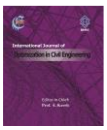
INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2022; 12(2):234-243



SIMULATED ANNEALING ALGORITHM FOR SELECTING SUBOPTIMAL CYCLE BASIS OF A GRAPH

A. Kaveh[*, †] and J. Jafari Vafa
School of Civil Engineering, Iran University of Science and Technology, P.O. Box 16846-13114, Iran

ABSTRACT

The cycle basis of a graph arises in a wide range of engineering problems and has a variety of applications. Minimal and optimal cycle bases reduce the time and memory required for most of such applications. One of the important applications of cycle basis in civil engineering is its use in the force method to frame analysis to generate sparse flexibility matrices, which is needed for optimal analysis.

In this paper, the simulated annealing algorithm has been employed to form suboptimal cycle basis. The simulated annealing algorithm works by using local search generating neighbor solution, and also escapes local optima by accepting worse solutions. The results show that this algorithm can be used to generate suboptimal and subminimal cycle bases. Compared to the existing heuristic algorithms, it provides better results. One of the advantages of this algorithm is its simplicity and its ease for implementation.

Keywords: suboptimal cycle basis; simulated annealing algorithm; graph theory; metaheuristic algorithms; sparse matrices.

Received: 25 January 2022; Accepted: 6 April 2022

1. INTRODUCTION

Cycle bases of graph have many applications in science, and in particular in the force method of structural analysis. Cycle basis satisfying certain conditions correspond to sparse flexibility matrices where the analysis can be performed more efficiently and accurately.

For cycle basis to be minimal, the corresponding cycle-edge incidence matrix **C** should be the sparsest, while for an optimal cycle basis the cycle adjacency matrix $\mathbf{D}=\mathbf{C}\mathbf{C}^{t}$ should have the

[*]Corresponding author: School of Civil Engineering, Iran University of Science and Technology, P.O. Box 16846-13114, Iran
[†]E-mail address: alikaveh@iust.ac.ir (A. Kaveh)

highest sparsity. On the other hand **C** and **D** are pattern equivalent to the equilibrium matrix $\mathbf{B}_1$ and flexibility matrix $\mathbf{G}=\mathbf{B}_1\mathbf{B}_1^t$ respectively. There are graph theoretical methods for the formation of subminimal and suboptimal cycle bases. However, selecting minimal cycle basis is time consuming and that of optimal cycle basis is impossible at present.

There are efficient algorithms for finding minimal cycle bases among which algorithms developed by Kaveh [1-4], Horton [5] and Berger [6] are noteworthy. Greedy algorithm was first employed by Kaveh using the cycle space of a graph. Horton used a subspace of this cycle space to reduce the size of the search space. This is perhaps the fastest candidate for the formation of minimal cycle basis. Unlike minimal cycles, algorithms developed for finding suboptimal cycle bases are limited to those of Kaveh [7-8], Kaveh and Rahami [9] and Kaveh [10].

Metaheuristic algorithms are used for selecting minimal fundamental cycle basis by Amaldi et al. [11] and Amaldi et al. [12] Kaveh and Jahanshahi [13] and Kaveh and Daei [14] employed ant Colony system algorithm for the formation of suboptimal cycle basis maintaining the submimality.

After this introduction defitions from graph theory is presented in section 2. Simulatted annealing is briefly discussed in section 3 and applied to the selection of suboptimal cycle basis is section 4. Examples are provided in section 5 and the paper is concluded in section 6.

2. DEFINITIONS FROM THEORY OF GRAPHS

In the following, some definitions from theory of graphs are provided when in main follow those of Kaveh [4].

