

Using Metaheuristic Algorithms to Improve k-Means Clustering: A Comparative Study

Sasan Harifi^{1*}, Madjid Khalilian¹, Javad Mohammadzadeh¹, Sadoullah Ebrahimnejad²

¹ Department of Computer Engineering, Karaj Branch, Islamic Azad University, Karaj 3149968111, Iran

² Department of Industrial Engineering, Karaj Branch, Islamic Azad University, Karaj 3149968111, Iran

Corresponding Author Email: s.harifi@kiau.ac.ir

<https://doi.org/10.18280/ria.340307>

Received: 12 April 2020

Accepted: 10 June 2020

Keywords:

k-means clustering, metaheuristic algorithms, particle swarm optimization, genetic algorithm, differential evolution algorithm

ABSTRACT

Finding the most suitable centroids for k-means clustering is one of the most important criteria for successful clustering operation. We are always looking for the best centroids. Since, clustering problem and finding best centroids are an NP-hard problems, using metaheuristic algorithms can be an appropriate tool to deal with these issues. Many authors have solved this issue with metaheuristic algorithms. Common and popular algorithms have very good solutions. But which of the metaheuristic algorithms really provides the best solution? To answer this question, in this comparative study, ten popular metaheuristic algorithms are compared. The comparisons are performed on synthetic and ten real-world datasets. To find significant differences between the results obtained by algorithms, statistical analysis is used. Comparison results are presented with suitable tables.

1. INTRODUCTION

Clustering or cluster analysis is one of the unsupervised learning categories and the process during which the samples are divided into groups whose members are similar to each other, which is called clusters [1]. The goal of clustering is to obtain clusters with very similar members and the maximum distance of each cluster than other clusters. For this purpose, one of the clustering methods is k-means clustering.

k-means method is a very simple and practical approach [2]. In fact, k-means is a heuristic method for partitioning clustering. In this method, the cluster centers are represented as cluster means. Then each data is checked and assigned to the nearest cluster center. After completing this, we can create cluster centers and then create new clusters by getting the meanings in each cluster. One of the problems in this method is that its optimality depends on the initial selection of the centers and therefore, not optimal.

Since, clustering problem is an NP-hard problem [3], using metaheuristic algorithms can be an appropriate tool to deal with these issues. Metaheuristic algorithms are defined as high-level methods that can be used as a guiding strategy for solving optimization problems and other issues. In other words, metaheuristic algorithms are advanced search strategies. These algorithms effectively search the solution space with special methods. If the solution space is large, the classical methods cannot find the solution in a reasonable time [4]. One way to overcome this problem is to use metaheuristic algorithms. The main problems of heuristic algorithms are their entrapment in local optimal points and premature convergence.

Metaheuristic algorithms are also used to solve the problems of heuristic algorithms. These algorithms significantly increase the ability to find high-quality solutions to difficult optimization problems [5]. k-means method, like other heuristic methods, has the problems mentioned above. The use of metaheuristic algorithms reduces its problems.

Many authors have used metaheuristic algorithms to solve the clustering problem. Shelokar et al. [6] used an ant colony approach for clustering. They showed that an ant colony approach had better solutions compared with genetic algorithm, simulated annealing, and tabu search. Parpinelli et al. [7] proposed an algorithm based on ant colony optimization called Ant-Miner. The Ant-Miner was used to extract classification rules from data. The algorithm was inspired by both researches on the behavior of real ant colonies and some data mining concepts as well as principles. In another study, Moh'd Alia et al. [8] employed the harmony search algorithm. They compared their method with random initialization mode. Their algorithm also responds well. Krishna and Murty [9], Murthy and Chowdhury [10], Maulik and Bandyopadhyay [11], and Cowgill et al. [12] all of them have used genetic algorithms in clustering. They claim that using the genetic algorithm improves the clustering operation.

Differential evolution algorithm is another algorithm that used widely in clustering. This algorithm works well in clustering. Das et al. [13] used the differential evolution in clustering and automatic clustering. Kwedlo [14] combined the differential evolution with k-means and compared it with the global k-means. He also demonstrated the superiority of his combined differential evolution algorithm. Paterlini and Krink [15] in their paper compared the differential evolution algorithm with the particle swarm optimization, genetic algorithm, random search algorithm, and k-means. They showed that the differential evolution algorithm is very fast in solving the clustering problem. In their other paper [16], they compared the genetic algorithm, particle swarm optimization, and differential evolution. From their experiments, it turned out that differential evolution is superior compared to genetic algorithm and particle swarm optimization both with respect to precision as well as the robustness of the results.

VanderMerwe and Engelbrecht [17] used the particle swarm optimization algorithm to find the centroids of a user-specified number of clusters. They showed that the particle

swarm optimization approaches have better convergence to lower quantization errors, and in general, larger inter-cluster distances and smaller intra-cluster distances. In another paper, Zhao et al. [18] also used particle swarm optimization for clustering and compared it with k-means algorithm. In their paper, the particle swarm optimization has better solutions compared with the k-means.

Karaboga and Ozturk [19] utilized artificial bee colony to classification benchmark datasets and then the performance compared with particle swarm optimization. Their results indicate that the artificial bee colony algorithm is successfully applied for clustering. Zhang et al. [20] also used the artificial bee colony approach for clustering and then compared with the ant colony, genetic algorithm, simulated annealing, and tabu search algorithm. They performed experiments on only three datasets. Their results indicate the superiority of the artificial bee colony algorithm.

Other algorithms have also been used by other authors to solve the clustering problem. For instance, Alami et al. [21] used the cultural algorithm, Chowdhury et al. [22] utilized invasive weed optimization algorithm, Satapathy and Naik [23] applied teaching-learning-based optimization algorithm. Hammouri and Abdullah [24] used biogeography based optimization for data clustering.

In the literature, metaheuristic algorithms used in the clustering problem, have the best performance compared with metaheuristic algorithms that have been tested with them. All authors claim that the metaheuristic algorithm used in clustering yields good results. But which one is better? Optimization operation to find the best centroid should be tested under the same conditions for different metaheuristic algorithms. Statistical analysis is the best way to find the difference between different performances. In this paper, we will answer the question of which one of the most common, state-of-the-art, and widely used metaheuristic algorithms has really the best performance in the clustering. The goal of this comparative study is to achieve the desired solutions and minimize the cost of clustering through the applying of metaheuristic methods. The main contribution of this paper is to improve the k-means by applying metaheuristic methods to achieve optimal centroids and to compare solutions by examining the three criteria of within-cluster distances, clustering error rate, and algorithm run time.

