Iterated Local Search with Tabu Search for the Bandwidth Reduction Problem in Graphs

Alexandre Augusto Alberto Moreira de Abreu[1,2,3(✉)] and Sanderson L. Gonzaga de Oliveira[2]

[1] Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, Canoinhas, SC, Brazil
[2] Universidade Federal de São Paulo, São José dos Campos, SP, Brazil
{alexandre.abreu,sanderson.oliveira}@unifesp.br
[3] Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil

Abstract. This paper addresses the bandwidth reduction problem in graphs, which is relevant in several applications, such as reducing memory consumption and computational cost in solving systems of linear equations. This problem consists of renumbering the vertices of a graph so that the difference between the labels of adjacent vertices is as low as possible. This paper shows a novel hybrid method based on the Iterated Local Search and Tabu Search (ILSTS) metaheuristics. The paper conducted the experiments using several graphs from the SuiteSparse Matrix Collection and compared the results with the current state-of-the-art Dual Representation Simulated Annealing (DRSA). The obtained results highlight that, although ILSTS demonstrates promising outcomes in terms of bandwidth reduction while maintaining low computational cost, it does not achieve competitive results compared to the quality of the solution delivered by DRSA.

Keywords: Bandwidth reduction problem · Graph · Metaheuristics

1 Introduction

The bandwidth reduction problem is a well-known and relevant graph layout problem. Researchers originally applied this problem to accelerate sparse matrix computations. The bandwidth reduction problem is to label the n vertices of a graph with distinct integers ranging from 1 to n such that the new layout reduces the maximum absolute difference between the labels of adjacent vertices. One can also use symmetric matrices to determine the problem. A method reorders the rows and columns of the matrix such that non-null elements are as close as possible to the main diagonal [25]. Bandwidth reduction is relevant in several

Supported by Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina and Universidade Federal de São Paulo, Brazil.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
O. Gervasi et al. (Eds.): ICCSA 2024, LNCS 14813, pp. 125–136, 2024.
https://doi.org/10.1007/978-3-031-64605-8_9

applications, such as hypertext navigation, small-world networks, analysis of data sets using visual similarity matrices, entropy rate minimization in graphs, symbolic model checking, mesh layout optimization, and the seriation problem [18].

This paper proposes a hybrid heuristic that explores the characteristics of both Iterated Local Search [1,2] and Tabu Search [10] metaheuristics. The computational experiment compares the results of the new metaheuristic algorithm with the DRSA heuristic [41] in 82 sparse matrices taken from the SuiteSparse Matrix Collection [5].

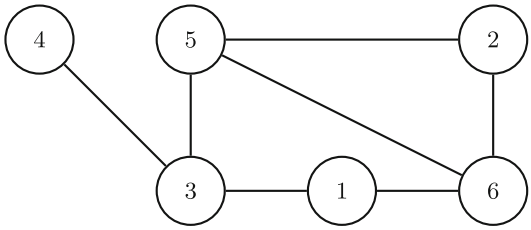
We describe the bandwidth reduction problem and some basic definitions in Sect. 2. Section 3 reviews recent approaches to bandwidth reduction, including the state-of-the-art methods and the most commonly used method in practical applications. Section 4 describes the proposed hybrid metaheuristic. Section 5 describes the data used and the conducted experiments. Finally, we present the results, analysis, conclusions, and future work in Sects. 6 and 7.

2 Bandwidth Reduction

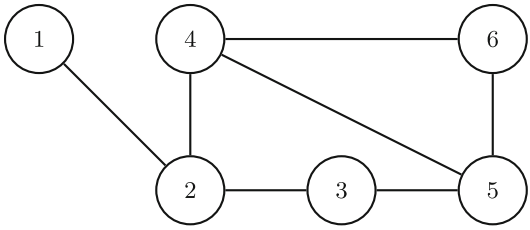
One can define the bandwidth of a matrix as the maximum distance of a non-zero element to the main diagonal. The bandwidth β of an $n \times n$ symmetric matrix $A = [a_{ij}]$ can be defined as $\beta = \beta(A) = \max_{1 \leq i \leq n} [\beta_i(A)]$, where $\beta_i(A) = i - \min_{1 \leq j \leq i, a_{ij} \neq 0} [j]$ represents the bandwidth of the i-th row of matrix A. Equivalently, one can directly determine the bandwidth for an undirected graph $G = (V, E)$, where V is a set of n vertices and E is a set of edges, corresponding to the symmetric matrix A, by considering a function $s : V \rightarrow \{1, 2, \ldots, n\}$ that labels the vertices, where $|V| = n$. Thus, a vertex labeling $S = \{s(v_1), s(v_2), \ldots, s(v_n)\}$ is defined as $\beta = \beta_S(G) = \max_{(u,v) \in E} [|s(u) - s(v)|]$.

