Asgarali Bouyer* and Nacer Farajzadeh

An Optimized *K*-Harmonic Means Algorithm Combined with Modified Particle Swarm Optimization and Cuckoo Search Algorithm

DOI 10.1515/jisys-2015-0009
Received January 27, 2015; previously published online June 25, 2015.

Abstract: Among the data clustering algorithms, the *k*-means (KM) algorithm is one of the most popular clustering techniques because of its simplicity and efficiency. However, KM is sensitive to initial centers and it has a local optima problem. The *k*-harmonic means (KHM) clustering algorithm solves the initialization problem of the KM algorithm, but it also has a local optima problem. In this paper, we develop a new algorithm for solving this problem based on a modified version of particle swarm optimization (MPSO) algorithm and KHM clustering. In the proposed algorithm, MPSO is equipped with the cuckoo search algorithm and two new concepts used in PSO in order to improve the efficiency, fast convergence, and escape from local optima. MPSO updates the positions of particles based on a combination of global worst, global best with personal worst, and personal best to dynamically be used in each iteration of the MPSO. The experimental result on eight real-world data sets and two artificial data sets confirms that this modified version is superior to KHM and the regular PSO algorithm. The results of the simulation show that the new algorithm is able to create promising solutions with fast convergence, high accuracy, and correctness while markedly improving the processing time.

Keywords: *K*-means, *k*-harmonic means clustering, particle swarm optimization, Lévy flight, local minimum.

1 Introduction

Data clustering is a popular data mining technique that is applied for extracting the reasonable organization of objects in a given data set. This technique classifies similar objects into different groups, or more precisely, the partitioning of a data set into subsets, so that each part (subset) has some similarities and common characters. In fact, a set of patterns are gathered into clusters based on the similarity among each cluster. Clustering is an important technique applied in many application domains, including document clustering [18], fraud detection [17], flow shop scheduling [29], machine learning [3], wireless mobile sensor networks [31], biomedical data [12], image processing [49], demand forecast [26], and financial classifications [34]. Many data clustering algorithms have been presented in the previous literatures with different approaches.

Clustering algorithms can be generally divided into two groups: hierarchical algorithms and partitional algorithms. Hierarchical algorithm finds nested clusters either in agglomerative or in divisive [19], and partitional algorithm divides the data sets into some clusters whose members have nothing in common with each other [16, 24, 43]. The most popular and extensively used algorithm among partitioning algorithms is the *k*-means (KM) algorithm. It easily clusters a large data set with the best runtime. However, the results of the KM algorithm are very sensitive to positions of the initial cluster centers in the problem space [50]. It

***Corresponding author: Asgarali Bouyer,** Faculty of Computer Engineering and Information Technology, Azarbaijan Shahid Madani University, 53751-71379 Tabriz, Iran, e-mail: a.bouyer@azaruniv.edu
Nacer Farajzadeh: Faculty of Computer Engineering and Information Technology, Azarbaijan Shahid Madani University, Tabriz, Iran

also has a local optimum problem [20] and does not have any criterion for computing the number of clusters. *K*-harmonic means (KHM) is an alternative algorithm to solve the sensitivity to initialization problem of KM methods [51]. This algorithm minimizes the harmonic average from all points in *N* to all centers in *K*. This approach proposes more robust results than KM with different initial configurations. KHM solves the problem of initialization using a built-in boosting function [48]. However, it easily runs into local optima like the KM algorithm. To overcome the shortcomings of the KM and KHM algorithms, some heuristic algorithms have been combined with these methods. Recently, evolutionary and meta-heuristics like genetic algorithm (GA) [13], ant colony optimization (ACO) [39, 40], artificial bee colony [23, 46], particle swarm optimization (PSO) [7], bacterial foraging optimization [37], cuckoo search (CS) optimization [45], and other optimization algorithms have been hybridized with standard and basic clustering algorithms, including KM, fuzzy KM, and KHM to reach the required quality and performance in clustering processes. These algorithms try to solve the weaknesses of the KM and KHM algorithms. However, they also have several limitations. For example, Tabu search and simulated annealing algorithms suffer from low-quality results and low convergence speed problems [14].

Xin-She and Deb proposed the CS algorithm via Levy flight in 2009 [45]. CS via Levy flights is based on the interesting breeding behavior such as brood parasitism of certain species of cuckoos. The basic ideas applied are the aggressive reproduction strategy of cuckoo and usage of Levy flights. The CS algorithm is being widely used in engineering optimization problems [17] with exceptionally good results. In addition, PSO and ACO have convergence problems. PSO is a versatile population-based stochastic optimization technique. The algorithm maintains a population of particles where each particle represents a potential solution to an optimization problem. In the regular PSO [25], the diversity loss is mainly due to the strong desirability of the global best particle, which results in that all the particles quickly converge on a local or global optimum where the global best particle locates [42].

To solve PSO weaknesses, we propose a modified version of PSO with better convergence that is combined with KHM, called KHM–MPSO, to meet the common factors. Generally, in most clustering algorithms, the main goals are to meet the required quality in clusters, such as processing time, *stdev* parameters, *F*-measure, and hError and kError [28]. In this paper, the following performance metrics are used in the comparative analysis: (i) the accuracy of final clustering results and (ii) the speed of convergence. The test suit chosen for this paper consists of eight real data sets and two artificial data sets (see Table 1). On the basis of the experimental results, it is found that the proposed KHM–MPSO performs cluster analysis with better quality and performance in comparison to PSO, KHM, and PSOKHM algorithms.

The rest of this paper is organized as follows. Section 2 briefly presents the current related works in clustering analysis and PSO. The CS via Levy flight, regular PSO, and KHM algorithm is presented in Section 3. In Section 4, the proposed KHM–MPSO clustering algorithm is explained. Section 5 shows the described experiment setting and results. Finally, the conclusion is presented in Section 6.

2 Related Works

The PSO has been used for clustering in many studies. An efficient hybrid clustering based on fuzzy PSO, ACO, and KM algorithms, called FAPSO–ACO–K, is presented in Ref. [36]. The results obtained from this technique are very notable in performance improvement of information clustering. The PSOKHM data clustering algorithm proposed a hybrid algorithm based on KHM and PSO [48]. This algorithm solves the KHM's local optima problem and PSO's slow convergence speed. The MOIMPSO clustering algorithm is a hybrid of multiobjective clustering algorithm and PSO that was presented to obtain a single best solution from the Pareto optimal archive [35]. By combining two genetic and PSO algorithms, Kao and Zahara invented a new method in which it has benefitted from jump and junction operator for genetic [21]. This approach could solve different problems of continual functions. In addition, significant changes have been obtained in finding the response to general optimization and convergence ratio. They also combined the KM algorithm, Nelder–Mead simplex search, and PSO, called K–NM–PSO [22]. The K–NM–PSO searches for cluster centers of an arbitrary

data set as does the KM algorithm, but it can effectively find the global optima. They used the KM algorithm alone to generate one particle in the initial population. It implemented a Nelder–Mead search only on the best $m + 1$ particles in each iteration, where m is the number of attributes, and then the rest of the population is moved toward the best particle of the whole population and toward the best neighbor. Van Der Merwe and Engelbrecht used PSO algorithm to solve the KM clustering problem. The algorithm is extended to use KM clustering to seed the initial swarm [44].

