Hacettepe Journal of Mathematics and Statistics
Volume 48 (3) (2019), 859 – 882
Research Article

Improving fuzzy c-means clustering via quantum-enhanced weighted superposition attraction algorithm

Adil Baykasoğlu*†, İlker Gölcük‡ and Fehmi Burçin Özsoydan§

Abstract

Fuzzy clustering has become an important research field in pattern recognition and data analysis. As supporting unsupervised mode of learning, fuzzy clustering brings about unique opportunities to reveal structural relationships in data. Fuzzy c-means clustering is one of the widely preferred clustering algorithms in the literature. However, fuzzy c-means clustering algorithm has a major drawback that it can get trapped at some local optima. In order to overcome this shortcoming, this study employs a new generation metaheuristic algorithm. Weighted Superposition Attraction Algorithm (WSA) is a novel swarm intelligence-based method that draws inspiration from the superposition principle of physics in combination with the attracted movement of agents. Due to its high converging capability and practicality, WSA algorithm has been employed in order to enhance performance of fuzzy-c means clustering. Comprehensive experimental study has been conducted on publicly available datasets obtained from UCI machine learning repository. The results point out significant improvements over the traditional fuzzy c-means algorithm.

Keywords: Fuzzy c-means clustering, Metaheuristics, Pattern recognition, Weighted superposition attraction.

Mathematics Subject Classification (2010): 03E72, 68T20

Received : 02.01.2018 *Accepted* : 10.02.2018 *Doi* : 10.15672/HJMS.2019.655

*Dokuz Eylül University, Faculty of Engineering, Department of Industrial Engineering, zmir, Turkey, Email: adil.baykasoglu@deu.edu.tr

†Corresponding Author.

‡Dokuz Eylül University, Faculty of Engineering, Department of Industrial Engineering, zmir, Turkey, Email: ilker.golcuk@deu.edu.tr

§Dokuz Eylül University, Faculty of Engineering, Department of Industrial Engineering, zmir, Turkey, Email: burcin.ozsoydan@deu.edu.tr

1. Introduction

Clustering is a prominent task of unsupervised machine learning and is defined as a process of grouping similar objects. Generally speaking, clustering aims to discover natural groupings of data points or instances based on the similarities [21]. Objects in the same cluster are said to be more similar than those in different groups. In this regard, a certain criterion is optimized in order to distribute objects into different clusters. A plethora of clustering methods has been developed in the literature during the past quarter-century and one can see that clustering has played a critical role in different application domains of science and engineering such as data mining, machine learning, fault diagnosis, pattern recognition and fault diagnosis [18].

Clustering algorithms can be classified into two categories, namely, hard (crisp) and fuzzy clustering. Typical hard clustering algorithms are ISODATA, LVQ, k-means clustering [23]. Hard clustering algorithms are easy to implement and practical. Despite the fact that hard clustering algorithms have played a central role in many application domains, they assume that clusters are strictly separated and overlapping between clusters are not allowed. In hard clustering algorithms, membership values of an object to clusters are either zero or one. On the other hand, fuzzy clustering algorithms assign membership degrees between objects and the different groups of the dataset [29].

Fuzzy c-means (FCM), proposed by Bezdek [7], is one of the most widely used fuzzy clustering methods. In FCM method, fuzzy memberships are distributed to clusters for each pattern. Unlike the hard clustering algorithms, degree of belongingness to a cluster is expressed by membership value between [0,1]. In fact, hard clustering algorithms can be seen as a special case of the fuzzy clustering algorithm in which an object takes the value of 1 as a membership degree for a cluster and a membership of 0 for the rest of the clusters. By utilizing maximum membership degrees for each cluster, crisp cluster structure can easily be obtained. The main advantage of using FCM is that when the clusters are overlapping in nature due to the dataset characteristics, FCM is generally more successful as a result of higher level information processing [28].

In FCM clustering, as with the K-means clustering [16], number of clusters are known a priori and objects are distributed in c clusters by iteratively minimizing an optimization criterion. The major drawback of these algorithms is that results are largely dependent on the initial cluster centers [2]. Also, they often can get trapped at local optima. Recently, many extensions and improvements have been reported in order to overcome deficiencies of the traditional FCM algorithm. Pimentel and de Souza [30] proposed multivariate memberships for FCM algorithm. Here, memberships are represented in a matrix form, varying one feature to another and one cluster to another. A weighted multivariate FCM method for handling interval-valued data is proposed by Pimentel and de Souza [31]. In the study, weights of each variable differ from one cluster to another. Zhaoet al. [38] proposed optimal selection based suppressed FCM algorithm with self-tuning capability in order to improve image segmentation performance. Zhanget al. [36] proposed a genetic algorithm and gradient-based optimization strategy for interval weighted FCM algorithm. Each attribute was weighted by interval values in order to obtain a reasonable data partition. Sabzekar and Naghibzadeh [32] proposed relaxed constraints support vector machines in order to solve the problem of FCM that data points assigned to some clusters might have low membership values. In order to solve the problem of assigning data points to clusters with low confidence, noise-aware implementation of supper vector machines was proposed.

Nature inspired metaheuristic approaches have also been employed in clustering applications. Many metaheuristic-based clustering approaches have been brought into the literature including ant colony optimization [33], simulated annealing [19], particle swarm

optimization (PSO) [22], artificial bee colony algorithm [37] to name a few. Especially swarm intelligence-based metaheuristics have enjoyed a visible position in the literature. Nayak et al. [26] proposed a hybrid chemical reaction based metaheuristic within FCM algorithm and performance metrics such as rate of error, inner and inter cluster distances, and etc. are calculated. Filho et al. [15] integrated self-adaptive PSO algorithm with FCM in order to achieve better results. Belacel et al. [6] combined variable neighborhood search with FCM clustering in order to obtain better quality results. For more information regarding metaheuristic-based enhancements of FCM algorithm can be found in [25, 1]

Weighted Superposition Attraction (WSA) algorithm is a recent swarm intelligence based metaheuristic developed by Baykasolu and Akpinar [3], [4]. WSA is inspired from the natural phenomenon of superposition principle and field attraction. WSA was designed to solve unconstrained and constrained global optimization problems. Recently, Baykasolu and Özsoydan [5] implemented WSA algorithm for dynamic binary optimization problems. Özbakr and Turna [27] evaluated performance of WSA algorithm for hard clustering and the results are compared with the other state of the art methods. In this study, WSA algorithm is carried out in order to model and solve fuzzy clustering problem. The contribution of the paper is twofold:

- WSA algorithm is implemented for enhancing traditional FCM first time in the literature. Comprehensive computational analysis is provided in order to test the performance of the WSA algorithm.
- WSA algorithm is further improved with the quantum particles [11] that perform local search within an intensifying quantum cloud.

The rest of the paper is organized as follows: Brief overview of the FCM algorithm and the performance metrics are given in Section 2. In Section 3, building blocks of the WSA algorithm are introduced. In Section 4, details of the proposed method are given. Finally, a comprehensive computational study and concluding remarks are given in Section 5 and Section 6, respectively.

2. Fuzzy c-means clustering

Fuzzy c-means clustering algorithm is one of the widely used objective function-based clustering techniques. The fundamental aspect in fuzzy clustering is to determine the similarity measure, in which distances between pair of data points are calculated. In fuzzy c-means clustering, patterns are treated as vectors in the Euclidean space. Fuzzy c-means algorithm attempts to find the most representative vector, which is considered as the prototype or centroid of the cluster, and thereby the grade of memberships of data points in each cluster are obtained. Various cluster validity measures are designed in order to assess the clustering results. Compactness and separation indices are considered in calculating cluster validity. In the next section, details of computational steps are given based on [7, 8].

2.1. Fuzzy c-means algorithm. The collection of N data $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N$ is partitioned into c clusters in the fuzzy c-means clustering, where $1 < c < N$. As a result, a collection of cluster centers $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c$ and a partition matrix, $U = [u_{ik}]$, $i = 1, 2, \ldots c$ and $k = 1, 2, \ldots, N$ are constructed. The partition matrix satisfies the following conditions: $u_{ik} \in [0, 1]$, $\sum_{i=1}^{c} u_{ik} = 1 \forall k$, and $0 < \sum_{k=1}^{N} u_{ik} < N \ \forall i$. The objective function of the fuzzy c-means clustering algorithm is as follows:

$$J_m = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^m \|\mathbf{v}_i - \mathbf{y}_k\|^2 \tag{2.1}$$

where symbol $\|\cdot\|$represents a distance function, and $m\,(m > 1)$ denotes a fuzzification coefficient. The fuzzification coefficient is usually taken as $m = 2$. The fuzzification coefficient influences the shape and overlapping of the membership function. Fuzzification coefficient higher than two yields spiky membership functions, which exhibits rippling effect. This behavior is highly influential on the results of fuzzy models.

