We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Partitional Clustering

Uğurhan Kutbay

Additional information is available at the end of the chapter

Abstract

People are living in a world full of data. Humans are collecting data from many measurements and observations in their daily works. The sorting of these numerous data is important and necessary in terms of analyzing, reasoning, and decision-making processes. For this reason, clustering has been used in many areas and has become very important in recent years. Feature selection and classifying the data in subsets can be changed data to data. As a result of these feature selection methods, some clustering methods have been revealed. Hierarchical clustering, partitional clustering, artificial system clustering, kernel-based clustering, and sequential data clustering are determined for different clustering strategies. This chapter examines some popular partitional clustering techniques and algorithms. Partitional clustering assigns a set of data points into k-clusters by using iterative processes. The predefined criterion function (J) assigns the datum into k^{th} number set. As a result of this criterion function value in k sets (maximization and minimization calculation), clustering can be done. This chapter starts with criterion function for clustering process. In addition, some applications will be done for each algorithm in this chapter.

Keywords: partitional clustering, K-means, fuzzy clustering, C-FCM, genetic algorithm

1. Introduction

The choice of feature types and measurement levels depends on data type. For this reason, many clustering methods have been developed. According to clustering strategies, these methods can be classified as hierarchical clustering [1–3], partitional clustering [4, 5], artificial system clustering [6], kernel-based clustering and sequential data clustering. This chapter examines some popular partitional clustering techniques and algorithms.
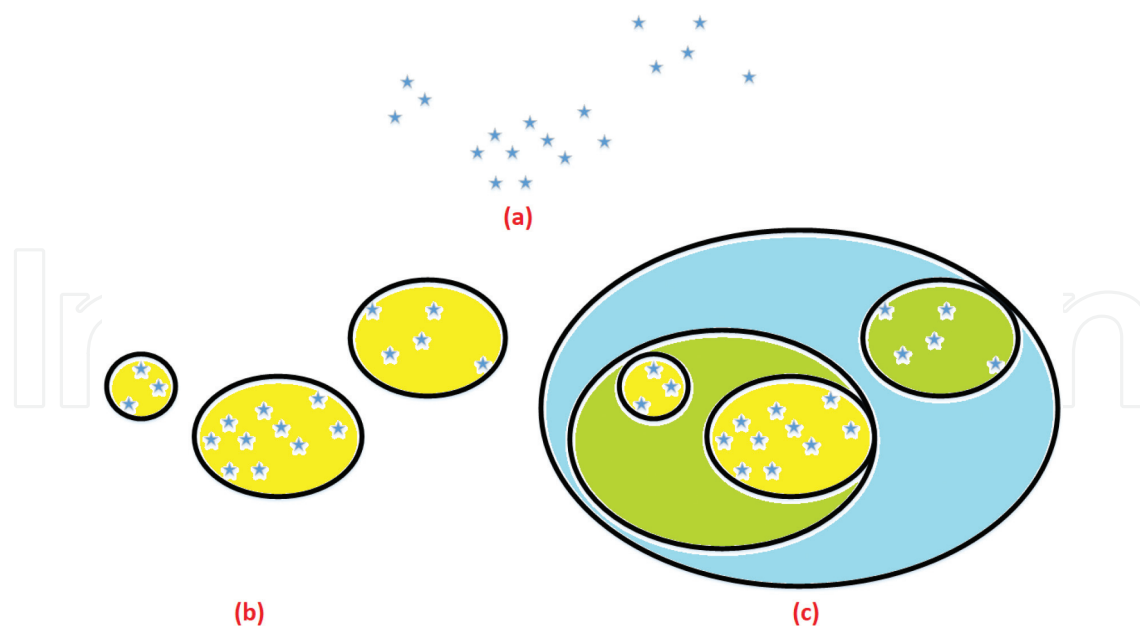
Figure 1. Clustering techniques: (a) data set; (b) partitional clustering; and (c) hierarchical clustering.

In contrast to hierarchical clustering methods, partitional clustering aims successive clusters using some iterative processes. Partitional clustering assigns a set of data points into k-clusters by using iterative processes. In these processes, n data are classified into k-clusters. The predefined criterion function J assigns the datum into k^{th} number set according to the maximization and minimization calculation in k sets.

Figure 1 represents the hierarchical clustering and partitional clustering. In addition, hierarchical clustering, all sub-clusters defined in another sub-cluster shown in **Figure 1**. **Figure 1a** represents the raw data, **Figure 1b** shows the partitional clustering and **Figure 1c** represents the hierarchical clustering. In hierarchical clustering, raw data are firstly clustered in some subgroups (three-clustered shape). After that procedure, subgroups hierarchically defined in two green clusters. Last procedure includes all these clusters which are defined in the union set.

This chapter starts with an introduction to clustering criteria and continues with K-Means algorithm, different fuzzy clustering techniques and genetic algorithm-based clustering.

2. Clustering criteria

Clustering aims to seek a partition of the data in the same homogenous clusters [7]. This homogeneity and finding the exact clustering are evaluated only by using criterion functions. One of the most popular techniques of the criterion function is the summing of squared-error (SSE) criterion and similar methods are used such as mean-square-error (MSE), normalized mean-square-error (NMSE), and so on. Sum of squared error criterion can be defined as:

$$J(\Gamma, \mathrm{M}) = \sum_{i=1}^{K} \sum_{j=1}^{N} \gamma_{ij} \|x_j - m_i\|^2 = \sum_{i=1}^{K} \sum_{j=1}^{N} \gamma_{ij} (x_j - m_i)^T (x_j - m_i) \qquad (1)$$

where Γ is a partition matrix of γ_{ij} and defined as;

$$\gamma_{ij} = \begin{cases} 1 & if \ \ x_j \in cluster \ \ i \\ 0 & otherwise \end{cases} \qquad (2)$$

M is the cluster prototype or centroid matrix and m_i defined as;

$m_i = \frac{1}{N_i} \sum_{i=1}^{N} \gamma_{ij} x_j$ is the sample mean for the ith number cluster corresponding to N_i objects.

The partition minimizes the sum-square-error (SSE). When the SSE regarded is minimum, minimum variance partition will be achieved. As a result of this calculation, optimum cluster is determined.