- *Graph*: a graph *S* consists of a set N(S) of elements called nodes and a set M(S) of elements called members together with a relation of incidence which associates with each member a pair of nodes, called its ends.
- *Walk*: a walk W_k of *S* is a finite sequence $W_k = \{n_0, m_1, n_1, , \ldots, m_p, n_p\}$ whose terms are alternately nodes n_i and members m_i of *S* for $1 \leq i \leq p$, and ni1 and n_i are two ends of m_i.
- *Trail*: a trail T_k in *S* is a walk in which no member of *S* appears more than once.
- *Path*: a path P_k is a trail in which no node appears more than once.
- *Cycle*: a cycle is a path $\{n_0, m_1, n_1, , \ldots, m_p, n_p\}$ for which $n_0 = n_p$ and $p \geq 1$; i.e. a cycle is a closed path.
- *Tree*: a tree *T* of *S* is a connected subgraph of *S* which contains no cycle. If a tree contains all the nodes of S, it is called a *spanning tree* of S. For simplicity it will be referred to as a *tree*, from now on.
- *Chord*: the complement of a tree T in S is called a *cotree*, denoted by T^*. The members of T are known as *branches* and those of T^* are called chords.
- *Cycle Basis*: a maximal set of independent cycles of a graph is known as its cycle basis.
- *Cycle-Member Incidence Matrix*: a cycle-member incidence matrix $\boldsymbol{C}$ has a row for each cycle and a column for each member. An entry c_{ij} of $\boldsymbol{C}$ is 1 if cycle C_i contains member m_j and it is 0 otherwise.
- *Cycle Adjacency Matrix*: the cycle adjacency matrix $\boldsymbol{D} = \boldsymbol{C}\boldsymbol{C}^t$ is a $b_1(S) \times b_1(S)$ matrix, each entry d_{ij} of which is 1 if C_i and C_j have at least one member in common and it is 0 otherwise. Here, $b_1(S) = M(S) - N(S) + 1$ is known as the first Betti number of *S*, and it is the

dimension of the cycle space of a graph *S*, where *M(S)* and *N(S)* are the cardinality of the node set and members set of *S*, respectively.

• *Minimal Cycle Basis*: a cycle basis $C = \{C_1, C_2 \dots C_{b_1(S)}\}$ is called minimal if it corresponds to a minimum value of

$$L(C) = \sum_{i=1}^{b_1(S)} L(C_i) \tag{1}$$

where $L(C_i)$ is he number of members of the cycle C_i. $L(C)$ shows the number of entries of the cycle-member incidence matrix $\boldsymbol{C}$. A cycle basis corresponding to near minimal $L(C)$ is called a subminimal cycle basis.

• *Sparsity Coefficient* χ: the sparsity coefficient χ of a matrix is defined to be its number of nonzero entries. A matrix of maximal sparsity has the minimum number of non-zero entries.

• *Optimal Cycle Basis*: a basis that corresponds to maximal sparsity of cycle adjacency matrix, $\boldsymbol{D}$, is called an optimal cycle basis. A cycle basis corresponding to near maximum sparsity of $\boldsymbol{D}$ is called a suboptimal cycle basis. An optimal cycle basis is not necessarily a minimal cycle basis and vice versa.

3. SIMULATED ANNEALING ALGORITHM

Simulated Annealing (SA) is commonly said to be the oldest among the metaheuristics and surely one of the first algorithms that had an explicit strategy to escape from local minima. The origins of the algorithm are in statistical mechanics (Metropolis algorithm) and it was first presented as a search algorithm for combinatorial problems in Kirkpatrick et al. [15] and Cerny [16]. The fundamental idea is to allow moves resulting in solutions of worse quality than the current solution (uphill moves) in order to escape from local minima. The probability of doing such a move is decreased during the search. Other studies on SA can be found in [17-19]. Different recent powerful metaheuristic algorithms can be found in Kaveh [20]. Efficient codes and examples can be found in Kaveh and Bakhshpoori [21].

The algorithm starts by generating an initial solution (either randomly or heuristically constructed) and by initializing the so-called temperature parameter *T*. Then, at each iteration a neighbor solution s' is randomly sampled and it is accepted as new current solution depending on $f(s)$, $f(s')$ and *T*. s' replaces s if $f(s') < f(s)$ or, in case $f(s') \geq f(s)$, with a probability which is a function of T and $f(s') - f(s)$. The probability is generally computed following the Boltzmann distribution $exp\left(-\frac{f(s')-f(s)}{T}\right)$.

The temperature T is decreased during the search process, thus at the beginning of the search the probability of accepting uphill moves is high and it gradually decreases, converging to a simple iterative improvement algorithm. This process is analogous to the annealing process of metals and glass, which assume a low energy configuration when cooled with an appropriate cooling schedule. Regarding the search process, this means that the algorithm is the result of two combined strategies: random walk and iterative improvement. In the first phase of the search, the bias toward improvements is low and it

permits the exploration of the search space; this erratic component is slowly decreased thus leading the search to converge to a (local) minimum. The probability of accepting uphill moves is controlled by two factors: the difference of the objective functions and the temperature. On the one hand, at fixed temperature, the higher the difference $f(s') - f(s)$, the lower the probability to accept a move from s to s'. On the other hand, the higher T, the higher the probability of uphill moves.

