy rate minimization and the compressibility of undirected binary graphs. In: *Proceedings of the IEEE Statistical Signal Processing Workshop (SSP)*. pp. 109–112. Ann Arbor, MI: IEEE.

CHAGAS GO & GONZAGA DE OLIVEIRA SL. 2015. Metaheuristic-based heuristics for symmetric-matrix bandwidth reduction: a systematic review. *Procedia Computer Science (ICCS 2015 International Conference on Computational Science)*, **51, 2015**: 211–220.

CHEN HK. 2017. Evaluation of triangular mesh layout techniques using large mesh simplification. *Multimedia Tools and Applications*, **76**(23): 25391–25419.

CHINN P, CHVATALOVA J, DEWDNEY AK & GIBBS NE. 1982. The Bandwidth Problem for Graphs and Matrices - a Survey. *Journal of Graph Theory*, **6**: 223–254.

CONCAS A, FENU C & RODRIGUEZ G. 2019. PQser: a Matlab package for spectral seriation. *Numerical Algorithms*, **80**(3): 879–902.

CZIBULA G, CRIŞAN GC, PINTEA CM & CZIBULA IG. 2013. Soft computing approaches on the bandwidth problem. *Informatica*, **24**(2): 169–180.

DAVIS TA & HU Y. 2011. The University of Florida sparse matrix collection. *ACM Transactions on Mathematical Software*, **38**(1): 1–25.

ESPOSITO A, CATALANO MS, MALUCELLI F & TARRICONE L. 1998. Sparse matrix bandwidth reduction: Algorithms, applications and real industrial cases in electromagnetics, high performance algorithms for structured matrix problems. *Advances in the theory of Computation and Computational Mathematics*, **2**: 27–45.

GEORGE A & LIU JW. 1981. *Computer solution of large sparse positive definite systems*. Englewood Cliffs: Prentice-Hall.

GEORGE A & LIU JWH. 1979. An implementation of a pseudoperipheral node finder. *ACM Transactions on Mathematical Software*, **5**(3): 284–295.

GIBBS NE, POOLE WG & STOCKMEYER PK. 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 SL, ABREU AAAM, ROBAINA DT & KISCHNHEVSKY M. 2016. A new heuristic for bandwidth and profile reductions of matrices using a self-organizing map. In: GERVASI O, MURGANTE B, MISRA S, ROCHA AMACV, TORRE CM, TANIAR D, APDUHAN BO, STANKOVA E & WANG S (Eds.), *The 16th International Conference on Computational Science and Its Applications (ICCSA), Lecture Notes in Computer Science, Part I, v. 9786*. pp. 54–70. Beijing: Springer.

GONZAGA DE OLIVEIRA SL, ABREU AAAM, ROBAINA DT & KISCHNHEVSKY M. 2017. An evaluation of four reordering algorithms to reduce the computational cost of the Jacobi-preconditioned Conjugate Gradient Method using high-precision arithmetic. *International Journal of Business Intelligence and Data Mining*, **12**(2): 190–209.

GONZAGA DE OLIVEIRA SL, ABREU AAAM, ROBAINA DT & KISCHNHEVSKY M. 2018a. Finding a starting vertex for the reverse Cuthill-McKee method for bandwidth reduction: a comparative analysis using asymmetric matrices. In: O GERVASI ET AL (Ed.), *The 18th International Conference on Computational Science and Its Applications (ICCSA), Lecture Notes in Computer Science*, vol. 10960. pp. 123–137. Cham: Springer International Publishing.

GONZAGA DE OLIVEIRA SL, BERNARDES JAB & CHAGAS GO. 2018b. An evaluation of low-cost heuristics for matrix bandwidth and profile reductions. *Computational & Applied Mathematics*, **37**(2): 1412–1471.

GONZAGA DE OLIVEIRA SL, BERNARDES JAB & CHAGAS GO. 2018c. An evaluation of reordering algorithms to reduce the computational cost of the incomplete Cholesky-conjugate gradient method. *Computational & Applied Mathematics*, **37**(3): 2965–3004.

GONZAGA DE OLIVEIRA SL & CARVALHO C. 2022. Metaheuristic algorithms for the bandwidth reduction of large-scale matrices. *Journal of Combinatorial Optimization*, **43**: 727–784.

GONZAGA DE OLIVEIRA SL & CHAGAS GO. 2015. A systematic review of heuristics for symmetric-matrix bandwidth reduction: methods not based on metaheuristics. In: *Proceedings of the Brazilian Symposium on Operations Research (SBPO 2015)*. Pernambuco, Brazil: Sobrapo.

GONZAGA DE OLIVEIRA SL & SILVA LM. 2019. Evolving reordering algorithms using an ant colony hyperheuristic approach for accelerating the convergence of the ICCG method. *Engineering with Computers*, **36**(4): 1857–1873.

GONZAGA DE OLIVEIRA SL & SILVA LM. 2020. An ant colony hyperheuristic approach for matrix bandwidth reduction. *Applied soft computing*, **94**: 106434.

GONZAGA DE OLIVEIRA SL & SILVA LM. 2021. Low-cost heuristics for matrix bandwidth reduction combined with a Hill-Climbing strategy. *RAIRO - Operations Research*, **55**(4): 2247–2264.

GUAN J, LIN G & FENG HB. 2019. Ant colony optimisation with local search for the bandwidth minimisation problem on graphs. *International Journal of Intelligent Information and Database Systems*, **12**(1-2): 65–78.

HIGHAM DJ. 2003. Unravelling small world networks. *Journal of Computational and Applied Mathematic*, **158**(1): 61–74.

