https://doi.org/10.3311/PPci.18519
Creative Commons Attribution

Periodica Polytechnica Civil Engineering, 66(2), pp. 323–334, 2022

Automatic Domain Decomposition in Finite Element Method – A Comparative Study

Ali Kaveh[1*] , Mohammad Reza Seddighian[1], Pouya Hassani[2]

[1] School of Civil Engineering, Iran University of Science and Technology, Narmak, 16846-13114 Tehran, Iran
[2] Department of Civil Engineering, Faculty of Engineering, Urmia University, 5756151818 Urmia, Iran
[*] Corresponding author, e-mail: alikaveh@iust.ac.ir

Received: 07 May 2021, Accepted: 08 October 2021, Published online: 13 October 2021

Abstract

In this paper, an automatic data clustering approach is presented using some concepts of the graph theory. Some Cluster Validity Index (CVI) is mentioned, and DB Index is defined as the objective function of meta-heuristic algorithms. Six Finite Element meshes are decomposed containing two- and three- dimensional types that comprise simple and complex meshes. Six meta-heuristic algorithms are utilized to determine the optimal number of clusters and minimize the decomposition problem. Finally, corresponding statistical results are compared.

Keywords

optimization, meta-heuristic algorithm, graph theory, finite element method, graph clustering optimization, k-median, automatic clustering, parallel processing

1 Introduction

Nowadays, the computational cost of numerical methods is increased, especially in dealing with real-size and practical problems. Hence, additional computational methods are used to improve the performance of the primary numerical approach. For instance, intelligent algorithms such as meta-heuristics are utilized to enhance the accuracy and efficiency of the numerical approaches [1–5] and/or parallel processing concepts have been applied widely to the finite element method in order to solve large-scale problems such as plastic analysis of structures, fracture mechanics, electrical networks, and fluid structures [6, 7].

There are various methods to prepare a finite element method (FEM) problem for parallel processing. One of the most efficient of them is to decompose the problem domain (is called mesh in the FEM) into various subdomains and analyze each of them by an individual processor. While a parallel approach is employed, the time to complete the analysis will be the time due to the longest sub-task. Therefore, it is vital to define each sub-domain in optimal condition in which the size of each of them is approximately equal.

The act of partitioning or decomposing an unlabeled data set into groups of similar objects means clustering [8]. Recently, Clustering concepts and methods widely are utilized in various research fields such as extended finite element method, shrinkage cracking, event recognition, satellite image classification, fuzzy optimization, black spot identification, cellular structures, and many other efforts.

In the 1980s, the initial efforts to the partitioning of the FEM meshes are made. Farhat [9] proposed a simple automatic domain decomposer of FEM meshes, and after a while, Farhat and Lesoinne [10] presented an automatic partitioning of unstructured type of meshes in FEM. In 1993, a simple and adaptive subdomain generation method for parallel finite element analysis was developed by Khan and Topping [11]. Kaveh and Roosta [12] employed graph theory concepts for optimal domain decomposition.

Gradually, by combining graph theory concepts such as *k*-median and *p*-median of a graph with the domain decomposition problem, the Artificial Intelligent (AI) approaches such as meta-heuristic algorithms are employed to optimize it. Kaveh et al. [13–16], Beitollahi et al. [17], Christofides and Beasley [18], Osman and Christofides [19], Kariv and Hakimi [20], and Estivill-Castro and Torres-Velázquez [21] are some of the scientists that utilized AI approaches as well as graph theory concepts in this manner.

Cite this article as: Kaveh, A., Seddighian, M. R., Hassani, P. "Automatic Domain Decomposition in Finite Element Method - A Comparative Study", Periodica Polytechnica Civil Engineering, 66(2), pp. 323–334, 2022. https://doi.org/10.3311/PPci.18519

Data Clustering methods can be Hierarchical or Partitional [22, 23] in each of which there is a wealth of subtypes and different methods for obtaining the clusters. In the first category, the output is a tree showing a sequence of clustering, with each cluster being a partition of the data set [24]. Hierarchical methods are divided into agglomerative (bottom-up) or divisive (top-down). Agglomerative methods start with each element as a separate cluster and merge them in successively larger clusters. Nevertheless, division types initiate with the whole set and proceed to divide it into successively clusters.

Although there are many advantages in hierarchical clustering techniques, such as independency of the initial conditions, however, the main drawback of them is they are static; that is, data points assigned to a cluster cannot travel to another cluster. Furthermore, it is possible for them to fail to separate overlapping clusters due to the absence of information about the global shape or size of the clusters [24].

On the other hand, partitional clustering techniques try to decompose the data set into a set of disjoint clusters directly. They attempt to optimize specific criteria, such as the square-error function. The partitional clustering problem has been developed by various fields of science, such as statistics (multivariate analysis) [25], graph theory [26], expectation-maximization methods [27], artificial neural networks [28–30], evolutionary computing [31, 32], and so on.

Enormous effort has been dedicated in the past few years to the complex data sets clustering. However, not tremendous research has been reported to determine the optimal number of clusters at the same time. Most of the well-known clustering methods, especially based on evolutionary algorithms, accept the number of classes k as an input instead of calculating the same on the process. Notwithstanding, in many practical cases, there is no appropriate number of groups in a data set, or it is impossible to approximate it.

In this paper, an automatic data set clustering method is presented. The introduced method is employed firstly, to obtain the optimal number of the clusters and secondly, to divide data into related groups. The considered problems in this study are two- and three- dimensional FEM meshes, which are decomposed into the optimal number of clusters. For this aim, the automatic clustering is stated as an optimization problem, and it is solved utilizing six meta-heuristic methods. The Particles Swarm Optimization (PSO) [33], and Genetic Algorithm (GA) [34] as well-known meta-heuristic algorithms, Grey Wolf Optimizer (GWO) [35], Sine Cosine Algorithm (SCA) [36] as the high used meta-heuristic algorithms and, Black Holes Mechanics Optimization (BHMO) [37], and Enriched Firefly Algorithm (EFA) [38] as newly developed meta-heuristic algorithms constitute the set of methods which are utilized to solve the proposed problem.

The rest of the current study is organized as follows. The basic definition of the graph theory and the k-median problem of a graph are outlined in Section 2. Section 3 is dedicated to automatic clustering concepts. The FEM examples are discussed in Section 4. Finally, Section 5 concludes the results of the study and recommends some suggestions for future directions.

2 Basic definitions of the graph theory

A set $N(G)$ of nodes and a set $E(G)$ of edges together with a relation of incidence which associate two distinct nodes with each edge, known as its *ends*, constitute a *simple graph G.* if two nodes of a graph are the end nodes of an edge then they are called *adjacent*. An *incident* edge with a node is one which contains it in its end nodes [39].

A *subgraph* G_i of a graph G is a graph for which $N(G_i) \subseteq N(G)$ and $E(G_i) \subseteq E(\mathrm{G})$, and each edge of G_i contains the same ends of G. A *path* of a graph is a finite sequence $P = \{n_0, e_1, n_1, e_2, \ldots, e_p, n_p\}$ whose terms are alternately distinct nodes n_i and distinct edges e_i for $1 \leq i \leq p$ and n_{i-1} and n_i are two ends of e_i.