FC–MOPSO is another research that combined the multiobjective particle swarm (MOPSO) approach with the fuzzy clustering (FC) technique [4]. In FC–MOPSO, the migration concept is used to exchange information between different subswarms and to ensure their diversity. A new approach based on PSO and radial basis function neural networks, PSO–OSD, has been developed in Ref. [11]. PSO–OSD used the PSO algorithm, which is not sensitive to the initial values of the cluster centers. Chuang et al. combined chaotic map PSO (CPSO) with an accelerated convergence rate strategy, and introduced this accelerated CPSO (ACPSO) in their research [8]. ACPSO searches through arbitrary data sets for appropriate cluster centers. Yang et al. introduced a hybrid method (called PSOKHM) based on combining PSO and KHM to enhance the global search ability of their algorithm [48]. PSOKHM repeats KHM four times in each generation for which it employs eight generations to improve particles within the population. Furthermore, the PSO algorithm repeats eight times in each generation. A new approach for clustering based on a particle swarm optimizer for dynamic optimization problems, CPSO, is presented in Ref. [6]. CPSO employs a hierarchical clustering method to track multiple peaks based on a nearest neighbor search strategy. Kiranyaz et al. proposed a PSO algorithm and fractional global best formation technique for multidimensional search in dynamic environment [27]. The GAI–PSO method is the combination of PSO, GA, and KM algorithm to find global optimum and fast convergence [1]. The GAI–PSO algorithm searches the solution space to find the optimal initial cluster centroids for the next phase. The next phase is a local refining stage utilizing the KM algorithm that can efficiently converge to the optimal solution. The GSOKHM algorithm is another method that has been presented to improve the efficiency of KHM using the PSO algorithm and GA [10]. Xin-She and Deb [45] has applied the CS algorithm for clustering. They have evaluated CS with GA and PSO using standard benchmark functions. In their study, the CS algorithm is used with Levy flight and is found to be performing better compared to the other two methods. ICAKHM is a novel method on the basis of a hybrid KHM algorithm and a modified version of the imperialist competitive algorithm (ICA) [2]. This version of the ICA method uses the genetic operators of crossover and mutation to prevent the premature convergence, helping KHM to evade the local optima problem similar to many other evolutionary algorithms. The evaluation result of the ICAKHM method [33] reveals that its results are often suitable. However, this algorithm usually is unstable, and its result may or may not be improved. We have compared our proposed algorithm with the ICAKHM method in Section 5. In addition, Ref. [33] presents a survey of the relevant literature in this field.

3 The Regular Cuckoo, PSO, and KHM Clustering Algorithms

Data clustering is aimed at finding out a reasonable organization for the objects of a given data set by identifying and quantifying similarities or dissimilarities among the objects [32]. In fact, clustering includes some qualities based on which a data set can be divided into parts (cluster) so that the components of each part have the most similarity with each other and the least similarity with members of the other parts. The goal of data clustering is to minimize the objective function, in this case a squared error function [41].

The cluster centers are represented by Eq. (2).

$$f(k) = \sum_{k=1}^{K} \sum_{i=1}^{n_k} (x_i - C_k)^2, \tag{1}$$

$$C_k = \frac{1}{n_k} \sum_{i=1}^{n_k} x_i, \tag{2}$$

where $k = 1, 2, \ldots, K$ is the number of clusters, $f(k)$ is the objective function, x_i, $i = 1, 2, \ldots, n_k$ are the patterns in the kth cluster, and C_k is center of the kth cluster.

Before the explanation of our proposed hybrid method (KHM–MPSO) for clustering, Lévy flight, regular PSO, and KHM algorithms are briefly discussed for immediate reference.

3.1 CS via Lévy Flight

The CS algorithm is a novel meta-heuristic technique [45]. The algorithm mimics the breeding behavior of cuckoos (to lay their eggs in the nests of other birds). CS is based on three idealized rules: (i) each cuckoo lays a single egg into a randomly chosen host nest from among n nests; (ii) the nests with better-quality eggs will join the next generation; (iii) the number of available hosts' nests is fixed, and the host bird discovers the egg laid with a probability $p_a \in [0,1]$.

On the basis of these three rules, the basic steps of CS can be summarized as the pseudo-code shown in Figure 1. When generating new solutions $x_i^{(t+1)}$ from the old one $(x_i^{(t)})$, Lévy flight is performed for a cuckoo ith with the parameter $1 < \lambda < 3$ as follows:

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus \text{Lévy}(\lambda), \tag{3}$$

$$\text{Lévy} \sim u = t^{-\lambda} \quad 1 < \lambda < 3, \tag{4}$$

where $\alpha > 0$ is the step size that should be related to the scales of the problem of interests. In most cases, we can use $\alpha = 1$. The product $\oplus$ means entry-wise multiplications. This entry-wise product is similar to those used in PSO, but here the random walk via Lévy flight is more efficient in exploring the search space as its step length is much longer in the long run. The Lévy flight essentially provides a random walk while the random step length is drawn from a Lévy distribution [Eq. (4)].

3.2 Particle Swarm Optimization

PSO was first introduced by Kennedy and Eberhart [25]. PSO incorporates the swarming behavior observed in flocks of birds, schools of fish, or swarms of bees, and even human social behavior. It is a population-based

Cuckoo search algorithm

Input: CN as stopping criterion

Output: optimal fitness value

1. Objective function $f(x)$, $x = (x_1, x_2, \ldots, x_d)^T$
   Generate initial population of n host nests x_i ($i = 1, 2, \ldots, n$)
   (One of the host nests is produced by the KHM at the first time)
2. Set cycle to 1
3. Repeat
   a. Get a cuckoo randomly by Levy flights
   b. Randomly select a cuckoo by Levy flight using Eq.3
   c. Calculate its fitness value (F_c) by the objective function
   d. Randomly select a nest
   e. Calculate its fitness value (F_n) by the objective function
   f. If ($F_c < F_n$) then Replace the nest with the cuckoo
   g. A fraction *pa* of nest are replaced by new nests
   h. Calculate fitness and keep best nests
   i. Rank the solutions and find the current best
   j. Store the best nest as optimal fitness value
4. Cycle=Cycle+1
5. Until Cycle ≤CN

Figure 1: Pseudo-code of Levy Flight Cuckoo Search Algorithm.

optimization tool and can be implemented and applied easily to solve various optimization problems. In PSO, a swarm of particles "fly" through the search space. Each particle follows the previous best position found by its neighbor particles and the previous best position found by itself [33]. Particles move through an n-dimensional search space. Each particle i maintains a record of the position of its previous best performance in a vector called *pbest*. The initial positions and velocities of the particles are chosen randomly. Each particle's position is updated at each iteration step according to its own personal best position and the best solution of the swarm. When a particle takes the entire population as its topological neighbors, the best value is a global best and is called *gbest*. All particles can share information about the search space. Representing a possible solution to the optimization problem, each particle moves in the direction of its best solution and the global best position discovered by any particles in the swarm. The evolution of the swarm is governed by the following equation:

$$v_i(t+1) \leftarrow \omega v_i(t) + c_1 rand_1(pbest_i(t) - x_i(t)) + c_2 rand_2(gbest(t) - x_i(t)), \tag{5}$$

where $x_i(t)$ is the position of the ith particle at the t moment and $v_i(t)$ is the velocity of the ith particle at the t moment. The factor ω is the inertia weight that denotes a proportion of the previous velocity, *pbest* is the best position of the particle, and *gbest* is the global best position of the swarm that has been found by the whole population thus far. In addition, $rand_1$ and $rand_2$ are variables ranging in random values between 0 and 1. The constants c_1 and c_2 are positive constants that determine the impact of the personal best solution and the global best solution on the search process, respectively.

The new position of a particle is calculated using the following equation:

$$x_i(t+1) \leftarrow v_i(t+1) + x_i(t)). \tag{6}$$

The updating of the particle position is performed with Eq. (6). Both Eqs. (5) and (6) are iterated until convergence of the search process is reached. The PSO algorithm is very fast, simple, and easy to understand and implement. Nevertheless, it has some shortcomings. PSO gives good results and accuracy for single objective optimization; however, for a multiobjective problem, it is stuck in the local optima. Another PSO problem is its nature of a fast and premature convergence in mid-optimum points [38].

3.3 KHM algorithm

KHM is a center-based data algorithm that has been developed to solve the clustering problem [51]. This algorithm uses the harmonic average of distance from each data point to the cluster center instead of the minimum distance in the KM algorithm. The basic KHM algorithm is shown as follows:

$X = (x_1, \ldots, x_n)$: the data to be clustered.

$C = (c_1, \ldots, c_k)$: the set of cluster centers.

$m(c_j|x_i)$: the membership function defining the proportion of data point that belongs to center c_j.

$w(x_i)$: the weight function defining how much influence data point x_i has in recomputing the center parameters in the next iteration.

Steps:

1. Initialize the algorithm with guessed centers C, i.e. randomly choose the initial centers.
2. Calculate the objective function value according to the following equation:

$$\mathrm{KHM}(X,C) = \sum_{i=1}^{n} \frac{k}{\sum_{j=1}^{k} \frac{1}{\|x_i - c_j\|^p}}, \tag{7}$$

where p is an input parameter and typically $p \geq 2$.