The rest of the paper is structured as follows: Section 2 describes clustering fundamentals. Section 3 includes selected algorithms for comparisons. Experimental results are presented in Section 4. Section 5 represents the statistical analysis, and in Section 6 conclusion is presented.

2. CLUSTERING FUNDAMENTALS

In partitioning-based clustering problems, two features are usually considered. These two characteristics are compactness and separation [25-27].

Compactness shows the greatest similarity between items in a group. In other words, the similarity between group items needs to be maximized. This causes the most compactness in the group.

Separation shows the differences between items of different groups. That is, items of one group have the least similarity to items of other groups.

When we are talking about the least or the most, clustering can be considered as an optimization problem. As a result,

clustering is a special case of optimization. This problem can be solved optimally by metaheuristic algorithms. Here the objective function is required. To define the objective function, one of the most convenient and appropriate criteria is within-cluster distance. Therefore, in a cluster, the total distance of each cluster member from the cluster center is equal to,

$$\sum_{x \in c_i} d(x, m_i) \quad (1)$$

where, d is the Euclidean distance, x is the sample data, m is the cluster centroid, and c is i^{th} cluster. If the above equation is considered for all clusters, the function of the cluster's center is obtained. In general,

$$\text{within cluster distance} = \sum_i \sum_{x \in c_i} d(x, m_i) \quad (2)$$

Hence, whatever the answer obtained from the above equation is minimal, this means that clustering is better done. Thus, the above equation is a cost function. This equation can be minimized by changing the centroids. The problem is mathematically expressed as follows.

Suppose there is a dataset: $X = \{x_1, x_2, \dots, x_n\}$ and the centroids are unknown: $M = \{m_1, m_2, \dots, m_n\}$. If x is in d -dimensional space, m is also in d -dimensional space. In overall, if $x_i \in \mathbb{R}^d$, so $m_j \in \mathbb{R}^d$. Therefore, the numbers of unknowns are k -center multiplication in d -dimension. In this problem, the number of clusters is predetermined. So the total objective function is as follows:

$$\text{obj func} = \sum_{j=1}^k \sum_{x \in c_j} d(x, m_j) = \sum_{i=1}^n \min_{1 \leq j \leq k} d(x_i, m_j) \quad (3)$$

In the problem presented in this paper, the Error Rate (ER) is considered as an external quality measurement. The error rate is expressed as the percentage, and the equation is as follows:

$$\text{ER} = \frac{f}{n} \times 100 \quad (4)$$

where, f is the number of misplaced objects, and n is the total number of objects within the dataset.

3. SELECTED ALGORITHMS FOR COMPARISONS

In this comparative study, ten well-known, common, and popular algorithms are chosen for comparison. These are Artificial Bee Colony (ABC) [28], Ant Colony Optimization (ACO) [29], Biogeography Based Optimization (BBO) [30], Cultural Algorithm (CA) [31], Differential Evolution (DE) [32], Genetic Algorithm (GA) [33], Harmony Search (HS) [34], Invasive Weed Optimization (IWO) [35], Particle Swarm Optimization (PSO) [36], and Teaching Learning Based Optimization (TLBO) [37]. The widespread use of the above algorithms by authors is the reason for choosing them.

In the ABC [28], agents who are called Artificial Bees are gathering together to be able to solve more difficult problems. All artificial bees are in the main hive at the beginning of the search process. In the search process, artificial bees

communicate directly with each other. Each artificial bee performs a series of local movements and will help them find a solution to their current problem. Then, with the integration of solutions, the main solution to the problem is achieved. The search process consists of sequential iterations. The first iteration ends when the first bee offers its solution to solve the main problem. In the next iterations, artificial bees are beginning to find new solutions, and this process continues. In this algorithm, the number of onlooker bees can be determined. Abandonment limit and acceleration coefficient are also parameters that have a direct impact on the performance of the algorithm.

The ACO [29] is inspired by the behavior of ants in its colony when searching for food. In the real world, the ants first randomly go to and fro to find food. They then return to the colony and mark the path with pheromone. When the other ants find this path, they abandon the roam and follow it. Then, if they reach the food, they return to the colony and mark the path again. Therefore, the path is strengthened. In this way, the best path to reach the food is determined. Although the ACO algorithm works well in finding good paths through graphs, it can be applied to other issues as well. Deviation to distance ratio and intensification factor are among the important parameters in setting up this algorithm.

Biogeography discusses the migration of animal species from one island to another, the evolution of new species and the extinction of species. Basically, in the biogeography, there is competition for survival and resources. Animals are trying to obtain resources exclusively. Because there are many resources in some areas, the animals are trying to migrate into these areas. The BBO algorithm [30] is based on a mathematical model, describing the migration of species between habitats. In this algorithm, the number of kept habitats must be determined according to the size of the population. Also, the number of new habitats and migration rates have a direct impact on the performance of the algorithm.

In the CA [31], there is a knowledge component called the belief space in addition to the population component. Hence, this algorithm can be considered a special extension of the genetic algorithm. The belief space defines the various knowledge of the population from the search space. The belief space is updated by the best member of the population after each iteration. The best member is selected through the fitness function. This fitness function is exactly like the fitness function of the genetic algorithm. In this algorithm, the acceptance ratio and the number of accepted individuals are effective in the operation of the algorithm.

In the DE [32], information about the direction and distance from the members is used to find the optimal solution. In this algorithm, to create a new generation, at first the Mutation operator and then the Crossover operator is applied. For the initial population, the uniform distribution is used to make the same distributed population in the space. At each step of the algorithm, the distance between the members is reduced to find the optimal solution. In this algorithm, determining the upper limit and the lower limit of the scaling factor is very important to determine the mutation. Determining the crossover coefficient is also important to get the desirable solutions.