In simple terms, the problem of bandwidth minimization in a matrix involves rearranging the non-zero coefficients of the matrix to be closer to the main diagonal. When considering the corresponding graph, the objective is to renumber the graph's vertices without altering the adjacencies, aiming to minimize $\beta(s)$. The goal is to swap the vertex labels to minimize the maximum difference between the new labels of adjacent vertices. For example, in Fig. 1, the label swapping in a graph G with $|V| = 6$ and $|E| = 7$ is presented, reducing the initial bandwidth $\beta = 5$ to $\beta = 2$. In the original graph, as shown in Fig. 1(a), $\beta(s) = \max[1, 2, 2, 3, 1, 5, 4] = 5$, considering the edges $\{3, 4\}, \{3, 5\}, \{1, 3\}, \{2, 5\}, \{5, 6\}, \{1, 6\}, \{2, 6\}$, respectively. After the new labeling, presented in Fig. 1(b), $\beta(s) = \max[1, 2, 1, 2, 1, 2, 1] = 2$, considering the edges $\{1, 2\}, \{2, 4\}, \{2, 3\}, \{4, 6\}, \{4, 5\}, \{3, 5\}, \{5, 6)$, respectively.

With bandwidth reduction, it is possible to optimize memory storage [6]. Additionally, one can obtain solutions for linear systems of equations with lower computational costs [16]. Therefore, over the last decades, hundreds of heuristic methods [3,15] have been developed and applied to the bandwidth reduction problem, as the bandwidth reduction problem is known to be $\mathcal{NP}$-Hard [33]. For a graph $G = (V, E)$ with $|V| = n$, there are $n!$ possible labelings.



(a) Original graph with $\beta = 5$.



(b) Final graph with $\beta = 2$.

Fig. 1. Example of bandwidth reduction from $\beta = 5$ to $\beta = 2$.

3 Related Works

The development of metaheuristic algorithms for bandwidth reduction began in the 1990 s [3]. Researchers have designed heuristics with the most different metaheuristics, including GRASP-PR [36], genetic algorithm [4,27,37], simulated annealing [38,41], ant colony optimization [4,16,19,20,22,34,35], variable neighborhood search [32], genetic programming [26], charged system search algorithm [23], colliding bodies optimization [21], brain storm optimization [28,29], biased random-key genetic algorithm [39], and iterated local search [17].

Silva et al. [39] proposed an approach grounded on the metaheuristic called Biased Random-Key Genetic Algorithm (BRKGA) [11]. BRKGA uses a permutation vector and vectors of random real numbers called keys to represent possible solutions. A deterministic decoding algorithm associates a solution vector with a feasible solution, allowing the algorithm to calculate the fitness function for the problem. The method is biased because one of the parents in the genetic algorithm is always selected from the elite group and has a high probability of passing its genes on [39]. The authors compared the results obtained with BRKGA with the metaheuristics Dual Representation Simulated Annealing (DRSA) [41], GRASP, and RCM. The results indicated that BRKGA did not perform as well as DRSA for bandwidth reduction but was better than the GRASP and RCM [7] approaches. DRSA dominated the other methods.

A previous investigation [17] proposed a heuristic method based on the iterated local search metaheuristic. The authors referred to the algorithm as the ILS-band heuristic. Gonzaga de Oliveira et al. [12,17] proposed a method

named FNCHC+ heuristic. The FNCHC+ heuristic dominated the VNS-band [32] (and, hence, DRSA), and ILS-band heuristics concerning the solution quality and running times when applied to large-scale matrices [17], i.e., with matrices with dimensions higher than one million. The FNCHC+ heuristic also surpassed the resulting heuristics from the ACHH approach [19] regarding the quality solution of large-scale matrices. Additionally, the FNCHC+ heuristic outperformed GPS [9] concerning the solution quality and running times when applied to large-scale matrices [12].