3. For each data point x_i, compute its membership $m(c_j|x_i)$ in each center c_j according to Eq. (8):

$$m(c_j \mid x_i) = \frac{\|x_i - c_j\|^{-p-2}}{\sum_{j=1}^{k} \|x_i - c_j\|^{-p-2}}, \quad m(c_j \mid x_i) \in [0, 1]. \tag{8}$$

4. For each data point x_i, compute its weight $w(x_i)$ according to Eq. (9):

$$w(x_i) = \frac{\sum_{j=1}^{k} \|x_i - c_j\|^{-p-2}}{\left(\sum_{j=1}^{k} \|x_i - c_j\|^{-p-2}\right)^2} \tag{9}$$

5. For each center c_j, recompute its location from all data points x_i according to their memberships and weights using Eq. (10):

$$c_j = \frac{\sum_{j=1}^{n} m(c_j \mid x_i) \cdot w(x_i) \cdot x_i}{\sum_{j=1}^{n} m(c_j \mid x_i) \cdot w(x_i)} \tag{10}$$

6. Repeat steps 2–5 with a predefined number of iterations or until KHM(X,C) does not change significantly.
7. Assign data point x_i to cluster j with the biggest $m(c_j|x_i)$.

The objective function of the KHM algorithm introduces the conditional probability of cluster center to data points and dynamic weights of data points at each iteration. Due to employing the membership function $m(c_j|x_i)$, the KHM algorithm is particularly useful when the boundaries of clusters are not well separated and ambiguous. The KHM algorithm alleviates the weakness of the KM algorithm, which is sensitive to the initial values. However, KHM still converges to the local optimum.

4 The Proposed Clustering Algorithm

In this section, we describe an improved clustering algorithm based on a modified version of the PSO (MPSO) algorithm and KHM, called KHM–MPSO. We have combined KHM and MPSO to form a hybrid clustering algorithm that maintains the qualities of MPSO and KHM and solves their convergence and sensitivity problems. The MPSO provides a partition of data points without any prior knowledge. Meanwhile, the KHM algorithm can obtain high-quality initializations from the MPSO, and provide better input to MPSO to accelerate its convergence.

In the MPSO algorithm, we use a one-dimensional array to encode cluster centers as particles. Every particle or candidate solution in the population consists of a one-dimensional array with length of $d \times k$ cells to show all cluster centers. KHM–MPSO tries to find an optimal partition of k optimal number of compactness and well-separated clusters. The proposed algorithm is built based on two main steps where at each step, only one type of move is done by particles. The first step is to escape from local optimums and migrate away from unsuitable places in the search space. The second stage is to converge to the global optimum. These two steps are repeated alternately until the termination criteria are satisfied (e.g. maximum number of iteration achieved or no change occur in certain number of iterations). KHM–MPSO applies KHM to the particles in the swarm every 10 generations such that the fitness value of each particle is improved. In the proposed algorithm, CS via Levy flight has been used, which is efficient to find new suitable neighbors and better solutions [47]. Sometimes, the best particle of PSO is selected by Levy flight CS instead of the PSO algorithm if the objective function of the generated PSO solution is weaker than that generated by the Levy flight Cuckoo solution. Besides, in comparison to basic or regular PSO, there are two new concepts in the modified version: (i) *gworst*: the worst point in the current population or "global worst" and (ii) *pworst*: the worst point in the memory of each particle or "personal worst." The global worst is the fitness value of that candidate solution that has the worst value for objective function (maximum value in minimization problems). This value is found by all particles in the swarm. The second concept is the worst place that every particle of the population has seen

X_{11}	X_{12}	…	X_{1d}	…	X_{k1}	X_{k2}	…	X_{kd}

Figure 2: Representation of a Particle.

during their move. This concept has been used differently in our previous research work with different impact and objective functions [15]. In the MPSO, the positions of particles depend on their own current worst solution and their group's previous worst. In our proposed algorithm, the *gbest* with *gworst* and *pbest* with *pworst* can dynamically be used instead of each other. For each particle, the worst fitness values, *pworst*, and for the whole swarm, *gworst*, are computed in each iteration. A particle is shown as Figure 2.

The best previous position of the particle in *i*th iteration is calculated as

$$pworst_i(t) = [\, pworst^t_{i,1},\ pworst^t_{i,2},\ \ldots,\ pworst^t_{i,m}\,]\ m \text{ is the number of solutions.}$$

In regular PSO, we had global best and personal best, which were the best values for objective function found by all particles in the swarm and the best values for objective function found by every particle thus far. At each iteration, after finding these specific positions, the particles move in the presence of two distinct steps. The logic of movements considered is first "escaping from bad points and areas" and then convergence to appropriate places. At the first step, which we call acceleration step, particles find the suitable area of search by moving away from unsuitable areas. In fact, this move causes particles to spread in the search space and search for good solutions in a wide area, and in case of entering local optima they can bypass it. At the next step, which we also call the convergence step, all particles try to move toward the global optimum based on their personal memory and best particle. The fitness function of the KHM–MPSO clustering is the objective function of the KHM algorithm. From the mathematical inference of PSO, a larger inertia weight performs more efficient global search and a smaller one means a more effective local search. Thus, Saatchi and Hung [40] used Eq. (11), which decreases inertia weight with increasing the number of iterations linearly:

$$\omega_{(t+1)} = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{t_{\max}} \times t. \tag{11}$$

This equation has also been used in MPSO. The main steps of the proposed MPSO algorithm are summarized as the pseudo-code in Figure 3.

In this algorithm, $x_i(t)$ and $v_i(t)$ respectively show the position and the velocity of particle i at time or iteration t. $\omega v_i(t+1)$ and $Sv_i(t+1)$ respectively calculate the velocity of particle i based on *pworst* and *pbest* solutions. $Pbest_i(t)$ is the best position found by particle i that keeps the fitness value of the best candidate solution encountered by the considered particle thus far. $Gbest_i(t)$ is the best position found by the whole swarm thus far, and ω is an inertia weight scaling the previous time step velocity. $Pworst_i(t)$ is the worst position found by particle i that keeps the fitness value of the worst candidate solution. The c_1 and c_2 coefficients are two constant coefficients [0, 2] that control the influence of the best personal position of the particle ($pbest_i(t)$) and the best global position ($gbest_i(t)$), where $c1+ c2 \leq 4$. $rand_1$ and $rand_2$ are random values in the range [0, 1]. K is a constriction factor for updating the particle's flying velocity. Through the constriction factor, the algorithm can have better convergence and stability. $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum of the inertia weights, respectively, and t is the iteration counter.

Our proposed hybrid algorithm (KHM–MPSO) maintains the merits of KHM and PSO and cuckoo algorithms. The pseudo-code of the proposed KHM–MPSO algorithm is represented in Figure 4.

5 Experimental Design

We test our proposed algorithm on seven data sets and compare it with other well-known algorithms. These data sets are eight real data sets and two artificial data sets that are named as Iris, Wine, Wisconsin breast

Algorithm: Pseudo-code for modified PSO

1. Initialize n particles (one of the particles is produced by KHM algorithm at the first time).
   - Set *psize* (population size), ω (the inertia factor), c_1, c_2 (the weight between the attraction to *Pbest* and *Gbest*).
2. Repeat
   a. Calculate fitness of each particle by the objective function
   b. Select the best particle of PSO (global best position) based on ***best fitness*** value($F_{best\_pso\_particle}$)
   c. Find a best nest by Cuckoo search optimization (algorithm in Fig.1)
   d. If ($F_{best\_pso\_particle} < F_{best\_Cuckoo\_nest}$) Then *// Use the PSO generation*
      i. Select solution from PSO
      ii. Store this solution as a best nest for Cuckoo search.
   e. else *// Use the Cuckoo generation*
      i. Select solution of best nest by Cuckoo search algorithm
      ii. Store this solution as a best particle of PSO
   f. Global best position is the best fit particle
   g. Update the velocity and position for each particle i^{th} based on following steps:
      i. $K = 2/|2-(c_1+c_2+\sqrt{(c_1+c_2)-4(c_1+c_2)}|)$
      ii. $\omega v_i(t+1) \leftarrow K[\omega v_i(t) + c_1 rand_1(x_i(t) - Pworst_i(t)) + c_2 rand_2(x_i(t) - Gworst_i(t))]$
      iii. $Sv_i(t+1) \leftarrow K[\omega v_i(t) + c_1 rand_1(pbest_i(t) - x_i(t)) + c_2 rand_2(gbest(t) - x_i(t))]$
      iv. Update the velocity using: $v_i(t+1) = MAX\ \{\omega v_i(t+1), Sv_i(t+1)\}$
      v. Update the position using: $x_i(t+1) \leftarrow v_i(t+1) + x_i(t)$
      vi. Recalculate $Pbest_i(t)$, $Pworst_i(t)$, $Gbest_i(t)$, and $Gworst_i(t)$

$$Pbest_i(t) = [Pbest^t_{i,1}, Pbest^t_{i,2}, ..., Pbest^t_{i,m}] \quad m \text{ is the number of solutions}$$

$$Pworst_i(t) = [Pworst^t_{i,1}, Pworst^t_{i,2}, ..., Pworst^t_{i,m}] \quad m \text{ is the number of solutions}$$