In the GA [33], it randomly selects people from the current generation as parents and uses them to create children who themselves are members of the next generation. During successive generations, the answers evolve and become optimized. The genetic algorithm works by using specific

principles, similar to genetic structures and chromosome behavior among a population of individuals. The two most important parameters in the adjustment of this algorithm are crossover and mutation rate.

The HS [34] is inspired by the process of harmonizing a piece of music with a musician. During the time, musicians produce a piece of music by playing different harmonies. After playing several pieces, the musicians memorize the pieces they played. The musicians try to match harmony with improvisations in the pitch played by him. In this algorithm, the number of new harmonies must be determined. Also, the amount of memory to maintain harmonies and the pitch adjustment rate are effective in setting the algorithm.

Table 1. Parameters values of competitor algorithms

Algorithm	Parameters	Values
ABC	Colony size	20
	Number of onlooker bees	20
	Abandonment limit parameter	10
	Acceleration coefficient upper bound	1
ACO	Colony size	20
	Sample size	40
	Intensification factor	0.5
	Deviation-Distance ratio	1
BBO	Number of habitats	20
	Keep rate	0.2
	Number of kept habitats	4
	Number of new habitats	16
	Immigration rates	[0,1]
CA	Population size	20
	Acceptance ratio	0.35
	Number of accepted individuals	7
DE	Population Size	20
	Lower bound of scaling factor	0.2
	Upper bound of scaling factor	0.8
	Crossover probability	0.2
GA	Population size	20
	Crossover percentage	0.8
	Mutation percentage	0.35
HS	Harmony memory size	20
	Number of new harmonies	20
	Harmony memory consideration rate	0.2
	Pitch adjustment rate	0.1
IWO	Population size	20
	Maximum population size	20
	Minimum number of seeds	0
	Maximum number of seeds	5
	Variance reduction exponent	2
	Initial value of standard deviation	0.5
	Final value of standard deviation	0.001
PSO	Swarm size	20
	Inertia weight	1
	Inertia weight damping ratio	0.99
	Personal learning coefficient	2
	Global learning coefficient	2
TLBO	Population size	20
	Teaching factor	[1,2]

Weeds are seen in almost all farms and gardens, and they are almost always the winners, regardless of how much we tried to eradicate them. In nature, weeds are heavily grown and this severe growth is a serious threat to the useful plants. One of the important characteristics of weeds is their very high sustainability and adaptability in nature. The IWO algorithm [35] is inspired by the proliferation, survival, and versatility of weeds. In this algorithm, the goal is to find the best point of a given environment for the seeding of the seeds. To do this, the minimum and the maximum number of seeds must be determined. Also, variance and standard deviation for determining new points are other settings of the algorithm.

In the PSO [36], inspired by the movement of the flock of birds, it is assumed that the flock in the problem space is randomly looking for food. There is only one piece of food in the problem space, and none of the birds do not know the location of food. One of the best strategies can be to follow a bird that has the least distance to the food. Each particle or bird has a fitness value calculated by the fitness function. Whatever particle in the search space is closer to the target or food in the bird's model, it has better fitness. Also, each particle in this swarm is defined by the velocity and position vector in the search space. In each iteration, the new particle position is defined according to the velocity vector and position vector in the search space. In this algorithm, the amount of personal learning and the amount of global learning have a direct impact on the performance of the algorithm.

The TLBO algorithm [37] has been inspired by the learning and teaching process. In the TLBO algorithm, a mathematical model for teaching and learning is considered, which ultimately runs in two phases and can lead to optimization. In the training phase, the best member of the community is selected as a teacher and leads the average population to his/her own. This is similar to what a real teacher really does in the world. In the learning phase, people in the community (who are colleagues together) work with each other to develop their knowledge. This is similar to what is really happening with friends and classmates. In this algorithm, the teaching factor is a value that is selected randomly.

The values of all parameters for each algorithm are shown in Table 1.

4. EXPERIMENTAL RESULTS

In the experiments, three criteria have been used for evaluation. The first is the objective function, as defined in Eq. (3). The total distance of each cluster member from the cluster center is considered as the objective function. The second is the Error Rate (ER), which is considered as an external quality measurement. The ER defined in Eq. (4). The Third criterion is Run Time. To evaluate performance, a simple synthetic dataset and real-world datasets were used. The synthetic dataset has been created by ourselves. The characteristics of the syntactic dataset are shown in Table 2. Also, this syntactic dataset can be found in the supplementary data files or publicly available at <http://www.harifi.com/>.

Also, ten selected real-world datasets for experiments are Balance Scale, Breast Cancer Wisconsin, Contraceptive Method Choice, Dermatology, Glass Identification, Haberman's Survival, Iris, MAGIC Gamma Telescope, Pima Indians Diabetes, and Wine, which are available in the repository of the machine learning databases [38]. Table 3 shows the characteristics of real-world datasets.

For this reason that the experiment results be comparable, the settings of all algorithms are similar to each other. Some parameters are selected for some algorithms through manual tuning. For example, the mutation and crossover rate in the GA are tune-up to get the best solution. It should be noted that the type of crossover used in the implemented DE algorithm is binomial crossover. Also, the crossover and mutation type used in the implemented GA are arithmetic crossover and Gaussian mutation, respectively. The number of iteration in each run is considered 150. The initial population is considered 20 in all algorithms. Each algorithm was run 30 times, as mentioned, each run has 150 iterations. So the mean of 30 results was considered. The standard deviation, best solution, worst solution, error rate, and run time were also reported. The label attributes are removed in some datasets for the experiments. All evaluation experiments have been run on an Intel® Pentium® processor CPU G645 2.90 GHz with 2 GB RAM. Implementations have been run on MATLAB R2015b for coding.

Table 2. Main characteristics of the synthetic dataset

Dataset	Number of Clusters	Number of Attributes	Number of Instances
Hypothetical dataset	5	2	500 (100, 100, 100, 100, 100)

Table 3. Main characteristics of the real-world datasets

Dataset	Number of Clusters	Number of Attributes	Number of Instances
Balance Scale	3	4	625 (49, 288, 288)
Breast Cancer Wisconsin	2	9	699 (458, 241)
Contraceptive Method Choice	3	9	1473 (629, 334, 510)
Dermatology	6	34	366 (112, 61, 72, 49, 52, 20)
Glass Identification	6	9	214 (70, 17, 76, 13, 9, 29)
Haberman's Survival	2	3	306 (225, 81)
Iris	3	4	150 (50, 50, 50)
MAGIC Gamma Telescope	2	10	19020 (12332, 6688)
Pima Indians Diabetes	2	8	768 (500, 268)
Wine	3	13	178 (59, 71, 48)

4.1 Experiment on synthetic dataset

By creating a hypothetical dataset, we can better comment on the performance of the algorithms. Because we know the answer. So we create a dataset to know what the answer is. For this synthetic dataset, we consider five centers (0, 0), (3, 3), (3, -2), (-3, -4), and (-4, 1). Data around these five centers have been distributed with normal distribution or specified variance.