The Euclidean distance from data point $\mathbf{y}_k$ to cluster center $\mathbf{v}_i$ is expressed as:

$$\|\mathbf{v}_i - \mathbf{y}_k\|^2 = \sum_{j=1}^{n} (y_{kj} - v_{ij})^2 \tag{2.2}$$

The minimization of objective function given in Eq. 2.1 can be attained by an iterative algorithm. For that aim, the cluster centers are updated as given in Eq. 2.3.

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^m \mathbf{y}_k}{\sum_{k=1}^{N} u_{ik}^m} \tag{2.3}$$

The partition matrix is updated as given in Eq. 2.4.

$$u_{ik} = \frac{1}{\sum_{h=1}^{c} \left(\frac{\|\mathbf{v}_i - \mathbf{y}_k\|}{\|\mathbf{v}_h - \mathbf{y}_k\|} \right)^{\frac{2}{m-1}}} \tag{2.4}$$

The Eqs. 2.3-2.4 are iterated, and $\mathbf{v}_i$ and u_{ik} are updated towards the direction that minimize the objective function J_m. When the $\mathbf{v}_i$ and u_{ik} lie within the tolerance, iteration is stopped. The pseudocode of the fuzzy c-means clustering algorithm is given in Algorithm 1.

Algorithm 1 A pseudo code of the fuzzy c-means clustering

```
define number of clusters c, tolerance ε
initialize partition matrix randomly
continue_iter ⟵ 1
while continue_iter = 1 do
    update cluster centers via Eq. 2.3
    update partition matrix via Eq. 2.4
    calculate objective function J_m
    if ||v^{t+1} − v^t|| ≤ ε or ||u^{t+1} − u^t|| ≤ ε then
        continue_iter ⟵ 0
    end if
end while
```

Although fuzzy c-means clustering has been implemented in wide variety of problems, FCM algorithm might be trapped at local optimal solutions due to its gradient based strategy. This entails the use of alternative methods, such as heuristics or stochastic search algorithms. Moreover, although the performance of clustering algorithm is assessed by using the objective function given in Eq. 1, there is a need for other measures to reveal detailed performance analysis. In the next section, cluster validity measures are introduced.

2.2. Cluster validity measures. Cluster validity measures are employed to evaluate clustering results. Significant number of approaches are devoted to test the quality of fuzzy partitioning. Two important criteria being used to evaluate clustering performance are compactness and separation. Compactness measures closeness of cluster elements.

Variance can be an example of compactness. If the elements in the cluster have low variance, this indicates closeness of the data points in that cluster. On the other hand, separation is related to how distinct the clusters are. Separation measures are based on the distances between different clusters. For instance, distance between centers of different clusters can be a good example of separation measures.

In this study, we have selected partition coefficient, partition entropy, validity index of Chen and Linkens, validity index of Fukuyama and Sugeno, and the validity index of Xie and Beni.

2.2.1. *Partition coefficient.* The partition coefficient validity (VPC) index is defined as [9]:

$$V_{PC} = \frac{1}{N}\sum_{i=1}^{c}\sum_{j=1}^{N} u_{ij}^2 \tag{2.5}$$

V_{PC} index evaluates the relative amount of membership being shared between pairs of subsets in partition matrix. The value of the index ranges in [1/c,1]. Cluster performance is high if the V_{PC} index is maximized.

2.2.2. *Partition entropy.* Partition entropy V_{PE} is defined as [10]:

$$PE = -\frac{1}{N}\sum_{i=1}^{c}\sum_{j=1}^{N} u_{ij}\log_b(u_{ij}) \tag{2.6}$$

where b is the base of the logarithm. PE takes its minimum value when the cluster structure is optimal.

2.2.3. *Chen and Linkens validity index.* Validity index of the Chen and Linkens VP is defined as [13]:

$$V_P = \frac{1}{N}\sum_{k=1}^{N}\max_i(u_{ik}) - \frac{1}{K}\sum_{i=1}^{c-1}\sum_{j=i+1}^{c}\left[\frac{1}{N}\sum_{k=1}^{N}\min(u_{ik},u_{jk})\right] \tag{2.7}$$

where $K = \sum_{i=1}^{c-1} i$. The first term represents the compactness within a cluster. When the kth pattern $\mathbf{x}_k$ is closer to the cluster center, then the maximum membership grades $\max_i(u_{ik})$ approach to 1. The second term makes use of intersection of two fuzzy sets and indicates separation measure. Here, fuzzy separation between $\mathbf{v}_i$ and $\mathbf{v}_j$. If data point $\mathbf{x}_k$ is close to the cluster center $\mathbf{v}_i$, then $\min(u_{ik}, u_{jk})$ approaches to 0, as a result of that ith and jth center can said to be clearly separated. Conversely, if the value of $\min(u_{ik}, u_{jk})$ is close to 1/c, then it means that $\mathbf{x}_k$ belongs to the all clusters with equal memberships and clusters are not well separated. VP indicates optimal cluster number when it is maximized

2.2.4. *Fukuyama and Sugeno index.* The validity index of Fukuyama and Sugeno is defined as [17]:

$$V_{FS} = J_m(u,v) - K_m(u,v) = \sum_{i=1}^{c}\sum_{j=1}^{N} u_{ij}^m \left\|\mathbf{y}_j - \mathbf{v}_i\right\|^2 - \sum_{i=1}^{c}\sum_{j=1}^{N} u_{ij}^m \|\mathbf{v}_i - \bar{\mathbf{v}}\|^2 \tag{2.8}$$

where $\bar{\mathbf{v}} = \sum_{i=1}^{c} \mathbf{v}_i/c$. The first term is the compactness of the representation of data in terms of cluster centers. The second term is related to separation that is expected to be maximized. Therefore, minimum V_{FS} indicates high quality of clustering.

2.2.5. *Xie and Beni index.* Xie and Beni index is calculated as [35]:

$$V_{XB} = \frac{J_m(u,v)/n}{Sep(v)} = \frac{\sum_{i=1}^{c}\sum_{j=1}^{N} u_{ij}^{m}\left\|\mathbf{y}_j - \mathbf{v}_i\right\|^2}{n\min_{i,j}\left\|\mathbf{v}_i - \mathbf{v}_j\right\|^2} \tag{2.9}$$

In the numerator part of the V_{XB} compactness of fuzzy partition is calculated, whereas the denominator part stands for strength of separation between clusters. Therefore, minimization of the index implies high performance of clustering.

3. Weighted Superposition Attraction Algorithm

In this section, first, the building blocks of WSA algorithm is described.

3.1. Initialization of agents. Initialization phase of WSA algorithm comprises of parameters setting and generating initial solutions. Parameters and their definitions will be given in subsequent sections where relevant. Initial positions of agents are determined randomly within the range of boundaries as given in Eq. 3.1.

$$x_{ij} = x_j^{\min} + rand(0,1)(x_j^{\max} - x_j^{\min}) \tag{3.1}$$

where $i = 1, 2, \ldots, nPop$, $j = 1, 2, \ldots, D$. The expressions rand(0,1), x_{ij}, $x_{ij}^{\min}$ and $x_{ij}^{\max}$ represent a uniform distributed random number $\in [0, 1]$, the value of the ith agent at the jth dimension, lower and upper bounds for the jth dimension, respectively. Number of search agents are denoted as $nPop$ and number of optimization parameters are represented by D.

After initialization, fitness of each solution vector (agent) is evaluated and global best solution is assigned. Then, population of search agents are subjected to repeat cycles of the search processes until the termination criterion (maximum number of iterations) is met. Subsequent to initialization stage, each iteration is initialized by the superposition generation phase.

3.2. Superposition generation. Guidance of the search process is one of the key steps in any metaheuristic optimization algorithm. The questions like how to choose moving patterns, or how to generate neighborhood solutions should be properly answered when designing an effective search algorithm. In WSA algorithm, movement of search agents are guided by a solution vector so called superposition which is indeed a physical term stating that the individual responses caused by more than one stimuli at a given place and time may be modeled as a single response. According to superposition principle, each currently discovered point by an agent is considered as stimuli in WSA algorithm. Each solution exerts attraction over the other agents in proportion to its fitness value. WSA algorithm is able to combine each stimulus (discovered point) in number of different ways. Initially, the attractiveness of each agent is transformed into weights. For this aim, the agents of the population are ranked according to their fitness values in ascending and descending order for minimization and maximization problems, respectively. Then, weights of the agents are calculated based on the Eq. 3.2.

$$weight(i) = i^{-\tau} \tag{3.2}$$

where $i = 1, 2, \ldots, nPop$. The weight of the ith-ranked agent is represented by $weight(i)$. The parameter $\tau \in [0, 1]$ determines relative differences of the weight values. The Eq. 3.2 states that lower the rank of an agent, the higher the weight it will be assigned.