3. K-means algorithm

The K-Means clusters were first developed by Mac Queen [8]. In the K-Means clusters, clusters are formed using Euclidean distance. In the K-Means algorithm, unsupervised learning is used and k classes are created which minimize the error function [9].

In K-Means clustering, k cluster centers are created from the selected data set. It is then placed at the nearest cluster using Euclidean distance. New cluster centers are formed according to the results of the clustering. From the calculations of the clustering, the cluster center is recalculated. The arithmetic average is used as the calculation method, and the new cluster center is determined. All samples are reclassified according to the new center. This process is repeated until it is determined that the samples in the set have not passed to another set.

The partitioning of the k pieces of data x is represented by the minimization of the J parameter as in (3).

$$J = \min \left(\sum_{k} \sum_{x \in c_k} w_x dist(x, o_k) \right) \qquad (3)$$

If the data are classified in a cluster near the center of the nearest cluster, the J value will be the minimum. If the data x are classified in the kth number cluster, the value can be optimized by changing the weighting value of w_x to obtain the minimum J value. $dist(x, o_k)$ is the notation which represents the distance function. In this formula, x represents the pixel data, o_k is the center of the cluster. k sets are shown as in (4).
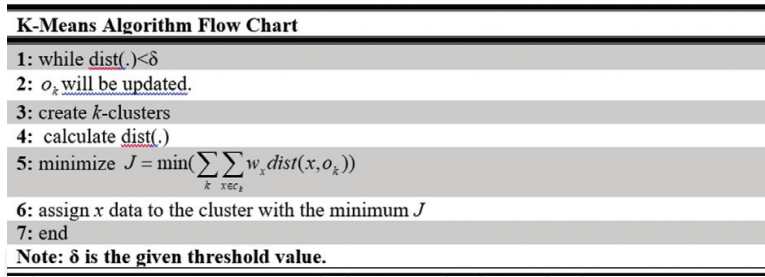
K-Means Algorithm Flow Chart

1: while $\underline{dist(.)} < \delta$
2: o_k will be updated.
3: create k-clusters
4:   calculate $\underline{dist(.)}$
5: minimize $J = \min(\sum_k \sum_{x \in c_k} w_x dist(x, o_k))$

6: assign x data to the cluster with the minimum J
7: end
Note: δ is the given threshold value.

Figure 2. Flow chart of the K-Means algorithm.

$$c_k = (c_{k,1}, c_{k,2}, \ldots, c_{k,n}) \tag{4}$$

The distance function can be calculated as in Euclidean distance for k clusters in the next formula (5).

$$dist\left(x_j^i, o_k\right) = \sqrt{\left(x_{j,1}^i - o_{k,1}\right)^2 + \left(x_{j,1}^i - o_{k,2}\right)^2 + \ldots + \left(x_{j,n}^i - o_{k,n}\right)^2} \tag{5}$$

The iterative operations are repeated as in the flow diagram in **Figure 2**. This flowchart represents the iterative distance controlled operations to minimize the J parameter.

When this algorithm is applied for three-cluster for the original image shown in **Figure 3a**, this image can be represented in three clusters as in **Figure 3b–d**, respectively. Cluster-1 shows the
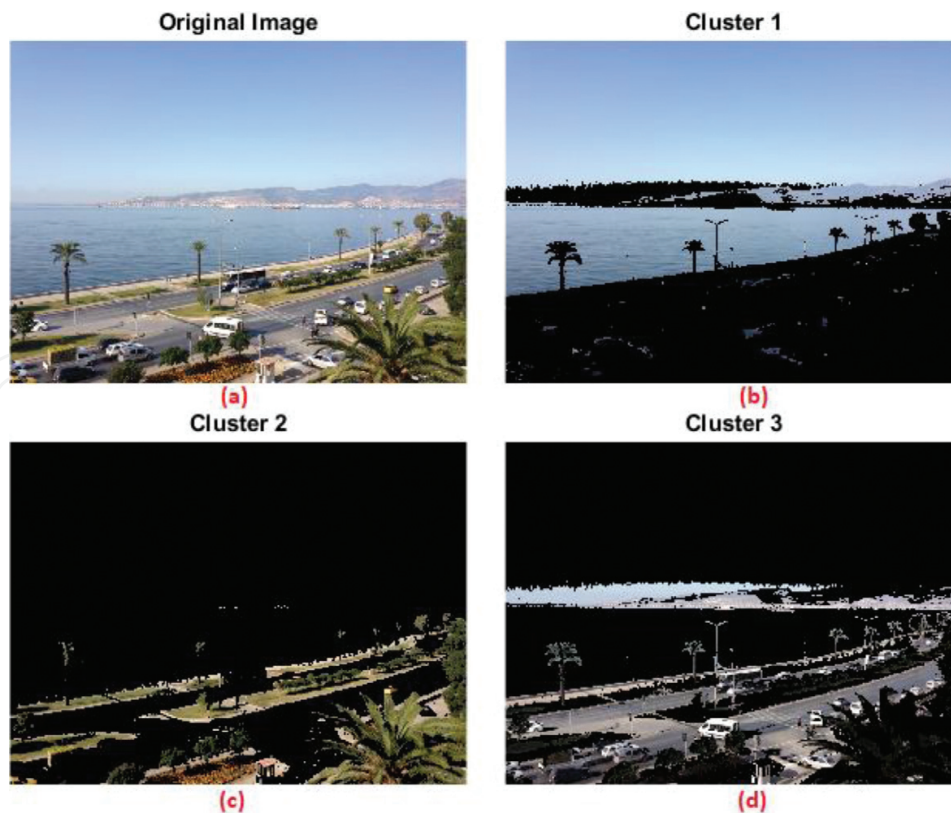

Figure 3. K-means algorithm results. (a) Original image; (b) K-Means Cluster No:1; (c) K-Means Cluster No:2; and (d) K-Means Cluster No:3.

sea for the given landscape, cluster-2 shows the the green areas in the landscape and cluster-3 shows the roads for the given landscape.

4. Fuzzy clustering

Fuzzy theory is firstly developed by Zadeh [10] for defining adjustable degrees of member-ships. Fuzzy theory creates intermediate sets rather than classical sets. In classical sets, each data item is assigned into only one cluster. In contrast, data in fuzzy clusters can be represented in multiple clusters. This multiset assignment can belong to all the clusters with a certain degree of membership defined by Bezdek [11]. This one item in multiset representation can be useful for sharply separated cluster boundaries.

