

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



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



Introductory Chapter: Clustering with Nature-Inspired Optimization Algorithms

Pakize Erdogmus and Fatih Kayaalp

1. Introduction

Humanity has been inspired from nature along its evolution, since ancient times. Each lively being has its own rules and magnificent knowledge. This capability of all lively beings gives inspiration to the human being in order to find solutions to the problems that he/she faces.

Most of the engineering designs have been inspired by nature. With the design of high-speed trains, the problem was “boom effect,” created by the trains, when entering the tunnel. This noise was because of the air pressure created on the front side of the train. This problem was solved with an excellent nature design, with kingfisher beak [1].

For more than half a century, algorithms have also been using inspirations from nature for computing and solving the problems related to computer science. The first optimization algorithm mimicking nature was genetic algorithm (GA). Genetic algorithm used the selection, mutation and crossover, finding the diverse solutions to complex problems [2]. Today, we have very powerful algorithms inspired by nature to optimize the problems.

Particle swarm optimization (PSO) is another population-based algorithm inspired by nature. Improved by James Kennedy and Russell C. Eberhart, PSO simulates the bird flocking and fish schooling foraging behaviors for the solution of a continuous optimization problem [3].

Biogeography-based optimization (BBO) algorithm is an evolutionary algorithm that simulates the formation of the biogeographies. BBO, improved by Simon [4], simulates the habitants’ immigration or emigration behaviors according to the suitability of the habitat for them.

Gray wolf optimizer (GWO) is also a nature-inspired population-based optimization algorithm originally proposed for the solution of the continuous optimization problems. GWO simulates the hunting behaviors of the gray wolves [5].

Optimization is a kind of programming, solving several problems including function minimization, clustering and feature selection. Clustering is an unsupervised machine learning method that groups the entities with a given number of categories according to their similarities. It is certain that clustering must maximize the similarities of the objects inside the same groups and also maximize the dissimilarity among the other groups’ objects. Clustering can be defined as an optimization problem with this perspective. A classical example of clustering is given in **Figure 1**.

Clustering is a very common technique used for data analysis especially for the applications of summarization, abstracting the data and segmentation [6].

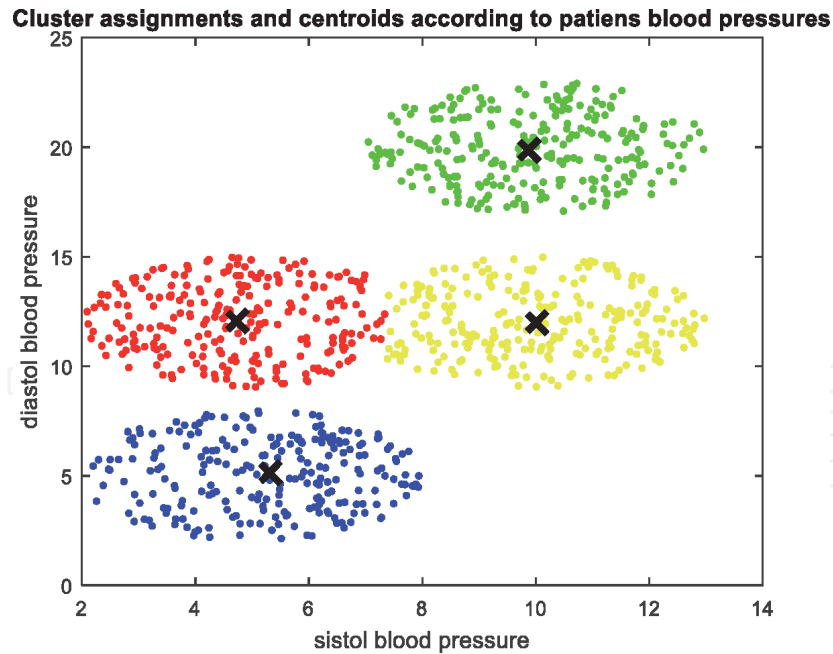


Figure 1.
Patient clusters according to their systole and diastole blood pressures.

A very common application of clustering is data analysis [7]. Cluster centroids give brief information for the attributes of each cluster. This knowledge is used for information discovery and general classification. Another application of clustering is collaborative filtering [8]. The users, grouped in the same cluster, are accepted similar likes and dislikes. Data and image segmentation are another application of clustering [9].

Today, clustering is commonly used for biological data [10], medical data [11], social network [12, 13] and wireless sensor network data [14] and big data [15] for different kinds of applications stated above.

In this chapter, the reader will learn how he/she can apply optimization algorithms for clustering problems. In the next section, the clustering is defined as an optimization problem. Nature-inspired optimization algorithms, genetic algorithm, particle swarm optimization algorithm, biogeography-based optimization algorithm and gray wolf optimization algorithm have been explained in Section 3. Clustering with nature-inspired algorithms has been studied for a very basic and popular dataset given in Section 4. And the results have been submitted in Section 5.