The weights of agents might be used in different ways to generate superposition. The first proposed approach is to calculate weighted sum of the agents position vectors in order to form superposition [3, 4]. However, weighted sum does not always lead to good quality

solutions, especially when each locus of the agent position stores only positive values. Another way to generate superposition is to employ roulette wheel selection [5]. Weights of agents in the population here create selection pressure, which means that the agent with a relatively higher weight has more chance of acting on the corresponding dimension of the superposition. The superposition generation steps of [5] can be summarized as follows: First, uniform random numbers are generated for each dimension. These random numbers are interpreted as threshold values for the candidate agents which are subjected to roulette wheel selection procedure. The agents whose weights are greater or equal to the generated random numbers (thresholds) for the corresponding dimension undergo roulette wheel selection. Roulette wheel selection is conducted based on fitness values of the chosen agents. The winner agent transfers its position value to the corresponding dimension of the superposition. This procedure is followed till all the dimensions are filled up.

A motivating example of the superposition generation mechanism is illustrated in Table 1. As given in the Table 1, weights are calculated with respect to different τ values. Suppose that the τ value is determined as 0.70 and generated random numbers for each dimension are 0.60, 0.36, 0.38, 0.32, 0.56, 0.33, and 0.80, respectively. According to these threshold values, the solution agents, which will undergo roulette wheel selection, are highlighted with gray color in Table 1. For instance, considering the first dimension, the first two agents compete in the roulette selection. Consequently, let us assume that the first agent wins the competition. Thus, the first dimension of the superposition becomes 0.25, which is indicated with star symbol. Similarly, the second dimension of the superposition turns out to be 5.32 as the first agent is selected in the roulette wheel selection against second, third, and forth agents. The same procedure is repeated for the rest of the dimensions.

Table 1. Randomized superposition determination procedure for WSA

								weight				
ranks	*dimensions*							$\tau = 0.1$	$\tau = 0.3$	$\tau = 0.5$	$\tau = 0.7$	$\tau = 0.9$
1	0.25*	5.32*	2.15	0.20	0.18*	0.45	1.44*	1.000	1.000	1.000	1.000	1.000
2	0.18	4.48	3.45	0.23*	0.13	1.21*	3.30	0.933	0.812	0.707	0.615	0.535
3	0.45	3.45	8.63*	0.30	0.12	1.63	6.12	0.895	0.719	0.577	0.463	0.372
4	0.96	7.45	2.14	0.28	0.09	2.44	2.15	0.870	0.659	0.500	0.378	0.287
5	0.15	6.48	6.45	0.16	0.21	0.16	3.22	0.851	0.617	0.447	0.324	0.234
6	0.48	2.14	3.15	0.48	0.06	1.32	1.45	0.835	0.584	0.408	0.285	0.199
7	0.54	6.47	6.25	0.54	0.54	0.48	2.63	0.823	0.557	0.377	0.256	0.173
8	0.14	3.15	3.48	0.32	0.14	1.33	3.85	0.812	0.535	0.353	0.233	0.153
random numbers	0.60	0.36	0.38	0.32	0.56	0.33	0.80					
superposition	0.25	5.32	8.63	0.23	0.18	1.21	1.44					

It should be emphasized here that the first ranked agent is always a candidate for the roulette wheel selection as any power of unity is equal to unity again. Chances of other candidates are determined via τ value. That is to say, lower values of τ give nearly equal chances to candidate agents. Increasing the τ value, the algorithm tends to exhibit greedy behavior that better solutions are given higher probabilities in the roulette wheel selection.

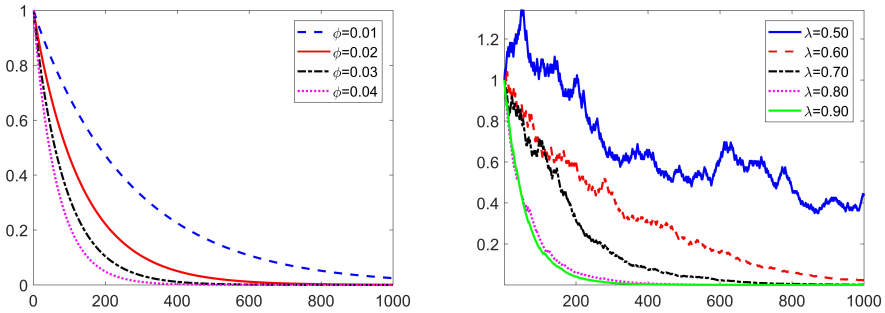
3.3. Search mechanism. Neighborhood generation and moving of agents are crucial steps in directing an effective search. Prior to details of movement of agents, the concept of step length is introduced first.

3.3.1. *Step sizing.* Determination of a proper step length contributes to have high level of intensification in the search process. Various approaches have been proposed for step length determination. These approaches can be categorized under the rubric of adaptive

and variable step sizing strategies. WSA algorithm follows variable step length strategy, which begins with an initial step length and changes the step length as the search progresses by using a proportional rule. The proportional rule requires a randomly generated number and a user defined parameter λ. The step size updating function is given in Eq. 3.3.

$$sl(t+1) = \begin{cases} sl(t) - e^{-t/(t+1)} \times \varphi_{step} \times sl(t), & \text{if } r \leq \lambda \\ sl(t) + e^{-t/(t+1)} \times \varphi_{step} \times sl(t), & \text{if } r > \lambda \end{cases} \tag{3.3}$$

where t is the iteration number and φ_{step} is a user defined parameter. The step sizing function given in Eq. 3.3 has a decreasing trend as the iterations continue, however, step length increases for some iterations during the search process. Due to this property, WSA is able to intensively explore the search space without trapping into local extremum points. General behavior of step size parameters is illustrated in Figure 1. Figure 1.a



(a) Behavior of ϕ **(b)** Behavior of λ

Figure 1. General behavior of the step sizing function

shows behavior of φ during the search process. The step length dramatically reduces as the φ value increases. On the other hand, small φ values give rise to gentle decrements in the step length. Effect of another parameter λ on step length is visualized in Figure 1.b. Small λ values bring about frequent increments in the step length, while high λ values produce consistent decrements of the step length. As the step length directly affects the intensification ability of the search algorithm, φ and λ values should be properly determined.

3.3.2. *Moving of agents.* The present study introduces new move mechanisms for WSA that differ from the first introduced ones [3, 4]. The proposed moves are given in the following.

Once the superposition is generated and the step lengths are calculated, each agent decides on its movement. Two types of movements are defined in the WSA algorithm. The first movement is moving towards superposition, and the second one is performing random walk. In the former case, each agent comes to its decision by comparing its own fitness with the fitness of the superposition. An agent certainly moves towards superposition if the fitness of the superposition is better than the agents fitness. The movement equation is given as in Eq. 3.4:

$$\mathbf{x}_i = \mathbf{x}_i + sl \times (\mathbf{x}_{sp} - \mathbf{x}_i) + \alpha\,(rand - 0.5) \tag{3.4}$$

where $\mathbf{x}_i = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ represents position vector of the ith agent, $\mathbf{x}_{sp}$ is the position vector of the superposition, α is the randomization parameter and rand is a random number uniformly distributed in [0,1].

If the fitness of the superposition is worse than an agent, then an agent may move towards the superposition or perform a random walk. This type of decision is made based upon generating a random number and comparing its value with the $e^{f(\mathbf{x}_i)-f(\mathbf{x}_{sp})}$. Here, fitness values of the superposition and ith agent are denoted by $f(\mathbf{x}_{sp})$ and $f(\mathbf{x}_i)$, respectively. If the randomly generated number is lower than or equal to the obtained value, agent moves towards the superposition. Otherwise, agent performs a random walk.

Preliminary work showed that the randomness parameter α play a critical role in finding good quality of solutions and it was seen that the performance of the proposed WSA might further be improved with the decreasing values of levels of α. Thus, by using the proposed formulation for α (Eq. 3.5), local optima traps are avoided at the prior iterations, whereas the effect of randomness can be decreased so as to encourage intensification towards the end.

$$\alpha(t+1) = \alpha(t) - e^{-t/(t+1)} \times \varphi_{randomness} \times \alpha(t) \tag{3.5}$$

As described earlier, agents might search for randomly selected points if it does not follow superposition. In this situation, number of different ways can be considered for performing random walk. In this study, Gaussian random numbers are used for performing random walk as given in Eq. 3.6:

$$\mathbf{x}_i = \mathbf{x}_i + sl \times N(0,1) \tag{3.6}$$

where $N(0,1)$ denotes Gaussian random number with mean value of 0 and standard deviation of 1.

3.4. Quantum particles based local search. The use of quantum particles was proposed by Blackwell and Branke [12] within the context of metaheuristic search algorithms as a means for maintaining a certain level of diversity in the population. However, it is modified as an intensifying local search procedure in the present work.

Quantum particles of Blackwell and Branke [12] have been indeed inspired by the atomic models, in which a number of electrons orbit a small ball of nucleons. Distinctive feature of the quantum particles is that the particles do not orbit in deterministic paths, similar to the particle behavior in quantum physics, they are distributed within a probability cloud around the nucleus.

In our implementation, global best solution found so far is considered as the nucleus, and quantum particles are used for performing an intensive local search around the best solution. Generating one quantum particle is given in Algorithm 2.