A *tree T* of a graph G is a subgraph that contains no *cycle* (a *cycle* is a closed path). A *spanning tree* is a tree containing all the nodes of a graph G. A *Shortest Route Tree* (*SRT*) rooted from a specified node n_0 (is called starting node) is a spanning tree for which the distance between every node n_j of T and n_0 is minimum. The distance between two nodes is taken as the number of edges in the shortest path between these nodes [39].

A *clique graph G* of a FEM mesh contains its nodes in one-to-one correspondence with the elements of the considered FEM mesh, and two nodes of G are connected by an edge if the corresponding elements have at least one common node [39].

The *k*-median problem of a graph G is to find a set S of k nodes that minimize the distance sum of the shortest paths from all the nodes to their closest node n_0. In other words, the *k*-median problem aims to decompose or to partition a node-set into k subsets. The k-median problem of graph G can be stated as an optimization problem in which the objective function can be defined as Eq. (1.)

$$\sigma_0(N_k) = \sum_{j \in N} v_j d(N_k, j) , \tag{1}$$

where $\sigma_0(N_k)$ is the out-transmission of nodes N_k, N_k is the median node number, and v_j is the weight of node j. Also, $d(N_k, j)$ is defined as Eq. (2.)

$$d(N_k, j) = \min[d(i', j)] : (i' \in N_k) , \tag{2}$$

where i' is the node of N_k, which minimizes the value for Eq. (2).

There are various algorithms to optimize the k-median problem of a graph G. The flowchart of an algorithm in which the introduced concepts are used is illustrated in Fig. 1.

It can be possible to define a FEM mesh as a graph and to decompose it into k subdomains that each subdomain can be considered as a super element, and its corresponding stiffness matrix can be created in a partitioning form using Eq. (3).

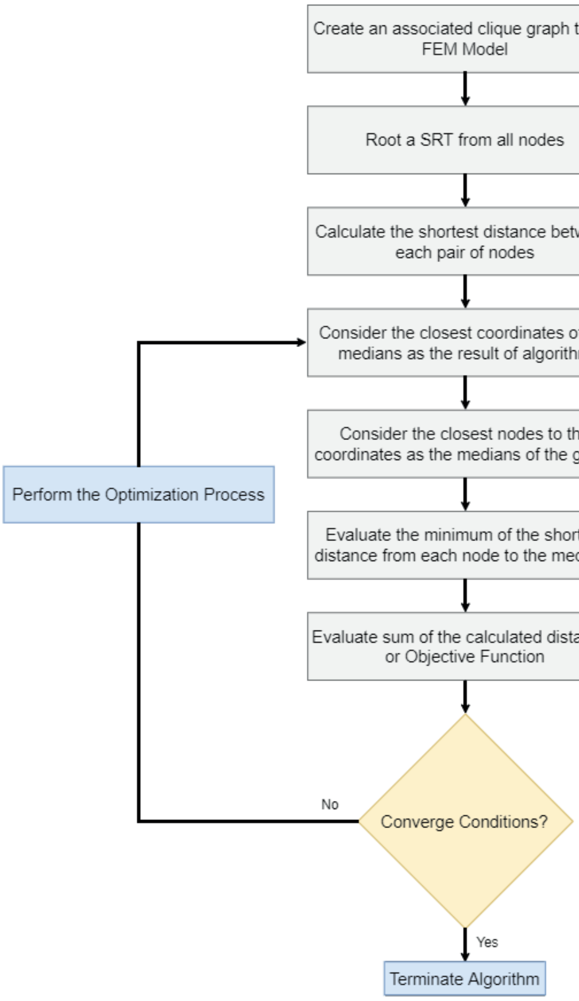
Create an associated clique graph to the FEM Model
Root a SRT from all nodes
Calculate the shortest distance between each pair of nodes
Consider the closest coordinates of the medians as the result of algorithm
Consider the closest nodes to the coordinates as the medians of the graph
Evaluate the minimum of the shortest distance from each node to the medians
Evaluate sum of the calculated distances or Objective Function
Converge Conditions?
No
Perform the Optimization Process
Yes
Terminate Algorithm

Fig. 1 The flowchart of the optimization procedure of the k-median problem of a graph

$$\begin{bmatrix} k_{ii} & k_{ib} \\ k_{bi} & k_{bb} \end{bmatrix} \begin{bmatrix} d_i \\ d_b \end{bmatrix} = \begin{bmatrix} F_i \\ F_b \end{bmatrix}, \tag{3}$$

where i and b denote the interior and boundary nodes, respectively. Also, k, d, and F are stiffness matrix, displacement, and the force vector.

A boundary node is one that is part of more than one subdomain. On the other hand, an interior node is defined as a node that is part of only one subdomain. The corresponding stiffness matrices, displacement, and force vectors used in Eq. (3) can be calculated using Eqs. (4)–(10).

The vector $\boldsymbol{F}_i$ (force vector of interior nodes) can be defined using static condensation.

$$[\boldsymbol{F}_i] = [k_{ii}][d_i] + [k_{ib}][d_b] , \tag{4}$$

or

$$[d_i] = [k_{ii}]^{-1}[\boldsymbol{F}_i - [k_{ib}][d_b]] . \tag{5}$$

Substituting Eq. (5) in the second part of Eq. (3) leads to Eq. (6).

$$[\boldsymbol{F}_b] = [k_{bi}][k_{ii}]^{-1}([\boldsymbol{F}_i] - [k_{ib}][d_b]) + [k_{bb}][d_b] , \tag{6}$$

or

$$[k_{bi}][k_{ii}]^{-1}[\boldsymbol{F}_i] - [k^*][d_b] = [\boldsymbol{F}_b] , \tag{7}$$

$$[k_{bi}][k_{ii}]^{-1}[k_{ib}] - [k^*] = [k_{bb}] . \tag{8}$$

Eq. (8) denotes the condensed super element stiffness matrix, and the corresponding modified force vector can be defined as Eq. (9).

$$[k_{bi}][k_{ii}]^{-1}[\boldsymbol{F}_i] - [\boldsymbol{F}^*] = [\boldsymbol{F}_b] \tag{9}$$

Therefore,

$$[\boldsymbol{F}^*] = [k^*][d_b] . \tag{10}$$

Eq. (10) can be used to assemble structural stiffness matrix, displacement, and the force vector.

3 Automatic clustering

A pattern is a physical or abstract structure of objects which is distinguished from others by a common set of attributes called features, which together represent a pattern [40]. Let $P = \{P_1, P_2, \ldots, P_n\}$ be a set of n data points or patterns, each of them containing d features in d-dimensional problem space. Also, these data points can be represented by a profile data matrix $X_{n \times d}$. the ith row vector X_i characterizes the ith object from the set P, and each element $X_{i,j}$ in X_i corresponds with jth real-value feature

(j = 1, 2, …, d) of the ith pattern (i = 1, 2, …, n). A partitional clustering technique attempts to obtain a partition $C = \{C_1, C_2, \ldots, C_k\}$ of k classes, such that the similarity of the patterns in each cluster is maximum and patterns from different clusters differ as far as possible [8].

The partitions must maintain three properties.

1. Each cluster should have at least one pattern assigned.
2. Two different clusters should have no pattern in common.
3. Each pattern should definitely be attached to a cluster.