2. Clustering as an optimization problem

Clustering is grouping the data into the clusters according to their similarities. Similarity is defined mathematically with a measure. The more the attributes of two data are near to each other, the less distance is between data. Namely, distance is inversely proportional to similarity. Different distance measures have been defined for clustering. Euclidean, Manhattan, Mahalanobis and Minkowski are the most [16, 17] used distance metrics. The most popular metric for continuous features is the Euclidean distance [18]. Euclidean distance is used while clusters are compact and the dimension of the data is low. For large dimension, Minkowski distance is preferred.

In this chapter, the objective function is defined based on Euclidean distance metric for comparison. Let us assume the two data points in D dimension space, X and Y . The distance between X and Y is calculated with Euclid and p th order of Minkowski distance, as given in Eqs. (1) and (2), respectively.

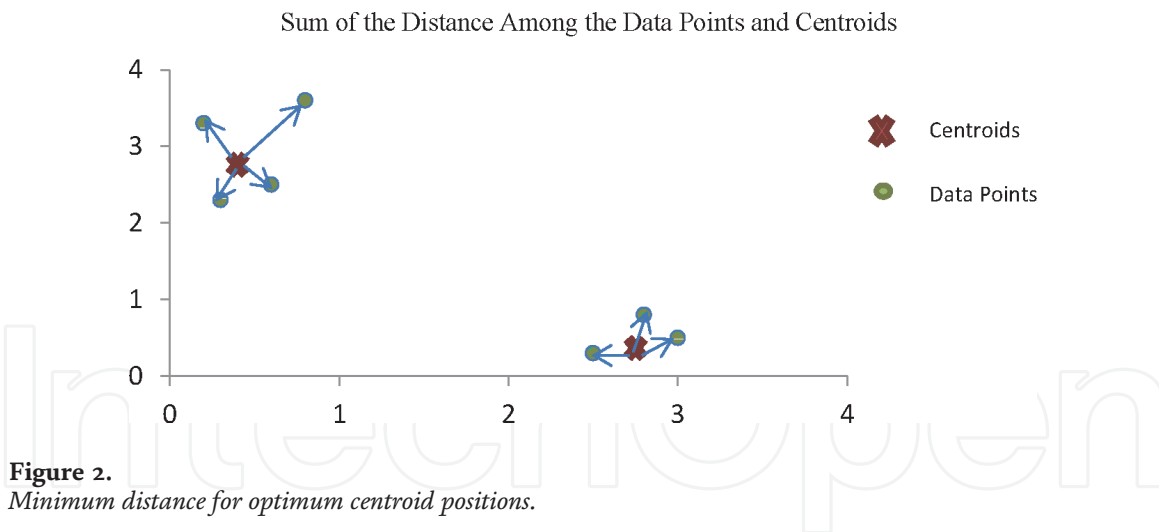


Figure 2.
 Minimum distance for optimum centroid positions.

$$De = \sqrt{\sum_{i=1}^D (X_i - Y_i)^2} \quad (1)$$

$$Dm = \sqrt[p]{\sum_{i=1}^D |X_i - Y_i|^p} \quad (2)$$

The object of clustering is to assign data to the clusters that minimize the sum of the distances from the data to centroids of the clusters. So the objective function with Euclid distance is defined as given in Eq. (3).

$$F_{obj} = \sqrt{\sum_{j=1}^K \sum_{k=1}^D (X_{ik} - C_{jk})^2} \text{ for } \forall X_i \in C_j \quad (3)$$

where N presents the number of data; K presents the number of fully separated clusters; D presents the number of dimension of data; X_{ik} presents the i th data k th feature; C_{jk} 1, 2, ..., K presents the center of the cluster j of k th feature.

The positions of centroids are independent variables. So if applied data dimension is D and the number of cluster is K, the number of independent variables for objective function is KXD. Namely, K centroid positions with D dimension are the independent variables of objective function. The objective is to find centroid positions that minimize the distance. The calculation of the objective function is shown schematically in **Figure 2**.

3. Nature-inspired optimization algorithms

In this chapter, some of the most cited and successful algorithms have been selected for comparing the clustering performances. Genetic algorithm, particle swarm optimization algorithm, biogeography-based optimization algorithm and gray wolf optimization algorithm have been selected. These algorithms have common features. All of them run a group of solutions.

Each solution is called as individual, particle, island and wolf, respectively. In this chapter, the number of solutions is given as S. Each solution has independent variables. So, the number of independent variables is called V, which is equal to KXD for clustering problem. The clustering problem is handled as an unconstrained optimization problem in this chapter as given in Eq. (4).

$$F_{obj} = f(C_1, C_2, \dots, C_V) \quad (4)$$

Algorithm	Solution	Group of solutions	Parameters
Genetic algorithm	Chromosome	Population	Crossover rate
	Individual		Mutation rate
Particle swarm optimization	Particle	Swarm	Inertia weight/constriction factor Social and cognitive parameters
Gray wolf optimization	Wolf	Group	A: linearly decreased from 2 to 0 C: random number between 0 and 2
Biogeography-based optimization	Habitat	Ecosystem	Migration parameters Mutation rate

Table 1.

Population-based optimization algorithms and their naming for common terms of optimization.