Algorithm 2 A pseudo code for generating quantum particles

1: define input parameters r_{cloud}
2: generate random Gaussian vector $\mathbf{g} = \{g_1, g_2, \ldots, g_d\}$ from $\mathcal{N}(0,1)$
3: calculate spatial distance of g_i to the origin $dist = \sqrt{\sum_{i=1}^{d} g_i^2}$
4: generate a uniform number $u \in [0,1]$
5: $\mathbf{x}_q = \mathbf{x}_{gb} + \mathbf{g} \times r_{cloud} \times \sqrt[d]{u}/dist$
6: repair infeasibilities in boundary constraints

There are two important parameters of the quantum particles namely, number of quantum particles and quantum radius, which are denoted by *nQuantum* and r_{cloud}, respectively. Through the quantum particle generations, r_{cloud} parameter is gradually

decreased (Eq. 3.7) so that the particles are tightly clustered around the global best solution. In Eq. 3.7, q is the local search counter and the maximum value of q will be equal to *nQuantum*. Thus, while a quantum particle is generated with the cloud $r_{cloud}(q)$, the next quantum particle will be generated with the $r_{cloud}(q+1)$, which is smaller than $r_{cloud}(q)$. The speed of decrement can be tuned via the parameter $\varphi_{quantum}$. Figure 2 illustrates quantum particle generation strategy.

$$r_{cloud}(q+1) = r_{cloud}(q) - e^{-l/(l+1)} \times \varphi_{quantum} \times r_{cloud}(q) \tag{3.7}$$

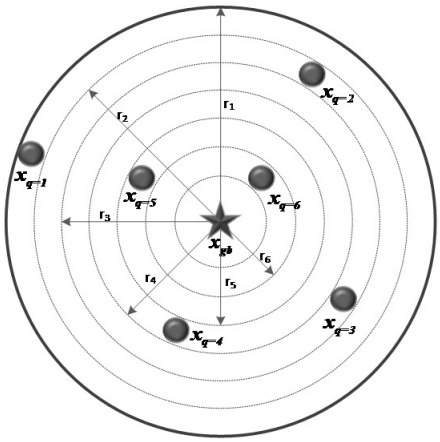


Figure 2. Quantum particles located around nucleus

In the Figure 2, The global best solution ($\mathbf{x}_{gb}$) is represented by star symbol and considered as the nucleus. The first quantum particle is shown by $\mathbf{x}_{q=1}$ is distributed around the nucleus within the diameter of r_1. Then, r_{cloud} is reduced and the second quantum particle $\mathbf{x}_{q=2}$ is located at the region within the diameter of r_2. This process continues and the quantum particles are drawn together around the nucleus in the latter generations.

4. Fuzzy clustering via WSA

This section presents the implementation details of the WSA algorithm for optimizing FCM clustering. As mentioned earlier, main disadvantage of the traditional FCM algorithm is that it is prone to getting trapped at local optima. In order to escape from local optima, WSA algorithm is used to optimize cluster centers so that best cluster centers are sought for minimization of the objective function.

In the WSA-based optimization of FCM clustering, cluster centers are $\mathbf{v} = [v_{ij}]_{c\times s}$ considered as decision variables and encoded as position of agents. The total of $c \times s$ decision variables are encoded. The position vector of the ith search agent is represented by:

$$\mathbf{x}_i = \left[x_{i,1}, x_{i,2}, \ldots, x_{i,s}, \ldots, x_{i,c\times(s-1)+1}, x_{i,c\times(s-1)+2}, \ldots, x_{i,c\times s}\right] \tag{4.1}$$

where s denotes number of features in the dataset and the first s elements represent first cluster, and the following s elements represent the second cluster and so on. By this way, cluster centers v lend themselves to fitness function evaluations. The cluster centers can be obtained by decoding search agents.

The position vector of each search agent is decoded in order to obtain cluster centers. When the cluster centers are acquired as a result of decoding procedure, fuzzy partition matrix is calculated. Then, fuzzy partition matrix, cluster centers and data points are used to calculate objective function J_m. Based on the objective function of the FCM clustering, fitness function is defined as:

$$f = \frac{1}{J_m + 1} \tag{4.2}$$

where f represents fitness function to be maximized.

Finally, putting all things together, a pseudo code for the proposed WSA algorithm is presented in Algorithm 3.

Algorithm 3 A pseudo code for the proposed WSA

```
read data
define input parameters
initialize population of nPop solution vectors x_i, i = 1,...,nPop
evaluate fitness of each solution vector f(x_i)
find the global best solution x_gb
while t < maxIter do
   sort the solution vectors in ascending order
   assign weights
   generate a superposition x_sp via roulette-wheel (Table 1)
   evaluate fitness of superposition f(x_sp)
   for i=1:nPop do
      if x_sp < x_i then
         // move towards the superposition (Eq. 3.4)
         x_i ⟵ superWalk(x_sp, x_i)
      else
         if rand ∈ [0,1] ≤ e^(f(x_i)−f(x_sp)) then
            // move towards the superposition (Eq. 3.4)
            x_i ⟵ superWalk(x_sp, x_i)
         else
            // perform random walk (Eq. 3.6)
            x_i ⟵ randomWalk(x_i)
         end if
      end if
      Update global best solution
   end for
   //Local search based on quantum particles
   quantumCloud ⟵ initial_quantumCloud
   for q=1:nQuantum do
      x_q ⟵ quantumWalk(x_gb, quantumCloud)
      if x_q ≤ x_gb then
         update the global best solution
      end if
      update quantumCloud (Eq. 3.7)
   end for
   update sl (Eq. 3.3), update α (Eq. 3.5)
end while
```

5. Experimental study

In this section, results of the experimental study are given. Total of 12 datasets are used to evaluate performance of the proposed WSA algorithm. The datasets used in the present study are described in the following.

5.1. Datasets. The datasets obtained from UCI machine learning repository [24] are used to evaluate performance of the proposed WSA algorithm in fuzzy clustering. There is no missing value in these datasets, and they are characterized by number of instances, number of attributes, and number of clusters as given in Table 2.

Table 2. Characteristics of datasets

dataset	*number of instances*	*number of attributes*	*number of clusters*
Balance scale	625	4	3
Cmc	1473	9	3
Glass	214	9	6
Haberman	306	3	2
Hayesroth	160	5	3
Heart (Statlog)	270	13	2
Iris	150	4	3
Lenses	24	4	3
Magic04	19020	10	2
Robot Navigation	5456	2	4
Spect	80	22	2
Wine	178	13	3

Brief description of the datasets can be summarized as follows [24]:

Balance scale weight and distance database was generated based on psychological experiments reported by Siegler [34]. Total of 625 samples are classified as having the balance scale tip to the right, left, or balanced. The attributes are left-weight, left-distance, right-weigh, and right-distance.

Contraceptive method choice dataset (cmc) is a subset of 1987 National Indonesia contraceptive prevalence survey. In this dataset, there are 9 attributes and the goal is to predict the current contraceptive method choice of women based on the demographic and socio-economic factors.

Glass dataset belongs to the USA Forensic Science Service, in which glasses are characterized by 9 attributes. The motivation is that a glass left can give valuable information in terms of criminological evidence at the scene of a crime. There are six classes and 214 instances.

Habermans survival dataset is concerned with survival of patients who had undergone surgery for breast cancer. The collected data belongs to University of Chicagos Billings hospital and instances were collected between 1958 and 1970. There are 306 instances, 3 attributes, and 2 clusters in the dataset.

Hayesroth dataset is based on the work of [20], where attributes are defined as name, hobby, age, educational level, and marital status. The goal is to classify objects into classes that take nominal value between 1 and 3.

Heart (Statlog) dataset has 13 representative attributes and 270 instances. The goal is to predict absence or presence of heart disease depending on the indicators given in attributes.

Iris dataset is one of the most commonly used dataset in the pattern recognition field. Iris dataset has 150 instances and 4 attributes which are representing sepal length, sepal width, petal length, and petal width in centimeters. There are 3 classes as Setosa, Versicolor, and Virginica, with 50 samples per class.

Lenses dataset has 24 instances and 4 attributes. Each instance represents a patient with age, spectacle prescription, being astigmatic or not, and tear production rate information. The goal is to classify patients for hard, soft, or no contact lenses categories.

Magic Gamma (magic04) dataset involves instances that are related to high energy gamma particles observed by Cherenkov gamma telescope. There are 19020 instances and 10 attributes in the dataset. There are two classes which are gamma and hadron.

Wall-following robot navigation (Robot Navigation) dataset comprises of sensor data and robot move classes. In the present study, 2 ultrasound readings are used as attributes and four classes are used for number of clusters. There are 5456 instances in the dataset.

SPECT Heart dataset (Spect) is based on cardiac single proton emission computed tomography images. Patients are classified as normal or abnormal. There are 80 instances and 22 attributes.

Wine dataset shows results of chemical analysis of wines grown in Italy. These wines are derived from three different cultivars. There are 13 constituents found in each type of wines. The dataset contains 178 instances and 3 classes.