One of the most recent works published by Koohestani [25] presents a study on uninformed search methods, such as breadth-first search and depth-first search. The author also analyzed informed search methods such as Tabu Search and Simulated Annealing, in which information about the problem was used to find more promising solutions. Koohestani [25] conducted numerical experiments to compare his proposal in a Genetic Algorithm implementation, called GA-GBP, with the Biased Random-Key Genetic Algorithm (BRKGA) [39], Greedy Randomized Adaptive Search Procedure (GRASP) [36], and the classic Reverse Cuthill-Mackee (RCM) method [7]. Koohestani [25] obtained the results for BRKGA, GRASP, and RCM from Silva et al. [39]. The author concluded that his proposal provided better guidance in the search employed by GA-GBP.

One could consider the proposal of Rodriguez-Tello et al. [38] the state-of-the-art method until 2010 when Mladenovic et al. [32] proposed the Variable Neighborhood Search for bandwidth reduction (VNS-band), which became the new state of the art until 2015. Gonzaga de Oliveira et al. [13,14] used a one-dimensional self-organizing map for reducing bandwidth and profile in directed and undirected graphs. The publication included experimental comparisons between this proposed method and the VNS-band heuristic. The experimentation involved 113 matrices from the SuiteSparse Matrix Collection and two sets of instances comprising linear systems with sparse symmetric positive-definite matrices that were solved using the Jacobi-preconditioned Conjugate Gradient Method. The results indicated that the VNS-band heuristic outperformed the proposed heuristic.

The DRSA metaheuristic [41], from 2015, differs from the original Simulated Annealing proposed by Kirkpatrick et al. [24] by maintaining a dual internal representation for the current problem solution and implementing three perturbation operations that allow a balance between searching for unknown solutions and exploring known and promising solutions [41]. In its dual representation, DRSA uses a vector of positive integers $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$, where π_i represents the new label of vertex i, and another vector of positive integers $\rho = (\rho_1, \rho_2, \cdots, \rho_n)$, where ρ_i represents the vertex to which label i is assigned. Torres-Jimenez et al. [41] defined three perturbation operations for each vector. Whenever a vector is modified, DRSA propagates the change to the other to maintain consistency between the two representations. The first and second operations are applied to the π vector and swap two positions i and j, randomly selected. In the second operation, the selected vertices must be adjacent. The third operation is a rotation applied to the ρ vector. It selects two positions, i

and j, with $i < j$. DRSA moves the element in position i to j and shifts to the left the elements between positions $i + 1$ and j. The authors conducted experiments with various parameter variations and compared their results with the metaheuristics GRASP [36], classic Simulated Annealing [38], and VNS-band [32]. After using the Wilcoxon test to evaluate their results, Torres-Jimenez et al. [41] presented a statistically significant difference in favor of DRSA, which has been considered the state-of-the-art method in very small matrices (with sizes up to approximately 1000) since then.

4 Iterated Local Search with Tabu Search

The proposed hybrid heuristic explores the characteristics of both Iterated Local Search [1,2] and Tabu Search [10] metaheuristics. The details of the heuristic, named here as Iterated Local Search with Tabu Search (ILSTS), are presented in Algorithm 1. In line 1, r_0 receives the initial labeling of vertices through a constructive method. The constructive method used is the Reverse Cuthill-Mackee method [7] starting with a pseudo-peripheral given by the George-Liu algorithm [8].

Algorithm 1: *Iterated Local Search with Tabu Search (ILSTS)*

input : graph $G = (V, E)$, maximum number of iterations $iter_max$, and tabu list size k.
output: bandwidth β