There are various ways for data partitioning, each of them maintaining all of the properties as mentioned above. Therefore, an objective function must be defined to obtain optimal clustering. Hence, the problem turns out to be one of getting a partition C^* of optimal or near-optimal adequacy, as compared to all other feasible solutions $C = \{C_1, C_2, \ldots, C^{N(n,k)}\}$, where $C^{N(n,k)}$ is the number of feasible partitions and can be defined as Eq. (11).

$$N(n,k) = \frac{1}{k!}\sum_{i=1}^{k}(-1)^i \binom{k}{i}^i (k-i)^i \tag{11}$$

Therefore, the current problem can be defined as an optimization one, as stated as Eq. (12).

$$\textbf{Optimize} \;\; f(X_{n\times d}, C), \tag{12}$$

where C is a single partition from the set C, and f is a statistical-mathematical function which quantifies the fitness of a partition based on the distance measure of the patterns. It has been shown that the current problem is NP-hard when the number of clusters exceeds 3 [41].

As introduced previously and according to the Eq. (12), defining a proper objective function or *Similarity Measure* [8] plays a crucial role in the clustering procedure. One of the most well-received approach to evaluate the similarity between patterns amounts is the use of distance criteria. The most popular distance criterion is the Euclidean distance that is stated for two d-dimensional patterns X_i and X_j as Eq. (13).

$$d(\vec{X}_i, \vec{X}_j) = \sqrt{\sum_{p=1}^{d}(X_{i,p} - X_{j,p})^2} = \left\|\vec{X}_i - \vec{X}_j\right\| \tag{13}$$

The Euclidean distance is a particular case of the Minkowski metric when α is equal to 2. The Minkowski metric can be defined as Eq. (14).

$$d^{\alpha}(\vec{X}_i, \vec{X}_j) = (\sum_{p=1}^{d}(X_{i,p} - X_{j,p}))^{1/\alpha} = \left\|\vec{X}_i - \vec{X}_j\right\|^{\alpha} \tag{14}$$

Also, when $\alpha = 1$, the distance is known as the Manhattan distance. Many other distance criteria can be found in mathematical or clustering references, such as [8].

For automatizing the clustering procedure, a Cluster Validity Index (CVI) must be defined. Ordinarily, a CVI serves two aims. Firstly, to determine the number of clusters, and secondly, to obtain the corresponding best partition. Hence, a CVI should take care of the two aspects of partitioning [23].

1. Cohesion: the patterns in a cluster should be as similar as possible to each other.
2. Separation: clusters should be well separated.

There are some well-known and robust CIVs, such as Dunn's index [41], the Calinski-Harabasz index [42], the DB index [43], the Pakhira Bandyopadhyay Maulik (PBM) index [44], and the CS measure [45]. In the current study, the DB Index is employed and described as follows.

The DB Index is a function of the ratio of the sum within-cluster scatter to between-cluster separation. It utilizes either the clusters or their sample means. The within ith cluster scatter and the between the ith, and jth cluster distance can be defined as Eqs. (15) and (16), respectively.

$$S_{i,q} = \left[\frac{1}{N_i}\sum_{\vec{X}\in C_i}\left\|\vec{X} - \vec{m}_i\right\|_2^q\right]^{1/q}, \tag{15}$$

$$d_{ij,t} = \left\{\sum_{p=1}^{d}\left|m_{i,p} - m_{j,p}\right|^t\right\}^{1/t} = \left\|\vec{m}_i - \vec{m}_j\right\|_t, \tag{16}$$

where m_i is the ith cluster center, q, $t \geq 1$, is an integer and can be independently selected. Also, N_i is the number of elements in the ith cluster C_i. If Eq. (17) states the $R_{i,qt}$, and then the DB Index can be defined as Eq. (18).

$$R_{i,qt} = \max_{j\in k, j\neq i}\left\{\frac{S_{i,q} + S_{j,q}}{d_{ij,t}}\right\} \tag{17}$$

$$DB(k) = \frac{1}{k}\sum_{i=1}^{k} R_{i,qt} \tag{18}$$

The smallest $DB(k)$ index indicates a valid optimal partition [8].

4 FEM examples

In this section, six FEM meshes, including two- and three-dimensional types, are decomposed to the optimum number of clusters employing introduced concepts and utilizing six meta-heuristic methods. The Particles Swarm Optimization (PSO) [33], and Genetic Algorithm (GA) [34] as well-known meta-heuristic algorithms, Grey Wolf Optimizer (GWO) [35], Sine Cosine Algorithm (SCA) [36] as high cited meta-heuristic algorithms and, Black Holes Mechanics Optimization (BHMO) [37], and Enriched Firefly Algorithm (EFA) [38] as newly developed meta-heuristic algorithms constitute the set of methods which are utilized to solve the current problem.

In each example, firstly, the FEM meshes are considered as a graph considering their corresponding element connectivity. Then, the proposed concepts are employed to obtain the optimum number of clusters and their relevant cost. Finally, the results of the algorithms are compared. In each case, the total number of function evaluations (NFE) is equal to 200000, and each problem is solved 50 times. For each example, the corresponding complete data to its mesh will be made available on request.

4.1 2D mechanical lever with corner crack

The first example is a two- dimensional mechanical lever model with a corner crack, as illustrated in Fig. 2. The corresponding mesh contains 501 nodes and 801 CST elements. The model is decomposed using meta-heuristic algorithms, and the final results are reported in Table 1.

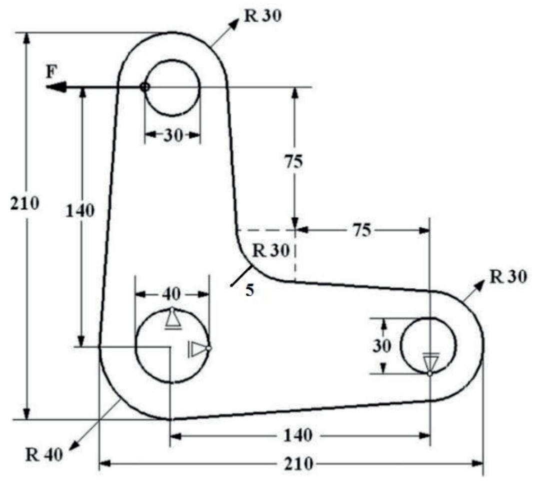

Fig. 2 mechanical-lever with a corner crack; geometry and boundary conditions

Table 1 Statistical Results of Example 1

	PSO	GA	SCA	GWO	BHMO	EFA
Best	30.94	20.22	22.86	24.45	**19.10**	19.29
Average	30.94	20.22	22.86	24.45	**19.10**	19.29
Std.	1.08E-14	7.23E-15	1.08E-14	7.23E-15	1.45E-14	**3.61E-15**
Cluster No. (k)	7	3	6	3	**3**	**3**
Time Ratio (%)	21.39	52.51	**5.18**	5.27	6.27	9.38

The comparison between the utilized algorithm is shown in Fig. 3. Also, the decomposed meshes obtained by different methods are illustrated in Figs. 4 to 6.