In the problem, the initial cluster centroids as independent variables are assigned randomly between the lower and upper values of data. After the independent variables are created randomly, the objective function value is calculated as given in Eqs. (3) and (4). Attaining for S initial solution, V random cluster centroids for each S solution are assigned. Solutions are called F_{obj1} , F_{obj2} , ..., F_{objS} . The optimization algorithm starts with these initial solutions and evaluates and improves the solutions, until the stopping condition is true. From one iteration to the other, the algorithm converges to the best solution.

Before the algorithms are explained in detail, the general properties of population (swarm)-based optimization algorithms and specific namings are listed in **Table 1**.

3.1 Genetic algorithm

Genetic algorithm is one of the most studied and powerful optimization algorithms, used for the solution of both combinatorial and continuous optimization problems. The main idea behind GA is “survival of the fittest.” So, the algorithm is based on the evolution of the individuals from one generation to the next.

After the optimization problem is modeled and its independent variables, constraints and objective function are specified, genetic algorithm parameters are adjusted for the problem. After the algorithm starts with an initial population, fitness value of each individual in the population is calculated. The selection process for the next generation is realized with some selection methods in such a way that best individuals have more chance than the worse ones. Tournament selection, roulette wheel selection and rank selection are some of the selection methods [2].

After selection of the parents, crossover is applied for the parents. In GA, in reverse to the real evolution, the number of population is constant, the number of child is selected as two, and the best individuals are copied like genetic cloning. Crossover operation is applied with a crossover rate. Zero crossover rate means the children will be the copy of their parents, one crossover rate means the children will be completely different from their parents. After crossover, mutation is applied with a very low mutation rate. Mutation is the permanent changes in genes, in order not to get trapped in local minimum. After the new generation is attained, the fittest ones are selected among the latest population. And algorithm stops after a number of generations. Stopping condition is generally selected as maximum number of generation.

The pseudocode of GA is given in **Table 2**.


```

    Generation = 1
    Specify max_generation value
    Generate S initial solution
    While Generation < max_generation
        Evaluate Fitness function values
        Select best solutions for the next generation
        Apply crossover for selected individuals
        Apply mutation for selected individuals
        New Population = selected individuals
        Generation = Generation + 1;
    end
    
```

Table 2.
 The pseudocode of GA.

3.2 Particle swarm optimization

PSO is another most studied optimization algorithm, used for the solution of continuous optimization problems introduced by Eberhart and Kennedy [19]. The bird flocking or fish schooling moves in a multidimensional space in such a way that they find the food in a shortest path. The main idea behind the PSO is the behavior of the particles in a swarm. Each particle has a position in a multidimensional space, and they exchange information among them. The particles move in a space using social and cognitive information. When the algorithm stops, the best position has been found.

The algorithm starts after initial positions and initial velocities of particles have been assigned. The dimension size of the particle position in PSO is the number of independent variables. Fitness value of each particle in the swarm has been calculated. The particles update their velocities according to velocity formula. Although two different velocity formulas have been defined, there are two main parameters in both formulas, representing the social and cognitive behaviors of the particles. In swarm, particles update their velocities according to both the best position in the swarm and to their best. In this way, from one iteration to the other, PSO converges the optimum solution of the problems. PSO is the fast convergent optimization algorithm and requires less memory and there are a few parameters to adapt. In the first velocity formula, there was no inertia weight [19]. Inertia weight is introduced by Shi and Eberhart [20]. Inertia weight balances the algorithm's local and global search ability. Inertia weight specifies the percentage of contribution of previous velocity to its current velocity. The velocity and position formulas for PSO are given in Eqs. (5) and (6), respectively.

$$v_i^{k+1} = wv_i^k + c_1 \text{rand}(p_{best_i} - x_i^k) + c_2 \text{rand}(g_{best} - x_i^k) \quad (5)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (6)$$

where w presents the inertia weight, v_i^k presents the velocity of ith particle for kth iteration, x_i^k presents the position of ith particle for kth iteration, p_{best_i} presents the local best solution of ith particle, g_{best} presents the global best solution, $\text{rand}()$ presents uniform random number, and c_1 and c_2 present the cognitive and social parameters.

Constriction factor (K) is used by Clerc [21]. Constriction factor assures the convergence of the PSO. The velocity and position formulas with constriction factor for PSO are given in Eqs. (7)–(9), respectively.

$$v_i^{k+1} = K(v_i^k + \varphi_1 \text{rand}(p_{best_i}^k - x_i^k) + \varphi_2 \text{rand}(g_{best} - x_i^k)) \quad (7)$$

$$K = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \varphi = \varphi_1 + \varphi_2 \varphi > 4 \quad (8)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (9)$$

where φ_1 and φ_2 are individual and social parameters. The pseudocode of PSO is given in **Table 3**.