(get the worst position of the particle in iteration i)

$$Gbest_i(t) = [Gbest^t_1, Gbest^t_2, ..., Gbest^t_m] \quad m \text{ is the number of solutions}$$

$$Gworst_i(t) = [Gworst^t_1, Gworst^t_2, ..., Gworst^t_m] \quad m \text{ is the number of solutions}$$

(get the worst in the whole particles till iteration i,)

$$\omega_{(t+1)} = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{t_{\max}} \times t$$

   vii. For each particle if (fitness of current position < fitness of personal best) then personal best = current position
   viii. Update $Pbest_i(t)$, $Pworst_i(t)$, $Gbest_i(t)$, and $Gworst_i(t)$ if the new values are better than the old ones.
3. Until stopping criteria met
4. Global best position is retained (as a cluster centre to KHM algorithm).

Figure 3: Pseudo-code of the Modified PSO Algorithm.

cancer (denoted as Cancer), contraceptive method choice (denoted as CMC), and Ripley's glass with different number of clusters, data objects, and features for every data object [5]. These data sets cover low, medium, and high dimensions. A brief description of these data sets is explained below.

5.1 The Datasets

- **ArtSet1 (n = 300, d = 2, k = 3):** This is an artificial data set. It is a two-featured problem with three unique classes. A total of 300 patterns are drawn from three independent bivariate normal distributions, where classes are distributed according to

$$N2 = \left(\mu = \begin{pmatrix}\mu_{i1}\\ \mu_{i2}\end{pmatrix}, \sum = \begin{bmatrix}0.4 & 0.04\\ 0.04 & 0.4\end{bmatrix}\right), i = 1,\ 2,\ 3$$

$$\mu_{11} = \mu_{12} = -2, \mu_{21} = \mu_{22} = 2, \mu_{31} = \mu_{32} = 6$$

  and μ being the mean vector and Σ is the covariance matrix.
- **ArtSet2 (n = 300, d = 3, k = 3):** This is an artificial data set with three features and three classes and 300 patterns, where every feature of the classes is distributed according to Class1~Uniform (10, 25), Class2~Uniform (25, 40), Class3~Uniform (40, 55).

Algorithm: Pseudo-code for KHM-MPSO hybrid algorithm

1. Set the initial parameters as follows:
   a. Set Max-Itr: maximum number of iterations (It often set to 10)
   b. Set *psize* (population size).
2. Initialize a population of size *Psize*.
3. Set iterative count count1= 0;
4. Set iterative counts count2=0, count3=0;
5. Execute the classic KHM algorithm for the first time
   a. Choosing the initial centers randomly
   b. Apply KHM algorithm (in section 3.3)
   c. Assign the result as one of the particles for PSO and one of the host nests for Cuckoo. (Other particles and host nests are initialized randomly)
6. Use the Modified PSO (MPSO) algorithm in Fig.2 to:
   a. Apply the MPSO operator to update the *psize* objects.
   b. count2 = count2 + 1. If count2 < 8, go to Step 6.1
7. (KHM Method) For each object *i*
   a. Take the result of MPSO algorithm as the initial cluster centers of the KHM algorithm.
   b. Recalculate each cluster center using the KHM algorithm.
   c. count3 = count3 + 1. If count3 < 4, go to Step 7.2.
8. count1 = count1 + 1. If count1 < Max-Itr, go to Step4.
9. Assign data point x_i to cluster j with the biggest $m(c_j|x_i)$.

Figure 4: Pseudo-code of the Proposed KHM–MPSO Algorithm.

- **Iris data set ($n = 150$, $d = 4$, $k = 3$):** This is perhaps the best-known database to be found in the pattern recognition literature. Fisher's paper is a classic in the field and is referenced frequently to this day. The data set contains three classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other two; the latter are not linearly separable from each other.
- **Wine data set ($n = 178$, $d = 13$, $k = 3$):** These data are the results of a chemical analysis of wines grown in Institute of Pharmaceutical and Food Analysis and Technologies in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.
- **Wisconsin breast cancer data set ($n = 683$, $d = 9$, $k = 2$):** In this data set, features are computed from a digitized image of a fine needle aspirate of a breast mass. They describe the characteristics of the cell nuclei present in the image. A few of the images can be found at http://www.cs.wisc.edu/~street/images/.
- **Ripley's glass data set ($n = 214$, $d = 9$, $k = 6$):** The study of classification of types of glass was motivated by a criminological investigation. At the scene of the crime, the glass left can be used as evidence, if it is correctly identified.
- **The CMC data set ($n = 1473$, $d = 10$, $k = 3$):** The samples consist of married women who were either not pregnant or not sure of their pregnancy at the time the interviews were conducted. It predicts the choice of the current contraceptive method (no contraception has 629 objects, long-term methods have 334 objects, and short-term methods have 510 objects) of a woman based on her demographic and socio-economic characteristics.
- **The thyroid gland data set ($n = 215$, $d = 3$, $k = 6$):** This data set contains three categories of human thyroid diseases, namely euthyroidism, hypothyroidism, and hyperthyroidism. In the thyroid gland data set, there are 215 samples with five attributes that were evaluated with various laboratory tests.
- **Vowel data set ($n = 871$, $d = 5$, $k = 3$):** This data set consists of 871 patterns. There are six overlapping vowel classes and three input features.
- **The Ecoli data set ($n = 336$, $d = 8$, $k = 8$).** The Ecoli data set, which contains 336 data objects, has eight clusters. The sizes of the eight clusters are 143, 77, 52, 35, 20, 5, 2, and 2, respectively.

Table 1: Properties of Eight Real Data Sets from UCI Data Repository and Two Artificial Data Sets.

	Number of attributes	Number of classes	Missing data	Number of instances
ArtSet1	2	3	No	300
ArtSet2	3	3	No	300
Iris	4	3	No	150
Wine	13	3	No	178
Wisconsin breast cancer	30	2	No	569
Ripley's glass	9	6	No	214
CMC	9	3	No	1437
Thyroid	3	6	No	215
Vowel	5	3	No	871
Ecoli	8	8	No	336

5.2 Simulation Setups

We compare the performance of the proposed algorithm on the selected data sets with traditional PSO, KHM, PSOKHM, and ICAKHM algorithms. The quality of solutions is compared by the sum of the intracluster distances, i.e. the distances between data objects within a cluster and its center. It is clear that the smaller the sum of the distances is, the higher the quality of clustering. The parameters of the proposed algorithm are adjusted based on Table 2.