5.2. Fine tuning of parameters. It is known that the performance of a stochastic search algorithm heavily depends on the used values of parameters. Therefore, first, the parameters of WSA are tuned in a step by step manner to find appropriate values for them. Throughout fine tuning tests, a medium-size dataset, namely glass dataset is used.

It is clear that the upcoming values of sl are indeed dependent to initial value of sl that is denoted by $sl(1)$. As the parameter sl is highly influential factor in the performance of metaheuristic optimization, the parameters $sl(1)$ and φ_{step} are tuned by fixing the rest of the parameters. Obtained results are given in Table 3.

Table 3. Tuning of $sl(1)$ and φ_{step}

φ_{step}=0.001		φ_{step}=0.0001		φ_{step}=0.00001	
sl(1)	best	sl(1)	best	sl(1)	best
0.1	165.0008506678658	0.1	162.9152808105720	0.1	162.7137662313837
0.5	160.1182663761601	0.5	157.7969927908699	0.5	159.5885258464963
0.8	159.4083777664926	0.8	158.8175975713733	0.8	158.4077071732732
0.9	157.8895222947028	0.9	157.5080298140393	0.9	158.3259830770111
1	157.3176492840064	1	**156.6663350043063**	1	157.8450722453701
1.1	156.9728330368572	1.1	157.7604209100807	1.1	159.1657531874061
1.2	157.6056963796496	1.2	158.7602132363775	1.2	158.8447250143040
1.5	157.4825978440570	1.5	160.6263498169422	1.5	160.9497977501265
2	157.9689968605096	2	211.8619918738082	2	234.3812022421562
3	181.5139850792915	3	313.5116043978085	3	330.8144636874062
5	352.8221653355770	5	322.9584472646028	5	327.2212187033614

It is again clear that upcoming values of α depend on the initial value of this parameter $\alpha(1)$. Fixing the values of sl(1) and φ_{step} to 1.00 and 0.0001, respectively (Table 3), varying values for $\alpha(1)$ and $\varphi_{randomness}$ are tested. Obtained results for these parameters are given in Table 4.

According to the results of Table 4, the proposed WSA can achieve better results while $\alpha(1)$ and $\varphi_{randomness}$ are fixed to 0.05 and 0.001, respectively.

Table 4. Tuning of $\alpha(1)$ and $\varphi_{randomness}$

$\varphi_{randomness} = 0.001$		$\varphi_{randomness} = 0.0001$		$\varphi_{randomness} = 0.00001$	
sl(1)	best	sl(1)	best	sl(1)	best
0	557.6814051755958	0	557.6814051755958	0	557.6814051755958
0.05	**154.1597167350611**	0.05	154.2290187052191	0.05	155.8121939085336
0.1	154.2092076125439	0.1	154.3157878545660	0.1	154.3749566095637
0.5	154.9364345965384	0.5	157.0216036382375	0.5	156.7409020343351
0.6	155.2312214822992	0.6	160.0692016716831	0.6	157.7692235639947
0.8	156.4727733394877	0.8	161.4094908196177	0.8	161.9529180806684
0.9	157.0137549584548	0.9	162.4027908280079	0.9	163.2516572684658
1	157.8092511779108	1	164.7461657222427	1	164.4016999273107
2	167.0668923236167	2	177.7129988173032	2	179.6105190632742
3	174.5040637869102	3	188.9542627575888	3	201.3224817191195
5	197.2488612799824	5	223.0493406420312	5	222.4804497545378
7	217.4706711473002	7	231.1562167624101	7	252.1146553423424
10	251.5662116210819	10	311.0214867170516	10	300.5899867746922

The next test is devoted to finding appropriate values for nPop andτ . It is clear that they have simultaneous effects of the generated superposition. Therefore, they were simultaneously analyzed in Table 5. As one can see from Table 5 that better results can be obtained when for nPop and τ are fixed to 20 and 0.50, respectively.

Table 5. Tuning of τ and *nPop*

nPop=15		nPop=20		nPop=25	
τ	best	τ	best	τ	best
0.1	163.2384329273312	0.1	165.9652409815041	0.1	154.2919487752122
0.2	163.5769417171133	0.2	154.2583721869968	0.2	154.2317009506931
0.3	155.4251137006843	0.3	154.1671873610296	0.3	154.1647376413734
0.4	154.2187703226765	0.4	154.1767479911774	0.4	154.1644531842730
0.5	154.7646289752577	**0.5**	**154.1597167350611**	0.5	154.1792165112592
0.6	154.1637529009309	0.6	154.1796126808663	0.6	154.1612183527501
0.7	154.1617916662650	0.7	154.1695894032582	0.7	154.1613007545423
0.8	154.1752374780770	0.8	154.1589153712213	0.8	154.1625626171038
0.9	154.6190886930279	0.9	154.1973253299350	0.9	154.1552273570345

The final test is devoted to finding the values of the quantum based local search parameters. All obtained results are given in Table 6. It is clear from this table that the efficient of WSA can be increased when of r_{cloud} and $\varphi_{quantum}$ are fixed to 1.00 and 0.70, respectively. The final parameter nQuantum is arbitrarily used as 50.

5.3. Performance evaluation of WSA. It is clear that a total of 70 fitness function evaluations are performed for each iteration in the proposed WSA. The termination criterion is determined as maximum number of iterations, which is equal to 2000. Therefore, 140,000 function evaluations are performed in total. All of the experiments are conducted on a PC with Intel I7 2.4 Ghz processor and 16 GB RAM and all results are evaluated over 30 independent replications. The resulting the objective function values are given in Table 7.

Table 6. Tuning of r_{cloud} and $\varphi_{quantum}$

$\varphi_{quantum} = 0.3$		$\varphi_{quantum} = 0.5$		$\varphi_{quantum} = 0.7$	
$r_{cloud}(1)$	best	$r_{cloud}(1)$	best	$r_{cloud}(1)$	best
0.01	154.1459871311321	0.01	162.5798261770892	0.01	154.9584394786354
0.05	154.1459934004529	0.05	154.1459876669529	0.05	154.1643027186621
0.1	154.1460010842752	0.1	154.1459868608848	0.1	154.1459872860223
0.2	154.1460345576582	0.2	154.1459868678999	0.2	154.1627937265761
0.3	154.1460896990897	0.3	154.1459876127277	0.3	154.1459902231412
0.5	154.1461861404444	0.5	154.1459928507705	0.5	154.1463733331513
0.7	154.1470005609310	0.7	154.1459871846886	0.7	154.1460082767046
1	159.9709013996618	1	154.1459872012569	**1**	**154.1459868575310**
2	162.9652576795842	2	154.1459890228997	2	154.1459870680219
3	162.9579731415267	3	154.1459897462816	3	154.1518714951608
5	162.6928536855918	5	154.1459912747030	5	154.1459870251029
10	163.1065324170578	10	162.9478662601923	10	162.5799215805601

According to resulting objective function values, WSA algorithm outranked classical FCM algorithm in all instances. Although small differences are observed in the best values of the results, the WSA algorithm produces smaller standard deviation in general. Also note that these smaller deviations can have great impact on the final outcomes in different application domains.

As the results are validated for objective function, performance of the WSA algorithm in terms of other metrics is examined. In Table 8, results for the partition coefficient index are given. According to resulting partition coefficient indices, WSA performance is better in 50% of the instances which are glass, haberman, heart, magic04, robot navigation, and spect. On the other hand, in balance scale, cmc, hayesroth, iris, lenses and wine datasets, traditional FCM yields slightly better results in comparison with the WSA.

Similarly, partition entropy indices are calculated for the WSA and traditional FCM results as given in the Table 9.

In 58% of the instances, partition entropy indices produced by WSA algorithm surpass traditional FCM results. On the other hand, 42% of the all instances, traditional FCM algorithm exhibits better performance with a very small differences.

In Table 10, resulting Chen and Linkens indices are given.

According to Chen and Linkens index, WSA has better performance in glass, haberman, heart, magic04, robot navigation, and spect datasets. Hence, WSA has better performance in 50% of the instances in terms of Chen and Linkens index.

Results of Fukuyama and Sugeno indices are given in Table 11.

As in the case with Chen and Linkens index, WSA performs better in 50% of the instances with respect to Fukuyama and Sugeno index. In the glass, haberman, heart, magic04, robot navigation and spect datasets, WSA shows better performance. On the other hand, in the balance scale, cmc, hayesroth, iris, lenses and wine datasets, traditional FCM algorithm exhibits better performance with slight differences.

In Table 12, results of Xie-Beni index are given.

According to Xie-Beni index, 67% of the instances, WSA outperforms the traditional FCM algorithm. FCM algorithm has better performance in balance scale, cmc, hayesroth and wine datasets.

In order to see the convergence behavior of the proposed approach, related plots for glass, haberman, iris, lenses, spect and wine datasets are illustrated in Figure 3. All data points in this figure are the average values over 30 runs.