KAVEH A & BIJARI S. 2015. BANDWIDTH OPTIMIZATION USING CBO AND ECBO. *Asian Journal of Civel Engineering*, **16**(4): 535–545.

KAVEH A & SHARAFI P. 2009. Nodal ordering for bandwidth reduction using ant system algorithm. *Engineering Computations*, **26**: 313–323.

KAVEH A & SHARAFI P. 2012. Ordering for bandwidth and profile minimization problems via charged system search algorithm. *IJST Transactions of Civil Engineering*, **36**: 39–52.

KOOHESTANI B. 2020. A graph representation for search-based approaches to graph layout problems. *International Journal for Computational Methods in Engineering*, **21**(3): 429–436.

KOOHESTANI B. 2022. On the solution of the graph bandwidth problem by means of search methods. *Applied Intelligence*, **53**: 7988–8004.

KOOHESTANI B & 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*. pp. 93–106. London, UK: Springer London.

LIM A, RODRIGUES B & XIAO F. 2006. Heuristics for matrix bandwidth reduction. *European Journal of Operational Research*, **174**(1): 69–91.

LIM A, RODRIGUES B & XIAO F. 2007. A fast algorithm for bandwidth minimization. *International Journal on Artificial Intelligence Tools*, **16**(03): 537–544.

LIVESLEY RK. 1960. The analysis of large structural systems. *The Computer Journal*, **3**(1): 34–39.

MAFTEIU-SCAI L, MAFTEIU E & MAFTEIU-SCAI R. 2019. Brain Storm Optimization Algorithms for Solving Equations Systems. In: CHENG S & SHI Y (Eds.), *Brain Storm Optimization Algorithms – Concepts, Principles and Applications*. pp. 189–220. A. ldaptation, Learning, and Optimization, v. 23. Y. S. Ong and M.-H. Lim series editors.

MAFTEIU-SCAI LO, MAFTEIU-SCAI E & VOINA T. 2017. Bandwidths Optimization on Sparse Matrices Using Brain Storm Optimization. In: *19th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. pp. 219–224. Timisoara, Romania: IEEE.

MARICHAL R, DUFRECHOU E & EZZATTI P. 2021. Optimizing Sparse Matrix Storage for the Big Data Era. In: NAIOUF M, RUCCI E, CHICHIZOLA F & DE GIUSTI L (Eds.), *Communications in Computer and Information Science (JCC-BD&ET 2021: Cloud Computing, Big Data & Emerging Topics)*, vol. 1444. pp. 121–135. Cham: Springer.

MLADENOVIC N, UROSEVIC D, PÉREZ-BRITO D & GARCÍA-GONZÁLEZ CG. 2010. Variable neighbourhood search for bandwidth reduction. *European Journal of Operational Research*, **200**: 14–27.

MUELLER C, MARTIN B & LUMSDAINE A. 2007. A comparison of vertex ordering algorithms for large graph visualization. In: *Proceedings of the 6th International Asia-Pacific Symposium on Visualization (APVIS'07)*. pp. 141–148. Sydney, Australia.

PAPADIMITRIOU CH. 1976. The NP-completeness of bandwidth minimization problem. *Computing Journal*, **16**: 177–192.

PIÑANA E, PLANA I, CAMPOS V & MARTÍ R. 2004. GRASP and path relinking for the matrix bandwidth minimization. *European Journal of Operational Research*, **153**(1): 200–210.

PINTEA CM, CRIŞAN GC & CHIRA C. 2010. A Hybrid ACO Approach to the Matrix Bandwidth Minimization Problem. In: GRAÑA ROMAY M, CORCHADO E & GARCIA SEBASTIAN M (Eds.), *Hybrid Artificial Intelligence Systems*, vol. 6076 of *Lecture Notes in Computer Science*. pp. 405–412. San Sebastián, Spain: Springer Berlin Heidelberg.

PINTEA CM, CRIŞAN GC & CHIRA C. 2012. Hybrid ant models with a transition policy for solving a complex problem. *Logic Journal of IGPL*, **20**(3): 560–569.

POP P, MATEI O & COMES CA. 2014. Reducing the bandwidth of a sparse matrix with a genetic algorithm. *Optimization: A Journal of Mathematical Programming and Operations Research*, **63**(12).

RODRIGUEZ-TELLO E, JIN JHH & TORRES-JIMENEZ J. 2008. An improved simulated annealing algorithm for bandwidth minimization. *European Journal of Operational Research*, **185**: 1319–1335.

SILVA PHG, BRANDÃO DN, MORAIS IS & GONZAGA DE OLIVEIRA SL. 2020. A Biased Random-Key Genetic Algorithm for Bandwidth Reduction. In: GERVASI O, MURGANTE B, MISRA S, GARAU C, BLECIC I, TANIAR D, APDUHAN BO, ROCHA AMAC, TARANTINO E, TORRE CM & KARACA Y (Eds.), *20th International Conference on Computational Science and its Applications - ICCSA. Proceedings, Part VI*, vol. 12254. pp. 312–321. L. lecture Notes in Computer Science. Theoretical Computer Science and General Issues.

TORRES-JIMENEZ J, IZQUIERDO-MARQUEZ I, GARCIA-ROBLEDO A, GONZALEZ-GOMEZ A, BERNAL J & KACKER RN. 2015. A dual representation simulated annealing algorithm for the bandwidth minimization problem on graphs. *Information Sciences*, **303**: 33–49.

How to cite

GONZAGA DE OLIVEIRA SL. 2023. An evaluation of heuristic methods for the bandwidth reduction of large-scale graphs. *Pesquisa Operacional*, **43**: e268255. doi: 10.1590/0101-7438.2023.043.00268255.