In the simulation process, C_1 and C_2 are adjusted with different values between [1, 2]; and $\omega_{\min}$, $\omega_{\max}$ are set with different values between [0, 1]. For instance, Table 2 shows the various assigned values for these parameters. These parameters have just been tested on the Iris and Wine data sets to find appropriate values. The obtained results show that in all PSO-based algorithms, the best setup for these parameters is $c_1 = 2$, $c_2 = 2$, $\omega_{\min} = 0.4$, and $\omega_{\max} = 0.9$. Therefore, in the following comparisons, we evaluate our proposed algorithm (KHM–MPSO) with other algorithms based on this mentioned setting. These algorithms are implemented using Matlab 2012, and evaluated based on the following measures:

1. The most common quality measurement for clustering algorithms is the *F*-measure criterion [9]. The *F*-measure uses the ideas of precision and recall from information retrieval [9]. In other words, the *F*-measure is provided to show the clustering accuracy of the algorithms. The higher the *F*-measure, the better the clustering due to the higher accuracy of the resulting clusters mapping to the original classes. Each class i (as given by the class labels of the used benchmark data set) is regarded as the set of n_i items desired for a query; each cluster j (generated by the algorithm) is regarded as the set of n_j items retrieved for a query; n_{ij} gives the number of elements of class i within cluster j. For each class i and cluster j *F*-measure, precision (p), and recall (r) are defined as follows:

$$F(i, j) = ((b^2 + 1) \cdot P(i, j) \cdot r(i, j)) / (b^2 P(i, j) + r(i, j)), \tag{12}$$

Table 2: Simulation Setups for PSO Parameters.

C_1	C_2	$\omega_{\min}$	$\omega_{\max}$
1	1	0.4, 0.3	0.9, 1
1.5	1	0.4, 0.5	0.9, 1
1	1.5	0.4	0.9, 1
2	1	0.4	0.9, 1
2	2	0.4	1, 0.9, 0.8, 0.7
2	2	0.3	1, 0.9, 0.8
2	2	0.2	1, 0.9, 0.8

where $b = 1$ to obtain equal weighting for P and r

$$p(i,j) = (n_{ij}/n_j), \tag{13}$$

$$r(i,j) = (n_{ij}/n_j), \tag{14}$$

$$F = \sum_i \frac{n_i}{n} \max_j \{F(i,j)\} \text{ (overall } F\text{-measure)}. \tag{15}$$

Clearly, the larger value for F-measure reveals the better quality for a clustering algorithm.

2. The average *stdev* is another criterion measure that is defined as follows:

$$stdev = \frac{1}{c}\sqrt{\sum_{i=1}^{n_c} \|\sigma(v_i)\|}, \tag{16}$$

where c is the number of clusters and v_i is the center of cluster ith.

3. Objective function value in best, average, and worst values: Best is the minimum objective function value among all runs, average is the average objective function value of all runs, worst is the maximum value among all times. The smaller the value for the objective function is, the higher the quality of the clustering algorithm.

6 Experimental Results

To compare the performance of our algorithm with those of other approaches, each algorithm is 100 times for each of the data sets and averaged at the end. The simulation results are demonstrated in Table 3.

The simulation results given in Table 3 show that KHM–MPSO and PSOKHM are very precise, and on average, KHM–MPSO is more precise than PSOKHM. Furthermore, in all other data sets, our algorithm has a small *stdev* compared to the other algorithms except the cancer data set (for PSOKHM). For instance, the results obtained on the Iris data set show that KHM–MPSO converges to the global optimum of 96.6228 in most of the runs, whereas the best solutions of KHM, PSO, ICAKHM, and PSOKHM are 97.8396, 98.7741, 96.6362, and 96.6301, respectively. Additionally, the obtained best, average, and worst solutions of the KHM, PSO, PSOKHM, and KHM–MPSO algorithms indicate that KHM–MPSO is the best one for all data sets, except the cancer data set. Nevertheless, the obtained results for best and average solutions by the PSOKHM algorithm are good and close to KHM–MPSO's results, whereas the worst solution of KHM–MPSO is of higher quality than that of other algorithms. In short, KHM–MPSO has minimum values of the KHM function in Iris, Wine, CMC, and Glass data sets.

On the other hand, the simulation results of Table 3 shows that the F-measure of the proposed algorithm absolutely is better than those of obtained by others in all data sets. It reveals that the clusters are spatially well separated by the KHM–MPSO algorithm.

The *stdev* of the proposed algorithm is less than that of the other algorithms. It means that KHM–MPSO can find optimal solutions in most of the cases, while other algorithms may be trapped in local optima. Moreover, it often can find high-quality solutions compared to the other algorithms. The best *stdev* in the Iris data set (with low dimension) belongs to ICAKHM and our proposed KHM–MPSO algorithm. The *stdev* of the fitness function for these algorithms is 0.01055 in the Iris data set, which is significantly less than that of the other methods. However, ICAKHM does not have a better *stdev* in all other data sets. For the Cancer data set (with high dimension), the PSOKHM and KHM–MPSO algorithms have better *stdev* than the other algorithms. Furthermore, the KHM–MPSO algorithm has better *stdev* than the other algorithms in Wine, CMC, and Glass data sets.

In general, the simulation results shown in Table 3 indicate that the proposed KHM–MPSO algorithm converges to the global optimum with an improved *stdev* and less function evaluations. This leads logically to the end that KHM–MPSO is a feasible and a robust clustering algorithm.

Table 3: Simulation Results of 100 Runs of the Following Clustering Algorithms ($p = 2.5$).

Data set	Criteria	KHM	PSO	ICAKHM	PSOKHM	KHM–MPSO
Iris	Best solution	97.8396	98.7741	96.6362	96.6301	96.6228
	Average	102.235	99.1629	96.6664	96.6355	96.6323
	Worst solution	108.4184	102. 9339	96.6919	96.6630	96.6382
	stdev	13.2517	0.3882	0.01055	0.09128	0.01055
	F-measure	0.8853	0.8861	0.356710	0.8891	0.8924
Wine	Best solution	16,552.38	16,344.38	16,293.9	16,297.17	16,293.15
	Average	18,057.74	16,415.51	16,295.6	16,302.59	16,293.43
	Worst solution	18,560.84	16,560.82	16,296.94	16,314.37	16,293.69
	stdev	789.998	82.55	1.002372	0.62	0.49
	F-measure	0.669	0.6781	0.6802	0.671	0.6885
Cancer	Best solution	2989.72	2964.50	2962.42	2961.98	2962.10
	Average	3233.46	3029.21	3022.81	3024.47	3024.49
	Worst solution	3545.81	3338.66	3150.15	3149.82	3148.90
	stdev	250.1	108.11	0.396	0.380	0.380
	F-measure	0.9617	0.9339	0.841	0.9617	0.9647
CMC	Best solution	5847.88	5701. 53	5699.2183	5698.73	5691.16
	Average	5899.48	5822.94	5705.1485	5700.04	5694.41
	Worst solution	5942.06	5918.93	5721.1779	5702.11	5695.72
	stdev	47.16	46. 96	1.268275	0.92	0.81
	F-measure	0.45034	0.4633	0.4446	0.4524	0.4731
Glass (Ripley's glass)	Best solution	215.23	271.63	199.86	199.47	199.425
	Average	234.95	276.85	202.41	199.503	199.438
	Worst solution	257.541	284.912	209.778	199.549	199.452
	stdev	12.465	4.551	0.26	0.141	0.139
	F-measure	0.6637	0.6429	0.6695	0.6648	0.6835
Vowel	Best solution	149,423.30	163,882.00	149,201.63	148,886.44	148,896.17
	Average	153,301.24	168,527.29	161,431.04	151,153.37	148,919.28
	Worst solution	159,099.82	173,821.58	165,804.67	158,725.91	149,004.82
	stdev	1272.67	3711.25	2746.041	2881.346	125.7
	F-measure	0.650	0.650	0.650	0.652	0.658
Thyroid	Best solution	1789.511	1978.570	1326.92	1385.926	1341.426
	Average	1803.328	3216.488	2164.466	1418.779	1393.080
	Worst solution	1928.0428	4354.114	4945.92	1501.239	1484.718
	stdev	13.146	235.046	53.119	12.062	1.159
	F-measure	1.577	1.477	1.485	1.585	1.58616
Ecoli	Best solution	207,231,226	201,485,217	123,692,659	124,338,350	123,198,852
	Average	216,737,953	216,231,820	214,496,232	127,560,119	127,127,784
	Worst solution	224,846,445	240,547,078	233,929,599	129,560,105	130,521,720
	stdev	3,326,399	3,631,463	7,334,211	4,959,607	1,010,064
	F-measure	0.793949	0.853926	0.8626622	0.937064	0.954263

Owing to the close similarity between PSOKHM and the proposed algorithm, we compare them considering more details that are presented in Tables 4 and 5. Tables 4 and 5 are the results of the objective function KHM(X,C), F-measure, and runtime criteria, which are in accordance with different p values, $p = 2.5$, $p = 3$, and $p = 3$. The tables show the means and *stdev* (in brackets) for 100 independent runs. Boldface indicates the best result out of the two algorithms.