Table 2 shows the characteristics of this dataset. In this dataset, each cluster has 100 members. The values in Table 4 shows the best, average, worst, standard deviation, error rate, and run time of solutions over 30 independent runs. Also, this table shows that all algorithms are set correctly and they work well. Now algorithms are ready for testing with real-world datasets.

4.2 Experiments on real-world datasets

To get reliable results, the collection of very diverse datasets is considered. There are small and simple or large and complex datasets in the selected datasets for experiments.

Table 5 shows the results of 30 independent runs of different algorithms on the selected datasets. In this table, the best, average, worst and standard deviation are specified.

According to the results of Table 5, it seems at a glance the

PSO algorithm has a better solution. But until the statistical analysis is done, the definitive opinion cannot be given. At a glance, the worst solution is related to the HS algorithm. Also according to the standard deviation, the PSO algorithm has the highest stability. Table 6 shows the mean error rate obtained through 30 independent runs of different algorithms on the different datasets. Table 7 shows the total run time and each iteration run time of all algorithms.

Table 4. The sum of within-cluster distances, error rate, and run time obtained by algorithms on synthetic dataset

Dataset	Criteria	Algorithms									
		ABC	ACO	BBO	CA	DE	GA	HS	IWO	PSO	TLBO
Hypothetical dataset	Best	649.40	614.88	659.81	615.52	632.38	614.91	732.41	614.88	614.88	614.88
	Average	694.48	638.35	717.31	658.76	663.97	619.53	812.04	621.61	621.29	616.34
	Worst	757.98	788.56	846.60	781.16	699.41	746.88	855.58	745.48	745.48	630.39
	Std	27.79	50.00	51.70	45.30	16.68	24.05	30.39	26.77	26.03	3.33
	ER (%)	2.79	1.18	3.70	3.24	1.66	1.29	7.35	1.53	1.49	0.73
	RT (s)	1.65	2.27	1.05	1.09	0.81	0.95	0.89	1.03	0.81	1.39

Table 5. The sum of within-cluster distances obtained by algorithms on real-world datasets

Dataset	Criteria	Algorithms									
		ABC	ACO	BBO	CA	DE	GA	HS	IWO	PSO	TLBO
Balance Scale	Best	1430.74	1423.82	1427.05	1426.00	1429.52	1423.86	1433.83	1423.82	1423.82	1424.13
	Average	1433.93	1432.50	1437.51	1430.23	1432.36	1426.84	1447.38	1423.82	1424.68	1425.85
	Worst	1437.91	1442.88	1447.71	1437.13	1437.74	1432.95	1460.13	1423.82	1425.99	1429.10
	Std	1.66	7.07	5.48	2.98	1.98	1.99	5.84	0.00	0.92	0.94
Breast Cancer Wisconsin	Best	3029.40	3028.28	3887.02	3754.08	3028.26	3030.29	4133.34	3028.15	3028.12	3028.19
	Average	3049.41	4623.91	4497.41	4342.42	3054.51	3038.04	4779.31	3028.25	3028.21	3035.09
	Worst	3084.91	4807.41	5455.91	5039.13	3121.33	3055.35	5132.80	3028.50	3028.52	3127.11
	Std	14.68	541.35	397.59	334.44	27.95	6.29	234.95	0.08	0.09	18.38
Contraceptive Method Choice	Best	5550.40	5590.18	5859.14	5789.33	5576.74	5557.26	6048.92	5532.34	5532.29	5540.52
	Average	5610.37	5991.51	6153.22	6144.16	5616.52	5611.90	6564.32	5623.12	5534.37	5588.19
	Worst	5715.76	7023.70	6694.81	7532.66	5654.00	5807.59	6887.17	7015.05	5550.90	5713.31
	Std	37.33	399.58	214.84	360.68	21.01	63.08	177.06	343.06	3.95	40.89
Dermatology	Best	2900.87	3248.20	2682.56	2898.96	2617.89	2548.37	3107.04	2254.84	2195.35	2321.36
	Average	3000.09	3404.99	2794.18	3030.68	2687.74	2644.54	3194.28	2360.61	2240.32	2395.20
	Worst	3116.26	3552.44	2983.79	3261.09	2791.23	2728.70	3232.11	2542.23	2291.50	2513.49
	Std	56.49	76.25	66.01	80.05	43.87	47.42	32.07	65.48	25.28	47.03
Glass Identification	Best	275.15	410.25	302.41	306.20	263.78	256.40	410.80	251.84	219.79	246.12
	Average	296.34	410.25	387.60	368.85	289.09	290.03	443.35	279.02	244.82	265.52
	Worst	332.00	410.25	495.89	498.28	307.81	333.68	496.47	341.63	261.96	301.02
	Std	14.59	0.00	42.95	38.09	12.28	15.84	21.45	19.54	11.56	11.82
Haberman's Survival	Best	2567.03	2566.99	2590.49	2567.02	2566.99	2566.99	2611.55	2566.99	2566.99	2566.99
	Average	2567.91	2567.10	2770.23	2651.84	2567.04	2568.05	2671.52	2734.54	2567.10	2567.58
	Worst	2569.47	2567.83	3259.44	3425.79	2567.82	2569.18	2727.21	3379.29	2567.82	2569.15
	Std	0.68	0.29	176.36	213.77	0.16	0.72	32.55	288.63	0.29	0.80
Iris	Best	97.45	96.66	99.77	99.61	96.99	96.69	125.39	96.66	96.66	96.66
	Average	102.28	123.72	130.42	111.22	102.70	99.24	144.06	97.71	97.46	97.50
	Worst	112.20	141.68	179.69	133.24	110.79	127.74	155.66	127.67	120.72	107.21
	Std	3.81	11.18	17.28	9.41	3.84	6.10	6.82	5.66	4.39	1.94
MAGIC Gamma Telescope	Best	1.64E+06	1.93E+06	1.81E+06	1.75E+06	1.63E+06	1.64E+06	2.14E+06	2.21E+06	1.62E+06	1.62E+06
	Average	1.66E+06	1.93E+06	2.17E+06	1.99E+06	1.63E+06	1.69E+06	2.41E+06	3.83E+06	1.62E+06	1.63E+06
	Worst	1.72E+06	1.93E+06	3.02E+06	2.30E+06	1.66E+06	1.83E+06	2.62E+06	5.66E+06	1.62E+06	1.64E+06
	Std	2.50E+04	0.00	2.80E+05	1.38E+05	5569.49	5.52E+04	1.57E+05	9.08E+05	403.71	4564.62
Pima Indians Diabetes	Best	4.76E+04	4.76E+04	5.06E+04	4.94E+04	4.76E+04	4.76E+04	5.33E+04	5.62E+04	4.76E+04	4.76E+04
	Average	4.76E+04	4.82E+04	5.97E+04	5.45E+04	4.76E+04	4.77E+04	5.85E+04	6.72E+04	4.76E+04	4.76E+04
	Worst	4.78E+04	6.78E+04	7.83E+04	6.20E+04	4.77E+04	4.81E+04	6.21E+04	8.14E+04	4.76E+04	4.82E+04
	Std	55.45	3697.98	7118.01	3230.39	49.00	119.81	1733.06	6528.23	2.52	125.57
Wine	Best	1.63E+04	1.65E+04	1.63E+04	1.63E+04	1.63E+04	1.63E+04	1.67E+04	1.66E+04	1.63E+04	1.63E+04
	Average	1.63E+04	1.67E+04	1.69E+04	1.65E+04	1.63E+04	1.63E+04	1.69E+04	1.93E+04	1.63E+04	1.63E+04
	Worst	1.63E+04	1.72E+04	1.88E+04	1.71E+04	1.64E+04	1.66E+04	1.71E+04	2.32E+04	1.63E+04	1.64E+04
	Std	11.31	173.31	563.49	180.69	21.63	59.18	144.96	1940.06	0.85	19.06