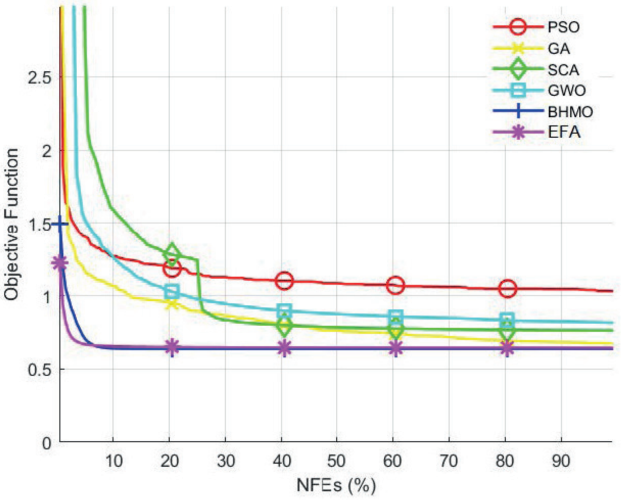

Fig. 3 The comparison between meta-heuristic algorithms (Example 1)

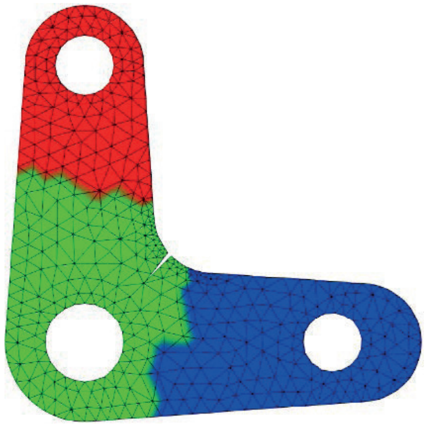

Fig. 4 The decomposed mesh into 3 clusters of Example 1 by GA, GWO, BHMO, and EFA algorithms

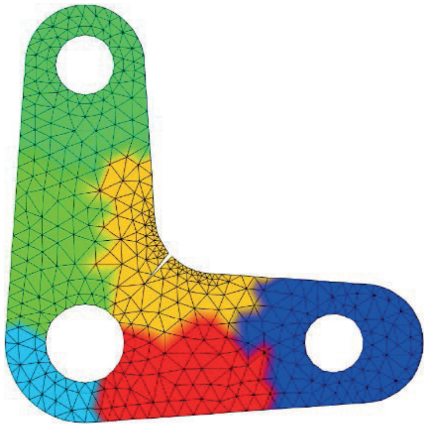

Fig. 5 The decomposed mesh into 6 clusters of Example 1 by the SCA algorithm

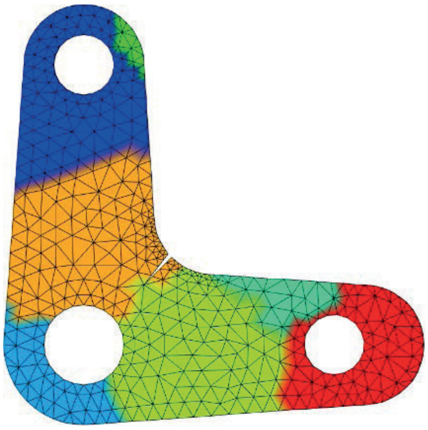

Fig. 6 The decomposed mesh into 7 clusters of Example 1 by the PSO algorithm

4.2 2D simple beam with non-central notch

The second example is a two-dimensional simple supported beam containing an initial notch near the left support, as shown in Fig. 7. The corresponding mesh comprises 584 nodes and 1056 CST elements. The statistical results are reported in Table 2.

The comparison between the utilized algorithm is shown in Fig. 8. Since all methods calculated the same number of clusters of the problems, the final corresponding decomposed meshes with them are very similar. Hence, only the final result of the EFA algorithm is shown in Fig. 9. Other outcomes will be available on request.

4.3 3D arc

The current example contains a three-dimensional arc, as shown in Fig. 10. The corresponding mesh comprises 468 nodes and 1445 tetrahedral elements. The statistical results are reported in Table 3.

The comparison between the utilized algorithm is shown in Fig. 11. Also, the decomposed meshes obtained by different methods are illustrated in Figs. 12 and 13.

Table 2 Statistical Results of Example 2

	PSO	GA	SCA	GWO	BHMO	EFA
Best	17.66	16.26	21.20	22.81	16.17	**16.12**
Average	17.66	16.26	21.20	22.81	16.17	**16.12**
Std.	1.08E-14	1.45E-14	**0.00E+00**	1.08E-14	7.23E-15	**0.00E+00**
Cluster No. (k)	**2**	**2**	**2**	**2**	**2**	**2**
Time Ratio (%)	20.33	53.79	**4.68**	5.15	6.80	9.25

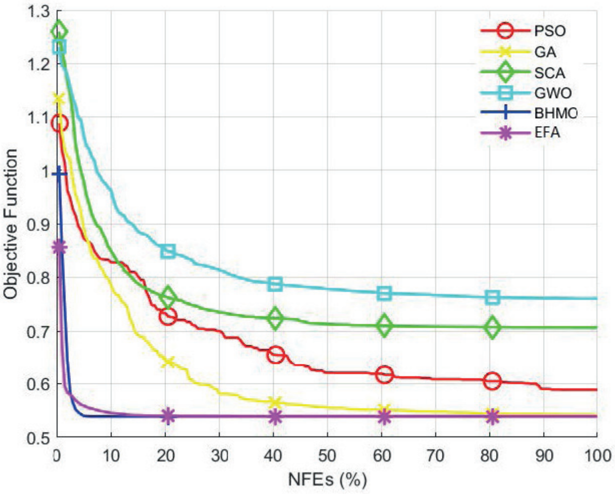

Fig. 8 The comparison between meta-heuristic algorithms (Example 2)

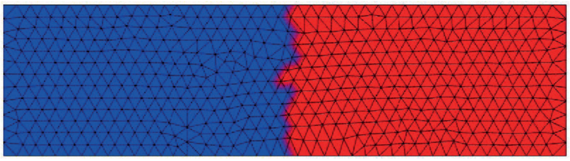

Fig. 9 The decomposed mesh into 2 clusters, of Example 2

4.4 3D mechanical motor

Example 4 contains a three-dimensional mechanical motor, as illustrated in Fig. 14. The corresponding mesh contains 5726 nodes and 2860 tetrahedral elements. The statistical results are reported in Table 4.

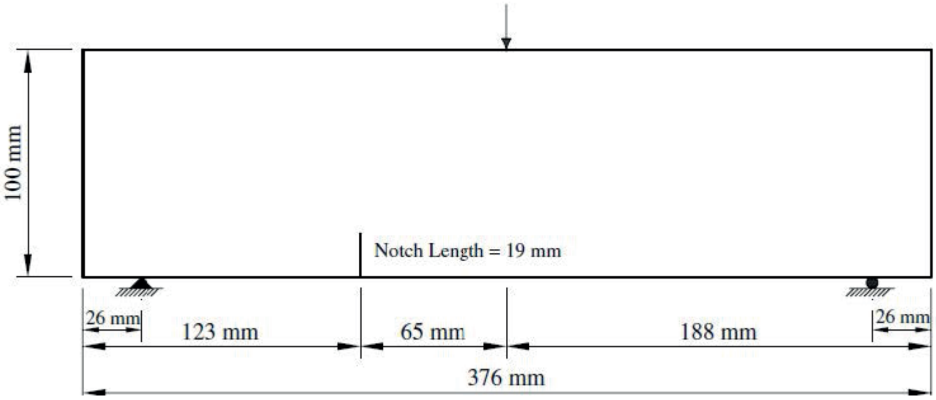

Fig. 7 Single edge notched beam with an eccentric crack; geometry and boundary conditions