1 $r_0 \leftarrow G.initial_labeling()$;
2 $tabu_list.size \leftarrow k$;
3 $iter \leftarrow 0$;
4 $level \leftarrow 1$;
5 $r \leftarrow G.local_search(r_0)$;
6 $\beta \leftarrow G.calculate_bandwidth(r)$;
7 **while** $iter < iter_max$ **do**
8 $\quad r' \leftarrow G.perturbation(r, level, tabu_list)$;
9 $\quad r'' \leftarrow G.local_search(r')$;
10 $\quad$ **if** $G.calculate_bandwidth(r'') < G.calculate_bandwidth(r)$ **then**
11 $\quad\quad r \leftarrow r''$;
12 $\quad\quad \beta \leftarrow G.calculate_bandwidth(r'')$;
13 $\quad\quad iter \leftarrow 0$;
14 $\quad\quad level \leftarrow 1$;
15 $\quad$ **else**
16 $\quad\quad level \leftarrow level + 1$;
17 $\quad$ **end if**
18 $\quad iter \leftarrow iter + 1$;
19 **end while**
20 **return** β; `// returns the bandwidth` β

The method initializes the *tabu_list* with size k in line 2. The variables controlling the number of iterations and the perturbation level are set to their initial values in lines 3 and 4, respectively.

Line 5 obtains the labeling r from a local search on r_0, and β receives the bandwidth considering this labeling in line 6. The local search method used is the *hill climbing* strategy proposed by Martí et al. [30], which uses the concept of critical vertex set defined by $C(S) = \{v : \beta_S(v) = \beta_S(G)\}$, where $\beta_s(v) = \max_{\{v,u\} \in E}[|s(v) - s(u)|]$ is the bandwidth of vertex v, and s is the labeling function of the vertices in G. It means that a vertex is critical if its bandwidth is equal to the bandwidth of G. The local search consists of exchanging the label of each critical vertex with the labels of vertices $u \in N'(v)$ in ascending order of the value $|mid(v) - s(u)|$ until it finds a better solution. The set of candidates to be exchanged with vertex v is defined by $N'(v) = \{u : |mid(v) - s(u)| < |mid(v) - s(v)|\}$, where $mid(v) = \lfloor \frac{\max[v]+\min[v]}{2} \rfloor$, $\max[v] = \max_{\{v,u\} \in E}[s(u)]$, and $\min[v] = \min_{\{v,u\} \in E}[s(u)]$.

The ILSTS heuristic executes lines 7 to 19 a maximum of *iter_max* times. The perturbation presented in Algorithm 2 is invoked in line 8. Thus, the method obtains a new candidate solution r'. The algorithm then applies the previously mentioned local search again in line 9. The criterion of acceptance, in line 10, checks if there is a reduction in the bandwidth for the new labeling r'' compared to labeling r. If this is the case, r'' becomes the new incumbent solution, and the values of β, *iter*, and *level* are updated. If not, the method increases the perturbation level in line 16. Afterward, the algorithm updates the iteration counter *iter* in line 18. Finally, at the end of the algorithm, in line 20, the found bandwidth β is returned.

The local search method allows for the improvements' intensification in the incumbent solution. In contrast, the approach attains diversification of ILSTS through the perturbation method outlined in the Algorithm 2, employing a tabu list to prevent the re-exploration of previously visited solutions. Initially, r_0 receives the incumbent solution r, the variable *iter* is initialized, and the method assigns the critical vertices $C(r_0)$ to the vector *critical*, in lines 1, 2, and 3, respectively.

The method repeats lines 4 to 12 while $iter \leq level$. Line 5 randomly chooses a critical vertex u. Another vertice v is randomly selected from vertices of G in line 7. The vertices u and v do not need to be adjacent, but $\{u, v\}$ cannot be in the tabu list. The method creates the tabu list with a HashMap data structure that allows insertion and access in $O(1)$.

The method exchanges labels of vertices u and v to generate a new candidate solution r' in line 9. In line 10, $\{u, v\}$ is appended to the *tabu_list*, and *iter* is incremented in line 11. Finally, in line 13, the candidate solution r' is returned.

Algorithm 2: *Perturbation method.*

input : labeling vector r, perturbation *level*, and *tabu_list*.
output: vector r' for labeling the vertices of G.