Owing to the reduction of the value of the KHM(X,C) function and the increase of the F-measure, the KHM–MPSO algorithm generates better clustering quality than the PSOKHM algorithm. In other words, KHM–MPSO improves the F-measure, runtime, and KHM(X,C) measures in most of the runs with different p values. The proposed KHM–MPSO has best runtimes in most of evaluations with different p values. The evaluations shown in Tables 4 and 5 clearly show that the KHM–MPSO algorithm has a small runtime in comparison with

Table 4: Obtained Results for PSOKHM Clustering on Eight Real Data Sets for $p = 2.5$, $p = 3$, and $p = 3.5$ Based on KHM(X,C), F-Measure, and Runtimes (for 100 Independent Runs).

PSOKHM algorithm			
	PSOKHM ($p = 2.5$)	**PSOKHM ($p = 3$)**	**PSOKHM ($p = 3.5$)**
ArtSet1			
KHM(X,C)	703.509 (0.050)	741.3861 (0.0023)	806.644 (0.0074)
F-measure	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
Runtime	0.7054 (0.0097)	0.7052 (0.0048)	0.7043 (0.0044)
ArtSet2			
KHM(X,C)	109,525.941 (0.152)	256,953.240 (13.183)	679,549.738 (283.234)
F-measure	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
Runtime	0.7461 (0.00440)	0.7411 (0.00532)	0.7363 (0.00443)
Iris			
KHM(X,C)	149.521122 (0.220046)	126.356025 (0.051715)	111.496432 (0.371611)
F-measure	0.889365 (0.001704)	0.891125 (0.000616)	0.890476 (0.000822)
Runtime	0.776514 (0.008286)	0.785798 (0.008178)	0.782373 (0.013236)
Wine			
KHM(X,C)	75,642,795.261 (123,127.311)	1,075,350,475.505 (5,934,548.867)	15,938,236,000.160 (375,608,578.016)
F-measure	0.678695 (0.008791)	0.647009 (0.008415)	0.631343 (0.007597)
Runtime	1.198114 (0.006380)	1.199792 (0.008373)	1.200532 (0.009848)
CMC			
KHM(X,C)	96,730.543291 (205.878047)	187,530.512796 (209.278023)	385,242.257966 (1077.221514)
F-measure	0.464650 (0.003114)	0.454853 (0.003122)	0.455401 (0.004125)
Runtime	7.942413 (0.013390)	7.892877 (0.033972)	8.032246 (0.034444)
Cancer			
KHM(X,C)	57,167.360619 (0.626255)	113,703.834625 (6.098736)	232,149.835544 (25.711355)
F-measure	0.961290 (0.000216)	0.964719 (0.000)	0.965644 (0.000188)
Runtime	2.846137 (0.014304)	2.848792 (0.026659)	2.860135 (0.012080)
Glass			
KHM(X,C)	1242.219883 (9.641556)	1741.945932 (19.074922)	2251.847572 (90.469432)
F-measure	0.647040 (0.019853)	0.663190 (0.017553)	0.672230 (0.016469)
Runtime	2.176758 (0.023808)	2.194541 (0.011367)	2.175558 (0.011030)
Vowel			
KHM(X,C)	149,430.2360 (0.8277)	149,015.2967 (1.5638)	148,967.8820 (1.8665)
F-measure	0.648 (0.0598)	0.650 (0.0038)	0.652 (0.0021)
F-measure	16.59 (0.02332)	16.58 (0.0649)	17.72 (0.1130)
Thyroid			
KHM(X,C)	1907.240771 (20.687668)	1596.359443 (17.436539)	1413.150447 (11.094568)
F-measure	1.594384 (0.013395)	1.589108 (0.005559)	1.593293 (0.009469)
F-measure	5.093505 (0.107767)	5.242260 (0.058426)	4.944705 (0.021232)
Ecoli			
KHM(X,C)	4,096,255.0352 (16,613.315)	32,410,697.522434 (174,022.118)	127,650,199.201421 (721,762.6053)
F-measure	0.880411 (0.029102)	0.887144 (0.035657)	0.908949 (0.035395)
Runtime	0.757109 (0.014003)	0.739773 (0.005437)	0.728822 (0.005773)

the PSOKHM algorithm in all data sets, except the Cancer data set. Consequently, the accuracy, correctness, and convergence of our proposed algorithm are more satisfactory and robust than the PSOKHM and other compared algorithms.

Finally, an execution of KHM–MPSO and other mentioned algorithms on Artset2 data set is shown in Figure 5.

The analysis of this figure also proves the improved quality of clustering by the proposed algorithm. It can be seen that KHM–MPSO can cluster objects more clearly with the best F-measure and KHM(X,C) values.

Table 5: Obtained Results for the Proposed KHM–MPSO Clustering on Eight Real Data Sets for $p = 2.5$, $p = 3$, and $p = 3.5$ Based on KHM(X,C), F-Measure, and Runtimes (for 100 Independent Runs).

Proposed KHM–MPSO Algorithm			
	KHM–MPSO ($p = 2.5$)	**KHM–MPSO ($p = 3$)**	**KHM–MPSO ($p = 3.5$)**
ArtSet1			
KHM(X,C)	668.703 (0.0061)	711.365 (0.2372)	762.157 (0.301)
F-measure	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
F-measure	0.7503 (0.0071)	0.7142 (0.0046)	0.6959 (0.0040)
ArtSet2			
KHM(X,C)	108,046.437 (0.1762)	256,469.498 (9.686)	646,197.154072 (198.852)
F-measure	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
F-measure	0.7397 (0.00466)	0.7411 (0.0053)	0.7325 (0.00493)
Iris			
KHM(X,C)	149.046166 (0.047562)	126.279684 (0.062980)	111.496936 (0.472)
F-measure	0.892268 (0.001425)	0.891129 (0.000616)	0.891775 (0.000)
F-measure	0.294792 (0.007)	0.281934 (0.004)	0.301901 (0.008)
Wine			
KHM(X,C)	74,944,203.249 (121,621.908)	1,066,602,515.373 (3,911,744.742)	15,668,540,757.021 (263,948,606)
F-measure	0.678521 (0.009)	0.647854 (0.010)	0.631675 (0.006)
F-measure	0.908909 (0.005)	0.9061102 (0.007)	1.044160 (0.019)
CMC			
KHM(X,C)	96,569.572990 (125.758)	187,350.068520 (132.273)	383,568.825231 (329.816)
F-measure	0.464754 (0.003)	0.464096 (0.002978)	0.462089 (0.002998)
F-measure	5.861913 (0.012)	5.910834 (0.020)	6.093088 (0.077)
Cancer			
KHM(X,C)	57,167.366038 (0.625)	113,716.138 (4.578)	232,137.293652 (24.844)
F-measure	0.961290 (0.000)	0.964719 (0.000)	0.965901 (0.000)
F-measure	2.088459 (0.013)	2.0761 (0.012)	2.283777 (0.035)
Glass			
KHM(X,C)	1241.6486 (9.659926)	1740.79643 (18.84145)	2251.524907 (90.139592)
F-measure	0.6471302 (0.019853)	0.672816 (0.016377)	0.67172101 (0.017489)
F-measure	2.03842 (0.017369)	1.904374 (0.01027)	2.00720816 (0.010112)
Vowel			
KHM(X,C)	148,999.8251 (0.82813)	148,995.2032 (1.58331)	148,976.0010 (1.82935)
F-measure	0.648 (0.056)	0.651 (0.002)	0.652 (0.001)
F-measure	16.03 (0.02215)	16.11 (0.0623)	17.23 (0.1102)
Thyroid			
KHM(X,C)	1863.913382 (25.397029)	1587.730451 (10.719377)	1381.654997 (14.112486)
F-measure	1.595105 (0.009447)	1.602765 (0.005239)	1.596271 (0.005473)
F-measure	4.932390 (0.008502)	5.018692 (0.063792)	4.903797 (0.007211)
Ecoli			
KHM(X,C)	4,082,027.0541 (11,661.250)	22,732,800.045764 (162,511.3850)	90,873,569 (501,099.769)
F-measure	0.998500 (0.044583)	0.965545 (0.043558)	0.928575 (0.018024)
Runtime	0.728618 (0.004084)	0.730831 (0.008129)	0.737341 (0.006761)

The main drawback for KHM–MPSO, ICAKHM, and PSOKHM is their running time in comparison to the KHM algorithm. The KHM algorithm has the best running time. However, KHM–MPSO has a better running time than ICAKHM and PSOKHM.