The fuzzy C-Mean algorithm (FCM) is frequently used because of its ease of operation and reliability in many applications [12–15]. Conventional Fuzzy C-Mean (FCM) works with the principle of minimizing the objective function [16] shown in the following formula (6):

$$J_{FCM} = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m} d_{ik}^{2} \tag{6}$$

where u_{ik}^{m} is the membership function in the range [0,1]. This membership represents the membership degree of x_k for the kth pixel. k is defined in the range of $k \in [1, n]$ for the ith numbered cluster. Total cluster size (c) given in the range of $[1, c]$. In the formula (7), d_{ik}^{2} represents the distance between x_k and i^{th}-cluster center (v_i).

$$d_{ik}^{2} = \|x_k - v_i\| \tag{7}$$

The fuzzy set theory aims that the membership function is $\sum_{i=1}^{c} u_{ik} = 1$ for each pixel. Using the FCM membership function u_{ik}^{m} and cluster center v_i, FCM targets to reach local minimums by using the equivalence (8) and (9), respectively.

$$u_{ik}^{*} = \left\{ \sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(m-1)} \right\}^{-1} \quad \forall i \in [1, c] \ \text{and} \ k \in [1, n] \tag{8}$$

$$v_{i}^{*} = \frac{\sum_{k=1}^{n} u_{ik}^{m} x_k}{\sum_{k=1}^{n} u_{ik}^{m}} \quad \forall i \in [1, c] \tag{9}$$

FCM algorithm flow chart is shown in **Figure 4**. This algorithm will be done in the given iteration step specified in this traditional clustering method. The cluster centers are updated in each iteration step to calculate the membership function.

When the FCM algorithm is applied for the given original image shown in **Figure 5a** for three clusters, this image can be represented in three clusters as in **Figure 5b–d**, respectively. This algorithm can only be applied for gray-scaled images. There is no crisp clustering for the given
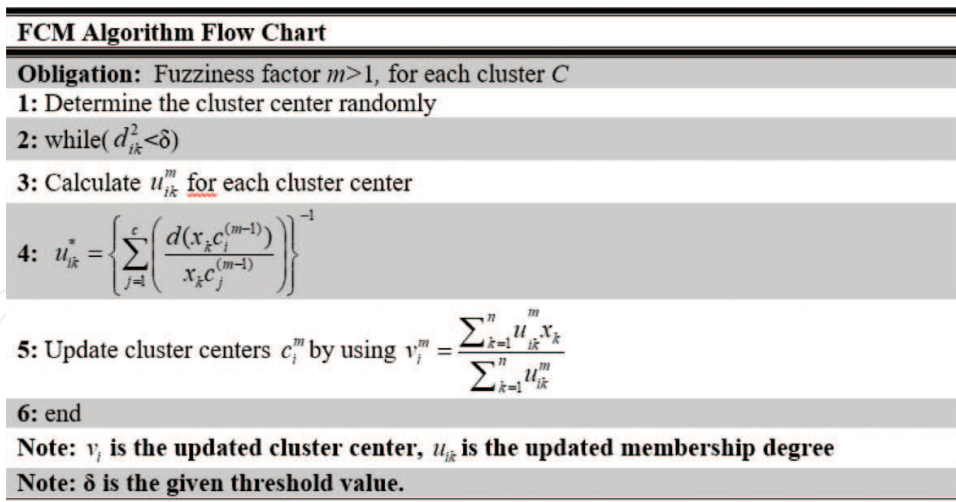
FCM Algorithm Flow Chart

Obligation: Fuzziness factor $m>1$, for each cluster C

1: Determine the cluster center randomly

2: while($d_{ik}^2 < \delta$)

3: Calculate u_{ik}^m for each cluster center

4: $u_{ik}^* = \left\{ \sum_{j=1}^{c} \left(\dfrac{d(x_k c_i^{(m-1)})}{x_k c_j^{(m-1)}} \right) \right\}^{-1}$

5: Update cluster centers c_i^m by using $v_i^m = \dfrac{\sum_{k=1}^{n} u_{ik}^m x_k}{\sum_{k=1}^{n} u_{ik}^m}$

6: end

Note: v_i is the updated cluster center, u_{ik} is the updated membership degree

Note: δ is the given threshold value.

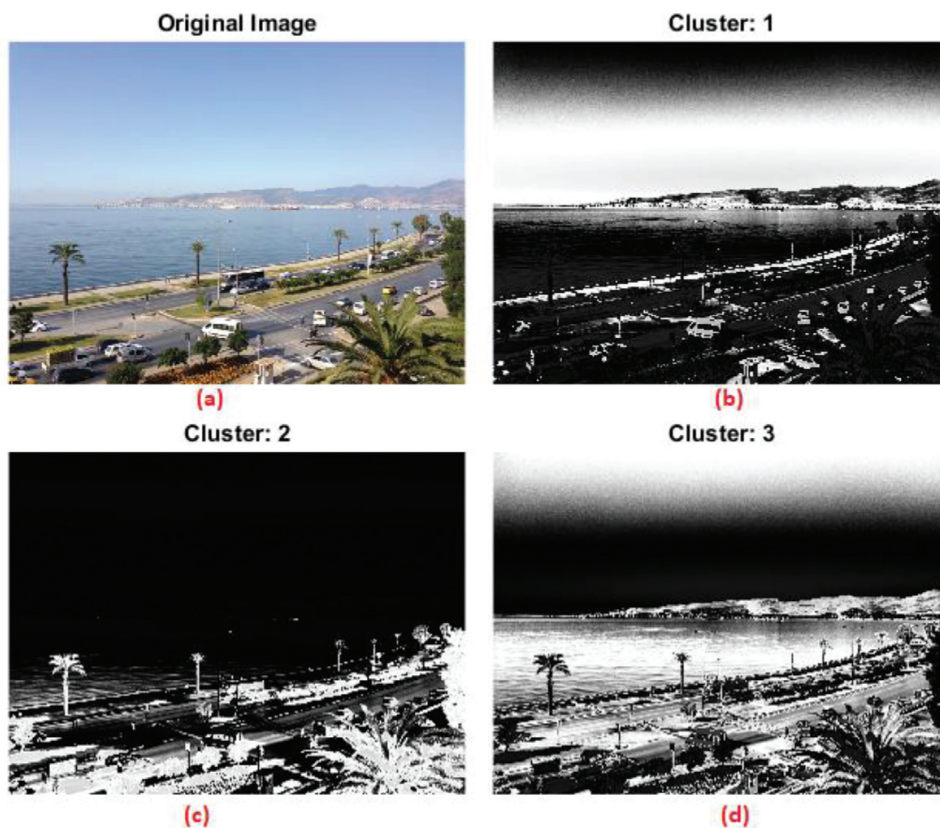**Figure 4.** Flow chart of the FCM algorithm.