$r_0 \leftarrow r$;
$iter \leftarrow 1$;
$critical \leftarrow G.critical_vertices(r_0)$;
while $iter \leq level$ **do**
  $u \leftarrow random_choice(critical)$;
  repeat
    $v \leftarrow random_choice(G.V \backslash \{u\})$;
  until $\{u, v\} \notin tabu_list$;
  $r' \leftarrow G.swap_labels(s(u), s(v))$;
  $tabu_list.Insert(\{u, v\})$;
  $iter \leftarrow iter + 1$;
end while
return r'; `// returns the labeling vector` r'

5 Description of the Data and Tests

Publicly available matrix instances from the SuiteSparse Matrix Collection [5] were used in the computational experiments. Torres-Jimenez et al. [41] conducted tests with 33 matrices considered tiny and 82 matrices with dimensions ranging from 207 to 1104. In this paper, we used these 82 matrices extracted from the SuiteSparse Matrix Collection to evaluate the results yielded by the hybrid metaheuristic algorithm proposed in this study.

We implemented the ILSTS method in the Rust programming language [31] version 1.69.0 and utilized a memory representation scheme known as Compressed Sparse Row (CSR) [40] or Yale format. We conducted all computational experiments with ILSTS on an Acer Aspire 3 computer with an Intel® Core™ i3-8130U 2.200 GHz processor, 8 GiB of main memory DDR4 1600 MHz, and GNU/Linux Deepin operating system with the Kernel 5.15.77-amd64.

After empirical testing, we established the parameters $iter_max = 200$ and $k = 50$ for ILSTS with all instances. The findings of DRSA reported by Torres-Jimenez et al. [41] were used to evaluate the results yielded by ILSTS.

6 Results and Analysis

Table 1 presents the results obtained by ILSTS and DRSA for the 82 instances used by Torres-Jimenez et al. [41]. Columns n and β_0 refer to the number of vertices and initial bandwidth of the instances. The table presents the execution times (t) of ILSTS in seconds.

Despite demanding reduced processing time compared to DRSA [41], ILSTS outperformed DRSA in only two instances (nos3 and gr_30_30) and achieved identical results in five matrices (nos1, saylr1, steam2, nos2, and sherman4). In

Table 1. Results for bandwidth reduction using DRSA and ILSTS.

Instance	n	β_0	DRSA	ILSTS	t	Instance	n	β_0	DRSA	ILSTS	t
impcol_a	207	167	32	130	0.13	pores_3	532	77	13	15	0.91
dwt_209	209	184	23	163	0.15	fs_541_1	541	540	70	537	0.57
gre_216a	216	36	21	36	0.17	dwt_592	592	259	29	189	0.47
dwt_221	221	187	13	150	0.17	steam2	600	331	63	**63**	18.86
impcol_e	225	92	42	92	0.28	west0655	655	564	157	562	0.73
dw_234	234	48	11	45	0.32	662_bus	662	335	39	334	0.41
nos1	237	4	3	**3**	0.38	shl_0	663	661	224	632	0.77
saylr1	238	14	14	**14**	0.40	shl_200	663	661	233	654	0.79
steam1	240	146	44	50	1.82	shl_400	663	662	229	613	1.56
dwt_245	245	115	21	114	0.16	nnc666	666	262	40	177	0.94
nnc261	261	64	24	36	0.51	nos6	675	30	16	30	0.32
bcspwr04	274	265	24	248	0.19	fs_680_1	680	600	17	600	0.35
ash292	292	24	19	24	0.22	685_bus	685	550	32	471	0.49
can_292	292	282	38	215	0.31	can_715	715	611	71	456	0.98
dwt_310	310	28	12	14	0.22	nos7	729	81	65	81	0.46
gre_343	343	49	28	49	0.30	fs_760_1	760	740	37	185	0.78
dwt_361	361	50	14	48	0.28	mcfe	765	187	126	187	17.79
plat362	362	249	34	248	1.01	bcsstk19	817	567	14	514	0.62
plskz362	362	248	18	247	0.27	bp_0	822	820	234	774	2.00
str_0	363	359	115	307	0.62	bp_1000	822	820	283	776	3.26
str_200	363	359	124	322	0.96	bp_1200	822	820	287	781	2.44
str_600	363	359	131	320	1.40	bp_1400	822	820	291	774	2.70
west0381	381	363	149	323	0.46	bp_1600	822	820	292	778	2.94
dwt_419	419	356	25	329	0.53	bp_200	822	820	257	783	6.07
bcsstk06	420	47	45	47	1.24	bp_400	822	820	267	787	3.08
bcsstm07	420	47	45	47	1.01	bp_600	822	820	272	777	2.51
impcol_d	425	406	39	213	0.35	bp_800	822	820	278	775	2.67
hor_131	434	421	54	72	1.99	can_838	838	837	86	624	1.19
bcspwr05	443	435	27	415	0.18	dwt_878	878	519	25	458	0.87
can_445	445	403	52	329	0.45	orsirr_2	886	554	85	113	1.79
nos5	468	178	63	178	0.59	gr_30_30	900	31	33	**31**	0.84
west0479	479	388	119	386	0.44	dwt_918	918	839	32	763	0.83
bcsstk20	485	20	13	20	0.28	jagmesh1	936	778	27	643	0.69
494_bus	494	428	28	403	0.22	nos2	957	4	3	**3**	0.42
mbeacxc	496	490	260	486	127.45	nos3	960	43	44	**43**	1.91
mbeaflw	496	490	260	485	123.47	west0989	989	855	207	855	0.83
mbeause	496	490	254	482	96.03	jpwh_991	991	197	88	175	2.29
west0497	497	416	85	363	0.41	dwt_992	992	513	35	513	3.14
dwt_503	503	452	41	358	0.90	saylr3	1000	100	46	54	1.30
lnsp_511	511	57	44	57	0.51	sherman1	1000	100	46	54	1.29
gre_512	512	64	36	64	0.55	sherman4	1104	368	27	**27**	1.48