3.3 Biogeography-based optimization

BBO applies biogeography mathematical foundations to solve the optimization problems. Biogeography observes the distribution of species in geographic space and tries to find the reason of the biodiversities in geography. Species migrates from one habitat to the other, trying to find the most suitable habitat. So if this biogeographic movement is simulated well, it can be applicable to solve an optimization problem. Geographical areas that are suitable for biological species are said to have a high habitat suitability index (HSI) [4]. The features, such as land area, temperature and rainfall show the suitability of the biogeography and called as suitability index variables (SIVs) independent variables of the optimization problem and HSI represents the fitness function. Species living in a geography that has high HSI emigrates to nearby habitats, which has low species, since this biogeography is already nearly saturated. BBO has been used for clustering in some studies [22, 23]. As seen in **Table 3**, since BBO algorithm uses three loops, BBO runs slower than the other algorithms like PSO and GWO. So some strategies have been used in the studies that make BBO run faster. The pseudocode of BBO is given in **Table 4**.

3.4 Gray wolf optimizer

Gray wolf optimizer, a population-based, nature-inspired algorithm, simulates the hunting behaviors of gray wolves [5]. Gray wolves live in groups, and there is a hierarchy among them. Their hunting strategy has three steps: encircling the prey, circling the prey and hunting the prey. This process is adapted for the optimization problem solution. The wolves move in d-dimensional space in order to search their prey. The position of the wolves presents d the independent variables. After they find the prey, they encircle their preys and lastly they hunt. Encircling behavior presents the converging of the solution and hunting presents the optimum point. The algorithms start with the creation of the initial positions of the wolves. The positions are evaluated with the fitness function. Since there is no knowledge about the position of the prey in problem, the best three solutions are selected, in order to

```

Create P initial particle position
Do
  For I = 1:P
    Evaluate Fitness function values
    If fitness(Pi) < Pbest(I)
      Pbest(I) = Pi
    end
    If fitness(Pi) < Gbest
      Gbest = Pi
    end
  end
Until stopping condition is true

```

Table 3.
The pseudocode of PSO.

```
Initialize the SIVs of N habitat
Calculate HSI values of each habitat
Sort them and find best HSI
for i = 1 to maximum_iteration
    for i = 1:N
        for k = 1:dim
            CandidateNewHabitat = Habitat
                Select Source Habitat
                Apply migration with a probability
                Apply mutation with a probability
        end
    end
    Calculate HSI values for new habitat
    Sort CandidateNew Habitat
    Create NewHabitat from Habitat bests and CandidateNewHabitat
    Update Best Solution Ever Found
End
```

Table 4.
The pseudocode of BBO.

```
Initialize the Gray Wolf Population
Initialize parameter A,a,C
Calculate each wolf fitness value
Specify first,second and third best solutions
while (t < max_iteration)
    for each wolf
        Update the position
    end
    Update a,A,C
    Update fitness of each wolf
    Update first, second and third best solutions
    t = t + 1;
end
```

Table 5.
The pseudocode of GWO.

update the next positions of the wolves. Instead of saving only one global best solution in memory, GWO saves three best solutions. This property makes the algorithm powerful for global best finding. GWO is applied successfully in feature selection [24], training multilayer perceptrons [25] and clustering [26–28].

The pseudocode of GWO is given in **Table 5**.

4. Clustering performances of the algorithms

As stated in another chapter, the object of clustering is to assign data to the clusters that minimize the sum of the distances from the data to centroids of the clusters. So the **objective function** value is accepted evolution metric for clustering. In **Tables 8** and **9**, F_{obj} column is given for the other algorithms' clustering performance comparison. In this section, the clustering performances of the algorithms have been compared for IRIS dataset. The parameters, used in the simulation, have been given in **Table 6**.

The benchmark dataset is quite well-known as IRIS dataset [29]. The dataset has four attributes and three class as given in **Table 7**.

All simulations have been implemented on a personal computer with Intel Core Duo 3.0 GHz and 8 GB RAM. Each algorithm has been simulated 30 times and

Parameters	PSO	GA	GWO	BBO
Population size	5–30	5–30	5–30	5–30
Maximum iteration	100–500	100–500	100–500	100–500
Crossover rate	—	0.8	—	—
Mutation rate	—	0.01	—	—
Self-adjustment rate	1.49	—	—	—
Social adjustment rate	1.49	—	—	—
Inertia weight	1.1	—	—	—
a	—	—	2 → 0	—
Habitat modification probability	—	—	—	1
Mutation probability	—	—	—	0.005
Elitism rate	—	—	—	0.05
Immigration probability	—	—	—	[0–1]

Table 6.
Parameters of the optimization algorithms.

Attributes	Classes
Sepal length in cm	<i>Iris setosa</i>
Sepal width in cm	<i>Iris versicolour</i>
Petal length in cm	
Petal width in cm	<i>Iris virginica</i>

Table 7.
The attributes and classes of the IRIS dataset.

results have been saved. The average, best and worst clustering performances have been calculated from 30 runs. As it has been seen, population size and the maximum iteration number are two important parameters, in order to get best solutions in the nature-inspired optimization algorithms. So in order to get optimum values, two parameters must be selected in such a way that both solution time and optimum value must be optimized. With this aim, firstly population size is selected as constant and the number of maximum iteration is changed as 100, 200, 300, 400 and 500. But only the results for iteration number 100, 200 and 300 have been shown in **Table 8**, so that the rows of the table aren't too many.

Secondly, population size is changed as 10, 20 and 30, while maximum iteration number is constant and equal to 200. The results have been shown in **Table 9**.