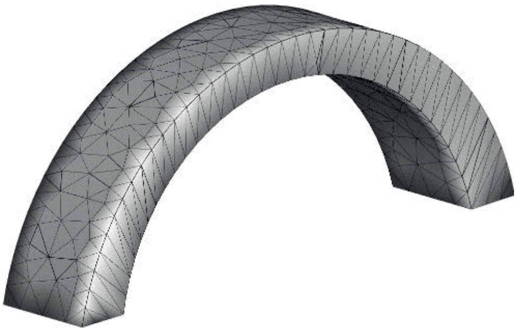

Fig. 10 3D view of Example 3 (3D Arc)

Table 3 Statistical Results of Example 3

	PSO	GA	SCA	GWO	BHMO	EFA
Best	33.54	17.47	24.72	26.42	17.09	**16.90**
Average	33.54	17.47	24.72	26.42	17.09	**16.90**
Std.	0.00E+00	3.61E-15	1.81E-14	1.81E-14	1.45E-14	**0.00E+00**
Cluster No. (k)	2	2	2	4	2	**2**
Time Ratio (%)	22.63	50.50	**5.20**	5.68	6.16	9.84

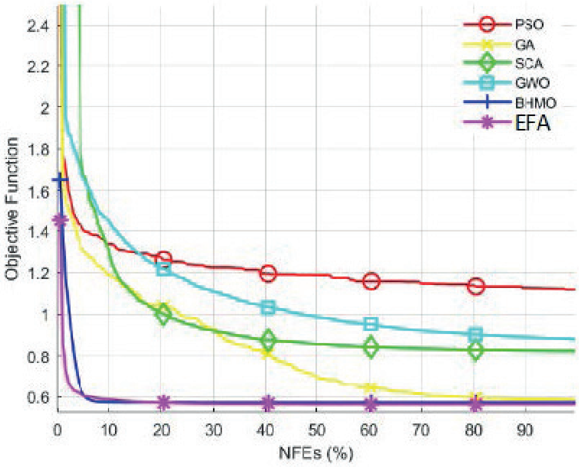

Fig. 11 The comparison between meta-heuristic algorithms (Example 3)

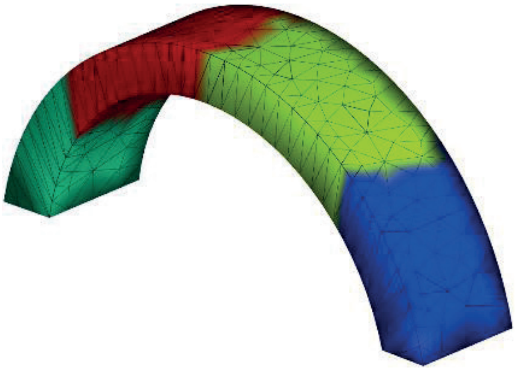

Fig. 12 The decomposed mesh into 4 clusters of Example 3 by the GWO algorithm

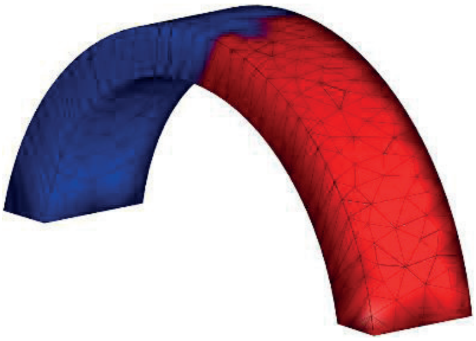

Fig. 13 The decomposed mesh into 2 clusters of Example 3 by EFA algorithm

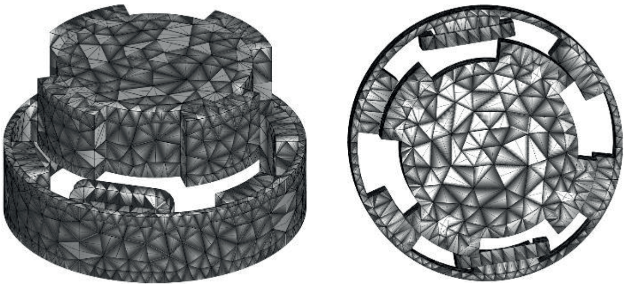

Fig. 14 3D mechanical motor; 3D, and top views

Table 4 Statistical Results of Example 4

	PSO	GA	SCA	GWO	BHMO	EFA
Best	34.73	28.20	26.36	27.96	23.75	**18.96**
Average	34.73	28.20	26.36	27.96	23.75	**18.96**
Std.	0.00E+00	1.45E-14	1.81E-14	1.81E-14	3.61E-15	**3.32E-15**
Cluster No. (k)	2	**3**	8	9	**3**	**3**
Time Ratio (%)	20.99	51.98	**5.62**	5.74	5.45	10.22

The comparison between the utilized algorithm is shown in Fig. 15. Also, the decomposed meshes obtained by different methods are illustrated in Figs. 16–19.

4.5 3D mechanical piece

Example 5 contains a three-dimensional mechanical piece, as illustrated in Fig. 20. The corresponding mesh contains 15506 nodes and 12700 tetrahedral elements. The statistical results are reported in Table 5.

The comparison between the utilized algorithm is shown in Fig. 21. Also, the decomposed meshes obtained by different methods are illustrated in Figs. 22–25.

4.6 3D plate with Two Holes

The last example contains a three-dimensional plate with two holes, as illustrated in Fig. 26. The corresponding mesh comprises 898 nodes and 2688 tetrahedral elements. The statistical results are reported in Table 6.

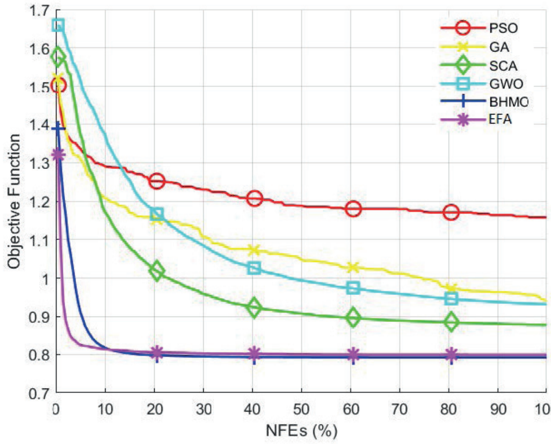

Fig. 15 The comparison between meta-heuristic algorithms (Example 4)