2 matrices (bcsstk06, bcsstm07), ILSTS found results 10% close to the results obtained by DRSA. In five instances (steam1, dwt_310, pores_3, saylr3, and sherman1), the results found by ILSTS were worse than those found by DRSA by a margin smaller than 20%. DRSA was considerably better than ILSTS in reducing bandwidth in all other instances. Figure 2 shows that DRSA performed better in most instances, even though there is no relationship between the size of the matrices and the difference between the bandwidths found.

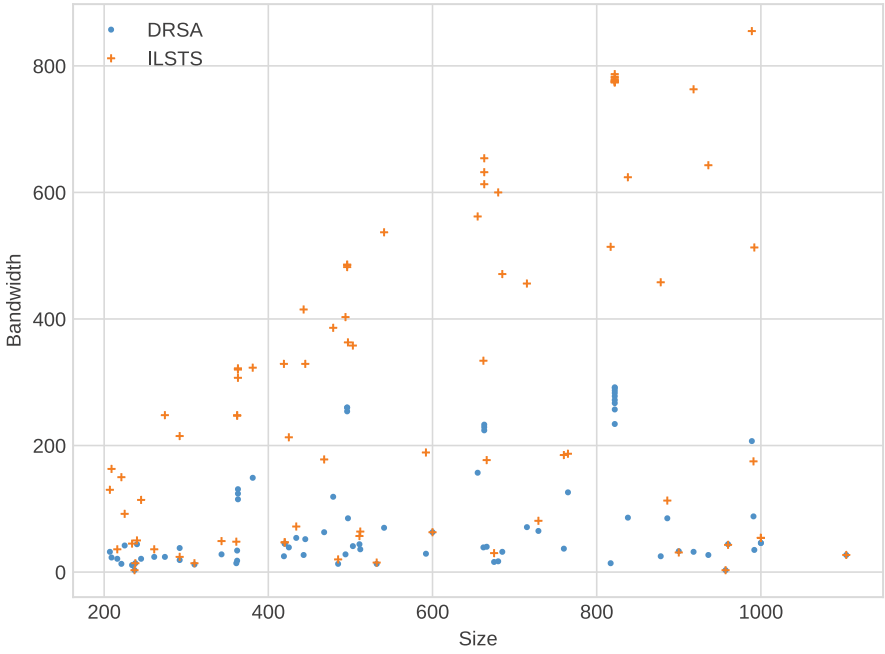


Fig. 2. Comparison of results for bandwidth reduction using DRSA and ILSTS.