As it has been seen in **Table 8**, the minimum objective value for iteration number = 100 and the population size = 5 belongs to GWO. PSO is the second best algorithm for clustering. These minimum values found with GWO and PSO are not far from the minimum distance found with k-means. But the average values are quite far from the minimum objective value. So it can be said that both the population size and iteration number are not enough for finding near optimum values for clustering problems [30]. So the algorithms are not stable for these parameters. Average objective values for iteration numbers have been shown in **Figure 3**.

As it has been seen in **Figure 3**, PSO and GWO are fast convergent algorithms. But GA and BBO are also showing similar characteristics, since they have a lot of parameters like mutation rate.

Algorithm	Iteration number		Time (s)	F _{obj}
K-means	100	Average	0,014958	85,24,339
		Min	0,002778	78,85,144
		Max	0,195,176	142,7541
GA		Average	12,675	890,8601
		Min	1,112,884	182,4538
		Max	1,139,856	2071,767
PSO		Average	1,373,846	268,877
		Min	1,338,447	97,33,318
		Max	1,335,599	681,3707
BBO		Average	2,637,282	749,6682
		Min	1,331,544	184,2395
		Max	2,717,847	2044,928
GWO		Average	1,365,047	243,7228
		Min	1,317,441	83,71,005
		Max	1,440,723	692,7803
GA	200	Average	2,693,518	771,0638
		Min	3,119,822	204,5146
		Max	2,432,806	2390,792
PSO		Average	343,235	190,5667
		Min	3,455,119	80,04687
		Max	1,558,096	681,3707
BBO		Average	6,591,394	769,89
		Min	4,468,819	125,669
		Max	5,675,291	1717,338
GWO		Average	3,352,038	204,6926
		Min	3,352,038	81,44,311
		Max	6,069,114	774,6732
GA	300	Average	3,190,949	876,8533
		Min	2,670,085	290,7814
		Max	0,584,012	2148,225
PSO		Average	3,582,182	301,8806
		Min	4,470,383	80,00451
		Max	111,881	862,6507
BBO		Average	7,955,762	674,7325
		Min	8,167,651	198,4093
		Max	8,294,291	1412,369
GWO		Average	3,901,675	169,6811
		Min	384,877	79,77,414
		Max	3,964,712	681,3854

Table 8.
 IRIS clustering results of the algorithms for population size = 5.

Algorithm	Pop size		Time (s)	F _{obj}
K-means	10	Average	0,014958	85,24,339
		Min	0,002778	78,85,144
		Max	0,195,176	142,7541
GA		Average	5,298,525	316,0649
		Min	5,094,453	99,18,655
		Max	5,589,196	931,0285
PSO		Average	552,843	126,6415
		Min	5,541,587	78,86,165
		Max	5,511,388	176,8169
BBO		Average	11,19,563	295,1141
		Min	10,73,245	99,54,019
		Max	13,60,106	837,8205
GWO		Average	6,707,655	141,2761
		Min	5,483,527	79,59,426
		Max	908,555	226,1199
GA	20	Average	11,11,173	124,9446
		Min	10,03661	78,8596
		Max	13,36,941	164,6155
PSO		Average	8,826,531	123,2611
		Min	5,475,998	78,85,145
		Max	13,67,359	152,348
BBO		Average	21,39,521	162,8367
		Min	21,10,746	79,57,156
		Max	22,08803	227,7017
GWO		Average	10,61,837	136,0126
		Min	10,43,832	78,90,892
		Max	11,68,964	237,9805
GA	30	Average	17,91,835	114,2972
		Min	14,41,476	78,85,246
		Max	23,24,536	152,3933
PSO		Average	13,68,435	105,8372
		Min	7,912,835	78,85,144
		Max	18,75,625	152,348
BBO		Average	37,57,738	106,0112
		Min	32,82,001	78,85,754
		Max	50,14,788	152,46
GWO		Average	17,19,025	124,1055
		Min	16,05447	78,988
		Max	19,14,221	153,6034

Table 9.
IRIS clustering results of the algorithms for iteration number = 200.

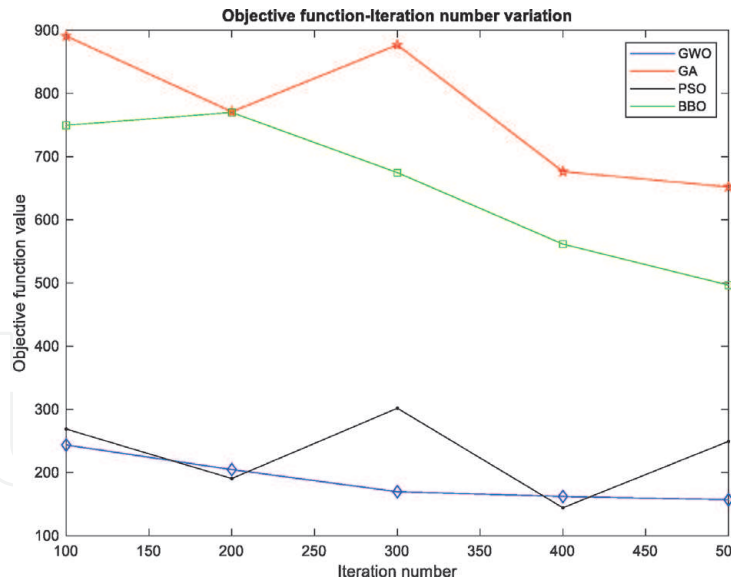


Figure 3.
 Average objective values for iteration numbers = 100, 200, 300, 400 and 500.