As one see from Figure 3 that WSA seems to be converging to promising values only within 50 iterations for haberman, iris, lenses, and spect datasets, which demonstrates

Table 7. Results of objective function value

data sets	*perf.*	*WSA*	*FCM*
balance scale	best	**1.66666666692E+03**	1.66666755513E+03
	mean	1.66666667028E+03	1.66666921515E+03
	std	2.50345720401E-06	1.09842913759E-03
cmc	best	**1.73842759127E+04**	1.73842759180E+04
	mean	1.73842759127E+04	1.73842759199E+04
	std	1.46768266630E-11	1.24825465112E-06
glass	best	**1.54145986857E+02**	1.54145990128E+02
	mean	1.55519457468E+02	1.54145994304E+02
	std	3.06519928090E+00	4.46033077336E-06
haberman	best	**2.15826290765E+04**	2.15826290781E+04
	mean	2.15826290765E+04	2.15826290794E+04
	std	4.16440504685E-12	8.75034743197E-07
hayesroth	best	**1.63710527427E+04**	1.63710527517E+04
	mean	1.63710527427E+04	1.63710527544E+04
	std	1.04000478825E-11	1.53420562635E-06
heart	best	**4.06868096383E+05**	4.06868096385E+05
	mean	4.06868096383E+05	4.06868096386E+05
	std	3.45377482473E-10	7.84870753214E-07
iris	best	**6.05057106295E+01**	6.05057110091E+01
	mean	6.05057106295E+01	6.05057153801E+01
	std	8.23781666227E-14	1.35229394782E-06
lenses	best	**1.05468487416E+01**	1.05468614506E+01
	mean	1.05468487416E+01	1.05468676584E+01
	std	4.93691014313E-15	3.61749698204E-06
magic04	best	**1.33807543985E+08**	1.33807543985E+08
	mean	1.33807544030E+08	1.33807543985E+08
	std	2.47113959284E-01	1.48075292333E-06
robot navigation	best	**3.43077123303E+02**	3.63698734051E+02
	mean	3.44607388717E+02	3.63871478848E+02
	std	3.61791975887E+00	9.46146008998E-01
spect	best	**1.43701203586E+02**	1.43701205491E+02
	mean	1.43701203586E+02	1.43701206711E+02
	std	2.23231956931E-13	1.02093341694E-06
wine	best	**1.79608275957E+06**	1.79608275958E+06
	mean	1.79608275957E+06	1.79608275958E+06
	std	5.72207146093E-08	2.03331614535E-06

the convergences capability of this algorithm. On the other hand, for larger scales data sets such as glass and vine datasets, it takes more time to achieve the printed results.

Table 8. Results of partition coefficient index

data sets	*perf.*	*WSA*	*FCM*
balance scale	best	3.34427658408E-01	**3.34664395310E-01**
	mean	3.34416440477E-01	3.33910515014E-01
	std	7.00175699474E-06	2.27587210299E-04
cmc	best	7.02434860790E-01	**7.02435917288E-01**
	mean	7.02434858636E-01	7.02434598879E-01
	std	1.54956139927E-09	9.22869715277E-07
glass	best	**6.07024829976E-01**	4.93019240193E-01
	mean	5.03391035693E-01	4.93015298402E-01
	std	2.78562676871E-02	1.26170132007E-06
haberman	best	**7.39772409625E-01**	7.39772167168E-01
	mean	7.39772407122E-01	7.39772097489E-01
	std	1.43170113425E-09	5.09930306272E-08
hayesroth	best	7.98825832426E-01	**7.98825841114E-01**
	mean	7.98825832366E-01	7.98825833369E-01
	std	3.58949503447E-11	7.72481965946E-09
heart	best	**7.12624906009E-01**	7.12624717599E-01
	mean	7.12624898628E-01	7.12624650031E-01
	std	3.55730457951E-09	4.04123699363E-08
iris	best	7.83397490182E-01	**7.83401609679E-01**
	mean	7.83397486489E-01	7.83392012133E-01
	std	1.84519743304E-09	3.10835603583E-06
lenses	best	4.23129963924E-01	**4.23133866511E-01**
	mean	4.23129957420E-01	4.23129354508E-01
	std	2.45481611147E-09	2.28936683179E-06
magic04	best	**6.56310749897E-01**	6.56310638081E-01
	mean	6.56310402064E-01	6.56310630978E-01
	std	1.57238821077E-06	4.73454231025E-09
robot navigation	best	**7.62723441074E-01**	6.92469967570E-01
	mean	7.59247820424E-01	6.91073239905E-01
	std	1.30906943295E-02	6.88041432555E-03
spect	best	**5.75772453635E-01**	5.75727924009E-01
	mean	5.75772436531E-01	5.75716208030E-01
	std	9.19721153255E-09	8.77043402433E-06
wine	best	7.90939872268E-01	**7.90940039156E-01**
	mean	7.90939863176E-01	7.90939844886E-01
	std	6.64530797983E-09	1.56317767490E-07

Finally, in order to demonstrate what intuition suggests, either parametric or non-parametric statistical tests are crucially required. Particularly, non-parametric tests are encouraged by Derracet al. [14] while comparing the performances of stochastic search

Table 9. Results of partition entropy index

data sets	*perf.*	*WSA*	*FCM*
balance scale	best	**1.58236705532E+00**	1.58237029056E+00
	mean	1.58239115857E+00	1.58365381141E+00
	std	1.50551573528E-05	5.07868748689E-04
cmc	best	7.76350641279E-01	**7.76348351582E-01**
	mean	7.76350646284E-01	7.76350788363E-01
	std	3.44288449963E-09	1.99151821234E-06
glass	best	**1.16718020985E+00**	1.43726385270E+00
	mean	1.41748136975E+00	1.43728122595E+00
	std	6.12891607344E-02	9.86853921895E-06
haberman	best	**5.96806315308E-01**	5.96806704068E-01
	mean	5.96806320517E-01	5.96806838216E-01
	std	3.01439109739E-09	9.46968854744E-08
hayesroth	best	5.32679515135E-01	**5.32679483959E-01**
	mean	5.32679515555E-01	5.32679510423E-01
	std	2.56664024969E-10	2.78531916182E-08
heart	best	**6.49284121541E-01**	6.49284478422E-01
	mean	6.49284135748E-01	6.49284581587E-01
	std	6.84184555184E-09	6.77863661882E-08
iris	best	5.70573735708E-01	**5.70571830994E-01**
	mean	5.70573743343E-01	5.70579878999E-01
	std	3.83115437467E-09	2.43239265544E-06
lenses	best	1.39120694367E+00	**1.39119847324E+00**
	mean	1.39120695785E+00	1.39120609392E+00
	std	5.35122298253E-09	4.44766750970E-06
magic04	best	**7.48445423462E-01**	7.48445626533E-01
	mean	7.48446034209E-01	7.48445637596E-01
	std	2.74317104109E-06	7.54379479268E-09
robot navigation	best	**6.74226306568E-01**	8.47503440696E-01
	mean	6.84387406744E-01	8.56075585829E-01
	std	3.36061851042E-02	2.49929136507E-02
spect	best	**8.80170342006E-01**	8.80250592345E-01
	mean	8.80170371680E-01	8.80273118437E-01
	std	1.60954903870E-08	1.58237462082E-05
wine	best	5.48812203163E-01	**5.48811931652E-01**
	mean	5.48812217562E-01	5.48812260733E-01
	std	1.04112993084E-08	2.49733889491E-07

algorithms. Because, the safe use of parametric tests is dependent to several conditions, which are not fulfilled in the present work.

In this regard, the non-parametric sign test, which is appropriate for pair-wise comparisons is applied here. In this test, first, the control algorithm that seems to be outperforming the alternative algorithm is assigned with the number of wins, which is simply the

