https://doi.org/10.24867/SYMOPIS-2024-51-102

Faculty of Technical Sciences
University of Novi Sad
51st Symposium on Operational Research
SYM-OP-IS 2024
Tara, 16-19. 9. 2024.

MAXIMIZING SPECTRAL RADIUS OF GRAPHS IS AN OPTIMIZATION PROBLEM

TATJANA DAVIDOVIĆ[1], DRAGAN STEVANOVIĆ[1], LUKA RADANOVIĆ[2], ABDELKADIR FELLAGUE[3], DRAGUTIN OSTOJIĆ[4]

[1] Mathematical Institute, Kneza Mihaila 36, Belgrade, Serbia, tanjad@mi.sanu.ac.rs, ORCID:0000-0001-9561-5339, dragan_stevanovic@mi.sanu.ac.rs, ORCID: 0000-0003-2908-305X

[2] Faculty of Mathematics, Studentski trg 16, Belgrade, Serbia, lukaradanovich@gmail.com, ORCID: 0009-0007-2135-9732

[3] SIMPA, Department of Computer Science, University of Sciences and Technology of Oran, Algeria, abdelkadir.fellague@univ-usto.dz, ORCID: 0009-0000-6911-9331

[4] Faculty of Science, University of Kragujevac, Serbia, dragutin.ostojic@pmf.kg.ac.rs, ORCID: 0000-0001-6704-353X

Abstract: *Threshold graphs (TGs) are important in graph theory for their special structure and numerous applications. Especially interesting are the TGs that maximize the index, i.e., the largest eigenvalue. Characterization of radius maximizers among connected TG with a given number of vertices and edges in the general case is still an open problem. Therefore, computer enumeration that would help researchers make and prove hypotheses about such graphs' structure seems promising. We consider this enumeration a combinatorial optimization problem and address it using metaheuristic methods, General Variable Neighborhood Search (GVNS) and improvement-based Bee Colony Optimization (BCOi). Preliminary results on moderate-sized examples showed that more systematic searches performed by GVNS performed slightly better than the random modifications utilized within BCOi.*

Our methodology defines the considered problem as an optimization task and utilizes two metaheuristic methods, Variable Neighborhood Search (VNS), which relies on iterative improvements of a single current best solution, and Bee Colony Optimization (BCO), a population-based metaheuristic from the Swarm Intelligence (SI) class. We use compact solution representation and several auxiliary data structures that should enable an efficient search of the solution space. In addition, we define several types of transformations that preserve the feasibility of the resulting solution. The proposed methods are compared on the graphs with a moderate number of vertices. Preliminary results are in favor of the VNS approach, however, we believe that both methods could be improved.

Keywords: *Spectral Graph Theory, Largest Eigenvalue, Building Hypothesis, Combinatorial Optimization, Metaheuristic Methods*

1. INTRODUCTION

Graph theory is a very important discipline of discrete mathematics that studies the mathematical objects (graphs) used to model various problems in medicine, engineering, science, industry, etc. Graphs can be represented in several ways, the most commonly used being the *Adjacency matrix A* [6]. It is a squared binary matrix of dimension n representing the number of vertices with elements $a_{ij} = 1$ if vertices i and j are connected by an edge, and $a_{ij} = 0$ otherwise. Graphs can be studied from the structural or spectral perspective. *Spectral graph theory* (SGT) [7, 15] studies spectral properties of graphs based on their adjacency matrix. More precisely, SGT investigates the *eigenvalues* and *eigenvectors* of adjacency matrix, defined in the following way. *Eigenvalues* λ_i, $i = 1,2,\dots,n$ for the graph G are the roots of its characteristic polynomial $P_G(x) = det(xI - A)$. The set of all eigenvalues of graph G is called *spectrum* and it contains real numbers if G is undirected. The largest eigenvalue of graph G is called *index* or *radius*. An array x such that $Ax = \lambda x$ is known as *eigenvector* (corresponding to the eigenvalue λ) of graph G. It is important to note that some other matrices associated with graphs are defined and analyzed in recent literature, such as Laplacian matrix and signless Laplacian matrix ([7], section 1.3). However, they will not be considered in this paper. SGT has important applications in various domains [9], in particular in computer science [8]. Majority of them include finding extremal graphs, i.e., the graphs that minimize (maximize) some invariant or even the combination of several invariants [1, 3, 5, 11].