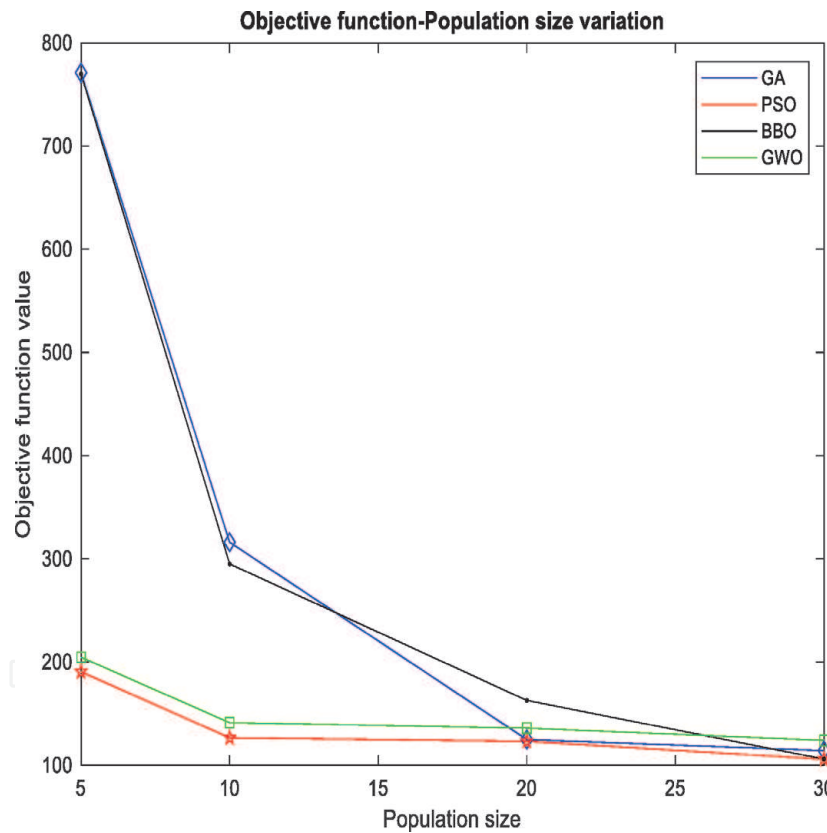


Figure 4.
 Average objective function value for population size = 5, 10, 20 and 30.

PSO is the best algorithm for clustering the data with minimum distance from centroid to each data for iteration number = 200. GWO is the second best algorithm for data clustering.

GWO is the best algorithm for clustering the data with minimum distance from centroid to each data for iteration number = 300 and PSO is the second. GWO and PSO are more stable than BBO and GA.

But it has been seen that this population size (population size = 5) is not enough for the algorithms' convergence to the minimum distance for clustering.

As it has been seen in **Table 9**, the best stable values belong to PSO and GWO. But four of the algorithms are working well under the conditions population

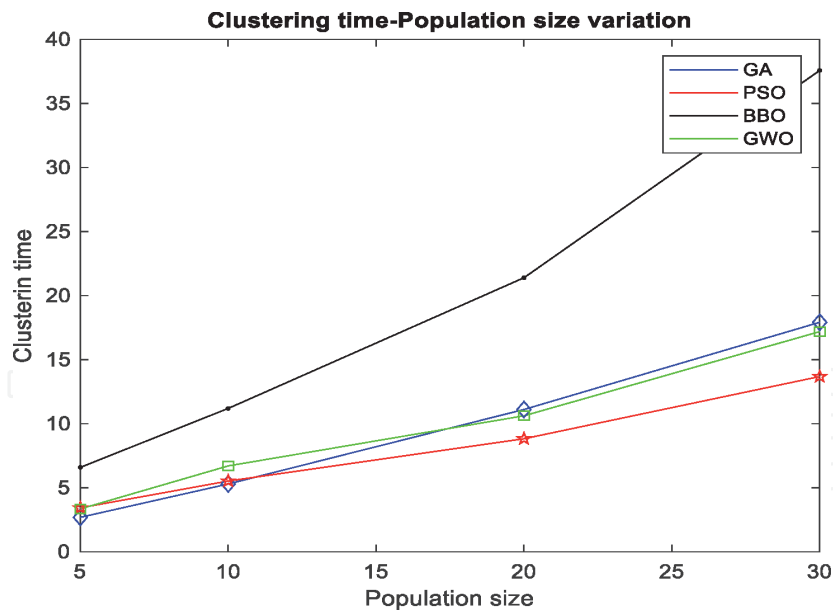


Figure 5.
Average clustering time values for population size = 5, 10, 20 and 30.