Table 6. The error rate (in percentage) obtained by algorithms on real-world datasets

Dataset	Algorithms									
	ABC	ACO	BBO	CA	DE	GA	HS	IWO	PSO	TLBO
Balance Scale	21.81	21.62	21.92	22.47	22.14	22.73	20.63	20.16	21.09	22.36
Breast Cancer Wisconsin	0.83	29.30	3.79	4.60	0.80	0.82	7.89	0.72	0.72	0.72
Contraceptive Method Choice	3.02	4.22	4.37	4.37	2.97	3.02	5.14	4.24	3.56	3.21
Dermatology	10.02	11.26	8.01	8.48	9.30	8.98	9.28	12.29	7.78	8.13
Glass Identification	39.39	64.49	36.68	43.99	35.90	36.79	47.01	39.84	34.70	36.73
Haberman's Survival	21.50	22.16	14.76	18.46	22.42	19.60	17.03	19.42	22.03	21.09
Iris	7.53	29.33	10.02	10.53	6.82	7.42	18.13	8.02	8.13	5.16
MAGIC Gamma Telescope	11.24	35.16	23.76	30.49	3.81	15.06	33.57	31.65	12.35	10.70
Pima Indians Diabetes	0.66	1.11	12.70	7.86	0.60	0.71	7.52	20.99	0.65	0.69
Wine	2.32	1.95	2.79	2.28	2.45	1.93	2.06	13.00	2.73	2.10

Table 7. The run time obtained by algorithms on real-world datasets. Total run time is the sum of 30 independent runs. Time is calculated in seconds

Dataset	Criteria	Algorithms									
		ABC	ACO	BBO	CA	DE	GA	HS	IWO	PSO	TLBO
Balance Scale	Total Run time	49.10	72.11	33.20	36.14	24.97	30.59	27.62	31.83	24.72	42.85
	Each Run	1.64	2.40	1.11	1.20	0.83	1.02	0.92	1.06	0.82	1.43
Breast Cancer Wisconsin	Total Run time	50.12	84.98	39.24	43.78	25.70	31.18	28.88	32.03	25.28	43.92
	Each Run	1.67	2.83	1.31	1.46	0.86	1.04	0.96	1.07	0.84	1.46
Contraceptive Method Choice	Total Run time	59.47	114.62	53.22	60.33	30.88	37.32	35.31	38.33	30.35	54.00
	Each Run	1.98	3.82	1.77	2.01	1.03	1.24	1.18	1.28	1.01	1.80
Dermatology	Total Run time	59.06	498.74	226.14	290.71	35.65	36.82	62.94	37.06	30.90	53.25
	Each Run	1.97	16.62	7.54	9.69	1.19	1.23	2.10	1.24	1.03	1.77
Glass Identification	Total Run time	48.33	183.12	82.90	103.69	25.98	29.40	34.84	30.02	24.46	42.13
	Each Run	1.61	6.10	2.76	3.46	0.87	0.98	1.16	1.00	0.82	1.40
Haberman's Survival	Total Run time	45.18	54.93	25.41	25.94	22.95	26.91	24.37	30.22	22.64	39.17
	Each Run	1.51	1.83	0.85	0.86	0.77	0.90	0.81	1.01	0.75	1.31
Iris	Total Run time	45.49	67.46	30.83	33.70	22.81	26.43	24.88	27.94	22.26	38.74
	Each Run	1.52	2.25	1.03	1.12	0.76	0.88	0.83	0.93	0.74	1.29
MAGIC Gamma Telescope	Total Run time	329.83	371.57	182.12	182.88	167.01	195.54	172.46	223.83	170.62	320.28
	Each Run	10.99	12.39	6.07	6.10	5.57	6.52	5.75	7.46	5.69	10.68
Pima Indians Diabetes	Total Run time	50.79	81.31	37.48	41.49	26.18	31.42	28.63	32.75	25.54	44.82
	Each Run	1.69	2.71	1.25	1.38	0.87	1.05	0.95	1.09	0.85	1.49
Wine	Total Run time	46.50	125.71	58.73	70.29	24.65	27.81	30.58	29.98	23.76	39.84
	Each Run	1.55	4.19	1.96	2.34	0.82	0.93	1.02	1.00	0.79	1.33