The problem considered in this study is referred to as *Maximization of Spectral Radius* (MSR). Its objective is to identify, among all connected graphs with given numbers of vertices n and edges m, the ones that maximize spectral radius (the largest eigenvalue, index). The problem for a general class of graphs (not necessarily connected) was defined in 1976 [2], (p. 438). The first theoretical characterisation results were related to the disconnected graphs. On the other hand, the problem of characterizing connected extremal graphs is still open in the general case. Brualdi and Solheid [4] showed that the adjacency matrix of a connected extremal graph must have a stepwise form, in the sense that its vertices can be ordered in such a way that $A_{ij} = 1$ (with $i < j$) implies $A_{hk} = 1$ for all $h \leq i$, $k \leq j$ and $h < k$. An alternative reasoning, showing that a connected extremal graph cannot have either the path P_4, the cycle C_4 or the pair of independent edges $2K_2$ as an induced subgraph, was suggested in [14]. This actually implies that a connected extremal graph has to be a *threshold graph*. Threshold graphs are attractive for investigation due to their numerous application, especially in medicine, psychology, computer science and many other fields [13].

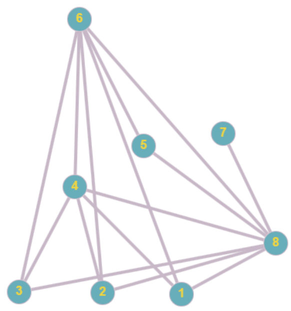
Threshold graphs can be described iteratively as it is proposed in [13]. We start with a single vertex and, in each step, add a new vertex that is either isolated or adjacent to all already included vertices. This process of sequentially building a threshold graph may be written in a more formal way as:

$$G_{p_1} = K_{p_1} \tag{1}$$

$$G_{p_1,p_2,\ldots,p_k} = \overline{G_{p_1,p_2,\ldots,p_{k-1}}} \vee K_{p_k} \tag{2}$$

where $p_1, p_2, \ldots, p_k$, are positive integers, $\overline{G}$ denotes the complement of G and $\vee$ denotes the join of two graphs. This notation compresses successive additions of p_1 vertices of one type (each isolated or each adjacent to all previous vertices), p_2 vertices of the opposite type, p_3 vertices of the first type, etc. Here the complement changes the types of previous vertices, while the join ensures that the p_k vertices added at the last step are adjacent to all previous vertices. Fig. 1 illustrates an example of threshold graph with $n = 8$ vertices and $m = 15$ edges. Given is also the corresponding value of its radius. However, this is not the extremal graph with respect to the radius value, the graph with maximum radius is presented in Fig. 2.

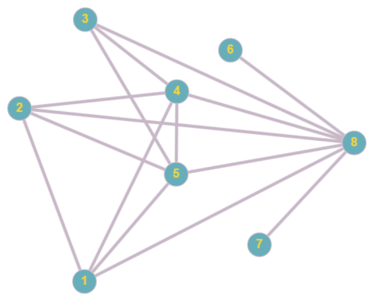
$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$



$n = 8, \ m = 15, \ \lambda_1 = 4.37$

Figure 1 An example of threshold graph

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$



$n = 8, \ m = 15, \ \lambda_1 = 4.52$

Figure 2 Extremal threshold graph with respect to the radius

Although we know that only threshold graphs are valid candidates for the spectral radius maximizer, the theoretical results in a general case are still missing. Exhaustive enumeration of all threshold graphs with a given n and m is an NP-hard problem, especially for the medium number of edges, i.e., for m close to $n(n-1)/4$. Therefore, the application of some incomplete search methods is more than welcome. One of the possible approaches is to use some general purpose software developed for generating conjectures in graph theory, such as GRAPH, 1984 (https://www.mi.sanu.ac.rs/novi_sajt/research/projects/GRAPH.zip), the three versions of AutoGraphiX, 1997, 2009, 2015 (https://www.autographix.ca), newGraph 2004 (https://www.mi.sanu.ac.rs/newgraph/), or maybe PHOEG 2008 (https://phoeg.umons.ac.be/phoeg), to mention a few. Another popular approach is to develop metaheuristic (MH) method, tailored for each particular optimization problem. We illustrate the application of Variable Neighborhood Search (VNS) and Bee Colony Optimization (BCO) metaheuristics. These two methods are representative of two distinct classes of algorithms, mathematical-based singe-solution and nature-inspired population-based, developed by Serbian scientists.

The remainder of this paper is organized as follows. MSR is described as a combinatorial optimization problem in Section 2. The implementation details about developed MH methods, the General Variable Neighborhood Search (GVNS) and Improvement-Based Bee Colony Optimization (BCOi), are given in Section 3. Experimental evaluation and comparison of the proposed methods are presented in Section 4. We provide results obtained by applying the implemented methods to some medium-sized threshold graphs. Concluding remarks and guidelines for future work are given in Section 5.