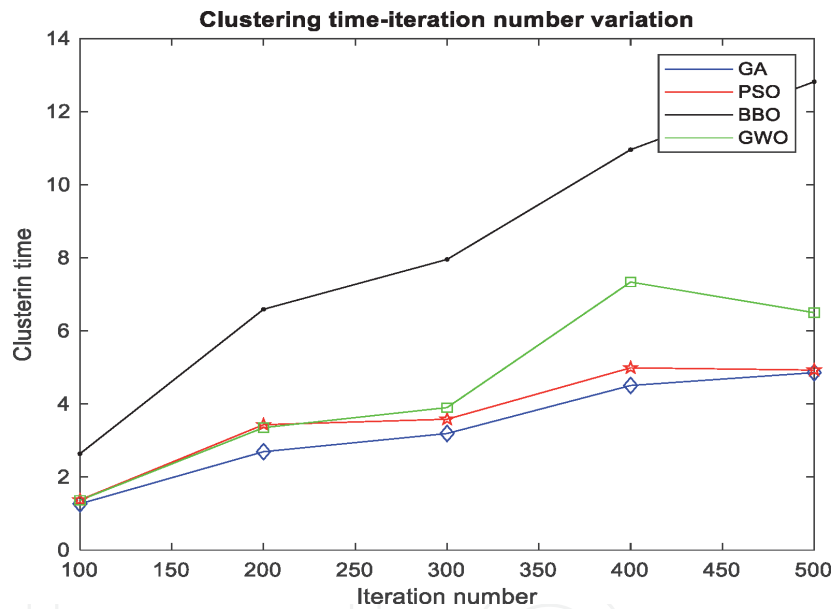


Figure 6.
Average clustering time values for iteration number = 100–500.

size = 30 and iteration number = 200. But clustering time is increasing with the number of population size. Clustering time and objective function value for population size = 5, 10, 20 and 30 have been shown in **Figures 4** and **5**, respectively.

As it has been seen in **Figure 4**, PSO and GWO can produce near optimal solutions for small population size.

However, BBO and GA require many people to effectively operate their mechanisms, such as crossing and mutation. GA and BBO catch the performances of the PSO and BBO after the population size is more than 20.

As it has been seen in **Figure 5**, BBO clustering time is highly increasing with the population size. Solution time for PSO, GWO and GA is changing less, while the population size increase.

Lastly, clustering time variation with iteration number has been shown in **Figure 6**. As it has been seen, GA and PSO clustering time are robust than BBO and GWO, depending on the number of iterations.

As an example, GWO convergence curves for 30 runs have been shown in **Figure 7**.

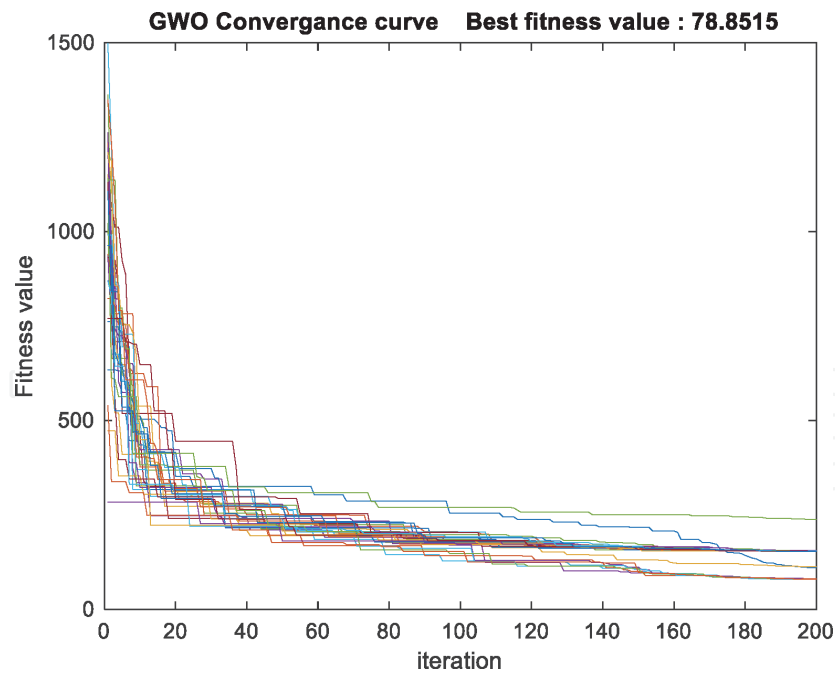


Figure 7.
GWO convergence curves for clustering IRIS data.

5. Results


Clustering is one of the unsupervised machine learning methods grouping data to the clusters. In this study, four well-known swarm-based, nature-inspired optimization algorithms have been used for clustering. In order to measure the clustering performance of the algorithms, sum of the distance values have been used. Clustering performance of the algorithms on IRIS dataset has been tested for comparison. As it has been seen in the tables, nature-inspired algorithms' solution time is not comparable with k-means. Nature-inspired algorithms are very slow because of the swarm-based run. According to the tables, PSO and GWO are faster than BBO and GA owing to the mutation and other parameters. Both PSO and GWO have fewer parameters to adapt, and they are faster and more stable than BBO and GA. In this study, no adaptation is applied for any algorithm. So if special adaptation is applied for those algorithms, the clustering performance of the algorithms will increase.