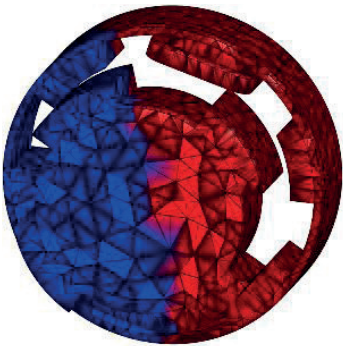

Fig. 16 The decomposed mesh into 2 clusters of Example 4 by the PSO algorithm

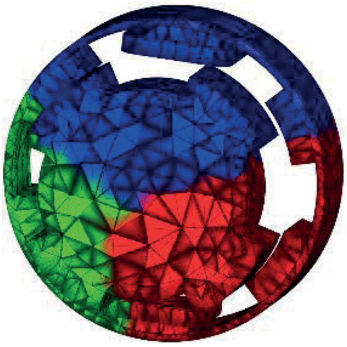

Fig. 17 The decomposed mesh into 3 clusters of Example 4 by the EFA algorithm

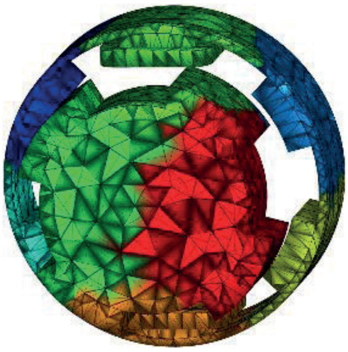

Fig. 18 The decomposed mesh into 8 clusters of Example 4 by the SCA algorithm

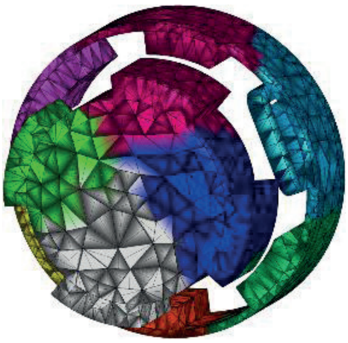

Fig. 19 The decomposed mesh into 9 clusters of Example 4 by the GWO algorithm

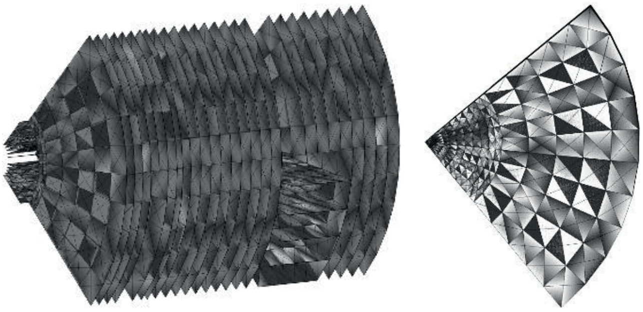

Fig. 20 The 3D mechanical piece; 3D and top views

Table 5 Statistical Results of Example 5

	PSO	GA	SCA	GWO	BHMO	EFA
Best	34.57	25.89	26.47	27.94	21.99	**18.92**
Average	34.57	25.89	26.47	27.94	21.99	**18.92**
Std.	1.45E-14	1.08E-14	1.81E-14	1.81E-14	**3.61E-15**	1.08E-14
Cluster No. (k)	6	**2**	4	7	**2**	**2**
Time Ratio (%)	22.75	49.97	5.77	6.40	**3.67**	11.44

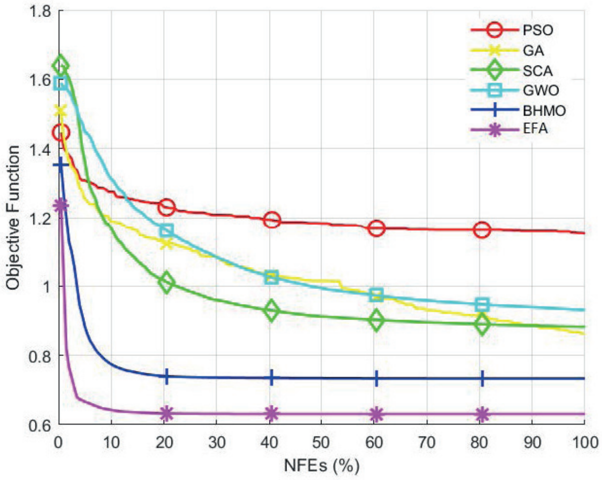

Fig. 21 The comparison between meta-heuristic algorithms (Example 5)

The comparison between the utilized algorithm is shown in Fig. 27. Also, the decomposed meshes obtained by different methods are illustrated in Figs. 28–30.

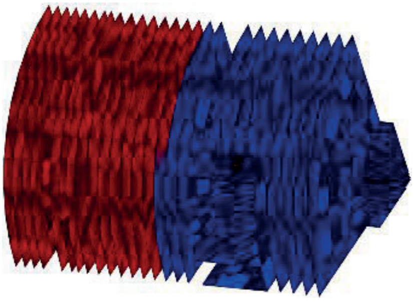

Fig. 22 The decomposed mesh into 2 clusters of Example 5 by the EFA algorithm

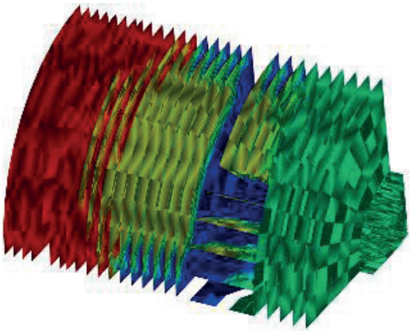

Fig. 23 The decomposed mesh into 4 clusters of Example 5 by the SCA algorithm

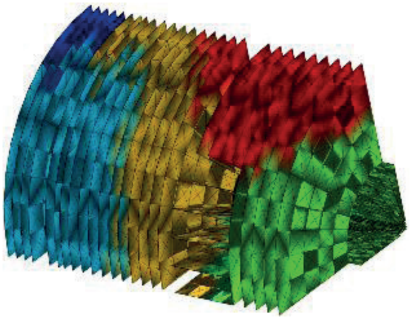

Fig. 24 The decomposed mesh into 8 clusters of Example 5 by the PSO algorithm

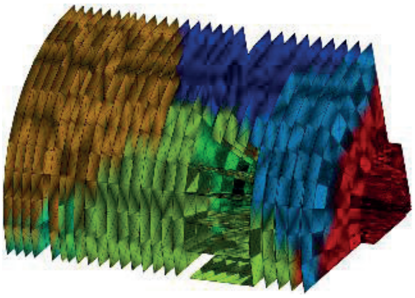

Fig. 25 The decomposed mesh into 9 clusters of Example 5 by the GWO algorithm

5 Conclusions

This paper has introduced an automatic data clustering technique in which some concepts of the graph theory are employs. In the presented method, a Cluster Validity Index (CVI) is used that is called DB Index. This CVI aids the algorithm to determine the number of clusters automatically in the optimization problem. Therefore, either the number of clusters or the decomposed set members will be

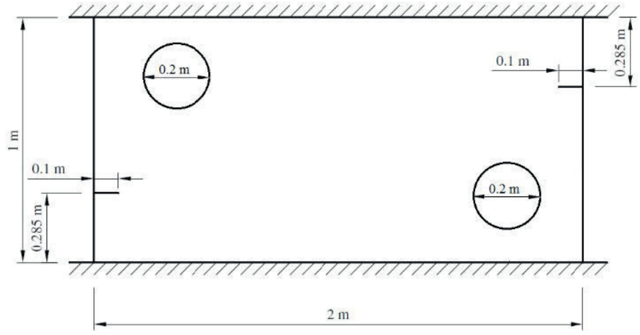

Fig. 26 The 3D plate with two holes; geometry in X-Y coordinate

Table 6 Statistical Results of Example 6

	PSO	GA	SCA	GWO	BHMO	EFA
Best	34.30	28.00	28.27	29.20	22.99	**18.92**
Average	34.30	28.00	28.27	29.20	22.99	**18.92**
Std.	2.17E-14	7.23E-15	7.23E-15	2.17E-14	3.61E-15	**0.00E+00**
Cluster No. (k)	**2**	**2**	7	8	**2**	**2**
Time Ratio (%)	20.43	52.47	5.39	5.66	10.85	**5.20**