7 Conclusions

This paper implemented a hybrid metaheuristic algorithm to leverage the characteristics of Iterated Local Search and Tabu Search (ILSTS). The paper compared the results yielded by the ILSTS with the Dual Representation Simulated Annealing (DRSA) [41], which is the current state-of-the-art metaheuristic algorithm for the bandwidth reduction problem in very small matrices. From the obtained results, we concluded that ILSTS demonstrates promising outcomes in terms of bandwidth reduction while maintaining low computational cost. However, ILSTS does not achieve competitive results when compared with DRSA when one only considers the bandwidth reduction and not the execution time.

Due to its reduced computational cost, ILSTS is a candidate for utilization in situations that require bandwidth reduction with low computational cost. In future work, we intend to evaluate the performance of ILSTS in conjunction with the conjugate gradient method to reduce the computational cost required to solve systems of linear equations. Furthermore, we intend to verify if the ILSTS presents competitive results when applied to large matrices. These future investigations are necessary to establish the potential of ILSTS as a viable alternative approach to bandwidth reduction since there is no record of the previous use of the metaheuristic Iterated Local Search with Tabu Search for this purpose.

Acknowledgment. The authors gratefully acknowledge the support of the FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) and IFSC (Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina).

References

1. Baum, E.B.: Iterated descent: a better algorithm for local search in combinatorial optimization problems. Technical report, Caltech, Pasadena, CA (1986)
2. Baxter, J.: Local optima avoidance in depot location. J. Oper. Res. Soc. **32**(9), 815 (1981)
3. Chagas, G.O., Gonzaga de Oliveira, S.L.: Metaheuristic-based heuristics for symmetric-matrix bandwidth reduction: a systematic review. Procedia Comput. Sci. (ICCS 2015 International Conference on Computational Science) **51**, 211–220 (2015)
4. Czibula, G., Crişan, G.C., Pintea, C.M., Czibula, I.G.: Soft computing approaches on the bandwidth problem. Informatica **24**(2), 169–180 (2013)
5. Davis, T., Hu, Y.: The university of florida sparse matrix collection. ACM Trans. Math. Softw. **38**, 1 (11 2011)
6. Freire, M., Marichal, R., Gonzaga de Oliveira, S.L., Dufrechou, E., Ezzatti, P.: Enhancing the sparse matrix storage using reordering techniques. In: Barrios H., C.J., Rizzi, S., Meneses, E., Mocskos, E., Monsalve Diaz, J.M., Montoya, J. (eds) High Performance Computing, CARLA 2023, CCIS, vol. 1887, pp. 66–76. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-52186-7_5
7. George, A., Liu, J.: Computer Solution of Large Sparse Positive Definite Systems. Prentice-Hall, Englewood Cliffs, New Jersey, USA (1981)
8. George, A., Liu, J.W.H.: An implementation of a pseudoperipheral node finder. ACM Trans. Math. Softw. **5**(3), 284–295 (1979)
9. Gibbs, N.E., Poole, W.G., Stockmeyer, P.K.: An algorithm for reducing the bandwidth and profile of a sparse matrix. SIAM J. Numer. Anal. **13**(2), 236–250 (1976)
10. Glover, F.: Future paths for integer programming and links to artificial intelligence. Comput. Oper. Res. **13**(5), 533–549 (1986)
11. Gonçalves, J.F., Resende, M.G.C.: Biased random-key genetic algorithms for combinatorial optimization. J. Heuristics **17**(5), 487–525 (2011)
12. 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)
13. Gonzaga de Oliveira, S.L., de Abreu, A.A.A.M., Robaina, D., Kischinhevsky, M.: A new heuristic for bandwidth and profile reductions of matrices using a self-organizing map. In: Gervasi, O., et al. (eds.) ICCSA 2016. LNCS, vol. 9786, pp. 54–70. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42085-1_5