In the end, to indicate a significant difference between the results of the proposed KHM–MPSO algorithm with other algorithms, statistical analysis was carried out. We applied the Friedman test to realize whether there are substantial differences in the results of the clustering algorithms. In this test, the α was set to 0.05

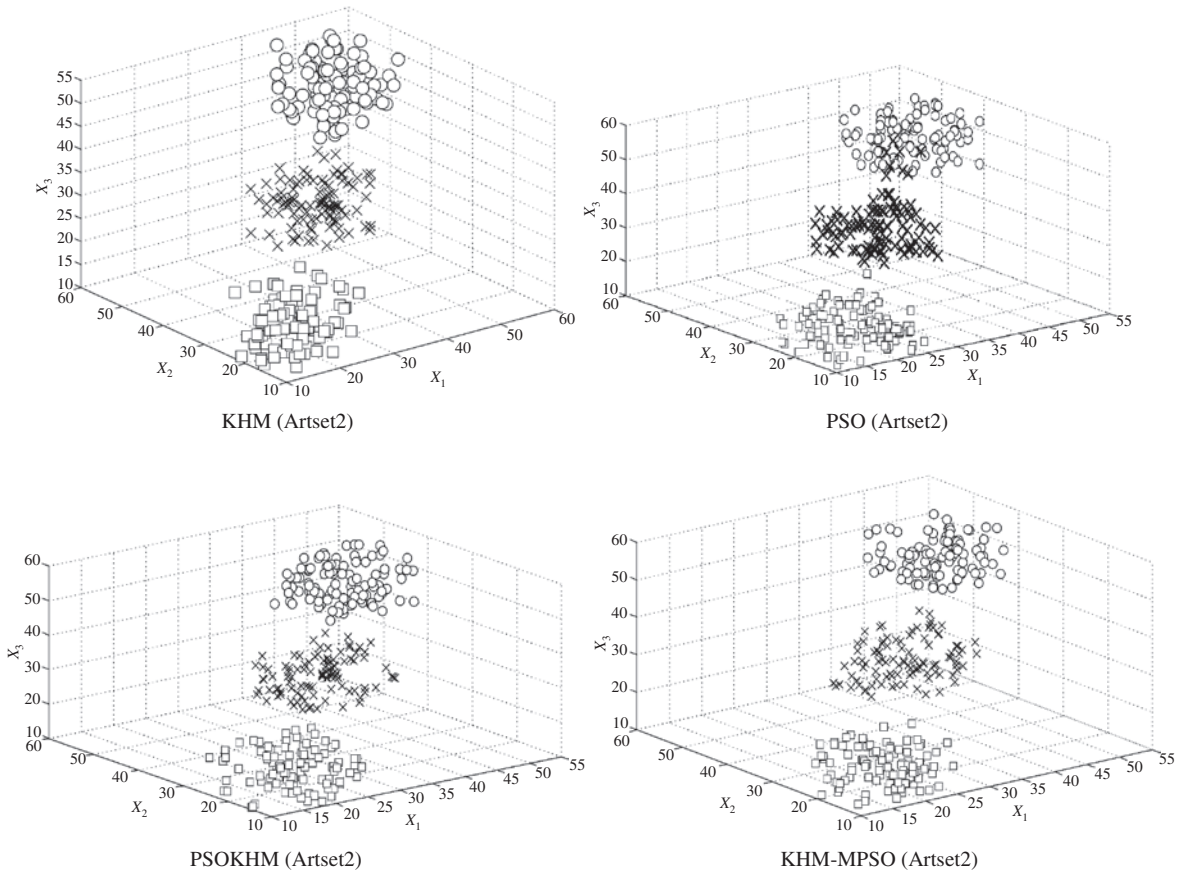

Figure 5: An Execution of KHM, PSO, PSOKHM, and KHM–MPSO Clustering Algorithms on Artset2 Data Set.

Table 6: Results of Friedman Tests Based on the Best and Average KHM() as well as 1/*F*-Measures.

Ranks			
Method name	**Mean rank (based on best KHM)**	**Mean rank (based on average KHM)**	**Mean rank (based on 1/*F*-measure)**
KHM–MPSO	1.38	1.25	1.00
PSOKHM	2.13	2.13	2.56
ICAKHM	2.50	2.88	3.75
PSO	4.50	4.38	3.75
KHM	4.50	4.38	3.94

Table 7: Test Statistics in the Friedman Test.

	N	**df**	χ^2	***p*-Value**	**Hypothesis**
Based on best KHM	8	4	26.100	0.000030	Rejected
Based on average KHM	8	4	24.400	0.000060	Rejected
Based on (1/*F*-measure)	8	4	20.464516	0.000404	Rejected

($\alpha = 0.05$) as the level of confidence in all cases. Table 6 reveals the obtained results of mean ranking of these algorithm by Friedman's test based on best and average KHM() function as well as *F*-measure. Table 7 shows the statistical test in the Friedman test. As shown in the table, the proposed KHM–MPSO algorithm is ranked

first, followed by PSOKHM, ICAKHM, PSO, and KM, successively. Furthermore, the Friedman test indicates that the proposed KHM–MPSO algorithm has a significant difference in the results of algorithms.

7 Conclusion

This paper proposed the KHM–MPSO algorithm, a hybrid clustering algorithm, by combining KHM and MPSO using CS via Levy flight algorithm. The MPSO used cuckoo optimization and two new concepts, *pworts* and *gworts*, in regular PSO algorithm. Therefore, the combination of the MPSO algorithm with KHM utilized the advantages of KHM and solved the KHM's shortcomings. It overcame the initialization sensitivity of KHM and achieved the global optima effectively. The new proposed algorithm was tested on several real and artificial data sets. The experiments confirmed that the proposed algorithm was accurate and robust compared to the PSO, KHM, PSOKHM, and ICAKHM algorithms. The proposed KHM–MPSO algorithm not only improved the *F*-measure and *stdev* parameters, but it also helped KHM escape from local optima. In the KHM–MPSO algorithm, because of obtaining high-quality initializations from the MPSO, the KHM algorithm provided better output and performance. Our proposed algorithm clustered large data sets faster and more accurately than other algorithms. Yet, it should be mentioned that one drawback of KHM–MPSO is its runtime compared to KHM. KHM has a better running time than other algorithms. As a future work, we investigate on combining PSO and artificial bee colony into KHM to reach a faster convergence, accuracy, and runtime.