Table 8. Friedman ranking for each dataset based on the objective function solution

Dataset	Algorithms									
	ABC	ACO	BBO	CA	DE	GA	HS	IWO	PSO	TLBO
Balance Scale	7.37	6.10	8.17	5.83	6.60	4.30	9.83	1.50	1.83	3.47
Breast Cancer Wisconsin	5.17	8.50	8.27	7.67	5.20	4.47	9.17	1.73	1.43	3.40
Contraceptive Method Choice	4.57	7.60	8.37	8.17	5.00	4.30	9.50	2.07	1.47	3.97
Dermatology	7.33	10.00	5.93	7.67	4.80	4.30	8.97	2.27	1.00	2.73
Glass Identification	4.90	8.57	8.10	7.50	4.47	4.47	9.80	3.63	1.13	2.43
Haberman's Survival	5.70	2.45	9.33	7.40	3.58	5.67	8.90	5.70	2.45	3.82
Iris	5.53	7.53	8.53	7.00	5.63	4.10	9.70	2.43	1.35	3.18
MAGIC Gamma Telescope	4.30	6.43	7.87	6.90	2.97	4.60	8.87	9.93	1.33	1.80
Pima Indians Diabetes	5.07	1.83	8.30	7.53	4.03	5.13	8.40	9.63	1.97	3.10
Wine	2.97	7.63	7.57	6.37	4.07	4.00	8.40	9.80	1.00	3.20

5. STATISTICAL ANALYSIS

To find significant differences between the results obtained by algorithms, statistical analysis is used. For this purpose, Friedman and Iman-Davenport tests are employed. Table 8 shows Friedman ranking for each dataset based on the objective function solution. In this table, for most datasets, the PSO algorithm has a better solution. The IWO and ACO algorithms have the best solution on Balance Scale and Pima Indians Diabetes datasets, respectively. However, the PSO algorithm is in the next rank. The IWO and ACO algorithms

have been fluctuating. So that the ACO algorithm in Dermatology dataset is in the last rank. Also, the IWO algorithm is in the last rank in MAGIC Gamma Telescope, Pima Indians Diabetes, and Wine. The HS algorithm in the clustering is an algorithm with an inappropriate solution because in most datasets is in the last ranks. This table shows that the GA algorithm is an average algorithm that usually has acceptable solutions. Also, the TLBO algorithm showed that is a good algorithm for clustering problem.

Table 9 shows the ranking of clustering algorithms based on the results of Table 5 (within-cluster distances) using the

Friedman test. As expected, the PSO algorithm is first in the ranking, then the TLBO algorithm is located. In the next ranks, the algorithms are DE, GA, ABC, IWO, CA, ACO, BBO, and HS, respectively.

Table 10 shows the results of the Friedman and Iman-Davenport tests. In this table, there is the Chi-Square value with nine degrees of freedom, and also there is the asymptotic significance of the test (p-value) with very close to zero value. Given the close to zero value of the asymptotic significance, the hypothesis is rejected. Therefore, it can be concluded that there is a significant difference in the performance of clustering algorithms.

Table 9. Ranking of clustering algorithms based on the sum of within-cluster distances

Algorithms										
	ABC	ACO	BBO	CA	DE	GA	HS	IWOPSO	TLBO	PSO
Ranking	4.85	7.15	8.45	6.904	104.409	25.5	6.60	1.60	2.70	

Table 10. Results of Friedman’s and Iman–Davenport’s tests based on the sum of within-cluster distances

Test method	Chi-Square	Degrees of freedom (DF)	p-Value	Hypothesis
Friedman	60.2226	9	1.21E-09	Rejected
Iman–Davenport	17.1440	9	1.77E-15	Rejected

Table 11. Ranking of clustering algorithms based on the error rate

Algorithms										
	ABC	ACO	BBO	CA	DE	GA	HS	IWOPSO	TLBO	PSO
Ranking	5.05	7.60	5.85	6.85	4.20	4.65	6.60	6.40	4.00	3.80

Table 12. Results of Friedman’s and Iman–Davenport’s tests based on the error rate

Test method	Chi-Square	Degrees of freedom (DF)	p-Value	Hypothesis
Friedman	17.6605	9	0.03932	Rejected
Iman–Davenport	2.1873	9	0.03121	Rejected

Table 13. Ranking of clustering algorithms based on the run time

Algorithms										
	ABC	ACO	BBO	CA	DE	GA	HS	IWOPSO	TLBO	PSO
Ranking	8.20	10.006	20.7	4.5	1.90	4.10	3.80	5.10	1.10	7.15

Table 14. Results of Friedman’s and Iman–Davenport’s tests based on the run time

Test method	Chi-Square	Degrees of freedom (DF)	p-Value	Hypothesis
Friedman	78.4675	9	3.26E-13	Rejected
Iman–Davenport	60.9480	9	2.20E-16	Rejected

Table 11 shows the ranking of clustering algorithms based on the results of Table 6 (Error Rate) using the Friedman test. In Table 11, the best rank is related to the TLBO algorithm, then the PSO algorithm is located. Table 12 shows the results of the Friedman and Iman-Davenport tests based on the error rate. In this table, the hypothesis is also rejected according to the p-value. There is a significant difference in the error rates.

In the clustering problem, in addition to the accuracy of clustering, the run time is also very important. Previously, in Table 7, the run time of each algorithm was shown. Table 13 shows that the PSO algorithm has a very high speed in solving the clustering problem based on the Friedman ranking. The DE and HS algorithms are in the next rank, respectively. Although the TLBO algorithm has high accuracy, the run time is not good enough. Also, the GA algorithm is placed in the category of intermediate in terms of run time. Table 14 shows the results of the Friedman and Iman-Davenport tests based on the run time. There is a significant difference in the run time because the hypothesis is rejected according to the p-value.