Table 10. Results of Chen and Linkens index

data sets	*perf.*	*WSA*	*FCM*
balance scale	best	6.72562208620E-03	**2.36987589319E-02**
	mean	5.35950135625E-03	1.82933974914E-02
	std	8.99879369166E-04	2.58934720546E-03
cmc	best	7.21245225621E-01	**7.21246117416E-01**
	mean	7.21245223718E-01	7.21244944890E-01
	std	1.35752243614E-09	7.64643542571E-07
glass	best	**6.93216617085E-01**	5.70296985406E-01
	mean	5.82413904614E-01	5.70283070841E-01
	Std	3.15312199482E-02	8.70419409257E-06
haberman	best	**6.36294394332E-01**	6.36294069611E-01
	mean	6.36294391355E-01	6.36293961470E-01
	std	1.67316432660E-09	7.43562895525E-08
hayesroth	best	8.11560041151E-01	**8.11560043418E-01**
	mean	8.11560041041E-01	8.11560040949E-01
	std	4.20784034292E-11	2.58155402703E-09
heart	best	**6.02110904240E-01**	6.02110714096E-01
	mean	6.02110896182E-01	6.02110657947E-01
	std	3.82658971352E-09	3.68558490200E-08
iris	best	8.01544415177E-01	**8.01561168345E-01**
	mean	8.01544410325E-01	8.01525001707E-01
	std	3.18282680557E-09	1.09273279155E-05
lenses	best	3.56057710604E-01	**3.56091608478E-01**
	mean	3.56057696206E-01	3.56082862761E-01
	std	5.54873613812E-09	1.04712609529E-05
magic04	best	**4.97407995118E-01**	4.97407812994E-01
	mean	4.97407417084E-01	4.97407801828E-01
	std	2.63261242250E-06	8.56419981569E-09
robot navigation	best	**8.05704291152E-01**	7.40361007694E-01
	mean	8.03120126643E-01	7.38753954660E-01
	std	1.19355184754E-02	7.84839815961E-03
spect	best	**3.00982613914E-01**	3.00882228799E-01
	mean	3.00982572877E-01	3.00854768786E-01
	std	2.03913783918E-08	2.06582624412E-05
wine	best	8.07291425668E-01	**8.07291691793E-01**
	mean	8.07291410621E-01	8.07291380260E-01
	std	1.10715004029E-08	2.50419341407E-07

total number of instances (data sets) for which the WSA outranks FCM. Next, according to total number instances and the number wins, a threshold level is defined based on the assumed significance level α to decide whether a significant difference exists [38]. All

Table 11. Results of Fukuyama and Sugeno index

data sets	*perf.*	*WSA*	*FCM*
balance scale	best	1.66656200506E+03	**1.66385601714E+03**
	mean	1.66660447517E+03	1.66539018889E+03
	std	2.65035256304E-02	4.80134416726E-01
cmc	best	-5.24998955353E+04	**-5.25003450336E+04**
	mean	-5.24998938088E+04	-5.24996943298E+04
	std	1.04886834248E-03	3.71187010251E-01
glass	best	**-8.78515544440E+02**	-4.98383719602E+02
	mean	-5.20671558626E+02	-4.98290544993E+02
	Std	7.97504711383E+01	4.53819767272E-02
haberman	best	**3.67663731222E+03**	3.67667523736E+03
	mean	3.67663809129E+03	3.67668584663E+03
	std	4.47448496040E-04	7.81057119278E-03
hayesroth	best	-1.33128931824E+05	**-1.33128935184E+05**
	mean	-1.33128930828E+05	-1.33128931056E+05
	std	5.87485791566E-04	3.88375103617E-03
heart	best	**1.23936322781E+05**	1.23936892588E+05
	mean	1.23936353656E+05	1.23937057499E+05
	std	1.44379388334E-02	1.11372239120E-01
iris	best	-4.50503493787E+02	**-4.50538297368E+02**
	mean	-4.50503476082E+02	-4.50467567423E+02
	std	9.74287782086E-06	1.40857197726E-02
lenses	best	6.22811347164E+00	**6.22803857347E+00**
	mean	6.22811398767E+00	6.22819382489E+00
	std	1.94739511761E-07	8.56222979222E-05
magic04	best	**9.35561664990E+07**	9.35562251187E+07
	mean	9.35564338606E+07	9.35562283885E+07
	std	1.28513216398E+03	2.38760717123E+00
robot navigation	best	**-2.25569636828E+03**	-1.76694752530E+03
	mean	-2.21497417675E+03	-1.72431400241E+03
	std	1.12750857318E+02	1.55844065111E+02
spect	best	**1.30256932848E+02**	1.30270296112E+02
	mean	1.30256938054E+02	1.30275036843E+02
	std	2.70095005998E-06	2.57480901585E-03
wine	best	-1.14603505211E+07	**-1.14603594335E+07**
	mean	-1.14603486381E+07	-1.14603472125E+07
	std	6.31871693584E-01	9.94547418168E+00

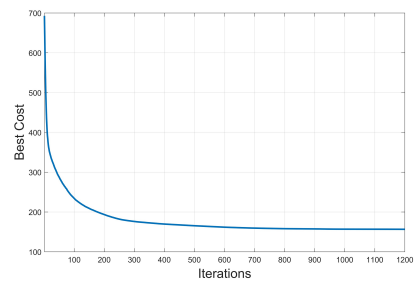
obtained results are presented in Table 13, where + and $\sim$ stand for significant difference and non-significant difference, respectively.

According to the results of Table 13, it can be concluded that WSA significantly outperforms FCM in terms objective function value (J), regardless of the α level. However, considering the rest of performance metrics, although WSA is at least as good as FCM

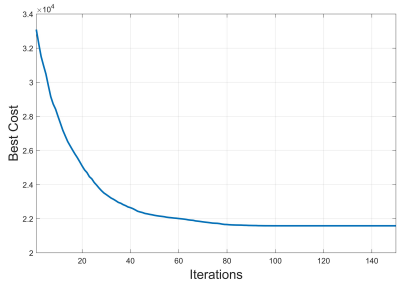
Table 12. Results of Xie Beni index

data sets	*perf.*	*WSA*	*FCM*
balance scale	best	3.54146830016E+03	**9.10820561093E+01**
	mean	7.82497747686E+03	8.09707508660E+02
	std	4.97216993083E+03	2.45613912957E+03
cmc	best	1.24860610637E-01	**1.24857387639E-01**
	mean	1.24860618254E-01	1.24860908722E-01
	std	3.25319691213E-09	2.83636572175E-06
glass	best	**3.95912911847E-01**	2.35740809265E+00
	mean	2.04203202513E+00	2.35772780214E+00
	std	7.17051101815E-01	2.56789030274E-04
haberman	best	**2.22917469864E-01**	2.22917865794E-01
	mean	2.22917478808E-01	2.22917991087E-01
	std	5.20766600592E-09	9.85294737493E-08
hayesroth	best	5.92920703715E-02	**5.92906860659E-02**
	mean	5.92920720899E-02	5.92923597671E-02
	std	1.33590940731E-09	1.28963527951E-06
heart	best	**2.56196339538E-01**	2.56196798815E-01
	mean	2.56196364841E-01	2.56196929580E-01
	std	1.18193841455E-08	9.25412552443E-08
iris	best	**1.36908139625E-01**	1.36909846306E-01
	mean	1.36908153252E-01	1.36910937405E-01
	std	8.55463329608E-09	1.79343752226E-06
lenses	best	**7.26920396501E-01**	7.27002697796E-01
	mean	7.26920549484E-01	7.33736820813E-01
	std	7.52545770229E-08	3.22635567064E-03
magic04	best	**5.45443007296E-01**	5.45443698923E-01
	mean	5.45446341307E-01	5.45443743812E-01
	std	1.61084773569E-05	2.93537621959E-08
robot navigation	best	**1.65387995591E-01**	2.80248786872E-01
	mean	1.69595271853E-01	2.80952346961E-01
	std	2.09999755581E-02	1.88899772942E-04
spect	best	**1.53855861675E+00**	1.53997782143E+00
	mean	1.53855917087E+00	1.54042998521E+00
	std	2.84714882024E-07	2.62887679483E-04
wine	best	1.25659903785E-01	**1.25659513073E-01**
	mean	1.25659951608E-01	1.25659835411E-01
	std	1.79791062653E-08	3.72933392789E-07

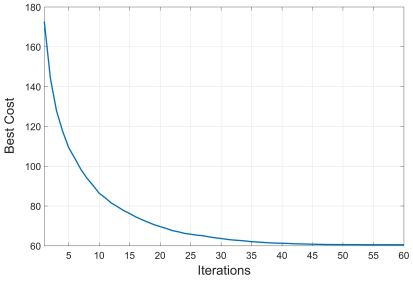
in the worst case, sign test do not point out significant differences for difference levels of α. This is indeed a usual circumstance, because WSA uses the value of J in all evaluations. Accordingly, one should recall the no free lunch theorem. Finally, putting things together, it can be put forward that WSA clearly outperforms FCM for the mentioned data sets.



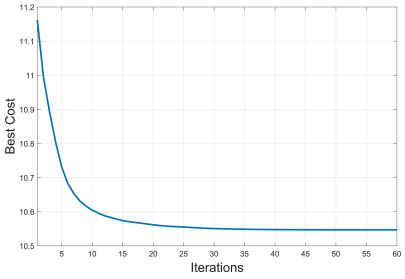
(a) Glass data set



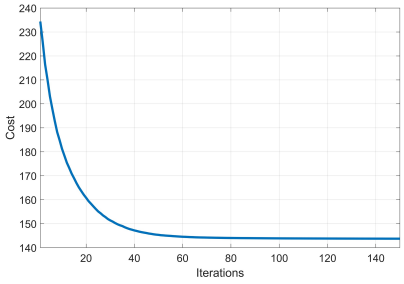
(b) Haberman data set



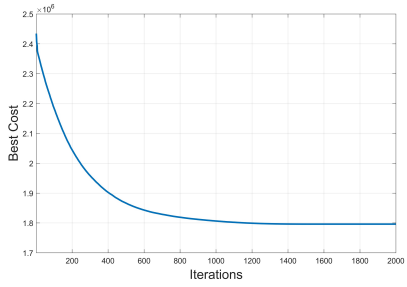
(c) Iris data set



(d) Lenses data set



(e) Spect data set



(f) Wine data set

Figure 3. Convergence graphs

Table 13. The results of the non-parametric sign test

WSA vs. FCM	*J*	*PC*	*PE*	*Chen Linkens*	*Fukuyama*	*Xie Beni*
wins (+)	12	6	7	6	6	8
losses (-)	0	6	5	6	6	4
outrank ratio	100%	50%	58%	50%	50%	67%
alpha=0.05	+	∼	∼	∼	∼	∼
alpha=0.1	+	∼	∼	∼	∼	∼

6. Conclusions

In this study, first time in the literature, WSA algorithm is used to contribute to fuzzy clustering techniques. In order to test the capability of WSA in tackling fuzzy clustering

problems, a set of computational experiments on the well-known datasets is carried out. The performance of the proposed approach is further improved by using a quantum-based local search procedure that provides and intensifying search with a shrinking quantum cloud.