Figure 5. FCM algorithm results. (a) Original image; (b) FCM Cluster No:1; (c) FCM Cluster No:2; and (d) FCM Cluster No:3.

landscape image. This image includes many objects in different colors, and these colors are generally indented in these objects.

FCM algorithm includes iterative procecesses. For the given image at the end of the 55 iteration steps, clustering processes are completed for the given threshold value which is 10^{-5} for FCM distance change.

5. Colored FCM

Colored Image Fuzzy C-Mean (C-FCM) involves color-based clustering using fuzzy sets. This 3D method is firstly given by Kutbay and Hardalaç [17] as Robust Colored Image FCM (RCI-FCM), but this presented method is different from RCI-FCM. In RCI-FCM, distances are calculated for each R, G and B channels, but in this method, the mean distance is calculated for RGB color spaces. This method represents the RGB color formed images in FCM, which uses FCM-based algorithm in colored images. The membership function calculates the centroids of the clusters for each R, G and B color spaces. After calculation of Euclidian distance for each channel, the mean distance could be calculated shown in equivalence (10).

$$d^2_{ik(R,G,B)} = mean\left(\left\|x_{k(R)} - v_{i(R)}\right\|, \left\|x_{k(G)} - v_{i(G)}\right\|, \left\|x_{k(B)} - v_{i(B)}\right\|\right) \tag{10}$$

Membership for each cluster can be calculated in the following formula (11).

$$u^*_{ik(R,G,B)} = \left\{ \sum_{j=1}^{c} \left(\frac{d_{ik(R,G,B)}}{d_{jk(R,G,B)}}\right)^{2/(m-1)} \right\}^{-1} \quad \forall i \in [1,c] \ \ and \ \ k \in [1,n] \tag{11}$$

where $u^*_{ik(R,G,B)}$ represents the membership degree for mean distance for each color. In this representation for each item, cluster's membership can be calculated and statistically membership could be defined for each cluster.

New cluster center can be calculated in the following equivalence (12) for each cluster:

$$v^*_{i(R,G,B)} = \frac{\sum_{k=1}^{n} u^m_{ik(R,G,B)} x_{k(R,G,B)}}{\sum_{k=1}^{n} u^m_{ik(R,G,B)}} \quad \forall i \in [1,c] \tag{12}$$

where $u^*_{ik(R,G,B)}$ is the membership function of each RGB color pixel. This function calculates for each R, G and B colors, for the each RGB space representation. c denotes the clusters for each n-pixel, and $d_{ik(R,G,B)}$ explains the distance between pixel x_k and the centroid v_i for i^{th} number cluster for each color domain. $v^*_{i(R,G,B)}$ represents the centers of the clusters for each RGB pixel value into the c-cluster. C-FCM algorithm's flow chart is shown in **Figure 6**. The proposed method aims to create c^3 cluster for the given stopping criteria. New cluster centroids are calculated from the results of the membership function for all RGB colors.

C-FCM algorithm flow chart is shown in **Figure 6**. The flowchart shows the C-FCM algorithm for colored images. For each iteration step, RGB distances are calculated and cluster centers are calculated. After updating the cluster centers in the given threshold value, each item will be assigned into the given cluster.

Figure 7 represents the C-FCM algorithm, which is applied for the given original image shown in **Figure 7a**. For each color, pixels assigned into two cluster. For RGB color space 2^3 cluster are created for this image. These clusters can be given in eight clusters as in **Figure 4b–i**, respectively.

C-FCM Algorithm Flow Chart

Obligation: Fuzziness factor $m>1$, for each cluster C

1: Initialize the cluster center randomly for each color RGB

2: while($d_{ik}^2 < \delta$)

3: Calculate $d_{ik(R,G,B)}^2$

3: Calculate u_{ik}^m for each cluster center

4: $u_{ik}^* = \left\{ \sum\limits_{j=1}^{c} \left(\dfrac{d(x_k)}{x_k c_j^{(m-1)}} \right) \right\}^{-1}$

5: Update cluster centers c_i^m by using $v_i^m = \dfrac{\sum_{k=1}^{n} u_{ik}^m x_k}{\sum_{k=1}^{n} u_{ik}^m}$

6: end

Note: v_i **is the updated cluster center,** u_{ik} **is the updated membership degree**