2. DESCRIPTION OF SPECTRAL RADIUS MAXIMIZATION PROBLEM

MSR can be considered as an optimization problem and treated by various optimization methods. The objective function to be maximized is $\max \lambda_i$, $i = 1, 2, \ldots, n$. To calculate the objective function value is not straightforward, we fist need to determine the spectrum of a given graph (which requires a polynomial number of operations) and then to identify its largest element. The constraints that need to be fulfilled involve the graph connectivity and a threshold property. These may also not be easy to verify.

Although, this problem cannot be formulated as a Mixed-Integer Program (MIP), it is easy to consider it as an combinatorial optimization problem and apply the metaheuristic methods. We decided to apply GVNS and BCOi, as they belong to different classes of algorithms and both have been mostly developed by researches from Serbia.

3. METAHEURISTIC APPROACH TO MSR

This section contains the implementation details of the General Variable Neighborhood Search (GVNS) [12] and Improvement-Based Bee Colony Optimization (BCOi) [10] that we consider the most suitable for the application to the considered problem. To implement any metaheuristic method as efficient as possible, it is necessary to use adequate data structures and to carefully design each step. The main steps in implementing MH methods are the definition of solution representation, auxiliary data structures, procedure for generation of initial solution and rules for transformation of solutions that actually represent neighborhoods of each particular feasible solution.

3.1. Solution representation

We use a compact binary representation for the solution of MSR problem, i.e., we represent a threshold graph on n vertices and m edges by a binary sequence $\mathbf{R} = \{r_1, r_2, \ldots, r_n\}$. Here, $r_i = 0$ means that node i is not connected to any vertex with index smaller than i, (type 0 vertex), while $r_i = 1$ denotes a vertex i that is adjacent to all vertices with index smaller than i (type 1 vertex). The value for r_1 can take both binary values and we decided to set it to 1. On the other hand, the last vertex should be connected to all the remaining vertices, and therefore, $r_n = 1$, because only connected threshold graphs are considered. To control the number of edges in the represented threshold graph, it is necessary that our binary sequence fulfills the following equality

$$\sum_{i=1}^{n} (i-1) \cdot r_i = m. \tag{3}$$

The representation of graph from Fig. 1 is $\mathbf{R}_1 = \{1,0,0,1,0,1,0,1\}$, while the graph from Fig. 2 is represented by the sequence $\mathbf{R}_2 = \{1,1,0,1,1,0,0,1\}$.

3.2. Initial solution generation

The initial solution/population for our MH methods are generated in a greedy manner. We first initialize the solution sequence: The first element by 1, while all remaining elements by 0. In addition, the number of remaining edges is initialized to m. Then we perform a loop that starts from the last element (vertex), and adds edges connecting that vertex to all previous if possible. If it is possible to add these edges, value of the corresponding element of **R** is changed to 1 and the number of remaining edges is updated properly. Otherwise, the value of sequence element and the number of remaining edges are not changed. The search continues from the previous vertex, i.e., the vertex with smaller index. When this loop is completed, a feasible solution of MSR problem is generated.

As BCOi is a population-based method, it is necessary to generate a set of feasible solutions at the beginning of each iteration. This is easily done by performing a random number of stochastic transformations described in the next subsection. However, our preliminary experiments showed such an initial solution is not appropriate. As the elements equal to 1 are concentrated at the end of sequence **R**, the number of possible transformation is small, and therefore, we always obtain similar solutions. A simple modification of the initialization procedure that moves twice across the binary sequence by the increment of 2 resolves this problem.

3.3. Solution transformations

Four transformations yielding to a feasible solution can be defined on the proposed solution representation.

1. Each combination $\{1...01...10...1\}$ can be replaced with $\{1...10...01...1\}$
2. Analogously, $\{1...10...01...1\}$ could be replaced with $\{1...01...10...1\}$
3. If $r_i = 1$, $r_j = r_k = 0$, and $i = j + k$, then it is possible to modify this solution in such a way that $r_i = 0$, $r_j = r_k = 1$
4. Analogously, if $r_i = 0$, $r_j = r_k = 1$, and $i = j + k$, then it is possible to modify this solution in such a way that $r_i = 1$, $r_j = r_k = 0$

The first two transformations preserve the number of elements having values 0 and 1 in the resulting sequences. Therefore, we need also the transformations of the second type (the last two) that enable to modify (increase or decrease) the number of 0 and 1 elements in **R**. As can be noticed from the definition of these transformations, they do not violate the number of edges. To simplify the implementation, the first two transformations are grouped into a single neighborhood, and the remaining two into the second neighborhood. This means that in the implemented GVNS, local improvement procedure, i.e., Variable Neighborhood Descent (VND) uses two neighborhoods. On the other hand, BCOi has a possibility to randomly select among the two types of transformations when it attempts to modify any solution from the population.