Comprehensive experimental study is conducted on the well-known benchmark data set. The results indicate the superiority of the proposed approach. It is further shown by non-parametric statistical tests WSA has significant improvements over the traditional FCM. Additionally, cluster validity indices show that WSA algorithm is as good as traditional FCM, yet two methods are not able to dominate each other.

It is clear that the developed procedures here can easily be adopted in any other metaheuristic. In this regard, hybridizing some building blocks of WSA with other algorithms is scheduled as the future work. Different performance metrics can also be implemented for FCM. For instance, validity indices used within the scope of the paper can be considered as an objective function during the training. Last but not least, different variants of the FCM can be improved by using proposed algorithm.

References

[1] Alam, S., Dobbie, G., Koh, Y.S., Riddle, P., and Rehman, S.U. *Research on particle swarm optimization based clustering: A systematic review of literature and techniques*, Swarm and Evolutionary Computation **17**, 1-13, 2014.

[2] Bandyopadhyay, S. and Maulik, U. *An evolutionary technique based on K-Means algorithm for optimal clustering in* R^N, Information Sciences **146** (1), 221-237, 2002.

[3] Baykasolu, A. and Akpinar, . *Weighted Superposition Attraction (WSA): A swarm intelligence algorithm for optimization problems Part 1: Unconstrained optimization*, Applied Soft Computing **56**, 520-540, 2017.

[4] Baykasolu, A. and Akpinar, . *Weighted Superposition Attraction (WSA): A swarm intelligence algorithm for optimization problems Part 2: Constrained optimization*, Applied Soft Computing **37**, 396-415, 2015.

[5] Baykasolu, A. and Ozsoydan, F.B. *Dynamic optimization in binary search spaces via weighted superposition attraction algorithm*, Expert Systems with Applications **96** 157-174, 2018.

[6] Belacel, N., Hansen, P., and Mladenovic, N. *Fuzzy J-Means: a new heuristic for fuzzy clustering*, Pattern Recognition **35** (10), 2193-2200, 2002.

[7] Bezdek, J.C., *Fuzzy Mathematics in Pattern Classification*, Cornell University: Ithaca, NY, 1973.

[8] Bezdek, J.C., Ehrlich, R., and Full, W. *FCM: The fuzzy c-means clustering algorithm*, Computers and Geosciences **10** (2), 191-203, 1984.

[9] Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms* (Plenum Press, New York, 1981).

[10] Bezdek, J.C. *Cluster Validity with Fuzzy Sets*, Journal of Cybernetics **3** (3), 58-73, 1973.

[11] Blackwell, T., Branke, J., and Li, X., *Particle swarms for dynamic optimization problems*, in Swarm Intelligence, Springer. p. 193-217, 2008.

[12] Blackwell, T. and Branke, J., *Multi-swarm Optimization in Dynamic Environments*, in *Applications of Evolutionary Computing: EvoWorkshops*, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 489-500, 2004.

[13] Chen, M.-Y. and Linkens, D.A. *Rule-base self-generation and simplification for data-driven fuzzy models.* in *10th IEEE International Conference on Fuzzy Systems*, 2001.

[14] Derrac, J., García, S., Molina, D., and Herrera, F. *A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms*, Swarm and Evolutionary Computation **1** (1), 3-18, 2011.

[15] Filho, T.M.S., Pimentel, B.A., Souza, R.M.C.R., and Oliveira, A.L.I. *Hybrid methods for fuzzy clustering based on fuzzy c-means and improved particle swarm optimization*, Expert Systems with Applications **42** (17), 6315-6328, 2015.

[16] Forgy, E.W. *Cluster analysis of multivariate data: efficiency versus interpretability models*, Biometrics **61** (3), 768-769, 1965.

[17] Fukuyama, Y. and Sugeno, M. *A new method of choosing the number of clusters for the fuzzy c-mean method.* in *Proc. 5th Fuzzy Syst. Symp*, 1989.

[18] Graves, D. and Pedrycz, W. *Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study*, Fuzzy Sets and Systems **161** (4), 522-543, 2010.

[19] Güngör, Z. and Ünler, A. *K-harmonic means data clustering with simulated annealing heuristic*, Applied Mathematics and Computation **184** (2), 199-209, 2007.

[20] Hayes-Roth, B. and Hayes-Roth, F. *Concept learning and the recognition and classification of exemplars*, Journal of Verbal Learning and Verbal Behavior **16** (3), 321-338, 1977.

[21] José-García, A. and Gómez-Flores, W. *Automatic clustering using nature-inspired metaheuristics: A survey*, Applied Soft Computing **41**, 192-213, 2016.

[22] Kao, Y.-T., Zahara, E., and Kao, I.W. *A hybridized approach to data clustering*, Expert Systems with Applications **34** (3), 1754-1762, 2008

[23] Li, C., Zhou, J., Kou, P., and Xiao, J. *A novel chaotic particle swarm optimization based fuzzy clustering algorithm*, Neurocomputing **83**, 98-109, 2012.

[24] Lichman, M., *UCI Machine Learning Repository*, University of California, School of Information and Computer Sciences, Irvine, CA, 2013.

[25] Nanda, S.J. and Panda, G. *A survey on nature inspired metaheuristic algorithms for partitional clustering*, Swarm and Evolutionary Computation **16** (Supplement C), 1-18, 2014.

[26] Nayak, J., Naik, B., Behera, H.S., and Abraham, A. *Hybrid chemical reaction based metaheuristic with fuzzy c-means algorithm for optimal cluster analysis*, Expert Systems with Applications **79**, 282-295, 2017.

[27] Özbakr, L. and Turna, F. *Clustering performance comparison of new generation metaheuristic algorithms*, Knowledge-Based Systems **130**, 1-16, 2017.

[28] Pakhira, M.K., Bandyopadhyay, S., and Maulik, U. *A study of some fuzzy cluster validity indices, genetic clustering and application to pixel classification*, Fuzzy Sets and Systems **155** (2), 191-214, 2005.

[29] Pal, N.R., Pal, K., Keller, J.M., and Bezdek, J.C. *A possibilistic fuzzy c-means clustering algorithm*, IEEE transactions on fuzzy systems **13** (4), 517-530, 2005.

[30] Pimentel, B.A. and de Souza, R.M.C.R. *A multivariate fuzzy c-means method*, Applied Soft Computing **13** (4), 1592-1607, 2013.

[31] Pimentel, B.A. and de Souza, R.M.C.R. *A weighted multivariate Fuzzy C-Means method in interval-valued scientific production data*, Expert Systems with Applications **41** (7), 3223-3236, 2014.

[32] Sabzekar, M. and Naghibzadeh, M. *Fuzzy c-means improvement using relaxed constraints support vector machines*, Applied Soft Computing **13** (2), 881-890, 2013.

[33] Shelokar, P.S., Jayaraman, V.K., and Kulkarni, B.D. *An ant colony approach for clustering*, Analytica Chimica Acta **509** (2), 187-195, 2004.

[34] Siegler, R.S. *Three aspects of cognitive development*, Cognitive psychology **8** (4), 481-520, 1976.

[35] Xie, X.L. and Beni, G. *A validity measure for fuzzy clustering*, IEEE Transactions on pattern analysis and machine intelligence **13** (8), 841-847, 1991.

[36] Zhang, L., Pedrycz, W., Lu, W., Liu, X., and Zhang, L. *An interval weighed fuzzy c-means clustering by genetically guided alternating optimization*, Expert Systems with Applications **41** (13), 5960-5971, 2014.

[37] Zhang, C., Ouyang, D., and Ning, J. *An artificial bee colony approach for clustering*, Expert Systems with Applications **37** (7), 4761-4767, 2010.

[38] Zhao, F., Fan, J., and Liu, H. *Optimal-selection-based suppressed fuzzy c-means clustering algorithm with self-tuning non local spatial information for image segmentation*, Expert Systems with Applications **41** (9), 4083-4093, 2014.