Bibliography

[1] R. F. Abdel-Kader, Genetically improved PSO algorithm for efficient data clustering, in: *2010 Second International Conference on Machine Learning and Computing (ICMLC)*, 2010.
[2] M. Abdeyazdan, Data clustering based on hybrid K-harmonic means and modifier imperialist competitive algorithm, *J. Supercomput.* **68** (2014), 574–598.
[3] A. R. Anaya, J. S. Boticario, Application of machine learning techniques to analyse student interactions and improve the collaboration process, *Expert Syst. Appl.* **38** (2011), 1171–1181.
[4] L. Benameur, J. Alami and A. El Imrani, A new hybrid particle swarm optimization algorithm for handling multiobjective problem using fuzzy clustering technique, in: *International Conference on Computational Intelligence, Modelling and Simulation, CSSim '09,* 2009.
[5] C. L. Blake, D. J. Newman and C. J. Merz, UCI repository of machine learning databases, Department of Information and Computer Sciences, University of California, Irvine, 1998.
[6] L. Changhe and Y. Shengxiang, A clustering particle swarm optimizer for dynamic optimization, in: *IEEE Congress on Evolutionary Computation, CEC '09*, 2009.
[7] C.-Y. Chen and Y. Fun, Particle swarm optimization algorithm and its application to clustering analysis, in: *2004 IEEE International Conference on Networking, Sensing and Control,* 2004.
[8] L.-Y. Chuang, C.-J. Hsiao and C.-H. Yang, Chaotic particle swarm optimization for data clustering, *Exp. Syst. Appl.* **38** (2011), 14555–14563.
[9] A. Dalli, Adaptation of the *F*-measure to cluster based lexicon quality evaluation, in: *Proceedings of the EACL 2003 Workshop on Evaluation Initiatives in Natural Language Processing: Are Evaluation Methods, Metrics and Resources Reusable?*, pp. 51–56, Association for Computational Linguistics, Budapest, Hungary, 2003.
[10] M. Danesh, et al., Data clustering based on an efficient hybrid of *k*-harmonic means, PSO and GA, in: *Transactions on Computational Collective Intelligence IV*, N. Nguyen, ed., pp. 125–140, Springer, Berlin, 2011.
[11] V. Fathi and G. A. Montazer, An improvement in RBF learning algorithm based on PSO for real time applications, *Neurocomputing* **111** (2013), 169–176.
[12] S. J. Fodeh, C. Brandt, T. B. Luong, A. Haddad, M. Schultz, T. Murphy, and M. Krauthammer, Complementary ensemble clustering of biomedical data, *J. Biomed. Inform.* **46** (2013), 436–443.
[13] R. Ghaemi, N. bin Sulaiman, H. Ibrahim and N. Mustapha, A review: accuracy optimization in clustering ensembles using genetic algorithms, *Artif. Intell. Rev.* **35** (2011), 287–318.
[14] A. Hatamlou, In search of optimal centroids on data clustering using a binary search algorithm, *Pattern Recogn. Lett.* **33** (2012), 1756–1760.
[15] A. Hatamlou and A. Bouyer, Application of modified PSO on clustering, in: *5th Postgraduate Annual Research Seminar 2009 (PARS'09)*, Malaysia, 2009.

[16] He, Q, A review of clustering algorithms as applied in IR, Graduate School of Library and İnformation Science, University of İllinois at Urbana-Compaign 6 (1999).
[17] C. S. Hilas and P. A. Mastorocostas, An application of supervised and unsupervised learning approaches to telecommunications fraud detection, *Knowl.-Based Syst.* **21** (2008), 721–726.
[18] G. Hu, S. Zhou, J. Guan and X. Hu, Towards effective document clustering: a constrained K-means based approach, *Inf. Process. Manage.* **44** (2008), 1397–1409.
[19] A. K. Jain, Data clustering: 50 years beyond K-means, *Pattern Recogn. Lett.* **31** (2010), 651–666.
[20] A. K. Jain, M. N. Murty and P. J. Flynn, Data clustering: a review, *ACM Comput. Surv.* **31** (1999), 264–323.
[21] Y.-T. Kao and E. Zahara, A hybrid genetic algorithm and particle swarm optimization for multimodal functions, *Appl. Soft Comput.* **8** (2008), 849–857.
[22] Y.-T. Kao, E. Zahara and I. W. Kao, A hybridized approach to data clustering, *Exp. Syst. Appl.* **34** (2008), 1754–1762.
[23] D. Karaboga and C. Ozturk, A novel clustering approach: artificial bee colony (ABC) algorithm, *Appl. Soft Comput.* **11** (2011), 652–657.
[24] F. Keller, *Clustering*, Computer University Saarlandes, Tutorial Slides.
[25] J. Kennedy and R. Eberhart, Particle swarm optimization, in: *Proceedings IEEE International Conference on Neural Networks*, 1995.
[26] M. S. Kıran, E. Özceylan, M. Gündüz and T. Paksoy, Swarm intelligence approaches to estimate electricity energy demand in Turkey, *Knowl.-Based Syst.* **36** (2012), 93–103.
[27] S. Kiranyaz, J. Pulkkinen and M. Gabbouj, Multi-dimensional particle swarm optimization in dynamic environments, *Exp. Syst. Appl.* **38** (2011), 2212–2223.
[28] M. Kumar and N. R. Patel, Clustering data with measurement errors, *Comput. Stat. Data Anal.* **51** (2007), 6084–6101.
[29] S. Kumar and C. S. P. Rao, Application of ant colony, genetic algorithm and data mining-based techniques for scheduling, *Robot. Comput.-Integr. Manuf.* **25** (2009), 901–908.
[30] P. Lévy, *The Lévy Distribution*, Available from: http://www.math.uah.edu/stat/special/Levy.html. Accessed September, 2014.
[31] C.-M. Liu, C.-H. Lee and L.-C. Wang, Distributed clustering algorithms for data-gathering in wireless mobile sensor networks, *J. Parallel Distrib. Comput.* **67** (2007), 1187–1200.
[32] O. Z. Maimon and L. Rokach, *Data Mining and Knowledge Discovery Handbook*, vol. 1, Springer, Berlin, 2005.
[33] V. Mangat, Survey on particle swarm optimization based clustering analysis, in: *Swarm and Evolutionary Computation*, L. Rutkowski, et al., eds., pp. 301–309, Springer, Berlin, 2012.
[34] Y. Marinakis, M. Marinak, M. Doumpos and C. Zopounidis, Ant colony and particle swarm optimization for financial classification problems, *Exp. Syst. Appl.* **36** (2009), 10604–10611.
[35] S. J. Nanda and G. Panda, Automatic clustering algorithm based on multi-objective immunized PSO to classify actions of 3D human models, *Eng. Appl. Artif. Intell.* **26** (2013), 1429–1441.
[36] T. Niknam and B. Amiri, An efficient hybrid approach based on PSO, ACO and *k*-means for cluster analysis, *Appl. Soft Comput.* **10** (2010), 183–197.
[37] K. M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, *IEEE Control Syst.* **22** (2002), 52–67.
[38] S. Rana, S. Jasola and R. Kumar, A review on particle swarm optimization algorithms and their applications to data clustering, *Artif. Intell. Rev.* **35** (2011), 211–222.
[39] T. A. Runkler, Ant colony optimization of clustering models, *Int. J. Intell. Syst.* **20** (2005), 1233–1251.
[40] S. Saatchi and C. C. Hung, Hybridization of the ant colony optimization with the *k*-means algorithm for clustering, in: *Image Analysis*, pp. 511–520, Springer, Berlin, Heidelberg, 2005.
[41] J. Senthilnath, V. Das, S. N. Omkar and V. Mani, Clustering using Levy flight cuckoo search, in: *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, J.C. Bansal, et al., eds., pp. 65–75, Springer, India, 2013.
[42] Y. Shengxiang and L. Changhe, A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments, *IEEE Trans. Evol. Comput.* **14** (2010), 959–974.
[43] C. Sung and H. Jin, A Tabu-search-based heuristic for clustering, *Pattern Recogn. Lett.* **33** (2000), 849–858.
[44] D. W. Van Der Merwe and A. P. Engelbrecht, Data clustering using particle swarm optimization, in: *The 2003 Congress on Evolutionary Computation, CEC '03,* 2003.
[45] Y. Xin-She and S. Deb, Cuckoo search via Levy flights, in: *World Congress on Nature & Biologically Inspired Computing, NaBIC 2009,* 2009.
[46] X. Yan, Y. Zhu, W. Zou and L. Wang, A new approach for data clustering using hybrid artificial bee colony algorithm, *Neurocomputing* **97** (2012), 241–250.
[47] X.-S. Yang and S. Deb, Cuckoo search via Lévy flights, in: *World Congress on Nature & Biologically Inspired Computing, NaBIC 2009,* IEEE, 2009.
[48] F. Yang, T. Sun and C. Zhang, An efficient hybrid data clustering method based on K-harmonic means and particle swarm optimization, *Exp. Syst. Appl.* **36** (2009), 9847–9852.

[49] S. Yang, R. X. Wu, M. Wang and L. Jiao, Evolutionary clustering based vector quantization and SPIHT coding for image compression, *Pattern Recogn. Lett.* **31** (2010), 1773–1780.
[50] K. R. Žalik, An efficient *k'*-means clustering algorithm, *Pattern Recogn. Lett.* **29** (2008), 1385–1391.
[51] B. Zhang, M. Hsu and U. Dayal, *K*-harmonic means – a spatial clustering algorithm with boosting, in temporal, spatial, and spatio-temporal data mining, in: *Temporal, Spatial, and Spatio-temporal Data Mining*, J. Roddick and K. Hornsby, eds., pp. 31–45, Springer, Berlin, 2001.