4. SIMULATED ANNEALING ALGORITHM FOR THE FORMATION OF A SUBOPTIMAL CYCLE BASIS

In this algorithm, an initial cycle basis is first generated, and then the optimal solution is searched by local search and generating neighbor solutions.

The novelty of this paper is how to generate a neighbor cycle basis (neighbor solution) for local search and use it in simulated annealing algorithm to obtain a suboptimal (or subminimal) cycle basis. The following are the steps of this algorithm:

(1) *Step 1*: Generate an initial cycle basis (randomly or heuristically), and evaluate it.
(2) *Step 2*: Consider the previous step solution as the best solution.
(3) *Step 3*: Set the initial temperature ($T = T_0$).
(4) *Step 4*: Generate a random solution in the neighborhood of the current solution and evaluate it.
(5) *Step 5*: Accept the new solution if it is better than the current solution.
(6) *Step 6*: Conditional (probable) acceptance if the new solution is not better.
(7) *Step 7*: Update the best solution ever found.
(8) *Step 8*: Decrease the temperature and return to Step 4 if the termination conditions are not met.

4.1 Generating the initial cycle basis

The initial cycle basis can be created in one of the following ways:

- Using the fundamental cycle basis, to improve the fundamental cycle basis, the used chords can be added to the tree branch set and also the chords can be used according to their distance from the root node Kaveh [1].
- Creating a minimal length cycle on each member and select independent cycles from them. If the number of independent cycles was not enough to form a basis, unused members can be selected and an independent cycle can be created on it.

4.2 Generating a neighbor solution

First, a cycle is selected randomly from the cycle basis elements and denoted by C_1. Then, from the set of neighboring cycles with cycle C_1, one cycle is randomly selected and denoted by C_2 (two cycles are called adjacent or neighbors if they have at least one member in common). The symmetric difference of the C_1 and C_2 creates a new cycle that denoted by C_{new}. Now, the cycle C_{new} is replaced with one of the cycles C_1 or C_2, and the neighbor cycle basis (neighbor solution) is generated. Creating a new cycle by the symmetric

difference of two adjacent cycles is shown in the Fig. 1.

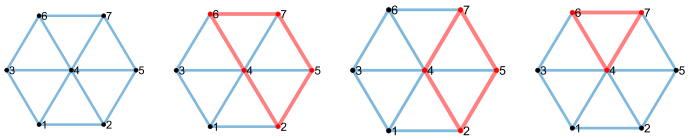


Figure 1. Creating a new cycle by the symmetric difference of two adjacent cycles

4.3 Reannealing (restarting)

In order the algorithm to be able to escape the local optima, it is sometimes better to increase the temperature so that the algorithm can escape the local optima by choosing worse solutions. This procedure is called restarting or reannealing. To do this, the initial solution to start over is usually equated to the best solution found, and the algorithm is restarted. The decision to restart can be based on several criteria. The most common restart methods: After a certain number of iterations, restart is due to overrun of system maximum energy and restarting randomly.

4.4 Algorithm parameters

The parameters used in the algorithm are presented in Table 1.

Table 1: SA algorithm parameters

parameter	**Value / formula**
Initial temperature	$T_0 = 10$
Temperature reduction rate	$\alpha = 0.995$
Temperature in each iteration (cooling schedule)	$T_{k+1} = \alpha . T_k$
Number of iterations at each temperature	$SubIt = 10$
ΔE	$\Delta E = f(s') - f(s)$
probability of accepting worse solutions	$p = e^{\frac{-\Delta E}{T}}$

The cost function (objective function) is defined as follows:

$$f(s) = \chi(D) \tag{2}$$

where $\chi(D)$ is the sparsity coefficient of the cycle adjacency matrix D.

5. EXAMPLES

In this section, some examples are provided to evaluate the simulated annealing algorithm.

5.1 Example 1

This example is a space structure (graph) with $b_1(S) = 21 - 12 + 1 = 10$, as shown in Fig. 2.

The proposed algorithm selects cycle basis with $\chi(D) = 42$, which is shown in Fig. 3. The best result of the heuristic algorithms for this example has $\chi(D) = 44$.