The main steps of GVNS and BCOi are performed while the stopping criterion is not fulfilled. The output is the Adjacency matrix (reconstructed from the **R** sequence corresponding to the identified threshold graph with maximum radius) and the spectrum of that graph. As both MH methods are stochastic search methods, it is necessary to repeat executions at least 30 (preferably 100) times (with different seed value) for each particular graph with fixed values for n and m and perform the statistical analysis of the obtained results.

4. COMPUTATIONAL EXPERIMENTS

Our MH methods are implemented in C++, executed on Intel Xeon E5-2620 v3, 2.40GHz, 32 GB RAM Under Linux 4.19.12, and compiled with GCC 4.8.3. To be able to control the experiment and to replicate the results, in the i-th execution we used seed value $n * i + m$. In the preliminary results presented here, 30 repetitions are performed as it is enough to reason about the performance of the proposed MH methods. Stopping criterion is set to 2000 evaluations of the objective function value. Each evaluation assumes the calculation of spectrum for currently examined graph. In every iteration of BCOi, the spectrum is calculated $NC \times B$ times, while the number of objective function calculation in GVNS depends on the number of neighbors in each examined neighborhoods and the search strategy (First- or Best-Improvement). This is the reason to use the number of function evaluations as the stopping criterion.

By the preliminary experimental evaluations we determine the parameter values for both methods as follows. In GVNS $k_{max} = n/2$. It is important to note that it is not always possible to perform the desired number of transformations and the actual distance between the starting and resulting binary sequence after the Shaking step may be smaller than expected. In Shaking, both neighborhood types are applied with the probability 0.5, the Local Search strategy in VND is First-Improvement. In BCOi, $B = 5$, $NC = 10$, $o = rnd(n/3, 2n/3)$. Here, o determines the number of transformations of a single solution in each forward pass and it is determined randomly from the given interval. Note that the distance between the initial and final solution of each transformation can be less than o. The initial population in each (except the first) iteration contains 2 best-so-far solutions (to attempt additional improvements and ensuring intensification of the search) and 3 completely new randomly generated solutions (for the diversification of the search process).

As the test examples we used some medium-size instances with 30 and 50 vertices. Having in mind that, for a fixed value of n, the number of connected (threshold) graphs with respect to m is a bell-shape function, we select the number of edges randomly from the middle of valid interval as we expect there the largest search space.

4.1. Results of experimental evaluation

The obtained results on larger graphs are presented in Table 1:. This table is organized as follows. The names of instances (containing the numbers of vertices and edges) are given in the first column. The second column contains the objective function value (spectral radius) of the initial solution. The next three columns contain the results provided by GVNS, number of best graphs (out of 30 repetitions), the objective function value corresponding to the best graph, and average value of the spectral radius (over 30 trials), respectively. The same data related to the execution of BCOi is provided in the remaining three columns of Table 1:.

Table 1: Comparing the results obtained by GVNS and BCOi on graphs with 30 and 50 vertices

Graph	Init.sol.	GVNS			BCOi		
	Obj.val.	#bests	best obj.	av. obj.	#bests	best obj.	av. obj.
$G_{30,100}$	10.96	30	12.34	12.34	13	12.34	12.10
$G_{30,220}$	18.12	30	20.03	20.03	30	20.03	20.03
$G_{30,300}$	22.16	**30**	23.65	**23.65**	26	23.65	23.64
$G_{30,400}$	27.04	30	27.58	27.58	30	27.58	27.58
$G_{50,100}$	10.38	30	10.87	10.87	30	10.38	10.38
$G_{50,300}$	19.58	**30**	**22.89**	**22.89**	25	22.50	22.19
$G_{50,500}$	26.80	**30**	**30.33**	**30.33**	1	30.18	30.08
$G_{50,1000}$	41.87	30	44.02	44.02	30	44.02	44.02

Comparing the preliminary results from Table 1: we can conclude that GVNS exhibits stable performance as it obtains the same result in all 30 repetitions. It outperforms the best BCOi results in 3 out of 8 tested graph examples. It is also evident that for small and large enough number of edges both algorithms perform equally good, which confirms our assumption that those are the easiest cases. When the number of edges approaches the middle of the examined interval, the search space becomes larger and systematic search performed by GVNS yields better results that the random perturbations among population members in BCOi.

The binary sequences that correspond to the best obtained solutions are as follows.