Note: δ **is the given threshold value.**

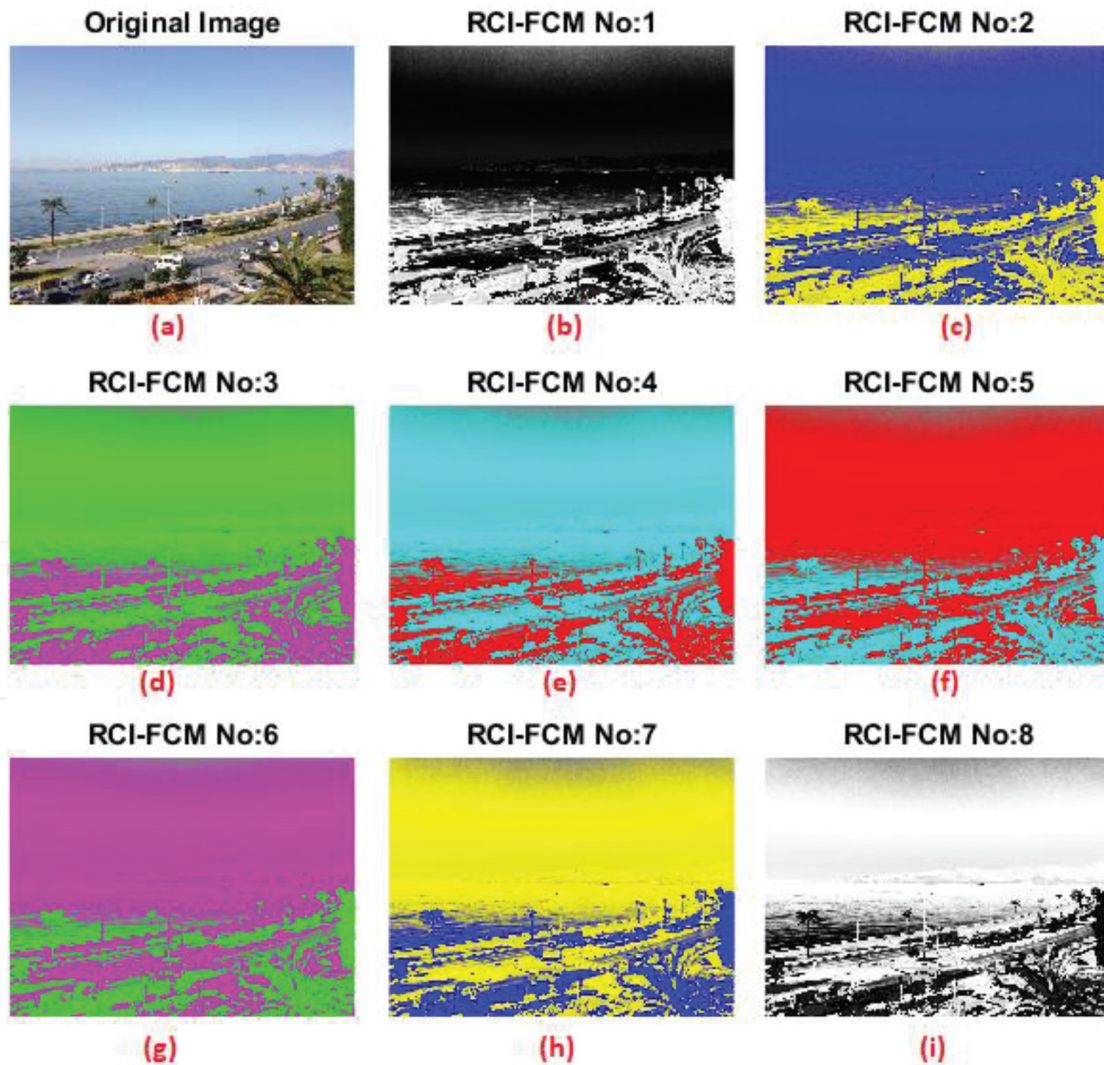**Figure 6.** Flow chart of the C-FCM algorithm.

Figure 7. C-FCM algorithm results. (a)Original image; (b)–(i). C-FCM Cluster No:1–8.

C-FCM algorithm includes iterative procecesses. For the given image at the end of the 26 iteration steps, clustering processes are completed for the given threshold value which is 10^{-5} for C-FCM distance change.

6. Genetic clustering

Genetic algorithm is very popular method in evolutionary computation processes. This method is firstly developed by Holland in 1975 [18]. This algorithm includes natural evolutionary processes. This method optimizes a population of the structure by using a set of evolutionary operators.

This method maintains a population of structures and these structures consisting of individuals. Each individual is evaluated by a function named as fitness function. These processes includes selection, recombination and mutation processes.

In genetic algorithms (GAs), each individual represents a candidate solution in binary form. This individual called as chromosome. After an initial population is generated, randomly crossover and mutation processes are executed for each iteration step.

For genetic algorithm examination, the following terms are useful for describing the concept of genetic algorithms. These are gene, chromosome, population (mass), reproduction process, and conformity value.

Gen is a unit that carries partial information. By bringing together these units, the chromosomal sequence that forms the solution cluster comes into play; for this reason, the genome decides well how to code it.

Chromosomes are structures that contain the information of the problem solving. Population is formed by the combination of chromosomes. At the initiative of the designer, what information is to be found on the chromosome.

The population is called the heap of possible solutions. In the GA process, the population size remains constant, but the bad chromosomes separate from the stack. The size of the heap is a very important concept, which must be well established, as the overcrowded heap increases the time of possible heuristic approach, while the small heap may cause no possible solution at all.

The reproduction process is the process of selecting the sequences to be transferred from the current stack to the next stack. The sequences carried are genetically the most appropriate sequences. The requirement for transition is whether the level of conformity specified has been achieved.

The fitness value, in genetic algorithms, which specifies which index will transfer the next generation, which index will be lost. Conformity value reflects the purpose of the problem.

Bandyopadhyay and Maulik [19] attempted to use GA to automatically determine the number of clusters K in 2002. The GA clustering aims assigning the data into k^{th} cluster using genetic processes.

Showing the basic concept of the GA-based clustering, genetic guided algorithm [20] is used. Fitness value is represented in fuzzy clusters as shown in equivalent (13);

$$j(M) = \sum_{j=1}^{N} \left(\sum_{i=1}^{K} D_{ij}^{1/(1-m)} \right)^{1-m} \tag{13}$$