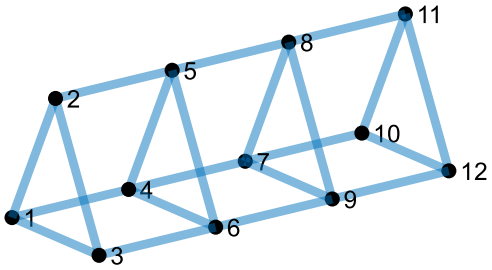


Figure 2. The graph model of a space frame S with $b_1(S) = 10$.

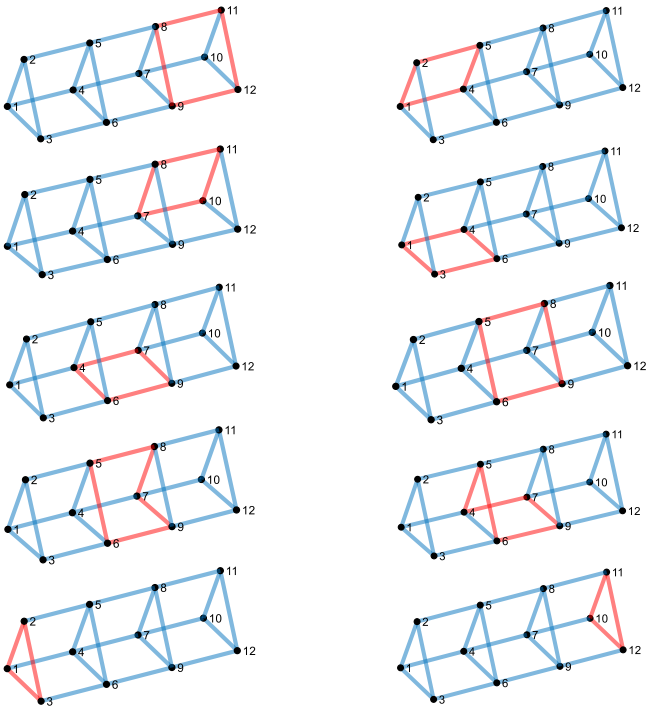


Figure 3. The selected suboptimal cycle basis

As shown in Fig. 4, restarting the algorithm (increasing the temperature) after a certain number of iterations can improve the result of the algorithm.

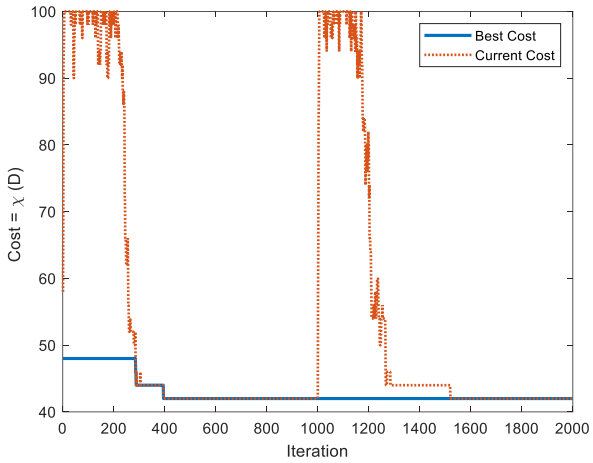


Figure 4. Diagram of the cost function (χ ($\boldsymbol{D}$)) relative to the number of iterations for Example 1

5.2 Example 2

This example is a space structure with $b_1(S) = 360 - 91 + 1 = 270$, as shown in Fig. 5.

The proposed algorithm selects cycle basis with $\chi(D) = 1482$. The best result of the heuristic algorithms for this example is $\chi(D) = 1504$.

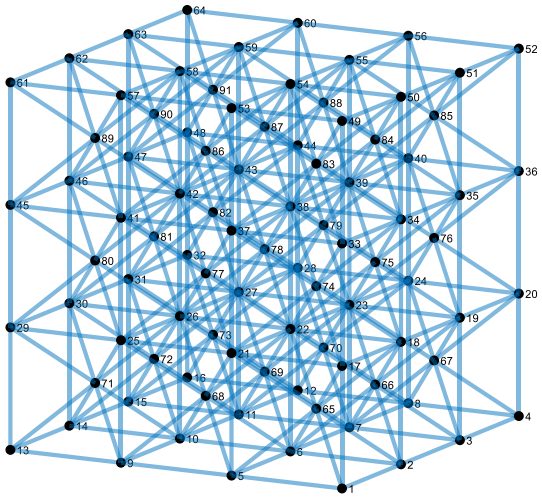


Figure 5. The model of a space structure with $b_1(S) = 360 - 91 + 1 = 270$

As shown in Fig. 6, restarting the algorithm (increasing the temperature) after a certain number of iterations can improve the result of the algorithm.