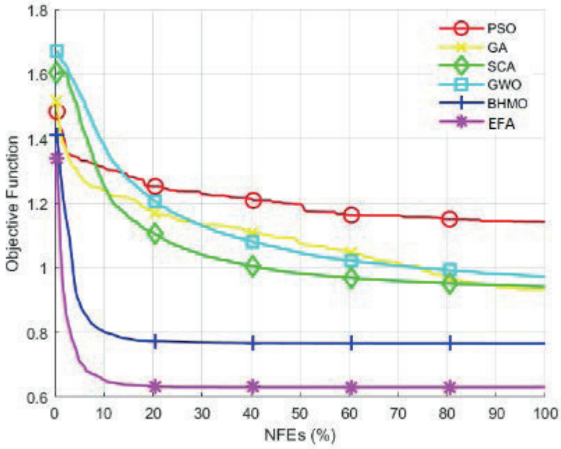

Fig. 27 The comparison between meta-heuristic algorithms (Example 6)

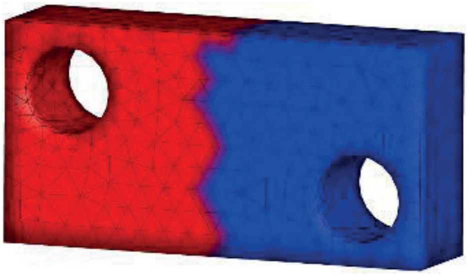

Fig. 28 The decomposed mesh into 2 clusters of Example 6 by EFA algorithm

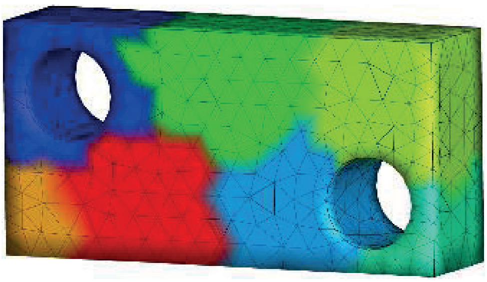

Fig. 29 The decomposed mesh into 7 clusters of Example 6 by the SCA algorithm

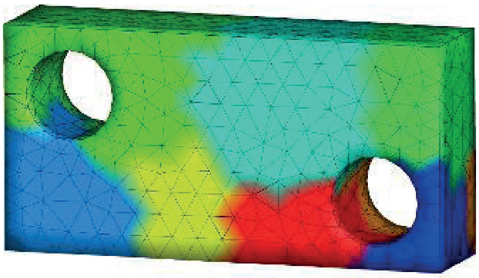

Fig. 30 The decomposed mesh into 8 clusters of Example 6 by the GWO algorithm

optimized. In order to evaluate the proposed method, two- and three- dimensional FEM meshes are considered. They have been decomposed into the sub-meshes utilizing six meta-heuristic algorithms. The set of examples comprises simple and more complicated FEM meshes. As reported in detail, the performance of the algorithms is approximately equal in simple problems such as the simple beam or mechanical lever. By increasing the complexity of the meshes, either due to geometry or type and number of elements, the performance of two newly developed algorithms, i.e., Enriched Firefly Algorithm (EFA) and Black Holes Mechanics Optimization (BHMO), is improving as well as their consumed time is decreasing. It may be due to graph-theoretical concepts that are employed in the principal procedure of these two algorithms.

For future works, it may be appropriate to develop a new objective function for domain decomposition based on FEM computations such as stress calculated stress fields in the domain. Also, it can be possible to utilize the adaptive version of algorithms to reduce the number of analyses and joint the proposed technique to the parallel processing codes to develop a general analysis program.

Compliance with ethical standards

Conflict of interest: No potential conflict of interest was reported by the authors.

References

[1] Csébfalvi, A. "A hybrid meta-heuristic method for continuous engineering optimization", Civil Engineering, 53(2), 93–100, 2009. https://doi.org/10.3311/pp.ci.2009-2.05

[2] Kaveh, A., Seddighian, M. R. "A New Nodal Stress Recovery Technique in Finite Element Method Using Colliding Bodies Optimization Algorithm", Periodica Polytechnica Civil Engineering, 63(4), pp. 1159–1170, 2019. https://doi.org/10.3311/ppci.14801

[3] Hasançebi, O., Kazemzadeh Azad, S. "Discrete Sizing of Steel Frames Using Adaptive Dimensional Search Algorithm", Periodica Polytechnica Civil Engineering, 63(4), pp. 1062–1079, 2019. https://doi.org/10.3311/ppci.14746

[4] Kaveh, A., Hoseini Vaez, S. R., Hosseini, P., Ezzati, E. "Layout Optimization of Planar Braced Frames Using Modified Dolphin Monitoring Operator", Periodica Polytechnica Civil Engineering, 62(3), pp. 717–731, 2018. https://doi.org/10.3311/ppci.11654

[5] Kaveh, A., Hoseini Vaez, S. R., Hosseini, P. "Performance of the Modified Dolphin Monitoring Operator for Weight Optimization of Skeletal Structures", Periodica Polytechnica Civil Engineering, 63(1), pp. 30–45, 2019. https://doi.org/10.3311/ppci.12544

[6] Khoei, A. R., Moslemi, H., Seddighian, M. R. "An efficient stress recovery technique in adaptive finite element method using artificial neural network", Engineering Fracture Mechanics, 237, Article number: 107231, 2020. https://doi.org/10.1016/j.engfracmech.2020.107231

[7] Kaveh, A., Seddighian, M. R., Ghanadpour, E. "Upper and lower bounds for the collapse load factor of rectangular grids using FEM", International Journal of Optimization in Civil Engineering, 9(3), pp. 543–554, 2019. http://ijoce.iust.ac.ir/article-1-406-en.html

[8] Das, S., Abraham, A., Konar, A. "Automatic Hard Clustering Using Improved Differential Evolution Algorithm", In: Metaheuristic Clustering. Studies in Computational Intelligence, Springer, Berlin, Heidelberg, 2007, pp. 218–237. https://doi.org/10.1007/978-3-540-93964-1_4

[9] Farhat, C. "A simple and efficient automatic FEM domain decomposer", Computers & Structures, 28(5), pp. 579–602, 1988. https://doi.org/10.1016/0045-7949(88)90004-1

[10] Farhat, C., Lesoinne, M. "Automatic partitioning of unstructured meshes for the parallel solution of problems in computational mechanics", International Journal for Numerical Methods in Engineering, 36(5), pp. 745–764, 1993. https://doi.org/10.1002/nme.1620360503

[11] Khan, A. I., Topping, B. H. V. "Parallel Adaptive Mesh Generation", Computing Systems in Engineering, 2(1), pp. 75–101, 1991.