where $D_{ij}i \in [1, K]$ and $j \in [1, N]$. D_{ij} represent the distance between the ith cluster prototype vector and the data object of the number of j. m represents the fuzzification coefficient. Calculation process of the genetic algorithm is shown in **Figure 8**.
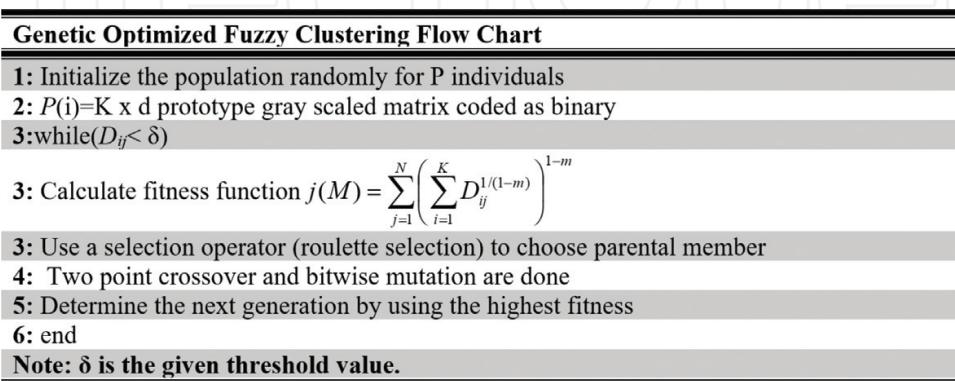
In this genetic parameter optimized algorithm, firstly P individuals are initialized, and each individual represents $K \ x \ d$ prototype matrix encoded as gray codes. After this prototype matrix representation, fitness values are calculated. After this calculated fitness value, tournament selection is used for parental member reproduction. For generating new parents, two-point crossover and bitwise mutation are done. For these new individuals, the highest fitness values are obtained using this fitness equation. All these processes are applied until termination condition ($D_{ij} < \delta$) is satisfied.

Chromosomes are randomly selected, and best parent are selected in tournament selection process. After tournament selection process, two-point crosover process and bitwise mutuation are done, respectively. These processes are shown in **Figures 9** and **10**, respectively.
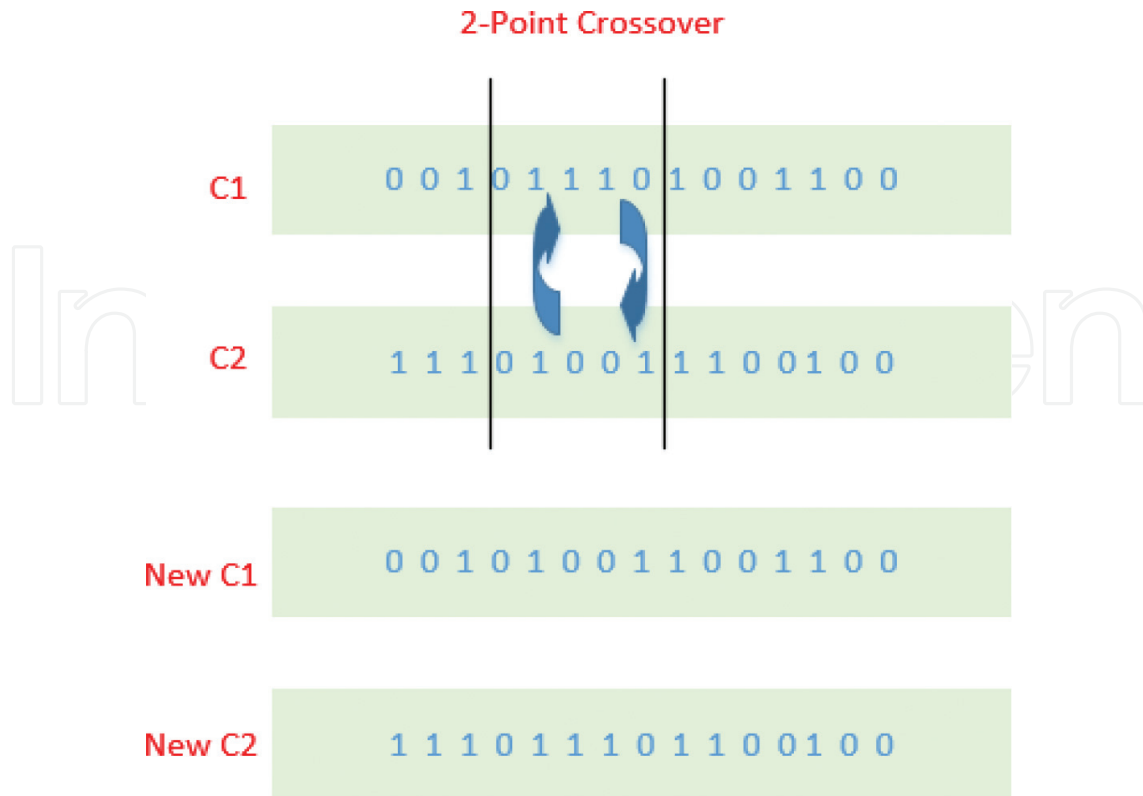
Figure 9 represents crossover process. In two-point crossover process, bitwise crossover points are determined. After determination process, crossover process will be done.

There are some crossover processes are used in different works [21–23]. These types are 1-point crossover, K-point crossover, shuffle crosover, and so on [24].
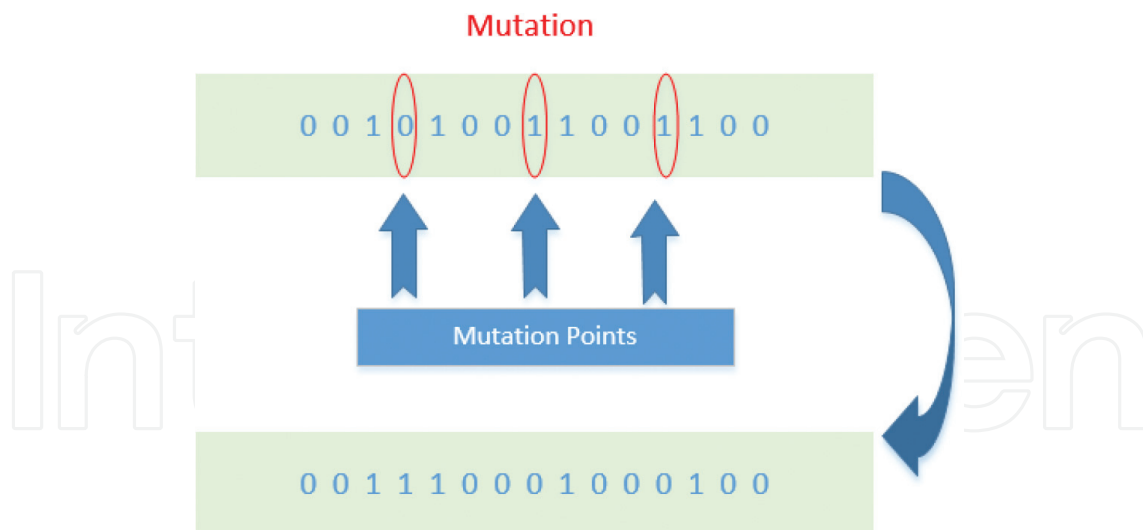
Mutation process helps the genetically variations from parent chromosomes to child chromosomes. There are many mutation operators defined in the literature. These are bit-flip-mutation, swap-mutation and inversion-mutation [25]. In mutation process, mutation points are given. In this example, three mutation points are determined shown in **Figure 10**. After mutation process, mutation points will be changed. If the mutation rate is high, the main