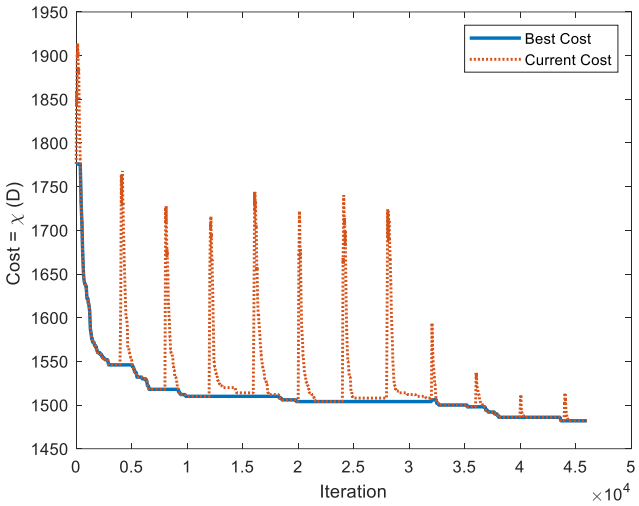


Figure 6. Diagram of the cost function (χ ($\boldsymbol{D}$)) relative to the number of iterations for Example 2

5.3. Example 3

This example is a space structure with $b_1(S) = 540 - 216 + 1 = 325$, as shown in Fig. 7.

The proposed algorithm selects cycle basis with $\chi(\boldsymbol{D}) = 2417$. The best result of the heuristic algorithms for this example is $\chi(\boldsymbol{D}) = 2565$.

As shown in Fig. 8, restarting the algorithm (increasing the temperature) after a certain number of iterations can improve the result of the algorithm.

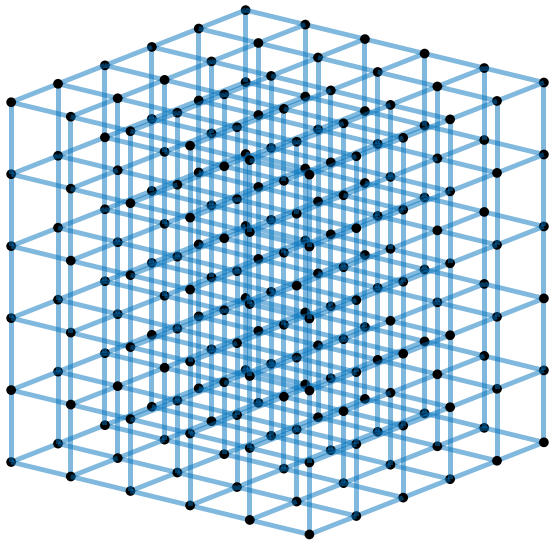


Figure 7. A space structure with 540 members and 216 nodes

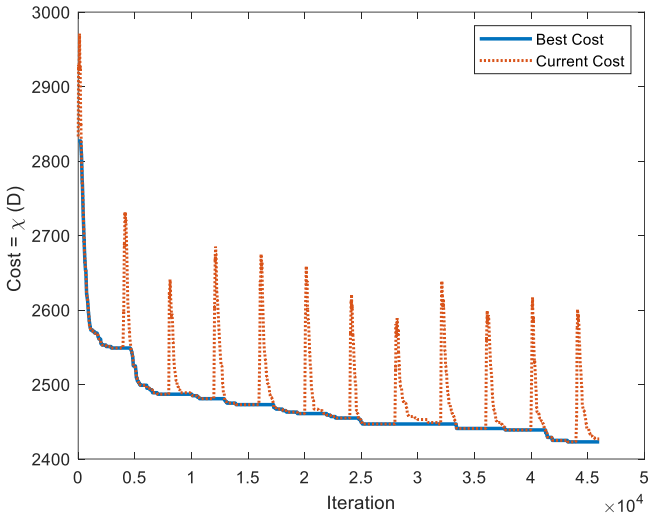


Figure 8. Diagram of the cost function (χ ($\boldsymbol{D}$)) relative to the number of iterations for Example 3