[12] Kaveh, A., Roosta, G. R. "An Algorithm for Partitioning of Finite Element Meshes", In: Topping, B. H. V. (ed.) Advances in Computational Structures Technology, Civil-Comp Press, Edinburgh, UK, 1996, pp. 289–296. https://doi.org/10.4203/ccp.38.7.8

[13] Kaveh, A., Shojaee, S. "Optimal domain decomposition via p-median methodology using ACO and hybrid ACGA", Finite Elements in Analysis and Design, 44, pp. 505–512, 2008.
https://doi.org/10.1016/j.finel.2008.01.005

[14] Kaveh, A., Seddighian, S. "Domain decomposition of finite element models utilizing eight meta-heuristic algorithms: A comparative study", Mechanics Based Design of Structures and Machines, 2020.
https://doi.org/10.1080/15397734.2020.1781655

[15] Kaveh, A., Sharafi, P. "Ant colony optimization for finding medians of weighted graphs", Engineering Computations, 25(2), pp. 102–120, 2008.
https://doi.org/10.1108/02644400810855931

[16] Kaveh, A., Mahdavi, V. R. "Colliding Bodies Optimization: Extensions and Applications", Springer, Cham, Switzerland, 2015.
https://doi.org/10.1007/978-3-319-19659-6_2

[17] Beitollahi, A., Kaveh, A., Mahdavi, V. R. (2018) "Locating Emergency Facilities Using the Weighted k-median Problem: A Graph-metaheuristic Approach", Periodica Polytechnica Civil Engineering, 62(1), pp. 200–205, 2018.
https://doi.org/10.3311/ppci.10665

[18] Christofides, N., Beasley, J. E. "A tree search algorithm for the p-median problem", European Journal of Operational Research, 10, pp. 196–204, 1982.
https://doi.org/10.1016/0377-2217(82)90160-6

[19] Osman, I. H., Christofides, N. "Capacitated clustering problems by hybrid simulated annealing and tabu search", International Transactions in Operational Research,1(3), pp. 317–336, 1994.
https://doi.org/10.1016/0969-6016(94)90032-9

[20] Kariv, O., Hakimi, S. L. "An algorithmic approach to network location problems. I: The p-centers", SIAM Journal on Applied Mathematics, 37(3), pp. 513–538, 1979.
https://doi.org/10.1137/0137040

[21] Estivill-Castro, V., Torres-Velázquez, R. "Hybrid genetic algorithm for solving the p-median problem", In: Simulated Evolution and Learning Second Asia-Pacific Conference on Simulated Evolution and Learning, SEAL'98 Canberra, Australia, 1998, pp. 18–25.
https://doi.org/10.1007/3-540-48873-1_4

[22] Frigui, H., Krishnapuram, R. "A robust competitive clustering algorithm with applications in computer vision", IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(5), pp. 450–465, 1999.
https://doi.org/10.1109/34.765656

[23] Leung, Y., Zhang, J.-S., Xu, Z.-B. "Clustering by scale-space filtering", IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(12), pp. 1396–1410, 2000.
https://doi.org/10.1109/34.895974

[24] Jain, A. K., Murty, M. N., Flynn, P. J. "Data clustering: a review", ACM Computing Surveys, 31(3), pp. 264–323, 1999.
https://doi.org/10.1145/331499.331504

[25] Forgy, E. "Cluster analysis of multivariate data: efficiency versus interpretability of classifications", Biometrics, 21, pp. 768–780, 1965.

[26] Zahn, C. T. "Graph-theoretical Methods for Detecting and Describing Gestalt Clusters", IEEE Transactions on Computers, C-20(1), pp. 68–86, 1971.
https://doi.org/10.1109/T-C.1971.223083

[27] Mitchell, T. M. "Does Machine Learning Really Work?", AI Magazine, 18(3), pp. 11–20, 1997 .
https://doi.org/10.1609/aimag.v18i3.1303

[28] Mao, J., Jain, A. K. "Artificial neural networks for feature extraction and multivariate data projection", IEEE Transactions on Neural Networks, 6(2), pp. 296–317, 1995.
https://doi.org/10.1109/72.363467

[29] Pal, N. R., Bezdek, J. C., Tsao, E. C.-K. "Generalized clustering networks and Kohonen's self-organizing scheme", IEEE Transactions on Neural Networks, 4(4), pp. 549–557, 1993.
https://doi.org/10.1109/72.238310

[30] Deboeck, G., Kohonen, T. "Visual Explorations in Finance with Self-Organizing Maps", Springer, London, UK, 2013.
https://doi.org/10.1007/978-1-4471-3913-3

[31] Falkenauer, E. "Genetic algorithms and grouping problems", Wiley & Sons, New York, NY, USA, 1998.
https://doi.org/10.1002/(SICI)1097-024

[32] Paterlini, S., Minerva, T. "Evolutionary approaches for cluster analysis", In: Bonarini, A., Masulli, F., Pasi, G. (eds.) Soft Computing Applications, Physica, Heidelberg, Germany, 2003, pp. 165–176, 2003.
https://doi.org/10.1007/978-3-7908-1768-3_15

[33] Kennedy, J., Eberhart, R. "Particle swarm optimization", In: Proceedings of ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942–1948.
https://doi.org/10.1109/ICNN.1995.488968

[34] Goldberg, D. E. "Genetic Algorithms in Search, Optimization & Machine Learning", Pearson Education, New Delhi, India, 2006.
https://doi.org/10.5860/choice.27-0936

[35] Mirjalili, S., Mirjalili, S. M., Lewis, A. "Grey Wolf Optimizer", Advances in Engineering Software, 69, pp. 46–61, 2014.
https://doi.org/10.1016/j.advengsoft.2013.12.007

[36] Mirjalili, S. "SCA: A Sine Cosine Algorithm for solving optimization problems", Knowledge-Based System, 96, pp. 120–133, 2016.
https://doi.org/10.1016/j.knosys.2015.12.022

[37] Kaveh, A., Seddighian, M. R., Ghanadpour, E. "Black Holes Mechanics Optimization: a novel meta-heuristic algorithm", Asian Journal of Civil Engineering, 21, pp. 1129–1149, 2020.
https://doi.org/10.1007/s42107-020-00282-8

[38] Kaveh, A., Seddighian, M. R. "Simultaneously multi-material layout, and connectivity optimization of truss structures via an Enriched Firefly Algorithm", Structures, 27, pp. 2217–2231, 2020.
https://doi.org/10.1016/j.istruc.2020.07.029

[39] Kaveh, A. "Optimal Analysis of Structures by Concepts of Symmetry and Regularity", Springer, Vienna, Austria, 2013.
https://doi.org/10.1007/978-3-7091-1565-7

[40] Konar, A. "Computational Intelligence: Principles, Techniques and Applications", Springer Science & Business Media, Berlin, Heidelberg, Germany, 2006.
https://doi.org/10.1007/b138935

[41] Brucker, P. "On the Complexity of Clustering Problems", In: Henn, R., Korte, B., Oettli, W. (eds.) Optimization and Operations Research, Springer, Berlin, Heidelberg, Germany, pp. 45–54.
https://doi.org/10.1007/978-3-642-95322-4_5

[42] Caliński, T., Harabasz, J. "A dendrite method for cluster analysis", Communications in Statistics, 3(1), pp. 1–27, 1974. https://doi.org/10.1080/03610927408827101

[43] Davies, D. L., Bouldin, D. W. "A Cluster Separation Measure", IEEE Transactions on Pattern Analysis and Machine Intelligence, 1(2), pp. 224–227, 1979.

[44] Pakhira, M. K., Bandyopadhyay, S., Maulik, U. "Validity index for crisp and fuzzy clusters", Pattern Recognition, 37, pp. 487–501, 2004. https://doi.org/10.1016/j.patcog.2003.06.005

[45] Chou, C.-H., Su, M.-C., Lai, E. "A new cluster validity measure and its application to image compression", Pattern Analysis and Applications, 7, pp. 205–220, 2004. https://doi.org/10.1007/s10044-004-0218-1