Genetic Optimized Fuzzy Clustering Flow Chart
1: Initialize the population randomly for P individuals
2: $P(i) = K \ x \ d$ prototype gray scaled matrix coded as binary
3: while($D_{ij} < \delta$)
3: Calculate fitness function $j(M) = \sum_{j=1}^{N} \left(\sum_{i=1}^{K} D_{ij}^{1/(1-m)} \right)^{1-m}$
3: Use a selection operator (roulette selection) to choose parental member
4: Two point crossover and bitwise mutation are done
5: Determine the next generation by using the highest fitness
6: end
Note: δ is the given threshold value.

Figure 8. Flow chart of the genetic clustering.

2-Point Crossover

C1   0 0 1 0 1 1 1 0 1 0 0 1 1 0 0

C2   1 1 1 0 1 0 0 1 1 1 0 0 1 0 0

New C1   0 0 1 0 1 0 0 1 1 0 0 1 1 0 0

New C2   1 1 1 0 1 1 1 0 1 1 0 0 1 0 0

Figure 9. Two-point crossover process.

Mutation

0 0 1 0 1 0 0 1 1 0 0 1 1 0 0
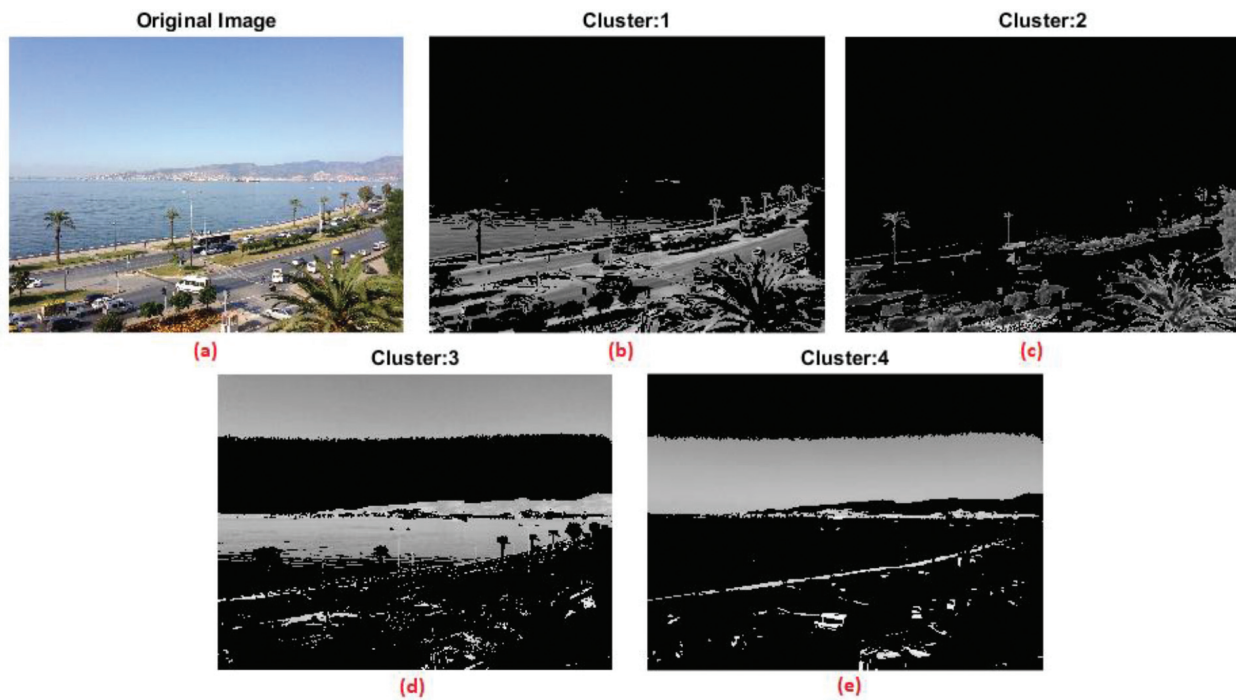
Mutation Points

0 0 1 1 1 0 0 0 1 0 0 0 1 0 0

Figure 10. Mutation process.

generation may be lost. In general terms, the rate of mutation in GA is between given 0.05 and 0.15% of the parent.

GA-based fuzzy clustering process in four clusters shown in **Figure 11**. GA is applied for the given original image shown in **Figure 11a** for four clusters. This image can be represented in

Figure 11. Genetic algorithm results. (a) Original image, (b) Cluster No:1, (c) Cluster No:2, (d) Cluster No:3, and (e) Cluster No:4.

four clusters as in **Figure 11b–e**, respectively. This algorithm can only be applied for gray-scaled images. Cluster-1 shows the roads for the given landscape, cluster-2 shows the green areas in the landscape, cluster-3 shows the sea for the given landscape and cluster-4 represents the air for the given landscape.

7. Conclusion

In this part, we discussed the fundamental partitional clustering algorithms and its applications. Partitional clustering produces a partition using *K* clusters but do not use hierarchical structures. This type of clustering process uses criterion function in a range of error such as mean square error, sum of squared error, and so on. For this reason, partitional clustering is an iterative process.

Some search techniques which include evolutionary process, provide seeking the global or local optima. Search techniques introduce more parameters than K-Means clustering process. There is no theoretical approval for selecting of the best approach. All the given methods can give the best results for different data.

The partitional clustering algorithms need high computational requirements. High computational requirements means that some complex algorithms need advanced operations. These requirements limit their applications in large data. Overcoming this situation, high speed and high-capacity memory-sized computers can be used for clustering processes.

Another method for overcoming high computational requirements is the genetic optimization. For large datasets, genetic optimization should be used with fuzzy clustering. In this technique, all individuals or pixels are genetically optimized so as to fuzzy sets.