Author details

Pakize Erdogmus* and Fatih Kayaalp
Computer Engineering Department, Engineering Faculty, Duzce University,
Duzce, Turkey

*Address all correspondence to: pakizeerdogmus@duzce.edu.tr

IntechOpen

© 2020 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Available from: <https://www.gtac.edu.au/the-kingfisher-and-the-bullet-train-in-the-news/>. Latest access
- [2] Goldberg DE, Deb K. A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms*. Vol. 1. Elsevier; 1991. pp. 69-93
- [3] Kennedy J, Eberhart R. *Particle Swarm Optimization*. 1995. pp. 1942-1948
- [4] Simon D. Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*. 2008;**12**(6): 702-713
- [5] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Advances in Engineering Software*. 2014;**69**:46-61
- [6] Aggarwal CC. *An introduction to cluster analysis*. Data Clustering. Chapman and Hall/CRC; 2018. pp. 1-28
- [7] Dubes R, Jain AK. Clustering methodologies in exploratory data analysis. In: *Advances in Computers*. Vol. 19. Elsevier; 1980. pp. 113-228
- [8] Twinkle G, Kumar D. Optimization of clustering problem using population based artificial bee colony algorithm: A review. *International Journal*. 2014;**4**(4)
- [9] Ashour AS et al. A hybrid dermoscopy images segmentation approach based on neutrosophic clustering and histogram estimation. *Applied Soft Computing*. 2018;**69**: 426-434
- [10] Milone DH et al. Clustering biological data with SOMs: On topology preservation in non-linear dimensional reduction. *Expert Systems with Applications*. 2013;**40**(9):3841-3845
- [11] Paul R, Hoque ASML. Clustering medical data to predict the likelihood of diseases. In: *2010 Fifth International Conference on Digital Information Management (ICDIM)*. IEEE. 2010
- [12] Breiger RL, Boorman SA, Arabie P. An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling. *Journal of Mathematical Psychology*. 1975;**12**(3): 328-383
- [13] Singh K, Shakya HK, Biswas B. Clustering of people in social network based on textual similarity. *Perspectives in Science*. 2016;**8**:570-573
- [14] Bandyopadhyay S, Coyle EJ. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In: *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE INFOCOM 2003 (IEEE Cat. No. 03CH37428)*, Vol. 3. IEEE. 2003
- [15] Fahad A et al. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE Transactions on Emerging Topics in Computing*. 2014; **2**(3):267-279
- [16] Greche L, Jazouli M, Es-Sbai N, Majda A, Zarghili A. Comparison between Euclidean and Manhattan distance measure for facial expressions classification. In: *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, Fez. 2017. pp. 1-4
- [17] Chouikhi H, Saad MF, Alimi AM. Improved fuzzy possibilistic C-means (IFPCM) algorithms using Minkowski distance. In: *2017 International Conference on Control, Automation and Diagnosis (ICCAD)*, Hammamet. 2017. pp. 402-405

- [18] Jain AK, Narasimha Murty M, Flynn PJ. Data clustering: A review. *ACM Computing Surveys (CSUR)*. 1999;**31**(3):264-323
- [19] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. In: *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. IEEE. 1995
- [20] Shi Y, Eberhart R. A modified particle swarm optimizer. In: *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence*. IEEE; May 1998. pp. 69-73. (Cat. No. 98TH8360)
- [21] Clerc M, Kennedy J. The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*. 2002;**6**(1): 58-73
- [22] Zhang X, Wang D, Chen H. Improved biogeography-based optimization algorithm and its application to clustering optimization and medical image segmentation. *IEEE Access*. 2019;**7**:28810-28825
- [23] Pal R, Saraswat M. Data clustering using enhanced biogeography-based optimization. In: *2017 Tenth International Conference on Contemporary Computing (IC3)*, IEEE. 2017
- [24] Emary E, Zawbaa HM, Hassanien AE. Binary grey wolf optimization approaches for feature selection. *Neurocomputing*. 2016;**172**: 371-381
- [25] Mirjalili S. How effective is the Grey wolf optimizer in training multi-layer perceptrons. *Applied Intelligence*. 2015; **43**(1):150-161
- [26] Kumar V, Chhabra JK, Kumar D. Grey wolf algorithm-based clustering technique. *Journal of Intelligent Systems*. 2017;**26**(1):153-168
- [27] Zhang S, Zhou Y. Grey wolf optimizer based on Powell local optimization method for clustering analysis. *Discrete Dynamics in Nature and Society*. 2015;**2015**(1)
- [28] Fahad M et al. Grey wolf optimization based clustering algorithm for vehicular ad-hoc networks. *Computers & Electrical Engineering*. 2018;**70**:853-870
- [29] Available from: <https://archive.ics.uci.edu/ml/datasets/Iris>
- [30] Fisher RA. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*. 1936;**7**(2):179-188