Some of the algorithms that were compared had a very good performance, some had a very low error rate, and some had a very good run time. To make better choices the more efficient algorithm, all criteria including within-cluster distances, error rate, and run time must be compared at the same time. For this purpose, Friedman's ranking has been performed with all these criteria. The results in Table 15 show that the PSO algorithm is much better than the other algorithms. After the PSO algorithm, the DE, GA, TLBO, IWO, ABC, HS, BBO, CA, and ACO are located, respectively. Table 16 shows the results of Friedman’s and Iman–Davenport’s tests based on the collection of within-cluster distances, error rate, and run time. In this table, the hypothesis is rejected according to the p-value, so there is a significant difference.

Because a significant difference has been observed, the pairwise comparisons of algorithms are performed by the Wilcoxon test to determine whether the difference is statistically significant. The Wilcoxon test is applied as a nonparametric test. Algorithms are compared in terms of within-cluster distances value, error rate, and run time for all datasets. In this method, the confidence interval is 95% ($\alpha=0.05$). The results are shown in Table 17 shows that the PSO algorithm has been successful in most cases. In the pairwise comparisons, most of the wins are related to the PSO.

Table 15. Ranking of clustering algorithms based on the collection of within-cluster distances, error rate, and run time

Algorithms										
	ABC	ACO	BBO	CA	DE	GA	HS	IWOPSO	TLBO	PSO
Ranking	6.03	8.25	6.83	7.07	3.40	4.38	6.55	5.70	2.23	4.55

Table 16. Results of Friedman’s and Iman–Davenport’s tests based on the collection of within-cluster distances, error rate, and run time

Test method	Chi-Square	Degrees of freedom (DF)	p-Value	Hypothesis
Friedman	100.4737	9	1.26E-17	Rejected
Iman–Davenport	16.9670	9	2.20E-16	Rejected

Table 17. Pairwise comparisons of algorithms based on Wilcoxon test

	ABC	ACO	BBO	CA	DE	GA	HS	IWO	PSO	TLBO
ABC	-	2.16E-05	0.02183	0.00241	0.00475	0.04039	0.00927	0.95899	1.43E-04	4.13E-05
ACO	2.16E-05	-	0.38203	0.03683	6.34E-06	4.29E-06	0.82901	0.08590	4.33E-06	4.29E-06
BBO	0.02183	0.38203	-	0.54486	4.07E-05	1.48E-04	0.18358	0.53718	1.02E-05	0.00148
CA	0.00241	0.03683	0.54486	-	8.19E-05	3.89E-05	0.02067	0.46528	1.24E-05	2.00E-04
DE	0.00475	6.34E-06	4.07E-05	8.19E-05	-	0.30430	7.51E-05	0.06268	0.02202	0.70065
GA	0.04039	4.29E-06	1.48E-04	3.89E-05	0.30430	-	4.71E-04	0.04714	7.11E-04	0.15662
HS	0.00927	0.82901	0.18358	0.02067	7.51E-05	4.71E-04	-	0.76550	5.30E-05	0.00385
IWO	0.95899	0.08590	0.53718	0.46528	0.06268	0.04714	0.76550	-	2.47E-04	0.16310
PSO	1.43E-04	4.33E-06	1.02E-05	1.24E-05	0.02202	7.11E-04	5.30E-05	2.47E-04	-	0.00476
TLBO	4.13E-05	4.29E-06	0.00148	2.00E-04	0.70065	0.15662	0.00385	0.16310	0.00476	-

6. CONCLUSIONS

In this comparative study, the performance of popular, general, and state-of-the-art metaheuristic algorithms in the clustering problem was compared. Three criteria including performance, error rate, and run time were investigated. Algorithms were compared in equal conditions. All algorithms were first tested on the synthetic dataset and then on real-world datasets. The real-world datasets contain ten datasets with a variety of features. Statistical analysis was also used to reveal significant differences. According to the results, we can answer the question of the paper's introduction. All the algorithms compared in this paper were indeed able to improve the k-means method well. However, according to the experiments performed in this paper and examining three evaluation criteria including performance, clustering error rate, and run time, it can be concluded that the PSO algorithm has been superior to other algorithms in improving the k-means method. DE and GA are in the next ranks, respectively.

REFERENCES

[1] Nanda, S.J., Panda, G. (2014). A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary Computation*, 16: 1-18. <https://doi.org/10.1016/j.swevo.2013.11.003>

[2] Zhao, W.L., Deng, C.H., Ngo, C.W. (2018). k-means: A revisit. *Neurocomputing*, 291: 195-206. <https://doi.org/10.1016/j.neucom.2018.02.072>

[3] Welch, W.J. (1982). Algorithmic complexity: three NP-hard problems in computational statistics. *Journal of Statistical Computation and Simulation*, 15(1): 17-25. <https://doi.org/10.1080/00949658208810560>

[4] Harifi, S., Mohammadzadeh, J., Khalilian, M., Ebrahimnejad, S. (2020). Giza pyramids construction: an ancient-inspired metaheuristic algorithm for optimization. *Evolutionary Intelligence*, 1-19. <https://doi.org/10.1007/s12065-020-00451-3>

[5] Harifi, S., Khalilian, M., Mohammadzadeh, J., Ebrahimnejad, S. (2019). Emperor Penguins Colony: A new metaheuristic algorithm for optimization. *Evolutionary Intelligence*, 12(2): 211-226. <https://doi.org/10.1007/s12065-019-00212-x>

[6] Shelokar, P.S., Jayaraman, V.K., Kulkarni, B.D. (2004). An ant colony approach for clustering. *Analytica Chimica Acta*, 509(2): 187-195. <https://doi.org/10.1016/j.aca.2003.12.032>

[7] Parpinelli, R.S., Lopes, H.S., Freitas, A.A. (2002). Data mining with an ant colony optimization algorithm. *IEEE*

Transactions on Evolutionary Computation, 6(4): 321-332. <https://doi.org/10.1109/TEVC.2002.802452>

[8] Moh'd Alia, O., Al-Betar, M.A., Mandava, R., Khader, A.T. (2011). Data clustering using harmony search algorithm. In *International Conference on Swarm, Evolutionary, and Memetic Computing*, pp. 79-88. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-27242-4_10

[9] Krishna, K., Murty, M.N. (1999). Genetic K-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3): 433-439. <https://doi.org/10.1109/3477.764879>