Clusters cannot be always sharply separated in some datasets. For this type of data sets, fuzzy-based clustering overcomes this crisp clustering. Fuzzy clustering allows to find the nearest optimum cluster to assign into the clusters. This technique provides more information about the data structure.

The most important point of the search techniques of the partitional clustering is the optimum parameter selection. Parameter selection is an optimization problem. Overcoming this optimization problem, parameter selection can be done by using genetic algorithms. Genetic algorithms can be useful solution for very-large scale data sets.

In partitional clustering, determination of cluster size is important. This selection differs from the data sets to data sets. If data set includes more features to classify in a cluster, more clusters will be needed. This cluster number unfortunately is not known for many clustering problems. Generally experiences give the cluster number. Estimation of the cluster number is one of the major problems for validation.

As a result, by using partitional clustering techniques, image understanding can be done by using the given techniques. For many applications such as biomedical image understanding, robotic applications and security applications, these techniques can be useful for pre-processing of some algorithms.

Acknowledgements

Competing interests

The author declares having no conflicts of interests.

Thanks

I would like to thank my wife Hande Kutbay who has given me a lot of support in my life. I thank my daughter, Gökçe Kutbay, who makes life difficult but makes sense. Special thanks to my mother Cemile KUTBAY and my father Hür Uğur KUTBAY for being in my life. In addition, I would like to thank Assoc. Prof. Dr. Fırat Hardalaç due to academic help.

Author details

Uğurhan Kutbay

Address all correspondence to: ukutbay@gazi.edu.tr

Electrical and Electronics Engineering Department, Gazi University Engineering, Ankara, Turkey

References

[1] Liu AA, Su YT, Nie WZ, Kankanhalli M. Hierarchical clustering multi-task learning for joint human action grouping and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017;**39**(1):102-114

[2] Bouguettaya A, Yu Q, Liu X, Zhou X, Song A. Efficient agglomerative hierarchical clustering. Expert Systems with Applications. 2015;**42**(5):2785-2797

[3] Cao K, Jiao L, Liu Y, Liu H, Wang Y, Yuan H. Ultra-high capacity lithium-ion batteries with hierarchical CoO nanowire clusters as binder free electrodes. Advanced Functional Materials. 2015;**25**(7):1082-1089

[4] Nanda SJ, Panda G. A survey on nature inspired metaheuristic algorithms for partitional clustering. Swarm and Evolutionary Computation. 2014;**16**:1-18

[5] Menendez HD, Barrero DF, Camacho D. A genetic graph-based approach for partitional clustering. International Journal of Neural Systems. 2014;**24**(03):1430008

[6] Pandeeswari N, Kumar G. Anomaly detection system in cloud environment using fuzzy clustering based ANN. Mobile Networks and Applications; **21**(3):494-505

[7] Xu R, Wunsch D. Clustering 2010. Vol. 10. John Wiley & Sons; 2008. pp. 63-110

[8] Mac-Queen J. Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability; Vol. 1: Statistics. California, USA; 1967. pp. 281-297

[9] Kutbay U, Ural AB, Hardalaç F. Underground electrical profile clustering using K-MEANS algorithm. In: 2015 23th IEEE Signal Processing and Communications Applications Conference (SIU); 2015. pp. 561-564

[10] Zadeh L. Fuzzy sets. Information and Control. 1965;**8**:338-353

[11] Bezdek J. Cluster validity with fuzzy sets. Journal of Cybernetics. 1974;**3**(3):58-72

[12] Abdulshahed AM, Longstaff AP, Fletcher S, Myers A. Thermal error modelling of machine tools based on ANFIS with fuzzy c-means clustering using a thermal imaging camera. Applied Mathematical Modelling. 2015;**39**(7):1837-1852

[13] Ji Z, Liu J, Cao G, Sun Q, Chen Q. Robust spatially constrained fuzzy c-means algorithm for brain MR image segmentation. Pattern Recognition. 2014;**47**(7):2454-2466

[14] Qiu C, Xiao J, Yu L, Han L, Iqbal MN. A modified interval type-2 fuzzy C-means algorithm with application in MR image segmentation. Pattern Recognition Letters. 2013;**34**(12): 1329-1338

[15] Kannan SR, Ramathilagam S, Devi R, Hines E. Strong fuzzy c-means in medical image data analysis. Journal of Systems and Software. 2012;**85**(11):2425-2438

[16] Ji Z, Xia Y, Chen Q, Sun Q, Xia D, Feng DD. Fuzzy c-means clustering with weighted image patch for image segmentation. Applied Soft Computing. 2012;**12**(6):1659-1667

[17] Kutbay U, Hardalaç F. Development of a multiprobe electrical resistivity tomography prototype system and robust underground clustering. Expert Systems. 2017;**34**(3):e12206

[18] Holland JH. Adaptation in Natural and Artificial Systems. An Introductory Analysis with Application to Biology, Control, and Artificial Intelligence. Ann Arbor, MI: University of Michigan Press; 1975

[19] Bandyopadhyay S, Maulik U. Genetic clustering for automatic evolution of clusters and application to image classification. Pattern Recognition. 2002;**35**(6):1197-1208

[20] Hall LO, Ozyurt IB, Bezdek JC. Clustering with a genetically optimized approach. IEEE Transactions on Evolutionary Computation. 1999;**3**(2):103-112

[21] De Jong KA, Spears WM. A formal analysis of the role of multi-point crossover in genetic algorithms. Annals of Mathematics and Artificial Intelligence. 1992;**5**(1):1-26

[22] Whitley D. An executable model of a simple genetic algorithm. Foundations of Genetic Algorithms. 2014;**2**(1519):45-62

[23] Yu F, Xu X. A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved BP neural network. Applied Energy. 2014;**134**:102-113

[24] Umbarkar AJ, Sheth PD. Crossover operators in genetic algorithms: A review. ICTACT Journal on Soft Computing. 2015;**6**(1):1083-1092

[25] Larranaga P, Kuijpers CM, Murga RH, Inza I, Dizdarevic S. Genetic algorithms for the travelling salesman problem: A review of representations and operators. Artificial Intelligence Review. 1999;**13**(2):129-170