6. CONCLUSIONS

Due to the importance of the optimal cycle basis, several methods have been introduced to create the suboptimal cycle basis of a graph. In order to improve the existing methods, in this research it is attempted to provide a method for producing a suboptimal cycle basis that has the desired performance by using metaheuristic algorithms, which are an effective tool for optimizing various problems. For this purpose, the Simulated Annealing algorithm is used to form the cycle basis. The simulated annealing algorithm works by using local search and generating neighbor solution, and also escapes local optima by accepting worse solutions. The results show that this algorithm can be used to generate suboptimal bases. In comparison to the existing heuristic algorithms, it leads to better results. One of the advantages of this algorithm is its simplicity requiring easily implementation.

The main purpose of this research was to form the suboptimal cycle basis, but the proposed algorithm can also be used to create the subminimal cycle basis or the suboptimal-subminimal cycle basis. For this purpose, the change of the objective (or cost) function in the simulated annealing algorithm is needed.

REFERENCES

1. Kaveh A. *Application of Topology and Matroid Theory to the Flexibility Analysis of Structures*, Ph. D. thesis, London University, Imperial College of Science and Technology, 1974.
2. Kaveh A. Improved cycle bases for the flexibility analysis of structures, *Comput Meth Appl Mech Eng* 1976; **9**: 267-72.
3. Kaveh A. Recent developments in the force method of structural analysis, *Appl Mech*

Review 1992; **45**(9): 401-18.
4. Kaveh A. *Structural Mechanics: Graph and Matrix Methods*, 3rd edition, *Springer*, 2021.
5. Horton JD. A polynomial-time algorithm to find the shortest cycle basis of a graph, *SIAM J Comput* 1987; **16**: 358-66.
6. Berger F, Gritzmann P, Sven de Vries. Minimum cycle bases for network graphs, *Algorith* 2004; **40**: 51-62.
7. Kaveh A. Suboptimal cycle bases of graphs for the flexibility analysis of skeletal structures, *Comput Meths Appl Mech Eng* 1988; **71**: 259-71.
8. Kaveh A. Suboptimal cycle bases of a graph for mesh analysis of networks, *Int J Networks* 1989; **19**: 273-79.
9. Kaveh A, Rahami H. Algebraic graph theory for sparse flexibility matrices, *J Math Model Algor* 2003; **2**: 171-82.
10. Kaveh A. *Optimal Structural Analysis*, John Wiley & Sons, 2006.
11. Amaldi E, Liberti L, Maffioli F, Maculan N. Algorithms for finding minimum fundamental cycle bases in graphs, *Electron Notes Discr Math* 2004; **17**: 29-33.
12. Amaldi E, Liberti L, Maffioli F, Maculan N. Edge-swapping algorithms for the minimum fundamental cycle basis problem, *Math Meth Oper Res* 2009; **69**: 205-33.
13. Kaveh A, Jahanshahi M. An efficient program for cycle basis selection and bandwidth optimization, *Asian J Civil Eng* 2006; **7**(1): 95-109.
14. Kaveh A, Daei, M. Suboptimal cycle bases of graphs using an ant colony system algorithm, *Eng Comput* 2010; **27**(4): 485-94.
15. Kirkpatrick S, Jr CD Gelatt, Vecchi MP. Optimization by simulated annealing, *Sci* 1983; **220**: 671-80.
16. Černý V. Thermo dynamical approach to the traveling salesman problem: An efficient simulation algorithm, *J Optim Theor Appl* 1985; **45**: 41-51.
17. Blum Ch, Roli A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *ACM Comput Surveys* (CSUR) 2003; **35**: 268-308.
18. El-Ghazali T. *Metaheuristics: from Design to Implementation*, John Wiley & Sons, 2009
19. Dan S. *Evolutionary Optimization Algorithms*, John Wiley & Sons, 2013.
20. Kaveh A. *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, Springer International Publishing, Switzerland, 3rd edition, 2021.
21. Kaveh A, Bakhshpoori T. *Metaheuristics: Outlines, MATLAB Codes and Examples*, Springer International Publishing, 2019.