$\mathbf{R}(G_{30,100}) = \{11111101111100000000000000001\}$
$\mathbf{R}(G_{30,220}) = \{11111111111111111101000000001\}$
$\mathbf{R}(G_{30,300}) = \{11111011111111111111111000001\}$
$\mathbf{R}(G_{30,400}) = \{11111101111111111111111111101\}$
$\mathbf{R}(G_{50,100}) = \{1111011111000000000000000000000000000000000000001\}$
$\mathbf{R}(G_{50,300}) = \{1101111111111111111111100000000000000000000000001\}$
$\mathbf{R}(G_{50,500}) = \{1111111111111101111111111111110000000000000000001\}$
$\mathbf{R}(G_{50,1000}) = \{1111111111111111111111111111111111111011111000001\}$

Analyzing the structure of final solution, the hypothesis can be stated that the largest spectral radius have graphs whose binary sequences contains 1s concentrated at the beginning. This contrasts the greedy procedure for creating initial solution.

5. CONCLUSION

We considered the maximization of spectral radius (MSR) as the combinatorial optimization problem and applied GVNS and BCOi as the incomplete search procedures. Preliminary results on medium-sized graphs are promising, however, they revealed the main challenges that need to be resolved for the application to large test examples. These issues are a large computational complexity of objective function value and a large memory requirements to store complex data structures needed to increase the efficiency of our search procedures. As the possible directions for future research may inclued the additional optimization of our implementation, increasing the set of possible neighborhoods/transformations, the analysis of previously visited solutions and its utilization for learning how to reduce the search space, and careful tuning of metaheuristic parameters.

Acknowledgement

This work has been partially supported by the Science Fund of the Republic of Serbia, Grant #6767, Lazy walk counts and spectral radius of graphs—LZWK, as well as by the Serbian Ministry of Science, Technological Development, and Innovations, Agreement No. 451-03-66/2024-03/200029.

REFERENCES

[1] Aouchiche, M., Bonnefoy, J.-M., Fidahoussen, A., Caporossi, G., Hansen, P., Hiesse, L., Lacheré, J., and Monhait, A. (2006). Variable neighborhood search for extremal graphs 14: The autographix 2 system. In *Global Optimization*, pages 281–310. Springer.

[2] Bermond, J.-C. and Fournier, J. C., L. M. S. D. (1978). *Problèmes combinatoires et théorie des graphes*, volume 260.

[3] Bollobás, B. (2004). *Extremal graph theory*. Courier Corporation.

[4] Brualdi, R. A. and Solheid, E. S. (1986). On the spectral radius of connected graphs. *Publ. Inst. Math.(Beograd)*, 39(53):45–54.

[5] Caporossi, G. and Hansen, P. (2000). Variable neighborhood search for extremal graphs: 1 the autographix system. *Discrete Mathematics*, 212(1-2):29–44.

[6] Cvetković, D. (1990). *Teorija grafova i njene primene*. Naučna knjiga, Beograd, treće dopunjeno izdanje.

[7] Cvetković, D., Doob, M., and Sachs, H. (1995). *Spectra of graphs: Theory and application*. Johann Ambrosius Barth Verlag, Heidelberg–Leipzig, 3rd edition.

[8] Cvetković, D. and Simić, S. (2011). Graph spectra in computer science. *Linear Algebra and its Applications*, 434(6):1545–1562.

[9] Cvetkovic, D. M. (2009). Applications of graph spectra: An introduction to the literature. In *Zbornik radova, special issue on the Applications of Graph Spectra*, pages 7–31. 13(21), Mathematical Institute SANU.

[10] Davidović, T., Teodorović, D., and Šelmić, M. (2015). Bee colony optimization Part I: The algorithm overview. *Yugoslav Journal of Operational Research*, 25(1):33–56.

[11] Erdös, P. (1964). On extremal problems of graphs and generalized graphs. *Israel Journal of Mathematics*, 2(3):183–190.

[12] Hansen, P., Mladenović, N., Todosijević, R., and Hanafi, S. (2017). Variable neighborhood search: basics and variants. *EURO Journal on Computational Optimization*, 5(3):423–454.

[13] Mahadev, N. V. R. and Peled, U. N. (1995). *Threshold graphs and related topics*. Elsevier.

[14] Simić, S. K., Li Marzi, E. M., and Belardo, F. (2004). Connected graphs of fixed order and size with maximal index: structural considerations. *Le Matematiche*, 59(1, 2):349–365.

[15] Spielman, D. (2012). Spectral graph theory. In Naumann, U. and Schenk, O., editors, *Combinatorial scientific computing*, number 18, pages 18:1–30. CRC Press.