[10] Murthy, C.A., Chowdhury, N. (1996). In search of optimal clusters using genetic algorithms. *Pattern Recognition Letters*, 17(8): 825-832. [https://doi.org/10.1016/0167-8655\(96\)00043-8](https://doi.org/10.1016/0167-8655(96)00043-8)

[11] Maulik, U., Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. *Pattern Recognition*, 33(9): 1455-1465. [https://doi.org/10.1016/S0031-3203\(99\)00137-5](https://doi.org/10.1016/S0031-3203(99)00137-5)

[12] Cowgill, M.C., Harvey, R.J., Watson, L.T. (1999). A genetic algorithm approach to cluster analysis. *Computers & Mathematics with Applications*, 37(7): 99-108. [https://doi.org/10.1016/S0898-1221\(99\)00090-5](https://doi.org/10.1016/S0898-1221(99)00090-5)

[13] Das, S., Abraham, A., Konar, A. (2007). Automatic clustering using an improved differential evolution algorithm. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 38(1): 218-237. <https://doi.org/10.1109/TSMCA.2007.909595>

[14] Kwedlo, W. (2011). A clustering method combining differential evolution with the K-means algorithm. *Pattern Recognition Letters*, 32(12): 1613-1621. <https://doi.org/10.1016/j.patrec.2011.05.010>

[15] Paterlini, S., Krink, T. (2004). High performance clustering with differential evolution. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753) (Vol. 2)*. IEEE. <https://doi.org/10.1109/CEC.2004.1331142>

[16] Paterlini, S., Krink, T. (2006). Differential evolution and particle swarm optimisation in partitional clustering. *Computational Statistics & Data Analysis*, 50(5): 1220-1247. <https://doi.org/10.1016/j.csda.2004.12.004>

[17] Van der Merwe, D.W., Engelbrecht, A.P. (2003). Data clustering using particle swarm optimization. In *The 2003 Congress on Evolutionary Computation, 2003. CEC'03*, pp. 215-220. <https://doi.org/10.1109/CEC.2003.1299577>

[18] Van der Merwe, D.W., Engelbrecht, A.P. (2003). Data clustering using particle swarm optimization. In *The 2003 Congress on Evolutionary Computation, 2003.*

- CEC'03, pp. 215-220. <https://doi.org/10.1109/CEC.2003.1299577>
- [19] Karaboga, D., Ozturk, C. (2011). A novel clustering approach: Artificial Bee Colony (ABC) algorithm. *Applied Soft Computing*, 11(1): 652-657. <https://doi.org/10.1016/j.asoc.2009.12.025>
- [20] Zhang, C., Ouyang, D., Ning, J. (2010). An artificial bee colony approach for clustering. *Expert systems with applications*, 37(7): 4761-4767. <https://doi.org/10.1016/j.eswa.2009.11.003>
- [21] Alami, J., El Imrani, A., Bourroumi, A. (2007). A multipopulation cultural algorithm using fuzzy clustering. *Applied Soft Computing*, 7(2): 506-519. <https://doi.org/10.1016/j.asoc.2006.10.010>
- [22] Chowdhury, A., Bose, S., Das, S. (2011). Automatic clustering based on invasive weed optimization algorithm. In *International Conference on Swarm, Evolutionary, and Memetic Computing*, pp. 105-112. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-27242-4_13
- [23] Satapathy, S.C., Naik, A. (2011). Data clustering based on teaching-learning-based optimization. In *International Conference on Swarm, Evolutionary, and Memetic Computing*, pp. 148-156. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-27242-4_18
- [24] Hammouri, A.I., Abdullah, S. (2014). Biogeography-Based Optimisation For Data Clustering. In *SoMeT*, pp. 951-963. <https://doi.org/10.3233/978-1-61499-434-3-951>
- [25] Halkidi, M., Batistakis, Y., Vazirgiannis, M. (2002). Clustering validity checking methods: Part II. *ACM Sigmod Record*, 31(3): 19-27. <https://doi.org/10.1145/601858.601862>
- [26] Legány, C., Juhász, S., Babos, A. (2006). Cluster validity measurement techniques. In *Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, pp. 388-393. World Scientific and Engineering Academy and Society (WSEAS) Stevens Point, Wisconsin, USA. <https://dl.acm.org/doi/abs/10.5555/1364262.1364328>
- [27] Liu, Y.C., Li, Z.M., Xiong, H., Gao, X.D., Wu, J.J. (2010). Understanding of internal clustering validation measures. In *2010 IEEE International Conference on Data Mining*, pp. 911-916. <https://doi.org/10.1109/ICDM.2010.35>
- [28] Karaboga, D., Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3): 459-471. <https://doi.org/10.1007/s10898-007-9149-x>
- [29] Dorigo, M., Birattari, M., Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4): 28-39. <https://doi.org/10.1109/MCI.2006.329691>
- [30] Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, 12(6): 702-713. <https://doi.org/10.1109/TEVC.2008.919004>
- [31] Reynolds, R.G., Peng, B. (2004). Cultural algorithms: Modeling of how cultures learn to solve problems. *16th IEEE International Conference on Tools with Artificial Intelligence*, Boca Raton, FL, USA, pp. 166-172. <https://doi.org/10.1109/ICTAI.2004.45>
- [32] Storn, R., Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4): 341-359. <https://doi.org/10.1023/A:1008202821328>
- [33] Sivanandam, S.N., Deepa, S.N. (2008). Genetic algorithms. In *Introduction to genetic algorithms*, pp. 15-37. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-73190-0_2
- [34] Geem, Z.W., Kim, J.H., Loganathan, G.V. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation*, 76(2): 60-68. <https://doi.org/10.1177/003754970107600201>
- [35] Mehrabian, A.R., Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics*, 1(4): 355-366. <https://doi.org/10.1016/j.ecoinf.2006.07.003>
- [36] Kennedy, J., Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, pp. 1942-1948. <https://doi.org/10.1109/ICNN.1995.488968>
- [37] Rao, R.V., Savsani, V.J., Vakharia, D.P. (2011). Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43(3): 303-315. <https://doi.org/10.1016/j.cad.2010.12.015>
- [38] Dua, D., Karra Taniskidou, E. (2017). UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science.