14. Gonzaga de Oliveira, S.L., Abreu, A.A.A.M., Robaina, D.T., Kischnhevsky, M.: An evaluation of four reordering algorithms to reduce the computational cost of the Jacobi-preconditioned conjugate gradient method using high-precision arithmetic. Int. J. Bus. Intell. Data Min. **12**(2), 190–209 (2017)
15. Gonzaga de Oliveira, S.L., Chagas, G.O.: 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), Sobrapo, Pernambuco, Brazil, August 2015
16. 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**(4), 1857–1873 (2019)
17. Gonzaga de Oliveira, S.L., Carvalho, C.: Metaheuristic algorithms for the bandwidth reduction of large-scale matrices. J. Comb. Optim. **43**, 727–784 (2022)
18. 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)
19. Gonzaga de Oliveira, S., Silva, L.M.: An ant colony hyperheuristic approach for matrix bandwidth reduction. Appl. Soft Comput. **94**, 106434 (2020)
20. Guan, J., Lin, G., Feng, H.B.: Ant colony optimisation with local search for the bandwidth minimisation problem on graphs. Int. J. Intell. Inf. Database Syst. **12**(1–2), 65–78 (2019)
21. Kaveh, A., Bijari, S.: Bandwidth optimization using CBO and ECBO. Asian J. Civel Eng. **16**(4), 535–545 (2015)
22. Kaveh, A., Sharafi, P.: Nodal ordering for bandwidth reduction using ant system algorithm. Eng. Comput. **26**, 313–323 (2009)
23. Kaveh, A., Sharafi, P.: Ordering for bandwidth and profile minimization problems via charged system search algorithm. IJST Trans. Civil Eng. **36**, 39–52 (2012)
24. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
25. Koohestani, B.: On the solution of the graph bandwidth problem by means of search methods. Appl. Intell. **53**(7), 7988–8004 (2022)
26. 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, SGAI 2011, pp. 93–106. Springer, London (2011). https://doi.org/10.1007/978-1-4471-2318-7_7
27. Lim, A., Rodrigues, B., Xiao, F.: Heuristics for matrix bandwidth reduction. Eur. J. Oper. Res. **174**(1), 69–91 (2006)
28. Mafteiu-Scai, L., Mafteiu, E., Mafteiu-Scai, R.: Brain storm optimization algorithms for solving equations systems. In: Cheng, S., Shi, Y. (eds.) Brain Storm Optimization Algorithms. Adaptation, Learning, and Optimization, vol. 23, pp. 189–220. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-15070-9_8
29. Mafteiu-Scai, L.O., Mafteiu-Scai, E., Voina, T.: 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. IEEE, Timisoara, Romania (2017)
30. Martí, 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)
31. Matsakis, N.D., Klock, F.S.: The rust language. In: Proceedings of the 2014 ACM SIGAda Annual Conference on High Integrity Language Technology, HILT 2014, pp. 103–104. Association for Computing Machinery, New York, NY, USA (2014)

32. Mladenovic, N., Urosevic, D., Pérez-Brito, D., García-González, C.G.: Variable neighbourhood search for bandwidth reduction. Eur. J. Oper. Res. **200**(1), 14–27 (2010)
33. Papadimitriou, C.H.: The NP-completeness of the bandwidth minimization problem. Computing **16**(3), 263–270 (1976)
34. Pintea, C.-M., Crişan, G.-C., Chira, C.: A hybrid ACO Approach to the Matrix Bandwidth Minimization Problem. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS (LNAI), vol. 6076, pp. 405–412. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13769-3_49
35. Pintea, C.M., Crişan, G.C., Shira, C.: Hybrid ant models with a transition policy for solving a complex problem. Log. J. IGPL **20**(3), 560–569 (2012)
36. Piñana, E., Plana, I., Campos, V., Martí, R.: Grasp and path relinking for the matrix bandwidth minimization. Eur. J. Oper. Res. **153**(1), 200–210 (2004)
37. Pop, P., Matei, O., Comes, C.A.: Reducing the bandwidth of a sparse matrix with a genetic algorithm. Optim. A J. Math. Program. Oper. Res. **63**(12), 1851–1876 (2014)
38. Rodriguez-Tello, E., Hao, J.K., Torres-Jimenez, J.: An improved simulated annealing algorithm for bandwidth minimization. Eur. J. Oper. Res. **185**(3), 1319–1335 (2008)
39. Silva, P.H.G., Brandão, D.N., Morais, I.S., Gonzaga de Oliveira, S.L.: A biased random-key genetic algorithm for bandwidth reduction. In: Gervasi, O., et al. (eds.) ICCSA 2020. LNCS, vol. 12249, pp. 312–321. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58799-4_23
40. Tinney, W., Walker, J.: Direct solutions of sparse network equations by optimally ordered triangular factorization. Proc. IEEE **55**(11), 1801–1809 (1967)
41. 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)